Redwane ZAGHOUINI
Mohamed Anouar BOUACHOUR

Projet Sécurité réseaux - Sécurité Wifi

A. CONFIGURATION ET ANALYSE DE WPA-PERSONNEL

1. Nous avons installé deux VM linux sous Debian 11 sans interface graphique.

```
root@debian:/home/luffyredz# sudo /sbin/modprobe mac80211_hwsim radios=3
```

Ce module nous permet de créer des cartes réseau virtuelles pour simuler des appareils Wi-Fi

```
3: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group
 default qlen 1000
    link/ether a2:ce:e6:55:6e:e8 brd ff:ff:ff:ff:ff:ff permaddr 02:00:00:00:00:0
0
4: wlan1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group
 default qlen 1000
    link/ether 5a:b1:4f:7d:07:f3 brd ff:ff:ff:ff:ff:ff permaddr 02:00:00:00:01:0
0
5: wlan2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group
 default qlen 1000
    link/ether 12:bc:33:93:d7:3a brd ff:ff:ff:ff:ff:ff permaddr 02:00:00:00:02:0
0
```

2. Ces commandes nous fournissent des informations sur la carte et ses capacités.

```
root@debian:/home/luffyredz# sudo apt-get install iw
```

```
root@debian:/home/luffyredz# sudo iw dev wlan0 info
Interface wlan0
        ifindex 3
        wdev 0x1
        addr a6:9e:74:7b:9d:0a
        type managed
        wiphy 0
        txpower 20.00 dBm
```

```
root@debian:/home/luffyredz# sudo iw phy0 info
Wiphy phy0
       wiphy index: 0
       max # scan SSIDs: 4
       max scan IEs length: 2186 bytes
       max # sched scan SSIDs: 0
       max # match sets: 0
       Retry short limit: 7
       Retry long limit: 4
       Coverage class: 0 (up to 0m)
       Device supports RSN-IBSS.
       Device supports AP-side u-APSD.
       Device supports T-DLS.
       Supported Ciphers:
               * WEP40 (00-0f-ac:1)
               * WEP104 (00-0f-ac:5)
               * TKIP (00-0f-ac:2)
               * CCMP-128 (00-0f-ac:4)
               * CCMP-256 (00-0f-ac:10)
               * GCMP-128 (00-0f-ac:8)
               * GCMP-256 (00-0f-ac:9)
               * CMAC (00-0f-ac:6)
```

```
root@debian:/home/luffyredz# sudo apt-get install hostapd
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
  hostapd
0 mis à jour, 1 nouvellement installés, 0 à enlever et 0 non
Il est nécessaire de prendre 822 ko dans les archives.
Après cette opération, 2 175 ko d'espace disque supplémentai
Réception de :1 http://deb.debian.org/debian bullseye/main a
:2.9.0-21 [822 kB]
822 ko réceptionnés en 1s (984 ko/s)
Sélection du paquet hostapd précédemment désélectionné.
(Lecture de la base de données... 144593 fichiers et réperto
)
Préparation du dépaquetage de .../hostapd_2%3a2.9.0-21_amd64
Dépaquetage de hostapd (2:2.9.0-21) ...
Paramétrage de hostapd (2:2.9.0-21) ...
Created symlink /etc/systemd/system/multi-user.target.wants/
ib/systemd/system/hostapd.service.
```

3. On créer le SSID (nom du réseau) avec un pilote open source intel

```
  GNU nano 5.4
interface=wlan0
ssid=luffyredz
hw_mode=g
channel=6
wpa=2
wpa_passphrase=luffyredz
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP CCMP
wpa_ptk_rekey=600
```

```
root@debian:/home/luffyredz# sudo hostapd /etc/hostapd/hostapd.conf
Configuration file: /etc/hostapd/hostapd.conf
nl80211: kernel reports: expected nested data
Using interface wlan0 with hwaddr 86:a3:af:12:2a:f4 and ssid "luffyredz"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED

^X
^Cwlan0: interface state ENABLED->DISABLED
wlan0: AP-DISABLED
wlan0: CTRL-EVENT-TERMINATING
nl80211: deinit ifname=wlan0 disabled_11b_rates=0
```

4. Nous allons des paquets avec « airodump-ng » :

Dans un premier temps, grâce à la commande ip a on vérifie que wlan2 est bien désactivé.

```
5: wlan2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN
```

