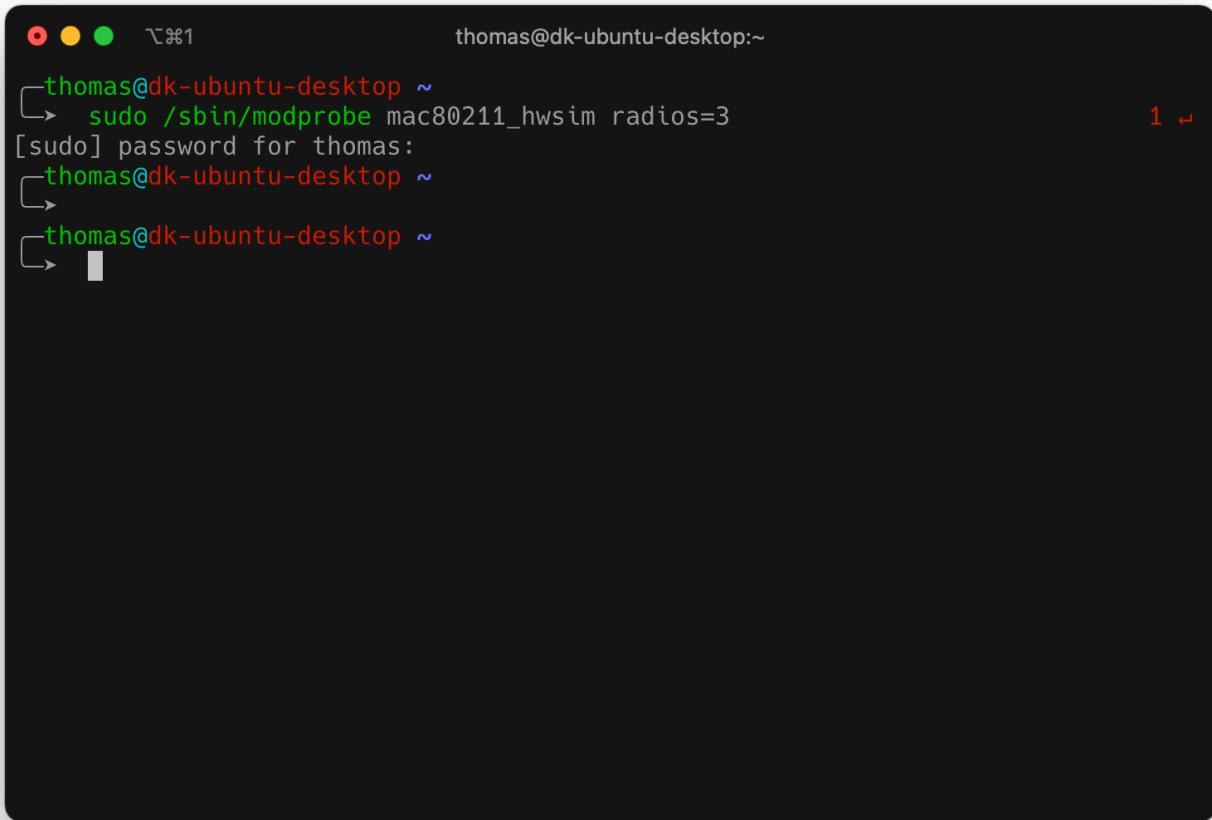


Cryptographie – Rendu TP03

Partie A

Question 1



A screenshot of a terminal window on a dark-themed desktop environment. The window title bar shows three colored circles (red, yellow, green) and the text "thomas@dk-ubuntu-desktop:~". The terminal itself has a black background with white text. It displays the following command and its execution:

```
thomas@dk-ubuntu-desktop ~
└> sudo /sbin/modprobe mac80211_hwsim radios=3
[sudo] password for thomas:
└thomas@dk-ubuntu-desktop ~
└>
└thomas@dk-ubuntu-desktop ~
└> █
```

The command `sudo /sbin/modprobe mac80211_hwsim radios=3` is run with superuser privileges. The user is prompted for their password. After entering the password, the command is executed successfully, indicated by the prompt returning to the user's home directory (~).

Le module a été correctement chargé.

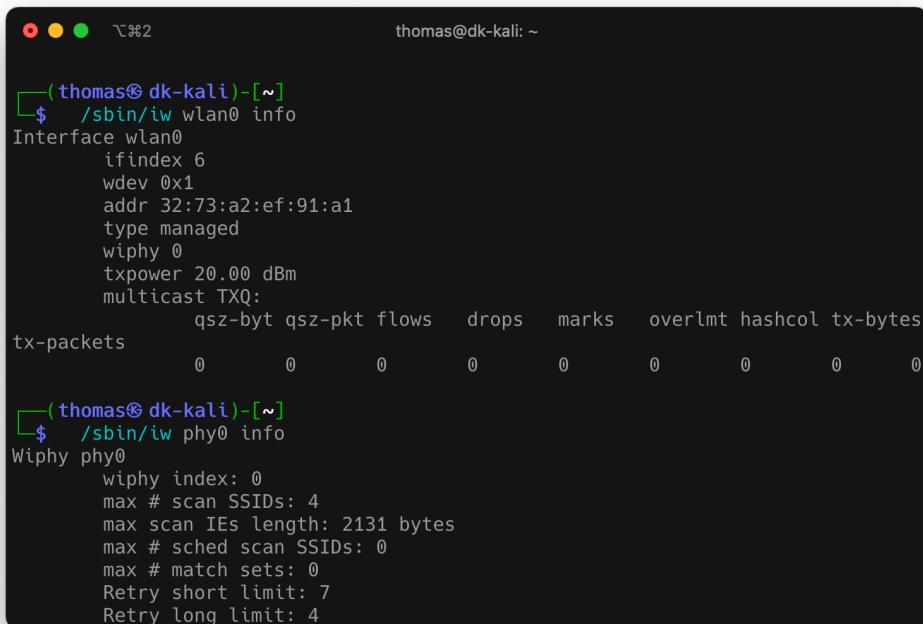
Question 2

Problème rencontré : Ubuntu ne connaît pas la commande `/sbin/iw`. Nous effectuons donc un clone de notre VM sous Kali (debian based, contenant visiblement l'exécutable).



A screenshot of a terminal window titled "thomas@dk-kali-metasploitable: ~". The window is mostly blank, with only the title bar and a few system icons visible at the top.

Une fois nos deux VMs mises à jour, nous avons le résultat suivant :



```
thomas@dk-kali: ~
(thomas@dk-kali)-[~]
$ /sbin/iw wlan0 info
Interface wlan0
    ifindex 6
    wdev 0x1
    addr 32:73:a2:ef:91:a1
    type managed
    wiphy 0
    txpower 20.00 dBm
    multicast TXQ:
        qszoq qsoq-pkt flows drops marks overlmt hashcol tx-bytes
tx-packets          0          0          0          0          0          0          0          0          0
(thomas@dk-kali)-[~]
$ /sbin/iw phy0 info
Wiphy phy0
    wiphy index: 0
    max # scan SSIDs: 4
    max scan IEs length: 2131 bytes
    max # sched scan SSIDs: 0
    max # match sets: 0
    Retry short limit: 7
    Retry long limit: 4
```

Nous pouvons noter que :

Interface wlan0 : Nom de l'interface sans fil.

ifindex 6 : Index d'interface, un numéro unique attribué à chaque interface réseau.

wdev 0x1 : Identificateur de périphérique sans fil (wdev) pour cette interface.

addr 32:73:a2:ef:91:a1 : Adresse MAC (Media Access Control) de l'interface wlan0.

type managed : Mode de fonctionnement de l'interface, "Managed" signifiant qu'elle se connecte à un point d'accès sans fil en tant que client.

wiphy 0 : Couche logicielle responsable des opérations sans fil, gérant cette interface.

