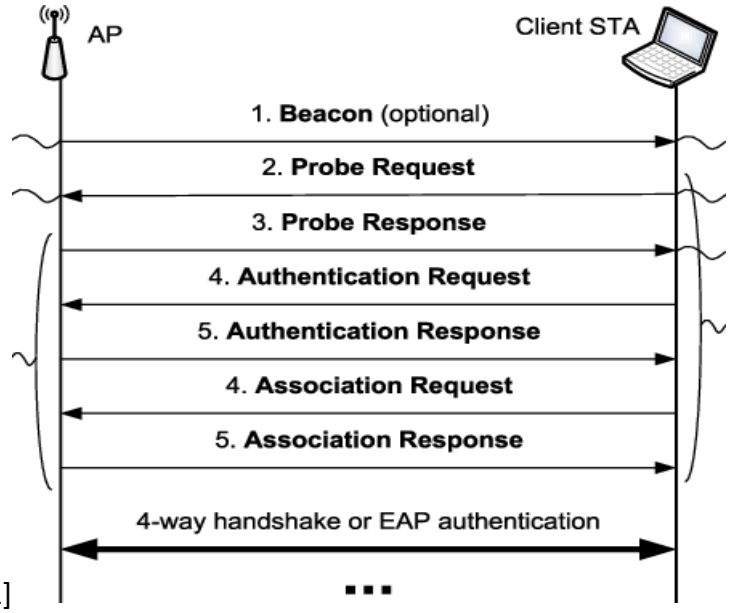
Ali Basma

Exigences – Choisir une attaque

L'attaque choisis sera celle 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.



[1]

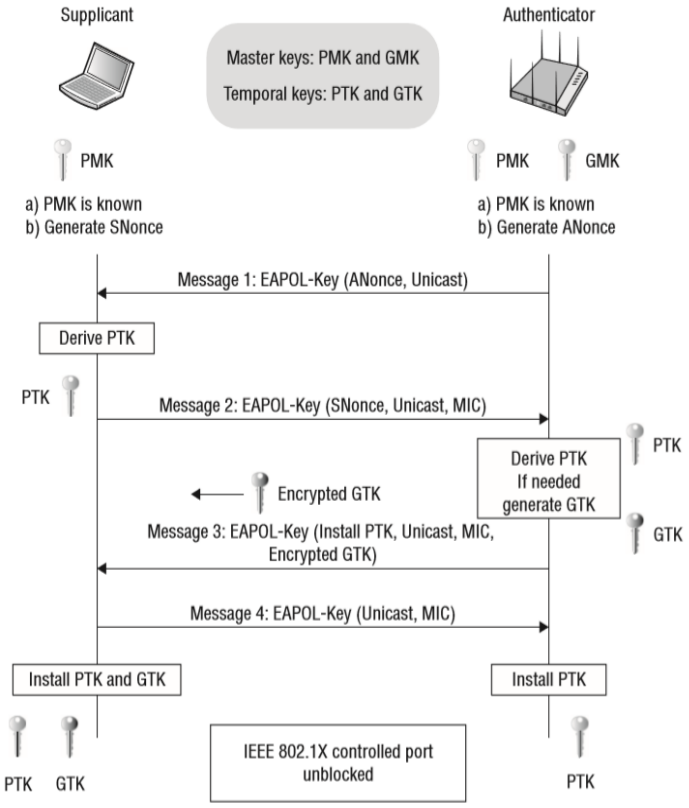
- Beacon (passive scanning) : Beacon est une trame qui contient toutes les informations sur le réseau. La trame sera envoyée périodiquement, elle est là pour annoncer la présence de l'accès point au LAN

- Probe (active scanning) : Les machines du LAN envoie une probe request pour voir quelles sont les points d'accès disponible au niveau du réseau (envoyé en broadcast). A contrepartie une fois le probe request reçu par le point d'accès celui si enverra un probe response (ressemble à la beacon trame). Enfin lorsque le client reçoit la probe response alors celui-ci enverra un ack

- Authentification : Une demande d'authentification sera envoyé du client vers le point d'accès et une reponse d'authentification sera envoyé du point d'accès au client.
  Ce processus va donc servir à voir si la machine client à les capacités requise pour se connecter a 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'acces et le client. Un schéma détailler ce trouve ci-dessus.

