Aircrack-ng

Tutoriel réalisé sur distro kali linux.

Pour pouvoir utiliser le kit aircrack-ng, il faut s'assurer que la carte wifi interne ou externe utilisée assure le mode appelé "monitor".

Pour vérifier cela, effectuez:

```
root@kl01:~/scripts/network# iw list
```

Si monitor apparait, c'est bon!

```
Supported interface modes:
         * IBSS
         * managed
         * AP
         * AP/VLAN
         * monitor
         * mesh point
         * P2P-client
         * P2P-GO
         * outside context of a BSS
```

Si ce mode n'apparait pas dans l'output, il n'est donc pas possible d'utiliser aircrack-ng ni aucun autre outil utilisant ce mode en prérequis.

Examples de chipsets qui supportent le mode monitor:

- Atheros AR9271
- Ralink RT3070
- Realtek RTL8187L

Il est facile de trouver ce type de cartes sur un site d'e-commerce tel que Amazon, vérifiez simplement que le mode est bien supporté, les prix sont facilement abordables également.

Votre carte vous permet d'utiliser le mode monitor?

La suite est donc pour vous!

Il existe six différents modes qui sont:

- Master:
  La carte réseau joue le rôle d'un point d'accès et transmet le signal aux clients.
- Managed:
  La carte réseau se connecte à un point d'accès en envoyant certains types de paquets pour faire partie du réseau.
- Ad-hoc:
  Permet aux appareils de communiquer entre eux sans utiliser de point d'accès.
- Mesh:
  Permet aux appareils de communiquer entre eux en établissant des routes dynamiques.
- Repeater:
  Amplifie le signal du réseau pour étendre la portée de l'infrastructure de celui-ci.
- Monitor:
  Décrit en tant que mode passif, les paquets ne sont plus transmis et les paquets entrants sont analysés par l'ordinateur de manière non filtrée.

Note: comme il a été mentionné auparavant, certains de ces modes ne sont pas supportés par défaut.

Maintenant que nous avons une compréhension minimale des différents modes disponibles et de ce qu'ils signifient, on peut passer à la pratique!

Prémièrement, on a besoin de passer notre adaptateur en mode monitor.

Pour faire cela, la première étape est d'arrêter le service network-manager et de kill tout processus interférant au bon fonctionnement du mode monitor:

```
root@kl01:~/scripts/network# airmon-ng check kill
```

Dans ce cas, le processus interférant est appelé wpa_supplicant.

```
Killing these processes:

   PID Name
  2625 wpa_supplicant
```

L’interface par défaut de la carte wifi (dans ce cas) est appelée wlan0 et peut être vue en faisant:

```
root@kl01:~/scripts/network# ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 22888  bytes 1853176 (1.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22888  bytes 1853176 (1.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether c6:e5:89:24:59:a8  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Pour qu’un appareil puisse être identifié sur un réseau, une adresse MAC est créee et est assignée à cet appareil par le fabriquant.

Dans ce cas, l’adresse MAC est c6:e5:89:24:59:a8.

Un procédé appelé spoof permet de remplacer l’adresse MAC par défaut par une fausse adresse MAC pour diverses raisons.
J’expliquerais plus tard comment réaliser cela.

(L’adresse MAC dans ces exemples a été spoof)

Lors de l’activation du mode monitor, une nouvelle interface appelée wlan0mon est créee et remplace l’interface par défaut wlan0.

Pour l’activer, effectuez

```
root@kl01:~/scripts/network# airmon-ng start wlan0
```

Un output similaire avec votre appareil devrait apparaitre

```
PHY     Interface       Driver          Chipset

phy0    wlan0           ath9k_htc       Atheros Communications, Inc. AR9271 802.11n

                (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
                (mac80211 station mode vif disabled for [phy0]wlan0)
```

Pour s’assurer que le mode monitor a bien été activé, on peut vérifier la nouvelle interface

```
root@kl01:~/scripts/network# iwconfig
wlan0mon  IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off

lo        no wireless extensions.
```

```
root@kl01:~/scripts/network# ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 27288  bytes 2209576 (2.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 27288  bytes 2209576 (2.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0mon: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        unspec 0E-C3-91-23-6C-86-30-3A-00-00-00-00-00-00-00-00  txqueuelen 1000  (UNSPEC)
        RX packets 4769  bytes 1457803 (1.3 MiB)
        RX errors 0  dropped 4769  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

wlan0 a bien été remplacée par wlan0mon!