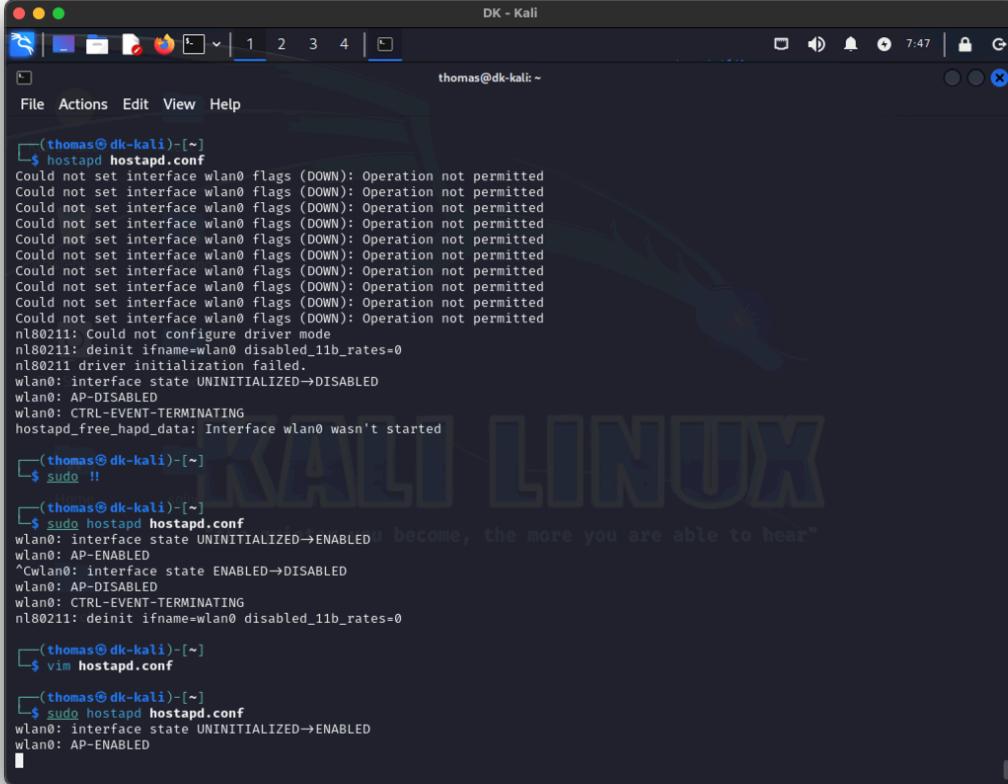
txpower 20.00 dBm : Puissance de transmission de l'interface en décibels milliwatts (dBm), réglée à 20.00 dBm.

Note : Ces informations sont issues de la documentation de /sbin/iw.

Par ailleurs, la VM actuellement utilisée est sur un hyperviseur Proxmox. Le dongle Wifi installé n'est pas configuré en USB PassThrough. L'interface WLAN0 est donc actuellement en status DOWN (depuis la commande **ip addr**).

Question 3

Après avoir une configuration minimale sur un fichier nommé `hostapd.conf`, nous obtenons le résultat suivant :



```
(thomas@dk-kali) [~]
$ hostapd hostapd.conf
Could not set interface wlan0 flags (DOWN): Operation not permitted
Could not set interface wlan0 flags (DOWN): Operation not permitted
Could not set interface wlan0 flags (DOWN): Operation not permitted
Could not set interface wlan0 flags (DOWN): Operation not permitted
Could not set interface wlan0 flags (DOWN): Operation not permitted
Could not set interface wlan0 flags (DOWN): Operation not permitted
Could not set interface wlan0 flags (DOWN): Operation not permitted
Could not set interface wlan0 flags (DOWN): Operation not permitted
Could not set interface wlan0 flags (DOWN): Operation not permitted
Could not set interface wlan0 flags (DOWN): Operation not permitted
Could not set interface wlan0 flags (DOWN): Operation not permitted
nl80211: Could not configure driver mode
nl80211: deinit ifname=wlan0 disabled_11b_rates=0
nl80211 driver initialization failed.
wlan0: interface state UNINITIALIZED->DISABLED
wlan0: AP-DISABLED
wlan0: CTRL-EVENT-TERMINATING
hostapd_free_hapd_data: Interface wlan0 wasn't started

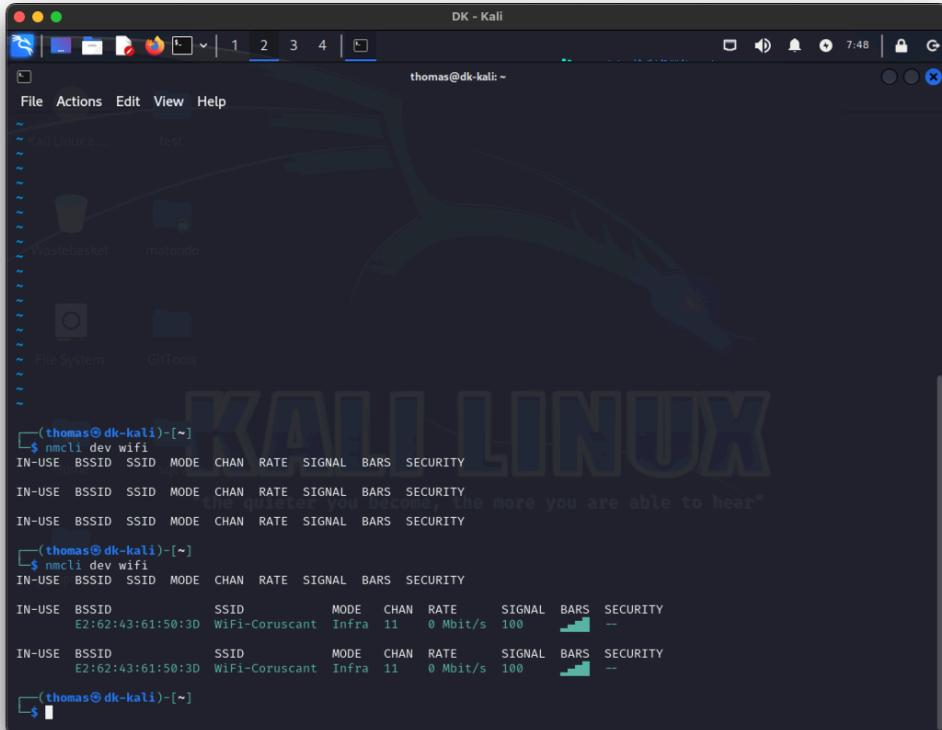
(thomas@dk-kali) [~]
$ sudo !!
(thomas@dk-kali) [~]
$ sudo hostapd hostapd.conf
wlan0: interface state UNINITIALIZED->ENABLED " become, the more you are able to hear"
wlan0: AP-ENABLED
^Cwlan0: interface state ENABLED->DISABLED
wlan0: AP-DISABLED
wlan0: CTRL-EVENT-TERMINATING
nl80211: deinit ifname=wlan0 disabled_11b_rates=0

(thomas@dk-kali) [~]
$ vim hostapd.conf

(thomas@dk-kali) [~]
$ sudo hostapd hostapd.conf
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED

[...]
```

Le SSID de notre réseau apparaît bien.



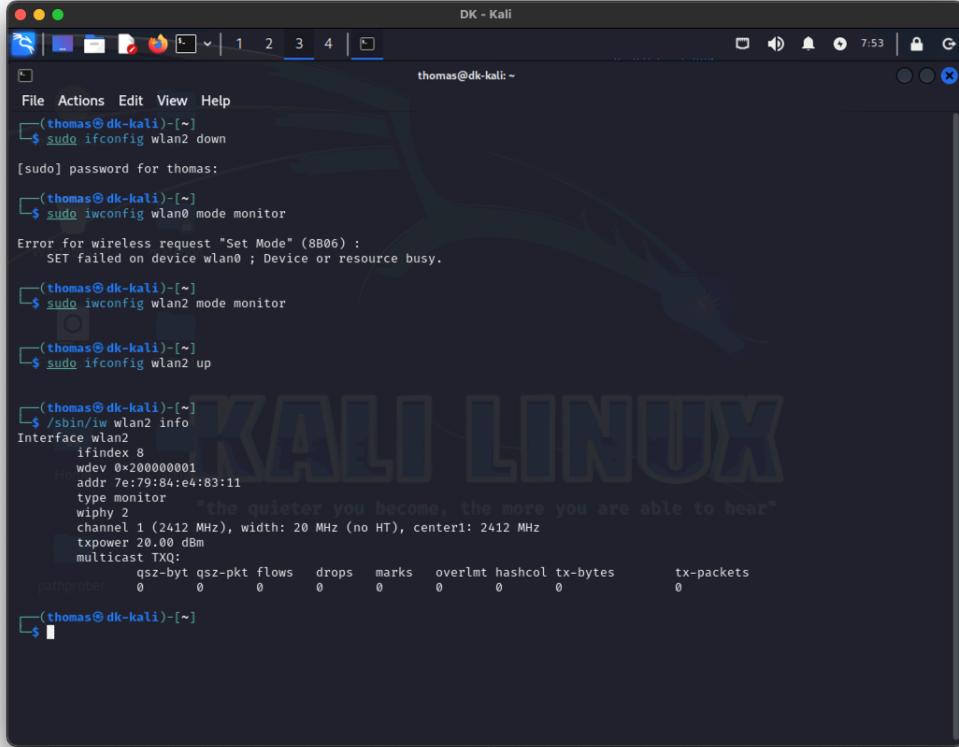
```
(thomas@dk-kali) [~]
$ nmcli dev wifi
IN-USE BSSID SSID MODE CHAN RATE SIGNAL BARS SECURITY
IN-USE BSSID SSID MODE CHAN RATE SIGNAL BARS SECURITY
IN-USE BSSID SSID MODE CHAN RATE SIGNAL BARS SECURITY

(thomas@dk-kali) [~]
$ nmcli dev wifi
IN-USE BSSID SSID MODE CHAN RATE SIGNAL BARS SECURITY
IN-USE E2:62:43:61:50:3D WiFi-Coruscant Infra 11 0 Mbit/s 100 ███ SECURITY
IN-USE E2:62:43:61:50:3D WiFi-Coruscant Infra 11 0 Mbit/s 100 ███ SECURITY

(thomas@dk-kali) [~]
```