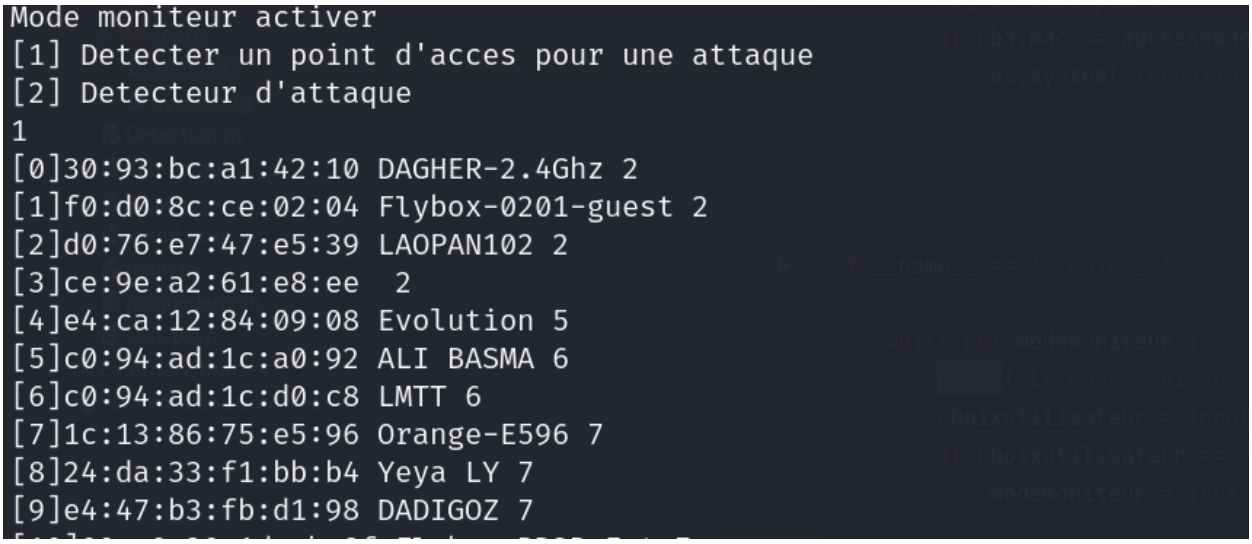
Fonctionnement de l'attaque :
Celle-ci se fera en plusieurs étapes, tout d'abord il faut configurer notre carte wifi en mode « Monitor », qui est par défaut en mode « management », le mode moniteur permettra à la carte wifi d'écouter tout le trafic réseaux
Nous cherchons maintenant à ce que l'utilisateur 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 desauthentifier du

réseaux, une fois que la desauthentification est faites nous allons arrêter l'envoie de paquets pour permettre au machine de ce reconnecter au point d'accès.
Lors de la tentative de connexion les poignées de mains (4 way handshake) se fera, et nous devrons 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.

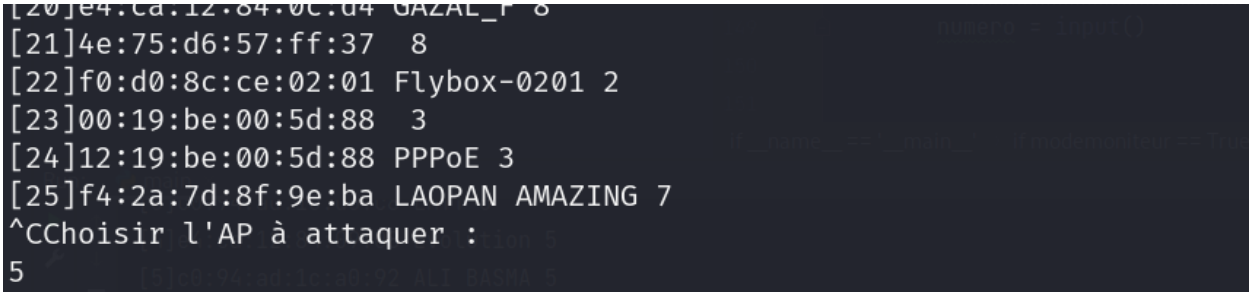
Exigences – Implémenter une attaque

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

Après que le mode moniteur soit activer 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.