On installe la commande airmond-ng puis on configure wlan en monitor.

```
root@debian:/home/luffyredz# airmon-ng start wlan2
bash: airmon-ng : commande introuvable
root@debian:/home/luffyredz# sudo apt-install aircrack-ng
sudo: apt-install : commande introuvable
root@debian:/home/luffyredz# sudo apt-get install aircrack-ng
```

```
root@debian:/home/luffyredz# sudo airmon-ng start wlan2

Found 4 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing
and sometimes putting the interface back in managed mode

    PID Name
    390 avahi-daemon
    393 NetworkManager
    408 wpa_supplicant
    411 avahi-daemon

PHY     Interface       Driver          Chipset

phy0    wlan0           mac80211_hwsim  Software simulator
r mac80211
phy1    wlan1           mac80211_hwsim  Software simulator
r mac80211
phy2    wlan2           mac80211_hwsim  Software simulator
r mac80211
                (mac80211 monitor mode vif enabled for [phy
2mon)
```

Cette commande créer une nouvelle interface virtuelle en mode monitor appelée dans mon cas **wlan2mon** à partir de mon interface **wlan2.**

```
7: wlan2mon: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UNKNOWN
roup default qlen 1000
```

Voici ce que donne la capture :

```
CH  8 ][ Elapsed: 3 mins ][ 2023-10-22 04:23

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSI

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Pro

(not associated)   56:3E:C4:6C:11:F7  -49    0 - 1      5       10
(not associated)   4E:37:55:4F:25:BD  -49    0 - 1      0        8
```

5. S
6. S
7. S
8. Les deux vulnérabilités conceptuelles sur le 4-way handshake :

- Attaque par dictionnaire offline : Les attaquants essaient de deviner la clé WPA en utilisant un dictionnaire de mots de passe pour déchiffrer le trafic capturé. L'attaquant capture le 4-way handshake et utilise des outils comme « **aircrack-ng »** pour tester différentes clés jusqu'à trouver la correspondance.

- KRACK : Il s'agit d'une vulnérabilité de sécurité qui permet à un attaquant de réinstaller la clé de chiffrement utilisée dans le 4-way handshake, créant ainsi une opportunité pour l'attaquant de déchiffrer le trafic Wi-Fi. Cette vulnérabilité a été découverte en 2017 et a nécessité des correctifs pour les dispositifs et les points d'accès Wi-Fi.

B. INSTALLATION ET CONFIGURATION DE FREERADIUS

1.

- On commence par update le système avec « apt update » sur la VM2

```
redwane@debian-REDWANE:~$ sudo apt install freeradius
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  freeradius-common freeradius-config freeradius-utils fre
  libfreeradius3 libtalloc2 libtevent0 libwbclient0 make
Paquets suggérés :
  freeradius-krb5 freeradius-ldap freeradius-mysql freerac
  libclone-perl libmldbm-perl libnet-daemon-perl libsql-st
Les NOUVEAUX paquets suivants seront installés :
  freeradius freeradius-common freeradius-config freeradiu
```

```
redwane@debian-REDWANE:~$ sudo systemctl status freeradius
● freeradius.service - FreeRADIUS multi-protocol policy serve
     Loaded: loaded (/lib/systemd/system/freeradius.service;
     Active: active (running) since Sun 2023-10-29 02:24:37 C
       Docs: man:radiusd(8)
             man:radiusd.conf(5)
             http://wiki.freeradius.org/
             http://networkradius.com/doc/
   Main PID: 1713 (freeradius)
     Status: "Processing requests"
      Tasks: 6 (limit: 1133)
     Memory: 81.8M (limit: 2.0G)
        CPU: 173ms
     CGroup: /system.slice/freeradius.service
             └─1713 /usr/sbin/freeradius -f
```

IP VM1

```
2: enp0s3: <BROADCAST,MULT
group default qlen 1000
    link/ether 08:00:27:98
    inet 192.168.1.34/24 b
```

IP VM2

```
2: enp0s3: <BROADCAST,MU
group default qlen 1000
    link/ether 08:00:27:
    inet 192.168.1.40/24
```

Le repertoire d'installation de freeradius est le suivant

```
root@debian:/home/luffyredz# cd /etc/freeradius
root@debian:/etc/freeradius# ls
3.0
```

- Le serveur FreeRadius écoute sur les ports UDP 1812 (authentification) et 1813 (comptabilité). (ss=netstat), hors dans notre cas il est en écoute sur le port 0.

```
root@debian:~# sudo ss -tuln | grep radius
```

```
radiusd: #### Opening IP addresses and Ports ####
listen {
        type = "auth"
        ipaddr = *
        port = 0
    limit {
        max_connections = 16
        lifetime = 0
        idle_timeout = 30
```

- Le serveur FreeRadius est lancé sous le compte "freerad" par défaut pour des raisons de sécurité.