Question 4

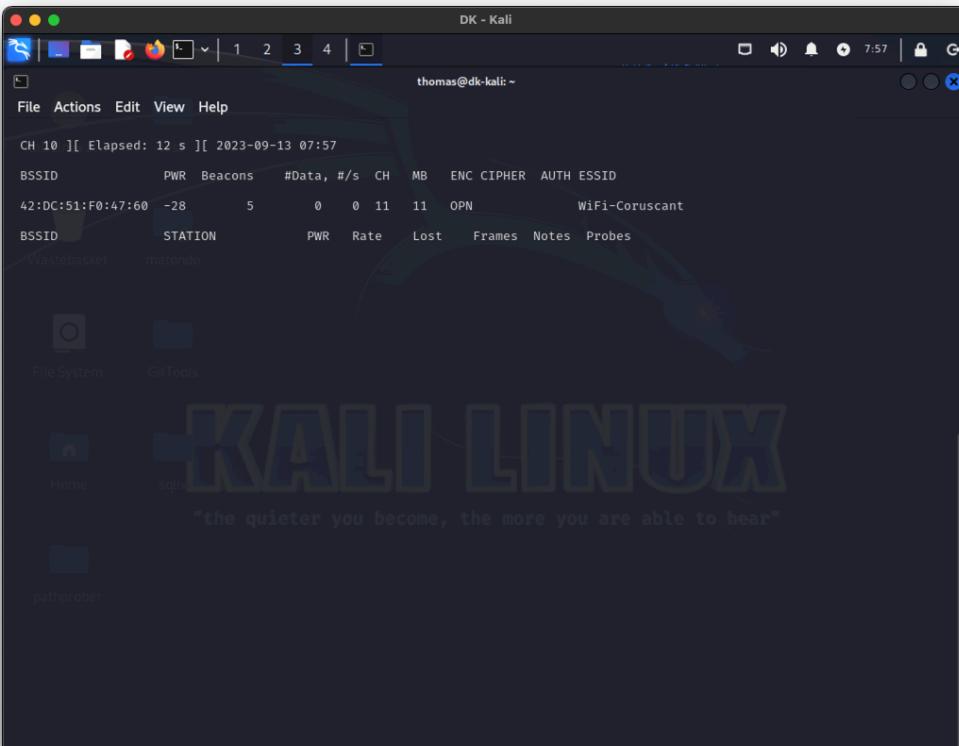
Par l'intermédiaire de `iwconfig`, nous avons paramétré notre deuxième interface wifi en mode monitor.
Statut vérifié par la commande `/sbin/iw wlan2 info`.



```
DK - Kali
File Actions Edit View Help
(thomas@dk-kali) [~]
$ sudo ifconfig wlan2 down
[sudo] password for thomas:
(thomas@dk-kali) [~]
$ sudo iwconfig wlan0 mode monitor
Error for wireless request "Set Mode" (8B06) :
        SET failed on device wlan0 ; Device or resource busy.
(thomas@dk-kali) [~]
$ sudo iwconfig wlan2 mode monitor
(thomas@dk-kali) [~]
$ sudo ifconfig wlan2 up

(thomas@dk-kali) [~]
$ /sbin/iw wlan2 info
Interface wlan2
    ifindex 8
    wdev 0x20000001
    addr 7e:79:84:e4:83:11
    type monitor
    wiphy 2           "the quieter you become, the more you are able to hear"
    channel 1 (2412 MHz), width: 20 MHz (no HT), center1: 2412 MHz
    txpower 20.00 dBm
    multicast TXQ:
        qsz-byt qsz-pkt flows drops marks overlmt hashcol tx-bytes tx-packets
        pathrober   0      0       0     0      0      0      0      0      0
(thomas@dk-kali) [~]
$
```

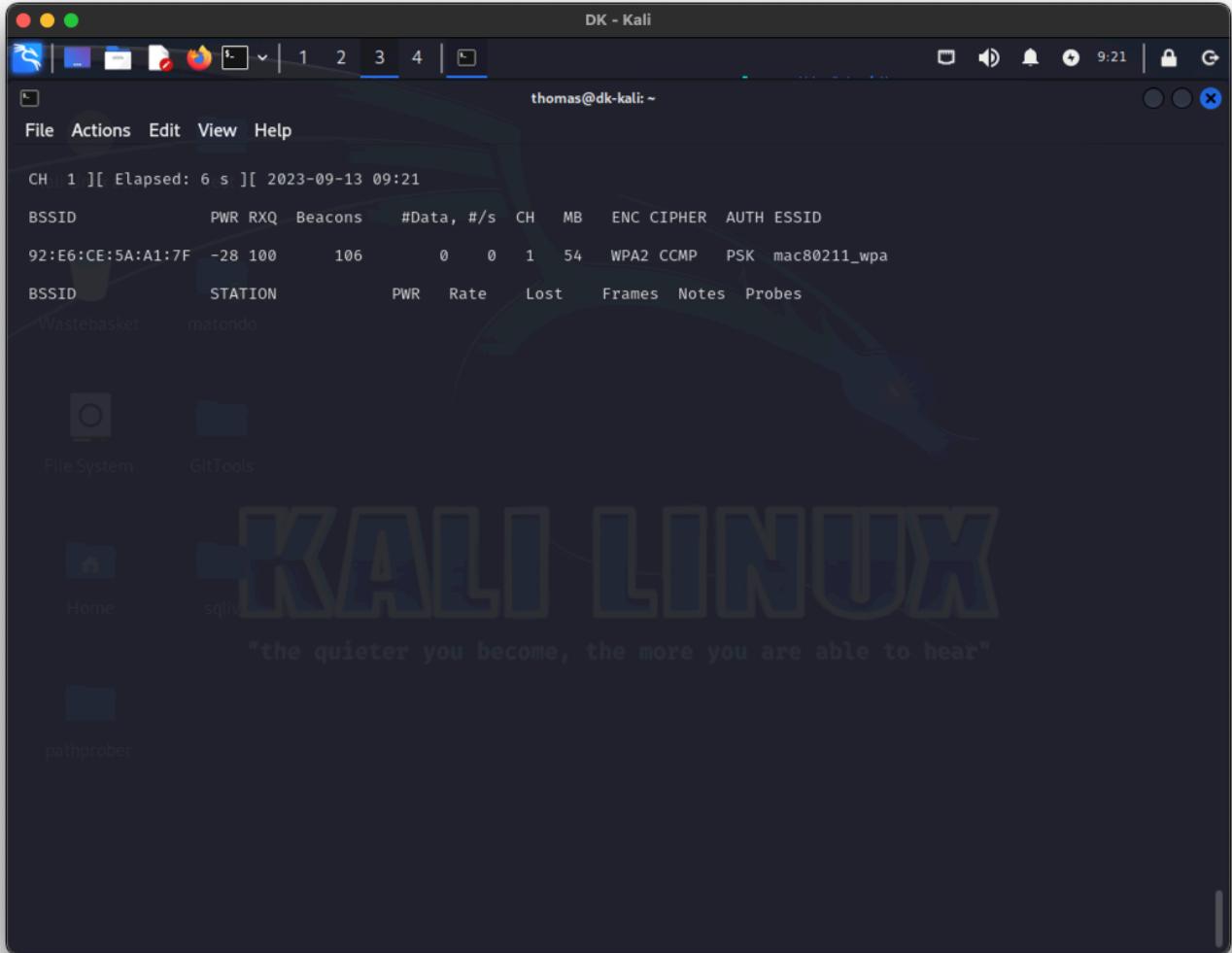
Après avoir lancé la commande `airodump-ng wlan2`, on obtient le résultat suivant, avec WiFi-Coruscant bien visible.



```
DK - Kali
File Actions Edit View Help
CH 10 ][ Elapsed: 12 s ][ 2023-09-13 07:57
BSSID          PWR  Beacons  #Data, /s CH  MB  ENC CIPHER AUTH ESSID
42:DC:51:F0:47:60 -28      5        0  0  11  11  OPEN           WiFi-Coruscant
BSSID          STATION  PWR  Rate  Lost  Frames  Notes  Probes
Wastebasket
matondo
File System
Git Tools
Home
sqlmap
pathrober
```

Question 5

Cette capture ci-dessus nous permet, dans un premier temps, de constater que notre réseau wifi n'est pas en WPA mais en Open. Nous changeons donc la configuration de hostapd.conf.