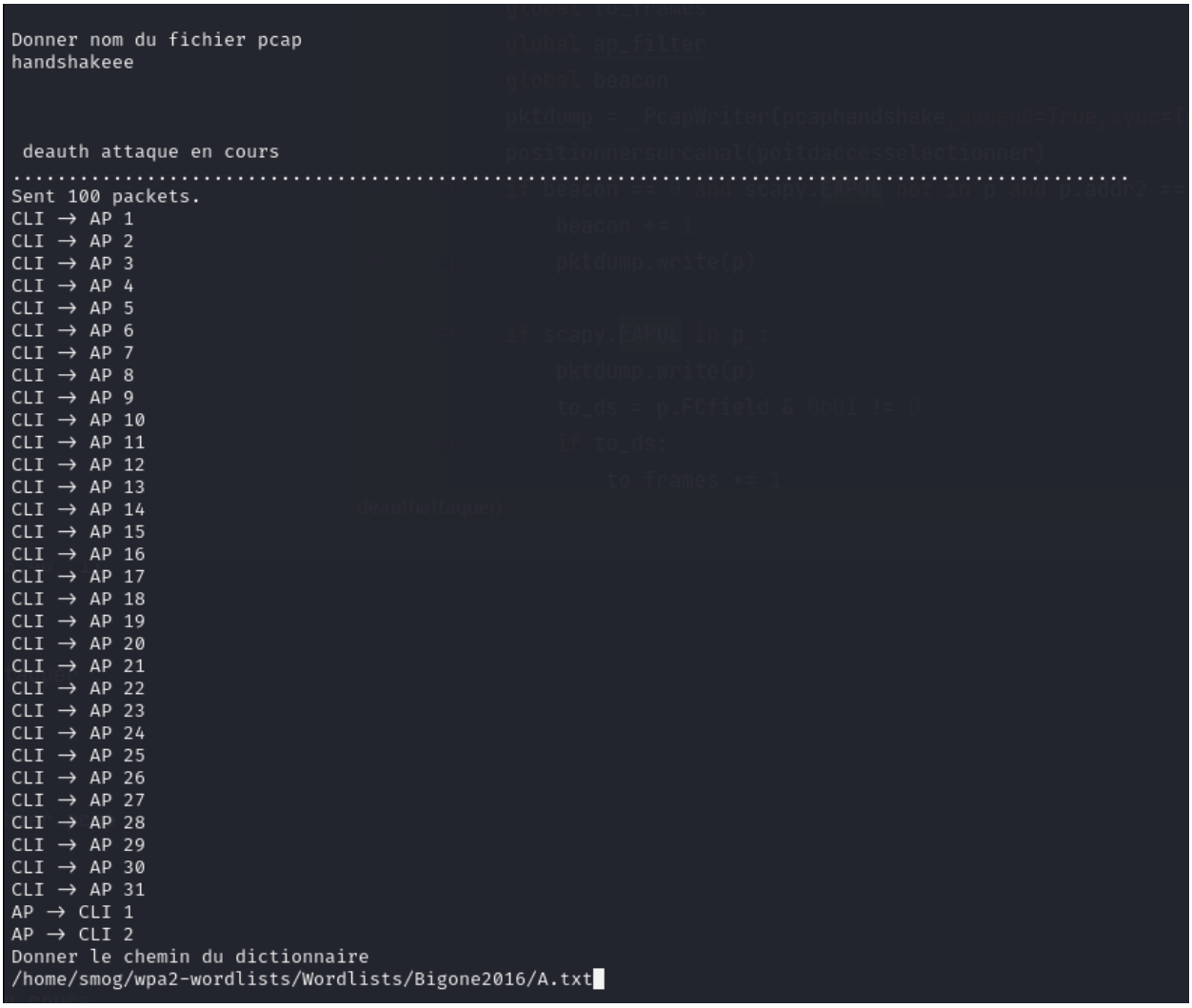


Pour arrêter le sniff il faudra faire un ctrl+c.
Puis choisir le point d'accès qu'on veut attaquer, ici c'est le 5 donc ALI BASMA