(adresse MAC spoofed également)

Maintenant que les bases pour activer le mode monitor ont été vues, on peut se pencher un peu plus en détails sur le spoofing.

Bien qu’utiliser le spoofing lors de l’utilisation d’aircrack-ng n’est pas obligatoire étant donné que l’adresse MAC de la carte (wlan0mon) n’est pas affichée dans les paquets (si quelqu’un examine les paquets que vous envoyez à partir de celle-ci, il ne verra que l’adresse MAC du point d’accès ciblé ainsi que de l’appareil ciblé), il est toujours intéressant de savoir comment cela fonctionne.

Par exemple:

Il existe une white liste dans la configuration d’un point d’accès qui autorise certains appareils via leur adresse MAC de s’y connecter et empêche ceux qui n’y sont pas présent. En utilisant les outils d’aircrack-ng, on peut déterminer les appareils présents dans la white liste facilement étant donné que ce sont ceux qui sont connectés. On peut donc spoof notre propre adresse MAC par une adresse présente dans la white liste pour avoir accès au réseau et être dans la white list à notre tour.

Une bonne utilisation de cela serait d'attendre que l'appareil que nous allons spoof se déconnecte du point d'accès pour pouvoir nous connecter à notre tour. Si deux appareils ayant la même adresse MAC sont connectés en même temps à un point d'accès, le point d'accès aura un comportement indéterminé. Pour être un peu plus précis, dans certains cas, certains paquets seront envoyés à l'appareil A puis à l'appareil B et à un certain moment un des appareils perdra la connection au réseau.

Pour revenir à notre cas en utilisant aircrack-ng, disons qu'utiliser une adresse MAC spoof est simplement un moyen de se réassurer un minimum.

Le spoof de wlan0 sera visible une fois connecté au réseau!

Spoof de wlan0:

```
root@kl01:~/scripts/network# service network-manager stop
```

Deuxième étape, désactiver l'interface wlan0

```
root@kl01:~/scripts/network# ifconfig wlan0 down
```

Une nouvelle adresse MAC est assignée à notre interface wlan0, l'option -r permet de créer une adresse aléatoire

```
root@kl01:~/scripts/network# macchanger -r wlan0
Current MAC:                [redacted] (unknown)
Permanent MAC:              [redacted] (ALFA, INC.)
New MAC:       2e:59:c3:38:77:b6 (unknown)
```

Réactivation de l'interface

```
root@kl01:~/scripts/network# ifconfig wlan0 up
```

Et finalement, on réactive le service network-manager

```
root@kl01:~/scripts/network# service network-manager start
```

Vérifiez que l'adresse MAC a bien été changée

```
root@kl01:~/scripts/network# ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 36348  bytes 2943016 (2.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 36348  bytes 2943016 (2.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.159  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::34d:5ea5:8153:ba12  prefixlen 64  scopeid 0x20<link>
        ether 2e:59:c3:38:77:b6  txqueuelen 1000  (Ethernet)
        RX packets 78  bytes 24986 (24.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40  bytes 5590 (5.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Si vous voulez vraiment être sûr que cela marche, vous pouvez aller sur la page de configuration de votre point d'accès et voir les appareils connectés, vous y verrez l'adresse MAC changée!

Spoof de wlan0mon maintenant, c'est plus ou moins le même procédé.

Pour cette interface, pas besoin de stopper le service network-manager.

```
root@kl01:~/scripts/network# ifconfig wlan0mon down
```

Pareil qu'avant, assignation d'une adresse MAC aléatoire avec -r pour wlan0mon.

```
root@kl01:~/scripts/network# macchanger -r wlan0mon
Current MAC:                (unknown)
Permanent MAC:              (ALFA, INC.)
New MAC:       c2:7b:92:51:80:62 (unknown)
```

Réactivation de l'interface.

```
root@kl01:~/scripts/network# ifconfig wlan0mon up
```

Vérifier avec ifconfig que le changement a eu lieu.