Voilà qui est mieux.

PWR : Cela représente la puissance du signal de chaque réseau Wi-Fi détecté. Plus le nombre est élevé, plus le signal est fort.

BEACONS : Il s'agit du nombre de balises de gestion (beacons) envoyées par le point d'accès. Les balises de gestion sont des paquets que les points d'accès envoient régulièrement pour annoncer leur présence et fournir des informations sur le réseau.

DATA : Cela représente le nombre de paquets de données qui ont été capturés depuis le point d'accès. Ces paquets contiennent généralement des données utiles transmises sur le réseau.

CH : Il s'agit du canal sur lequel le point d'accès fonctionne. Les réseaux sans fil peuvent utiliser différents canaux pour éviter les interférences.

MB : Cela indique la vitesse maximale de transmission des données prise en charge par le point d'accès, en mégabit par seconde (Mbps). Il peut s'agir de la vitesse de la norme Wi-Fi utilisée, comme 802.11n, 802.11ac, etc.

ENC : Cela indique si le réseau est chiffré ou non. Les valeurs courantes incluent "WEP" (Wired Equivalent Privacy), "WPA" (Wi-Fi Protected Access), "WPA2", etc.

CIPHER : Cela spécifie l'algorithme de chiffrement utilisé pour sécuriser les communications sur le réseau, tel que "TKIP", "CCMP", etc.

AUTH : Cela indique le type d'authentification utilisé pour se connecter au réseau, tel que "PSK" (Pre-Shared Key) pour un mot de passe partagé ou "EAP" (Extensible Authentication Protocol) pour une authentification plus complexe.

ESSID : Il s'agit du nom du réseau sans fil (SSID) tel qu'il est diffusé. C'est le nom que vous voyez lorsque vous recherchez des réseaux Wi-Fi disponibles.

Question 6

On a bien le 4 ways handshake.

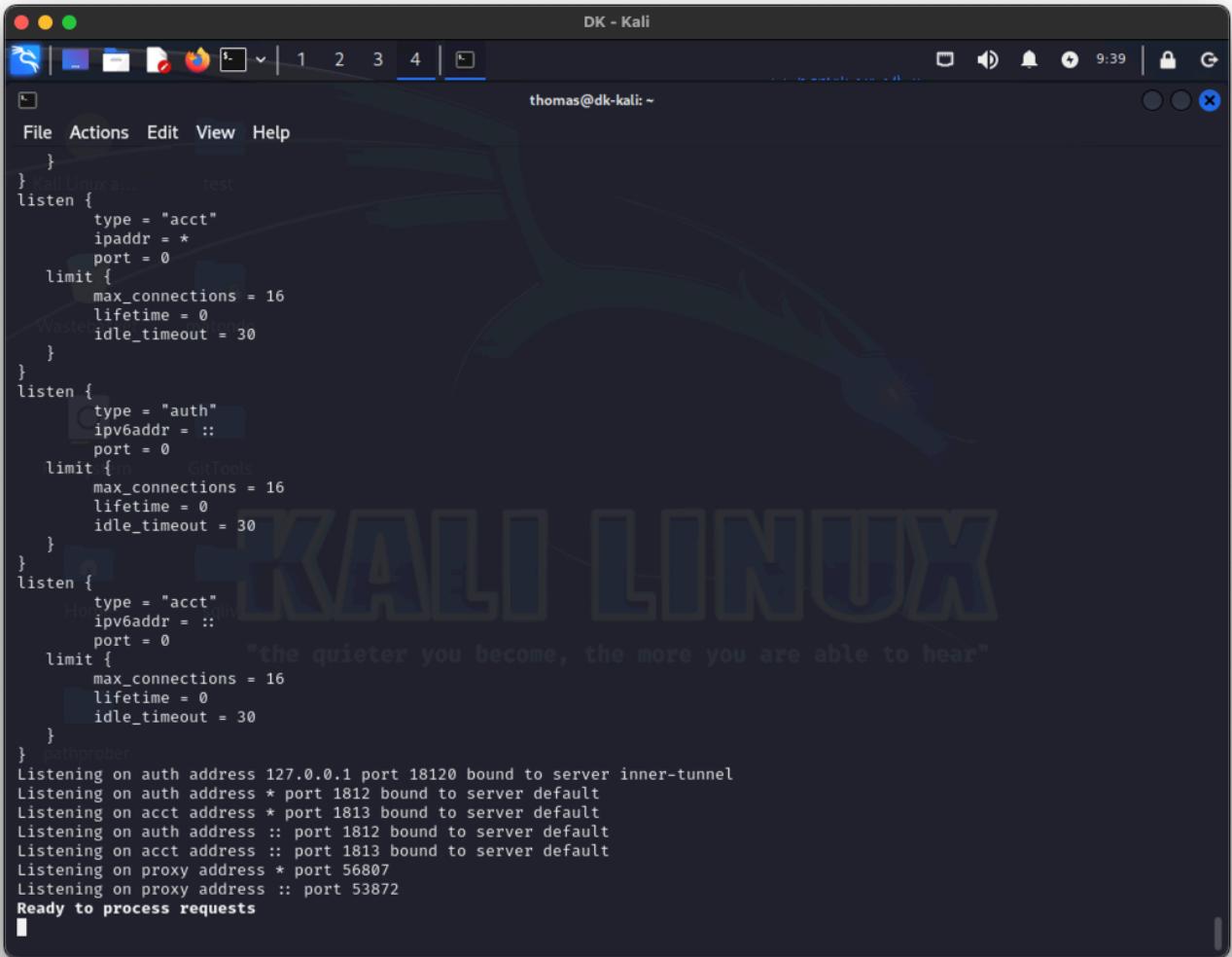
Question 7

Question 8

Partie B

Question 1

On a bien "Ready to process requests".



The screenshot shows a terminal window on a Kali Linux desktop environment. The title bar reads "DK - Kali". The terminal window displays the following content:

```
File Actions Edit View Help
}
} Kali Linux test
listen {
    type = "acct"
    ipaddr = *
    port = 0
    limit {
        max_connections = 16
        lifetime = 0
        idle_timeout = 30
    }
}
listen {
    type = "auth"
    ipv6addr = ::1
    port = 0
    limit {
        max_connections = 16
        lifetime = 0
        idle_timeout = 30
    }
}
listen {
    type = "acct"
    ipv6addr = ::1
    port = 0
    limit {
        max_connections = 16
        lifetime = 0
        idle_timeout = 30
    }
}
pathprobe
Listening on auth address 127.0.0.1 port 18120 bound to server inner-tunnel
Listening on auth address * port 1812 bound to server default
Listening on acct address * port 1813 bound to server default
Listening on auth address :: port 1812 bound to server default
Listening on acct address :: port 1813 bound to server default
Listening on proxy address * port 56807
Listening on proxy address :: port 53872
Ready to process requests
```

Le serveur écoute sur le port **18120**. L'utilisateur qui lance le serveur est **freerad**. On peut confirmer cette observation en utilisant la commande **ps**. Il est actuellement en correctement en fonction.