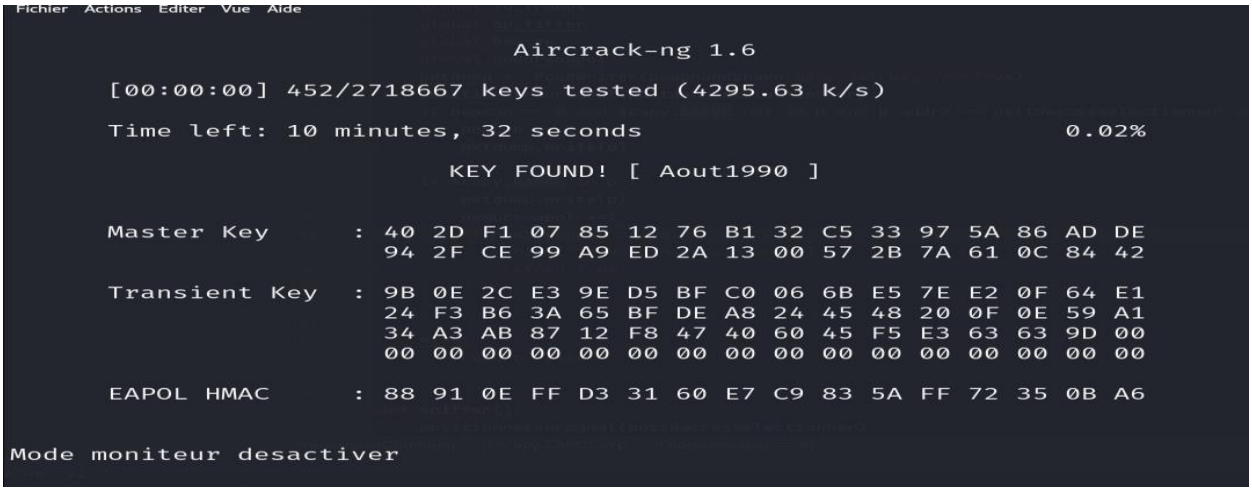


Une fois le choix du point d'accès fais nous allons donner un nom à notre fichier pcap où seront écris les paquets du 4 way hand shake (nous avons besoin des 2 qui partent du client au point d'accès et 2 qui partent du point d'accès au client).
Cependant pour provoquer cette capture, l'utilisateur doit s'authentifier, s'il est déjà authentifier 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 envoyer et que l'utilisateur n'est plus authentifier, nous devons nous sniffer les paquets avons la tentative d'authentification, pour cela nous utiliserons la méthode sniff() de scapy, nous récupérons tous les paquets que nous allons ensuite filtrer (protocole eapol) pour écrire tous cela dans le fichier pcap.(Ici nous pouvons voir qu'on a capturer 31 paquet du client vers le point d'accès et 2 paquets du point d'accès vers le client, nous avons donc à parmi c'est paquets capturer nos 4 paquets nécessaire pour le crack)
Pour terminer nous allons renseigner le chemin du fichier avec nos mot de passe.