- Pour lancer FreeRadius en mode debug, On utilise la commande freeradius -X. Cela permettra de voir la sortie de débogage en temps réel.

```
root@debian:~# sudo freeradius -X
FreeRADIUS Version 3.0.21
Copyright (C) 1999-2019 The FreeRADIUS server project and contributors
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE
You may redistribute copies of FreeRADIUS under the terms of the
GNU General Public License
For more information about these matters, see the file named COPYRIGHT
Starting - reading configuration files ...
including dictionary file /usr/share/freeradius/dictionary
including dictionary file /usr/share/freeradius/dictionary.dhcp
including dictionary file /usr/share/freeradius/dictionary.vqp
including dictionary file /etc/freeradius/3.0/dictionary
including configuration file /etc/freeradius/3.0/radiusd.conf
including configuration file /etc/freeradius/3.0/proxy.conf
including configuration file /etc/freeradius/3.0/clients.conf
including files in directory /etc/freeradius/3.0/mods-enabled/
including configuration file /etc/freeradius/3.0/mods-enabled/files
```

2.

- Dans le fichier de configuration clients.conf, on configure :
  - Le client_name : c'est un identifiant arbitraire que vous choisissez pour le client.
  - ipaddr : L'adresse IP (ou une plage d'adresses IP) du client.
  - secret : Le mot de passe partagé (secret) entre le client (point d'accès) et le serveur RADIUS.

```
client localhost {
        ipaddr = 192.168.1.34 # Adresse IP du client
        secret = root          # Mot de passe partagé
}
```

-
```
root@debian:~# cat /etc/freeradius/clients.conf | grep localhost
client localhost {
```
```
root@debian:~# cat /etc/freeradius/clients.conf | grep monitor
client monitor {
```

- On modifie paramètre secret par testing123

```
client localhost {
        ipaddr = 192.168.1.34 #
        secret = testing123
```

- Lorsque le point d'accès envoie des demandes d'authentification au serveur RADIUS, il inclut ce mot de passe partagé chiffré dans ces demandes.

  Le point d'accès s'authentifie auprès du serveur RADIUS en incluant le paramètre "secret" dans les demandes d'authentification qu'il envoie au serveur RADIUS. Le serveur RADIUS utilise ce mot de passe partagé pour valider l'authenticité du point d'accès.

Si le mot de passe partagé inclus dans la demande d'authentification du point d'accès correspond à celui configuré dans la déclaration du client dans le fichier clients.conf, alors

l'authentification est réussie, et le point d'accès est autorisé à communiquer avec le serveur RADIUS. Si le mot de passe partagé ne correspond pas, l'authentification échoue et le point d'accès n'est pas autorisé à se connecter au réseau.

3.

- ```
  root@debian:~# sudo nano /etc/freeradius/users
    GNU nano 5.4                    /etc/f
  luffyredz MD5-Password := root
  ```

  - **root** : Il s'agit du nom d'utilisateur que nous avons ajouter en tant que user.
  - **MD5**-Password : il s'agit du type de mot de passe
  - **root** : Le deuxième root est le mot de passe en clair que nous avons défini pour l'utilisateur root

- Vous pouvez utiliser le fichier **/etc/passwd** comme source d'utilisateurs, mais cela n'est pas recommandé pour les raisons de sécurité mentionnées précédemment. Les fichiers **/etc/passwd** stockent les mots de passe en texte clair, ce qui n'est pas sécurisé. Il est préférable d'utiliser des méthodes d'authentification plus robustes et sécurisées avec FreeRadius, comme la configuration d'une base de données externe, LDAP, ou des méthodes de hachage sécurisées pour stocker les mots de passe.

4.

```
root@debian:~# radtest root root localhost 1812 testing123
Sent Access-Request Id 69 from 0.0.0.0:50478 to 127.0.0.1:1812 length 74
        User-Name = "root"
        User-Password = "root"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 1812
        Message-Authenticator = 0x00
        Cleartext-Password = "root"
Received Access-Reject Id 69 from 127.0.0.1:1812 to 127.0.0.1:50478 length 20
(0) -: Expected Access-Accept got Access-Reject
```

Nous avons essayé l'authentification avec le port par défaut 1812, l'acces n'as pas fonctionner dans un premier temps.
Nous avons donc essayé avec le port qui était assigner pour notre machine

```
root@debian:~# radtest luffyredz root localhost 0 testing123
Sent Access-Request Id 200 from 0.0.0.0:41360 to 127.0.0.1:1812 length 79
        User-Name = "luffyredz"
        User-Password = "root"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 0
        Message-Authenticator = 0x00
        Cleartext-Password = "root"
Received Access-Reject Id 200 from 127.0.0.1:1812 to 127.0.0.1:41360 length 20
(0) -: Expected Access-Accept got Access-Reject
```