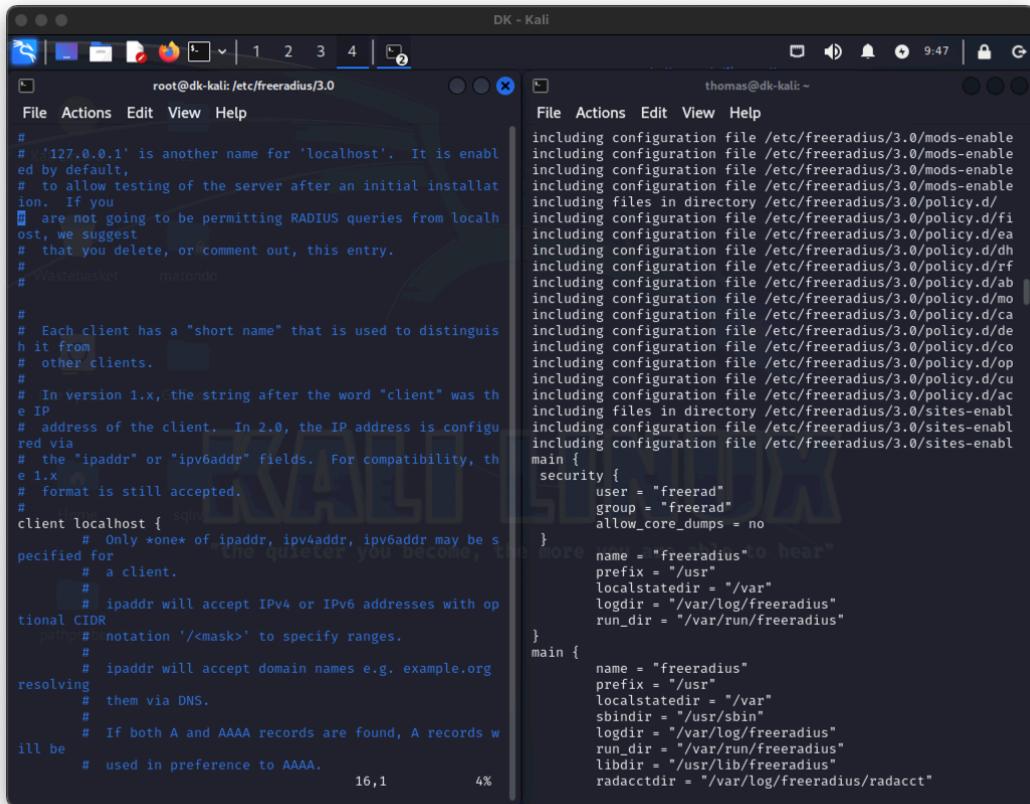
Question 2

C'est dans ce fichier que l'on déclare les différents clients (switchs par exemple) qui vont pouvoir requêter le serveur freeradius.

Le paramètre `testing123` est le mot de passe par défaut du serveur freeradius permettant de s'authentifier.

Les différents clients s'authentifient au serveur par l'intermédiaire d'un objet `client{}` créé dans le fichier `clients.conf`. Cet objet contient le nom du client, son secret et son IP. Si son IP n'est pas définie, la connexion sera refusée.

Le client `localhost` existe bel et bien.



The screenshot shows two terminal windows side-by-side. The left window, titled 'root@dk-kali:/etc/freeradius/3.0', displays the configuration file `clients.conf`. It contains several comments and one entry for the 'localhost' client, which has a secret of 'testing123'. The right window, titled 'thomas@dk-kali: ~', displays the configuration file `radacctdir`. It defines paths for various components like `freeradius`, `freerad`, and `radacct`.

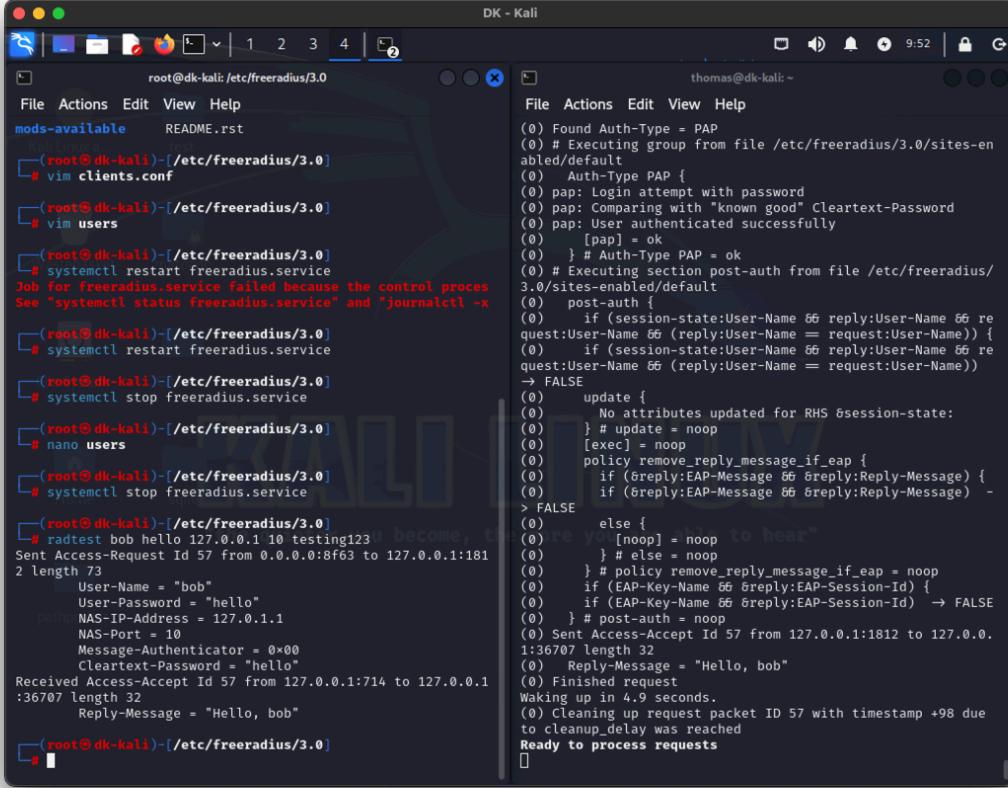
```
# '127.0.0.1' is another name for 'localhost'. It is enabled by default,
# to allow testing of the server after an initial installation. If you
# are not going to be permitting RADIUS queries from localhost, we suggest
# that you delete, or comment out, this entry.
# Wastebasket matondo

#
# Each client has a "short name" that is used to distinguish it from
# other clients.
#
# In version 1.x, the string after the word "client" was the IP
# address of the client. In 2.0, the IP address is configured via
# the "ipaddr" or "ipv6addr" fields. For compatibility, the 1.x
# format is still accepted.
#
client localhost {
    # Only *one* of ipaddr, ipv4addr, ipv6addr may be specified for
    # a client.
    #
    # ipaddr will accept IPv4 or IPv6 addresses with optional CIDR
    # notation '/<mask>' to specify ranges.
    #
    # ipaddr will accept domain names e.g. example.org resolving
    # them via DNS.
    #
    # If both A and AAAA records are found, A records will be
    # used in preference to AAAA.
}

including configuration file /etc/freeradius/3.0/mods-enable
including configuration file /etc/freeradius/3.0/mods-enable
including configuration file /etc/freeradius/3.0/mods-enable
including configuration file /etc/freeradius/3.0/mods-enable
including files in directory /etc/freeradius/3.0/policy.d/
including configuration file /etc/freeradius/3.0/policy.d/fi
including configuration file /etc/freeradius/3.0/policy.d/ea
including configuration file /etc/freeradius/3.0/policy.d/dh
including configuration file /etc/freeradius/3.0/policy.d/rf
including configuration file /etc/freeradius/3.0/policy.d/ab
including configuration file /etc/freeradius/3.0/policy.d/mo
including configuration file /etc/freeradius/3.0/policy.d/ca
including configuration file /etc/freeradius/3.0/policy.d/de
including configuration file /etc/freeradius/3.0/policy.d/co
including configuration file /etc/freeradius/3.0/policy.d/op
including configuration file /etc/freeradius/3.0/policy.d/cu
including configuration file /etc/freeradius/3.0/policy.d/ac
including files in directory /etc/freeradius/3.0/sites-enabled
including configuration file /etc/freeradius/3.0/sites-enabled
including configuration file /etc/freeradius/3.0/sites-enabled
main {
    security {
        user = "freerad"
        group = "freerad"
        allow_core_dumps = no
    }
    name = "freeradius"
    prefix = "/usr"
    localstatedir = "/var"
    sbindir = "/usr/sbin"
    logdir = "/var/log/freeradius"
    run_dir = "/var/run/freeradius"
}
main {
    name = "freeradius"
    prefix = "/usr"
    localstatedir = "/var"
    sbindir = "/usr/sbin"
    logdir = "/var/log/freeradius"
    run_dir = "/var/run/freeradius"
    libdir = "/usr/lib/freeradius"
    radacctdir = "/var/log/freeradius/radacct"
```

Question 3

On vient de se connecter avec l'utilisateur **bob** avec le serveur **freeradius** en mode debug.