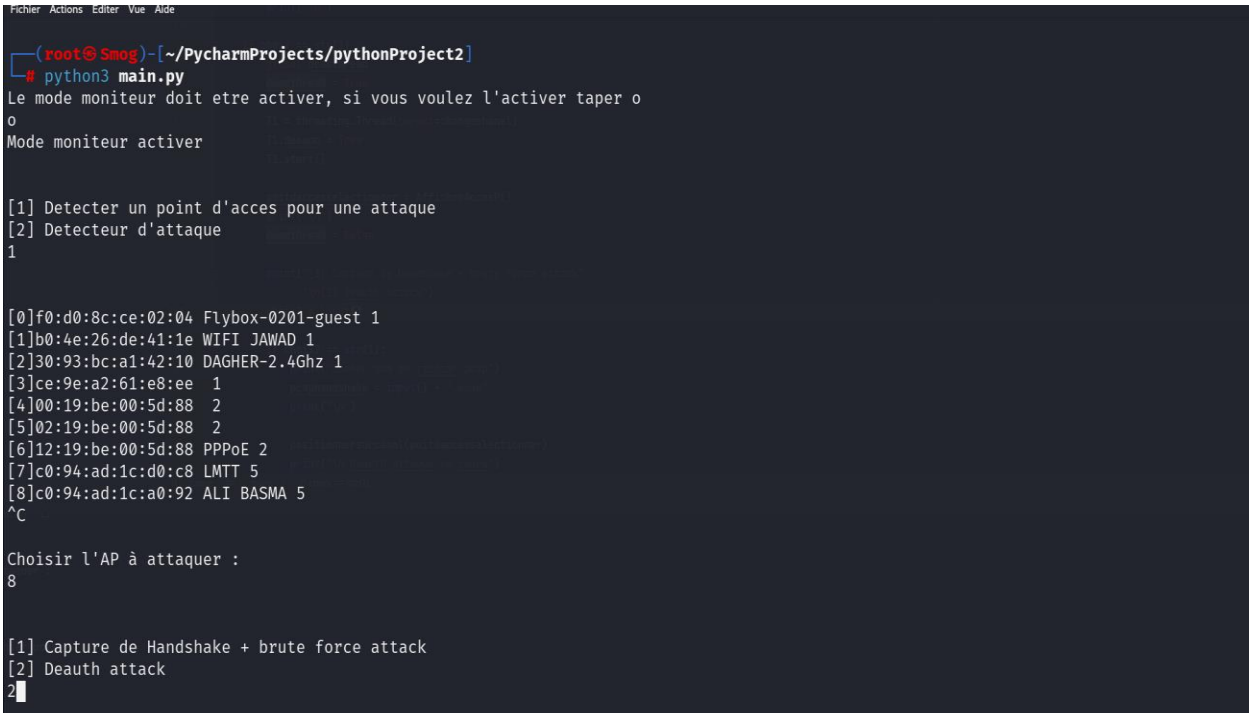


Une fois le chemin renseigner nous allons utiliser aircrack pour trouver le mot de passe, nous utiliserons donc os.system() pour avoir accès à air crack à partir de notre code. Qui utilisera notre dictionnaire de mot de passe et le fichier pcap qu'on a généré