Même problème, donc on met les 2 machines sur le même réseau et on reconfigure les 2 machines
```

```
luffyredz@debian:~$ ping 192.168.150.4
PING 192.168.150.4 (192.168.150.4) 56(84) bytes of data.
64 bytes from 192.168.150.4: icmp_seq=1 ttl=64 time=0.449 ms
64 bytes from 192.168.150.4: icmp_seq=2 ttl=64 time=0.763 ms
64 bytes from 192.168.150.4: icmp_seq=3 ttl=64 time=0.738 ms
64 bytes from 192.168.150.4: icmp_seq=4 ttl=64 time=0.239 ms
^C
--- 192.168.150.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3062ms
rtt min/avg/max/mdev = 0.239/0.547/0.763/0.216 ms
```

```
root@debian:/home/luffyredz# ping 192.168.150.3
PING 192.168.150.3 (192.168.150.3) 56(84) bytes of data.
64 bytes from 192.168.150.3: icmp_seq=1 ttl=64 time=0.545 ms
64 bytes from 192.168.150.3: icmp_seq=2 ttl=64 time=0.734 ms
64 bytes from 192.168.150.3: icmp_seq=3 ttl=64 time=0.725 ms
^C
--- 192.168.150.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2042ms
rtt min/avg/max/mdev = 0.545/0.668/0.734/0.087 ms
```

```
client localhost {
        ipaddr = 192.168.150.3 # Adresse IP du client
        secret = testing123          # Mot de passe partagé
}
```

```
root@debian:/home/luffyredz# radtest luffyredz root 192.168.150.3 1812 testing12
3
        User-Name = luffyredz
        User-Password = root
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 1812
        Message-Authenticator = 0x00
        Cleartext-Password = root
Received Access-Accept Id 119 from 192.168.150.4:1812 to 127.0.0.1:41055 length
20
root@debian:/home/luffyredz#
```

Voici donc le résultat de la capture :

```
1  0.000000000 192.168.150.4 -> 127.0.0.1 RADIUS 240 Access-Request (1) luf
z root 127.0.1.1
0.000000000 127.0.0.1 -> 192.168.150.4 RADIUS 178 Access-Accept (2) luffyre
7.0.1.1 InternetAccessFilter
0.001256856 192.168.150.4 -> 127.0.0.1 RADIUS 240 Access-Request (1) luffyr
oot 127.0.1.1
0.001256856 127.0.0.1 -> 192.168.150.4 RADIUS 178 Access-Accept (2) luffyre
7.0.1.1 InternetAccessFilter
root@debian:/home/luffyredz#
```

C. CONFIGURATION ET ANALYSE DE WPA-ENTREPRISE

1. Configuration du point d'accès AP :

• Nous allons configurer hostapd pour annoncer WPA-Entreprise aux stations sans fil.

On ouvre le fichier de configuration hostapd.conf pour définir les paramètres de notre point d'accès.

```
  GNU nano 5.4                /etc/hostapd/hostapd.conf
interface=wlan0
ssid=luffyredz
hw_mode=g
channel=6
ieee8021x=1
auth_algs=1
eap_server=1
wpa=2
wpa_passphrase=luffyredz
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP CCMP
wpa_ptk_rekey=600
wpa_strict_rekey=1
wpa_gmk_rekey=86400
eapol_version=2
eap_user_file=/etc/hostapd/eap_users
```

• Configurer les paramètres freeradius dans le fichier de configuration de hostapd.

```
  GNU nano 5.4                /etc/freeradius/3.0/users *
#       Login-Service = Rlogin,
#       Login-IP-Host = shellbox.ispdomain.com

# #
# # Last default: shell on the local terminal server.
# #
# DEFAULT
#       Service-Type = Administrative-User


# On no match, the user is denied access.


#########################################################
# You should add test accounts to the TOP of this file! #
# See the example user "bob" above.                     #
#########################################################
luffyredz    Cleartext-Password := root
```

• On déclare l'adresse de l'AP dans le fichier clients.conf de Freeradius

```
  GNU nano 5.4              /etc/freeradius/3.0/clie
#  You can have as many per-socket client lists as
#  sections, or you can re-use a list among multip
#
#  Un-comment this section, and edit a "listen" se
#  "clients = per_socket_clients".  That IP addres
#  will then accept ONLY the clients listed in thi
#
#clients per_socket_clients {
#       client socket_client {
#               ipaddr = 192.0.2.4
#               secret = testing123
#       }
#}

client AP {
        ipaddr = 127.0.0.1
        secret = testing123
}
```

2. Configuration du wpa_supplicant et capture Wireshark :

• Configurer wpa_supplicant afin de se connecter à votre SSID en utilisant la méthode EAP/PEAP.