The screenshot shows two terminal windows on a Kali Linux desktop. The left window, titled 'root@dk-kali:/etc/freeradius/3.0', displays the contents of the 'clients.conf' and 'users' files. The right window, titled 'thomas@dk-kali: ~', shows the output of a radius debug session. The user 'bob' is attempting to log in with the password 'hello'. The server responds with an Access-Accept message containing the text 'Hello, bob'.

```
root@dk-kali: /etc/freeradius/3.0
File Actions Edit View Help
mods-available README.rst
[root@dk-kali ~]# vim clients.conf
[root@dk-kali ~]# vim users
[root@dk-kali ~]# systemctl restart freeradius.service
Job for freeradius.service failed because the control process
See "systemctl status freeradius.service" and "journalctl -x"
[root@dk-kali ~]# systemctl restart freeradius.service
[root@dk-kali ~]# systemctl stop freeradius.service
[root@dk-kali ~]# nano users
[root@dk-kali ~]# systemctl stop freeradius.service
[root@dk-kali ~]# radtest bob hello 127.0.0.1 10 testing123 become, the
Sent Access-Request Id 57 from 0.0.0.0:8f63 to 127.0.0.1:181
2 length 73
    User-Name = "bob"
    User-Password = "hello"
    NAS-IP-Address = 127.0.1.1
    NAS-Port = 10
    Message-Authenticator = 0x00
    Cleartext-Password = "hello"
Received Access-Accept Id 57 from 127.0.0.1:714 to 127.0.0.1
:36707 length 32
    Reply-Message = "Hello, bob"

[root@dk-kali ~]# vim
thomas@dk-kali: ~
File Actions Edit View Help
(0) Found Auth-Type = PAP
(0) # Executing group from file /etc/freeradius/3.0/sites-enabled/default
(0)   Auth-Type PAP {
(0)     pap: Login attempt with password
(0)     pap: Comparing with "known good" Cleartext-Password
(0)     pap: User authenticated successfully
(0)       [pap] = ok
(0)   } # Auth-Type PAP = ok
(0) # Executing section post-auth from file /etc/freeradius/3.0/sites-enabled/default
(0)   post-auth {
(0)     if (session-state:User-Name && reply:User-Name && request:User-Name) {
(0)       if (session-state:User-Name && reply:User-Name && request:User-Name && reply:User-Name && (reply:User-Name = request:User-Name)) → FALSE
(0)         update {
(0)           No attributes updated for RHS &session-state:
(0)         } # update = noop
(0)         [exec] = noop
(0)         policy remove_reply_message_if_eap {
(0)           if (&reply:EAP-Message && &reply:Reply-Message) {
(0)             if (&reply:EAP-Message && &reply:Reply-Message) -> FALSE
(0)               else {
(0)                 [noop] = noop
(0)               } # else = noop
(0)             } # policy remove_reply_message_if_eap = noop
(0)           if (EAP-Key-Name && &reply:EAP-Session-Id) {
(0)             if (EAP-Key-Name && &reply:EAP-Session-Id) → FALSE
(0)           } # post-auth = noop
(0)         Sent Access-Accept Id 57 from 127.0.0.1:1812 to 127.0.1:36707 length 32
(0)         Reply-Message = "Hello, bob"
(0)       Finished request
(0)       Waking up in 4.9 seconds.
(0)       Cleaning up request packet ID 57 with timestamp +98 due to cleanup_delay was reached
Ready to process requests
```

D'après la documentation, il est possible d'utiliser un fichier avec un module ayant une syntaxe similaire à celle de `/etc/passwd`. Il ne semble donc pas possible d'utiliser directement le fichier en tant que tel.

rlm_passwd

Synopsis

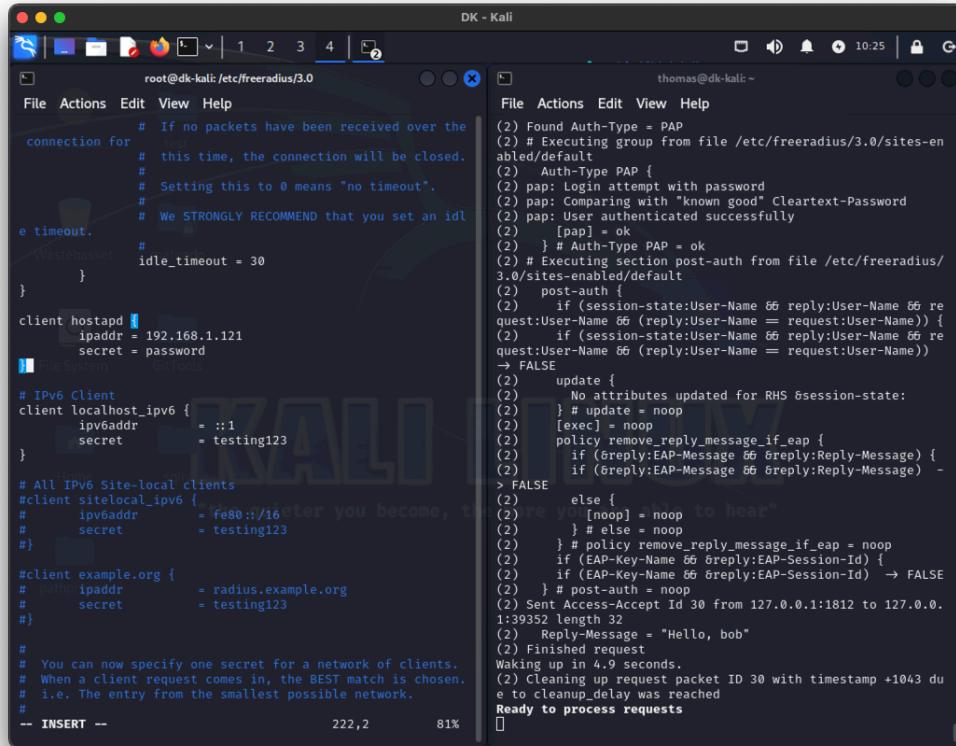
The `passwd` module reads and caches line-oriented files that are in a format similar to '`/etc/passwd`'. It assumes that each line is composed of a series of records, separated by a delimiter. The records are read from the file, cached, and then placed into one of the packet

The `passwd` module allows for authorization via any `passwd`-like file and for extraction of any attributes from these files. See the `smbpasswd` and `etc_group` files for more examples.

Question 4

Ajoutons un nouveau client, permettant de se connecter depuis ce dernier au lieu de localhost.