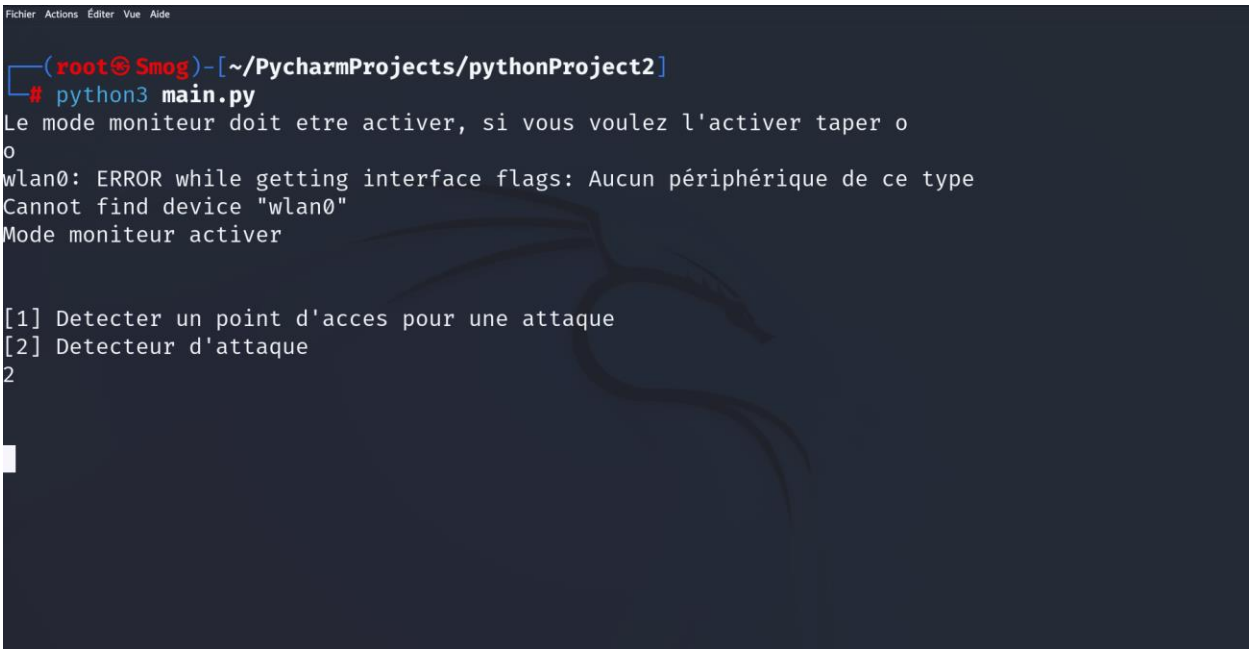


Exigences – Implémenter un détecteur d'attaque

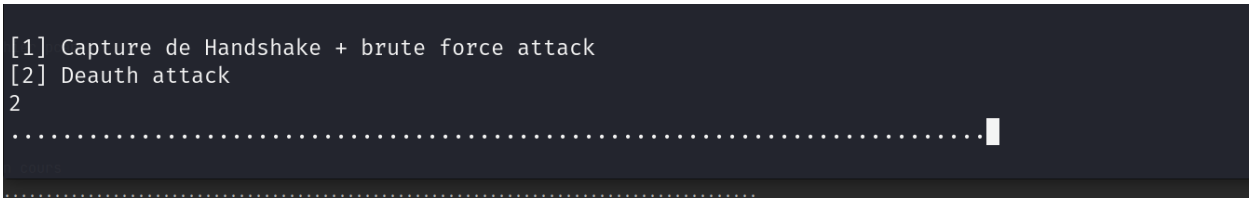
Pour cette partie nous allons detecter la deauth attaque. Nous aurons donc deux terminal un qui detectera l'autre qui attaquera.
Choisir l'ap que nous voulons attaquer et choisissons la deauth attaque simple sans capture



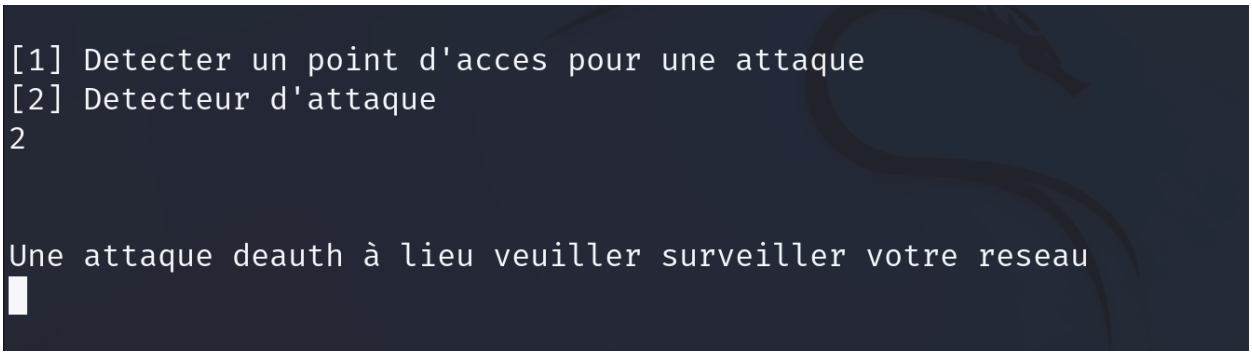
De l'autre cote nous choisissons le détecteur d'attaque



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



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