```
root@debian:/home/luffyredz# sudo apt install wpasupplicant
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
wpasupplicant est déjà la version la plus récente (2:2.9.0-21).
wpasupplicant passé en « installé manuellement ».
0 mis à jour, 0 nouvellement installés, 0 à enlever et 10 non mis à jour.
root@debian:/home/luffyredz# sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

```
  GNU nano 5.4          /etc/wpa_supplicant/wpa_supplicant.conf
network= {
        ssid=enp0s3
        key_mgmt=WPA-EAP
        eap=PEAP
        identity=luffyredz
        password=root
        phase1="peaplabel=0"
        phase2="auth=MSCHAPV2"
}
```

•

```
root@debian:/home/luffyredz# sudo wireshark
```

•

PEAP-MSCHAPv2 : Méthode couramment utilisée, basée sur un nom d'utilisateur et un mot de passe, compatible avec de nombreux clients et serveurs, notamment Windows.
TTLS-PAP : Protège le mot de passe PAP en le chiffrant avec une couche de sécurité TTLS, adapté lorsque la sécurité est moins critique.
TTLS-MSCHAPv2 : Combinaison de TTLS et MSCHAPv2, sécurise MSCHAPv2 avec une couche de chiffrement TTLS pour une sécurité accrue, adapté aux environnements nécessitant MSCHAPv2 mais exigeant une sécurité supplémentaire.

3. Freeradius est un serveur AAA.

• Que signifie ce terme ?
FreeRadius est en effet un serveur AAA, ce qui signifie qu'il gère l'authentification, l'autorisation et la comptabilité. Voici ce que signifie ce terme :

1. **Authentification (Authentication)** : FreeRadius permet de vérifier l'identité des utilisateurs ou des dispositifs qui tentent de se connecter à un réseau. Il s'assure que seules les personnes ou les dispositifs autorisés ont accès au réseau.
2. **Autorisation (Authorization)** : Une fois qu'un utilisateur ou un dispositif est authentifié, FreeRadius peut déterminer quelles ressources ou services sont autorisés pour cette personne ou ce dispositif. Il définit les droits d'accès en fonction des politiques de sécurité définies.

3. **Comptabilité (Accounting)** : FreeRadius permet de suivre et d'enregistrer les activités des utilisateurs ou des dispositifs sur le réseau. Cela inclut la collecte de données sur les connexions, la durée des sessions, la quantité de données transférées, etc. Ces informations sont souvent utilisées à des fins de facturation, de suivi de l'utilisation du réseau, de sécurité, etc.

• Voici les étapes pour interdire l'accès d'un utilisateur pour une plage horaire de notre choix.

- Créez un profil d'autorisation qui spécifie les heures d'accès autorisées et les heures de restriction.
- Associez ce profil d'autorisation à l'utilisateur en question en configurant le fichier de configuration des utilisateurs (par exemple, **users** dans FreeRadius).
- Assurez-vous que FreeRadius est correctement configuré pour prendre en compte ces règles d'autorisation.
- Testez l'accès de l'utilisateur pour vous assurer que les restrictions d'horaires sont correctement appliquées.

4. Installer maintenant WPA3 sur hostapd et faire des tests de connexion.

On installe hostapd avec WP3 et on configure le fichier de configuration suivant : /etc/hostapd/hostapd.conf

Il y'a une erreur lors du test de connexion relative au fichier de configuration

5.

1. **Chiffrement renforcé**
2. **Chiffrement individuel**
3. **Authentification renforcée**
4. **Protection contre les attaques par force brute**
5. **Sécurité renforcée des réseaux publics**
6. **Amélioration de la confidentialité**
7. **Compatibilité descendante**

Voici les changements majeurs de WPA3 qui apporte des améliorations significatives en matière de chiffrement, d'authentification et de sécurité des réseaux Wi-Fi par rapport à WPA2. Il vise à renforcer la sécurité des communications sans fil et à mieux protéger la vie privée des utilisateurs, notamment sur les réseaux publics. Cependant, pour tirer pleinement parti des avantages de WPA3, il est nécessaire d'utiliser des dispositifs compatibles WPA3 à la fois du côté de l'AP (Point d'Accès) et des dispositifs clients.