```
root@kl01:~/scripts/network# ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 36352  bytes 2943256 (2.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 36352  bytes 2943256 (2.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0mon: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        unspec C2-7B-92-51-80-62-30-3A-00-00-00-00-00-00-00-00  txqueuelen 1000  (UNSPEC)
        RX packets 4778  bytes 1292668 (1.2 MiB)
        RX errors 0  dropped 4778  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Le mode monitor ajoute 2x10 sets de charactères à l'adresse MAC affichée dans ifconfig (30-3A-00… c'est normal). Lorsque vous utiliserez l'interface, seulement les 2x6 premiers caractères seront utilisés (C2-7B-92-51-80-62).

Pour les curieux, si vous essayez d'assigner manuellement une nouvelle adresse MAC, les deux premiers caractères (C2) doivent finir par un chiffre pair sinon ca ne marchera pas! (C1 ne marcherait pas, 11 non plus, etc!)

On peut à presént créer deux shells scripts pour automatiser la procédure d'activation et de désactivation du mode monitor ainsi que le spoof, je n'irais pas plus en détails car ce n'est qu'un récapitulatif de ce qui a été expliqué.

Créez deux fichiers avec l'éditeur de votre choix.

- monitor_on.sh
- monitor_off.sh

Donnez les droits puis ajoutez le code affiché en dessous

```
chmod +x monitor_on.sh
chmod +x monitor_off.sh
```

monitor_on.sh:

```
#!/bin/bash

# Colors.
GREEN=$(tput setaf 46)
NORMAL=$(tput sgr0)

# Kill interfering processes.
airmon-ng check kill

# Create monitor mode interface wlan0mon.
airmon-ng start wlan0
printf "${GREEN}------ Monitor mode interface wlan0mon created ------ ${NORMAL}\n\n"

# Spoof MAC address of wlan0mon.
ifconfig wlan0mon down
macchanger -r wlan0mon
ifconfig wlan0mon up
printf "${GREEN}\n------ MAC address of wlan0mon correctly spoofed ------ ${NORMAL}\n"
~
```

monitor_off.sh:

```
#!/bin/bash

# Colors.
GREEN=$(tput setaf 46)
NORMAL=$(tput sgr0)

# Disable monitor mode.
airmon-ng stop wlan0mon
printf "${GREEN}------ Monitor mode disabled successfully ------\n\n${NORMAL}"

# Reset network for default usage.
service network-manager restart
printf "${GREEN}------ Network back to default usage ------\n${NORMAL}"

# Spoof wlan0 MAC address.
service network-manager stop
ifconfig wlan0 down
printf "\n"
macchanger -r wlan0
ifconfig wlan0 up
service network-manager start
printf "${GREEN}\n------ wlan0 MAC address successfully spoofed ------${NORMAL}\n"
~
```

Exemple d’éxecution:

monitor_on.sh:

```
root@kl01:~/scripts/network# ./monitor_on.sh

Killing these processes:

   PID Name
  5747 wpa_supplicant



PHY     Interface       Driver          Chipset

phy0    wlan0           ath9k_htc       Atheros Communications, Inc. AR9271 802.11n

                (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
                (mac80211 station mode vif disabled for [phy0]wlan0)

------ Monitor mode interface wlan0mon created ------

Current MAC:   [redacted] (ALFA, INC.)
Permanent MAC: [redacted] (ALFA, INC.)
New MAC:       d2:d1:4c:8b:3b:95 (unknown)

------ MAC address of wlan0mon correctly spoofed ------
root@kl01:~/scripts/network#
```

monitor_off.sh:

```
root@kl01:~/scripts/network# ./monitor_off.sh

PHY     Interface       Driver          Chipset

phy0    wlan0mon        ath9k_htc       Atheros Communications, Inc. AR9271 802.11n

                (mac80211 station mode vif enabled on [phy0]wlan0)

                (mac80211 monitor mode vif disabled for [phy0]wlan0mon)

------ Monitor mode disabled successfully ------

------ Network back to default usage ------

Current MAC:   [redacted] (ALFA, INC.)
Permanent MAC: [redacted] (ALFA, INC.)
New MAC:       9a:bc:25:bf:60:f1 (unknown)

------ wlan0 MAC address successfully spoofed ------
root@kl01:~/scripts/network#
```