Notre nouveau client ci-dessous.



```

root@dk-kali: /etc/freeradius/3.0
File Actions Edit View Help
# If no packets have been received over the
connection for
# this time, the connection will be closed.
#
# Setting this to 0 means "no timeout".
#
# We STRONGLY RECOMMEND that you set an idle
timeout.
idle_timeout = 30
}

client hostapd {
    ipaddr = 192.168.1.121
    secret = password
}

# IPv6 Client
client localhost_ipv6 {
    ipv6addr = ::1
    secret = testing123
}

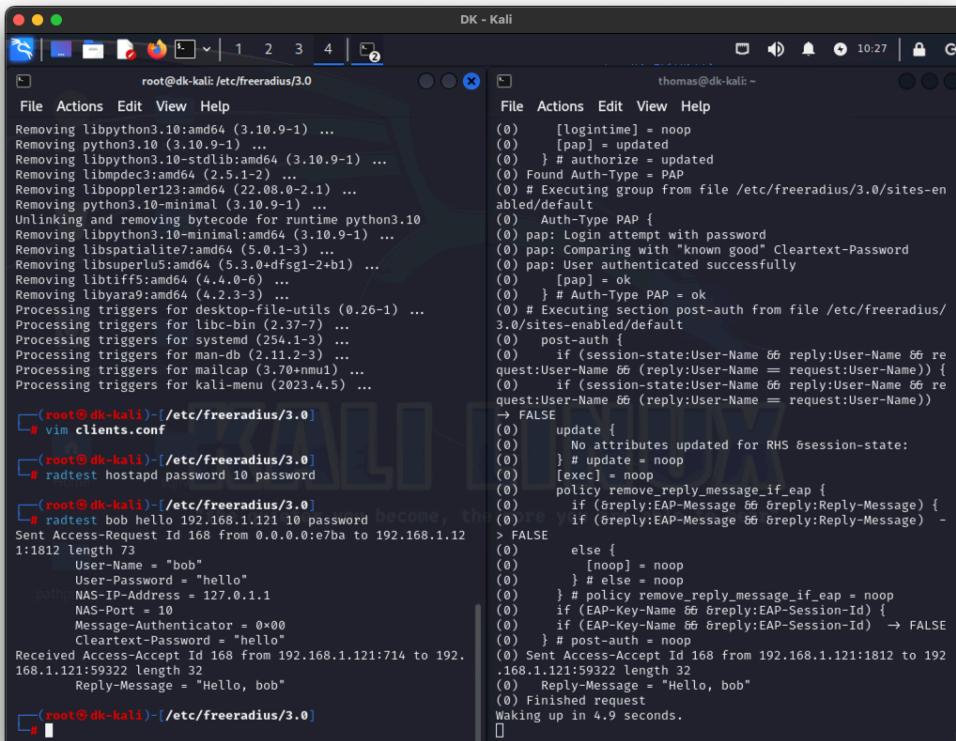
# All IPv6 Site-local clients
client sitelocal_ipv6 {
    ipv6addr = fe80::/16
    secret = testing123
}

client example.org {
    ipaddr = radius.example.org
    secret = testing123
}

#
# You can now specify one secret for a network of clients.
# When a client request comes in, the BEST match is chosen.
# i.e. The entry from the smallest possible network.
#
-- INSERT --

```

Le résultat, succès.



```

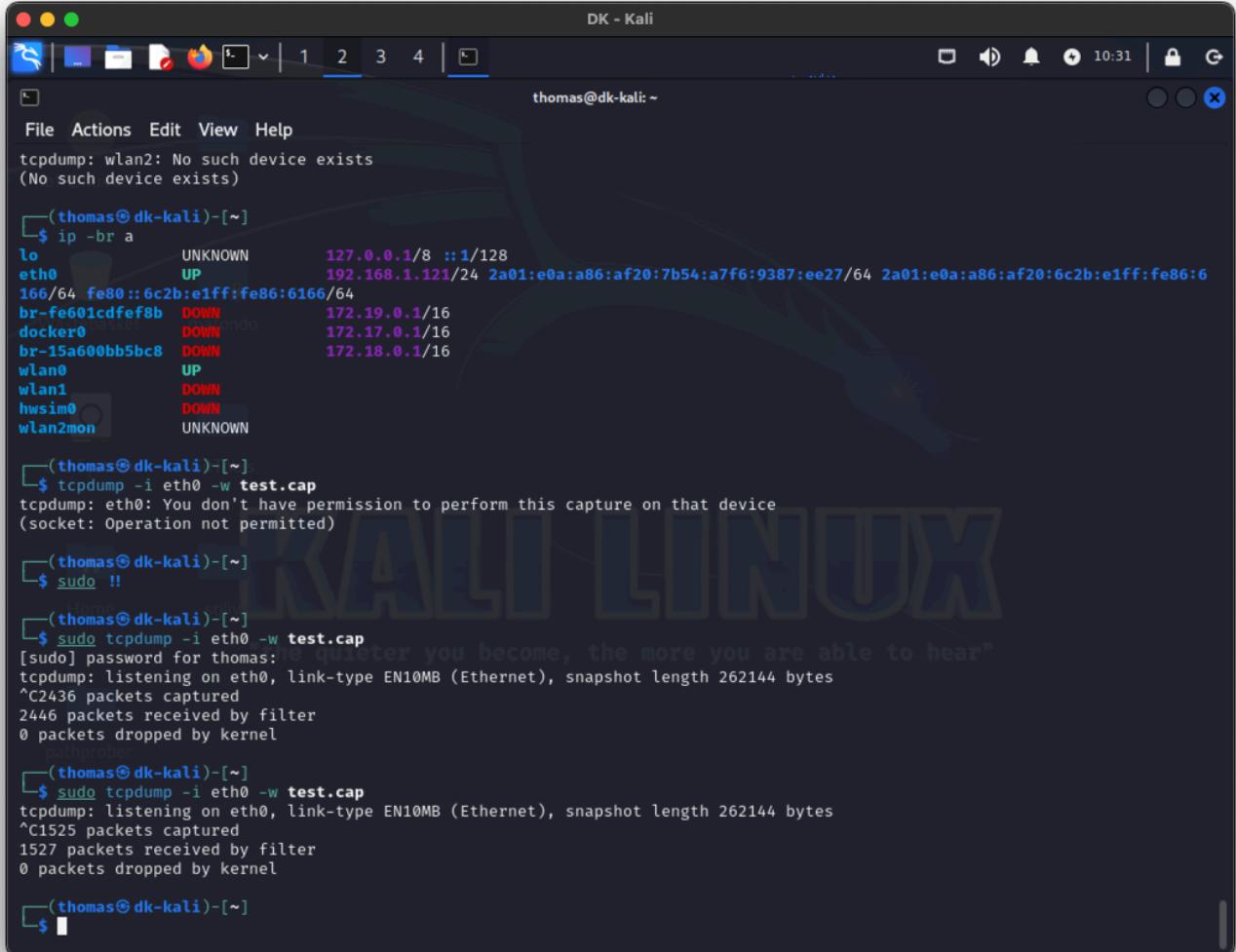
root@dk-kali: /etc/freeradius/3.0
File Actions Edit View Help
Removing libpython3.10:amd64 (3.10.9-1) ...
Removing python3.10 (3.10.9-1) ...
Removing libpython3.10-stdlib:amd64 (3.10.9-1) ...
Removing libmpdec3:amd64 (2.5.1-2) ...
Removing libpoppler123:amd64 (22.08.0-2.1) ...
Removing python3.10-minimal (3.10.9-1) ...
Unlinking and removing bytecode for runtime python3.10
Removing libpython3.10-minimal:amd64 (3.10.9-1) ...
Removing libspatialite7:amd64 (5.0.1-3) ...
Removing libspatialite5:amd64 (5.3.0+dfsg1-2+b1) ...
Removing libtiff5:amd64 (4.4.0-6) ...
Removing libyaml9:amd64 (4.2.3-3) ...
Processing triggers for desktop-file-utils (0.26-1) ...
Processing triggers for libc-bin (2.37-7) ...
Processing triggers for systemd (254.1-3) ...
Processing triggers for man-db (2.11.2-3) ...
Processing triggers for mailcap (3.70+nmu1) ...
Processing triggers for kali-menu (2023.4.5) ...
Processing triggers for kali-menu (2023.4.5) ...

[root@dk-kali] - /etc/freeradius/3.0
# vim clients.conf
[root@dk-kali] - /etc/freeradius/3.0
# radtest hostapd password 10 password
[root@dk-kali] - /etc/freeradius/3.0
# radtest bob hello 192.168.1.121 10 password
Sent Access-Request Id 168 from 0.0.0.0:e7ba to 192.168.1.12
1:1812 length 73
    User-Name = "bob"
    User-Password = "hello"
    NAS-IP-Address = 127.0.1.1
    NAS-Port = 10
    Message-Authenticator = 0x00
    Cleartext-Password = "Hello"
Received Access-Accept Id 168 from 192.168.1.121:714 to 192.
168.1.121:59322 length 32
    Reply-Message = "Hello, bob"

[root@dk-kali] - /etc/freeradius/3.0
#

```

La capture, avec la commande tcpdump ci-dessous :



The screenshot shows a terminal window on a Kali Linux desktop environment. The title bar says "DK - Kali". The terminal window has a dark background with light-colored text. It displays the following session:

```
tcpdump: wlan2: No such device exists
(No such device exists)

(thomas@dk-kali)-[~]
$ ip -br a
lo      UNKNOWN      127.0.0.1/8 ::1/128
eth0     UP          192.168.1.121/24 2a01:e0a:a86:af20:7b54:a7f6:9387:ee27/64 2a01:e0a:a86:af20:6c2b:e1ff:fe86:6
166/64 fe80::6c2b:e1ff:fe86:6166/64
br-fe601cddef8b DOWN        172.19.0.1/16
docker0   DOWN        172.17.0.1/16
br-15a600bb5bc8 DOWN        172.18.0.1/16
wlan0    UP          127.0.0.1/8 ::1/128
wlan1    DOWN
hwsim0   DOWN
wlan2mon UNKNOWN

(thomas@dk-kali)-[~]
$ tcpdump -i eth0 -w test.cap
tcpdump: eth0: You don't have permission to perform this capture on that device
(socket: Operation not permitted)

(thomas@dk-kali)-[~]
$ sudo !!
[dropped]                                          "the quieter you become, the more you are able to hear"
(thomas@dk-kali)-[~]
$ sudo tcpdump -i eth0 -w test.cap
[sudo] password for thomas:
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C2436 packets captured
2446 packets received by filter
0 packets dropped by kernel

(thomas@dk-kali)-[~]
$ sudo tcpdump -i eth0 -w test.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C1525 packets captured
1527 packets received by filter
0 packets dropped by kernel

(thomas@dk-kali)-[~]
$
```

Partie C

Question 1

Noter que le client hostapd a déjà été créé dans les étapes précédentes, en ayant précisé son IP. Les connexions ne seront donc pas bloquées.

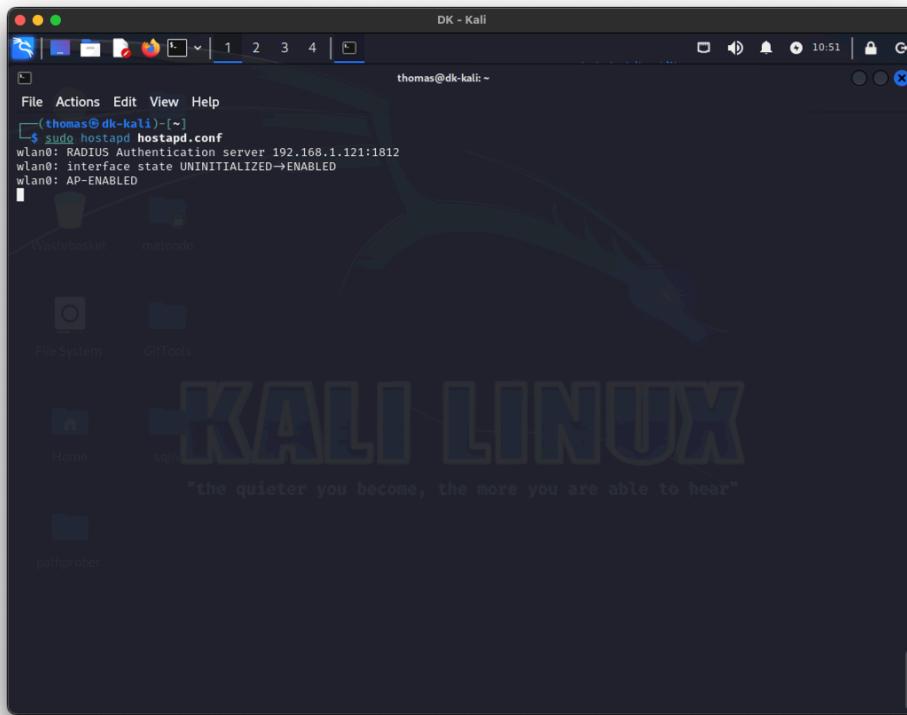


```
DK - Kali
File Actions Edit View Help
interface=wlan0
driver=nl80211
hw_mode=g
channel=6
ssid=RSD

ieee8021x=
wpa=2
wpa_key_mgmt=WPA-EAP
wpa_pairwise=CCMP
auth_algs=1
auth_server_addr=192.168.1.121
auth_server_port=1812
auth_server_shared_secret=password

File System GitTools
Home
pathrober
-- INSERT --
thomas@dk-kali: ~
```

On peut constater qu'il y a bien un serveur d'authentification qui est sur l'ip **192.168.1.121**.



```
DK - Kali
File Actions Edit View Help
(thomas@dk-kali: ~) $ sudo hostapd hostapd.conf
wlan0: RADIUS Authentication server 192.168.1.121:1812
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED

File System GitTools
Home
pathrober
-- INSERT --
thomas@dk-kali: ~
```

Question 2

Configuration du fichier `wpa2_supplicant.conf` ci-dessous.

The screenshot shows a Kali Linux desktop environment. The terminal window at the top has the title "DK - Kali" and the command "thomas@dk-kali: ~". The terminal content is a configuration file for wpa_supplicant:

```
network={  
    ssid="RSID"  
    key_mgmt=WPA-EAP  
    eap=PEAP  
    identity="bob"  
    password="hello"  
    phase1="peaplabel=0"  
    phase2="auth-MSCHEPv2"  
}  
Wlan0
```

The desktop interface includes a file browser with a sidebar showing "File System" and "GitTools", and a dock with icons for Home, SQLMap, and Pathower.

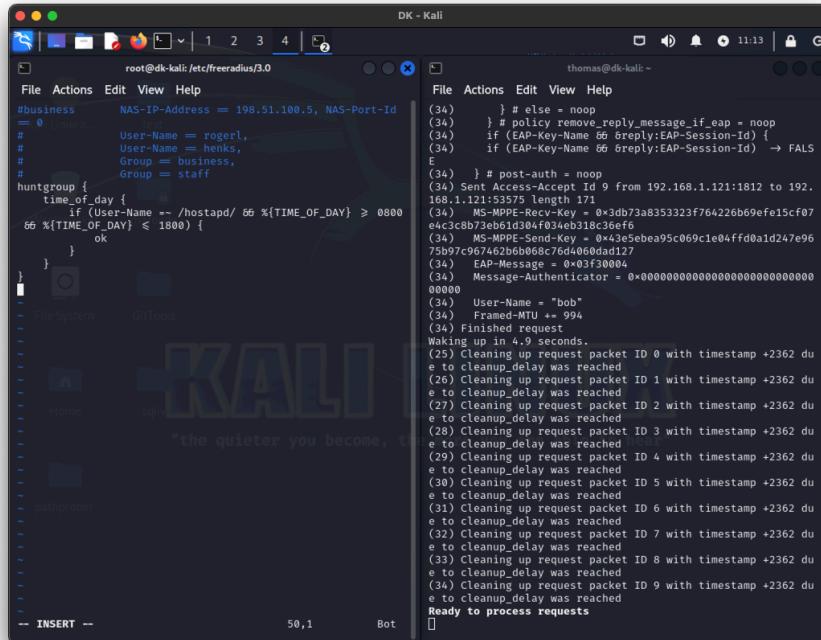
Après avoir démarré `hostapd`, on exécute `wpa_supplicant`, la connexion fonctionne !

Question 3

Le terme AAA signifie Authentication, Authorization, Accounting.

Voici les fichiers de configurations, ci-dessous, pour n'autoriser les connexions que pendant certaines heures (8h à 18h).

HuntGroup `time_of_day` :



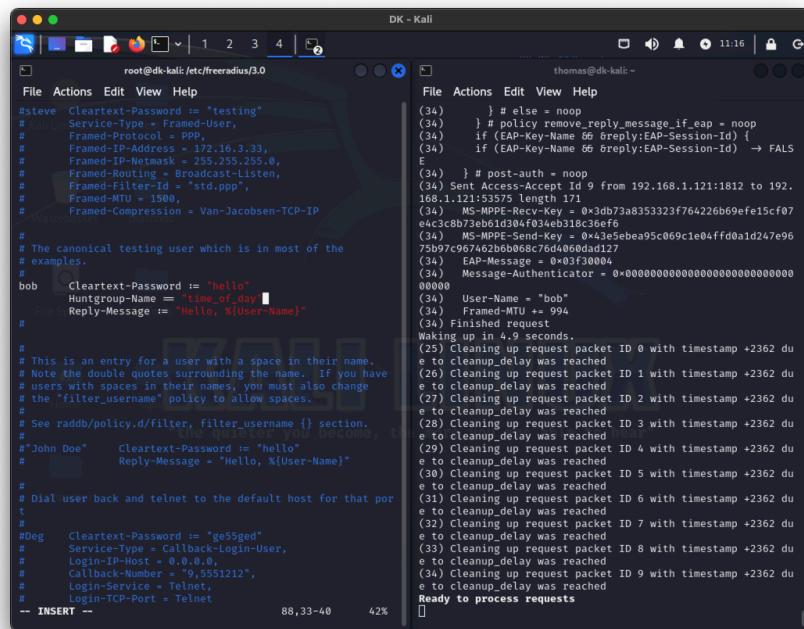
The screenshot shows two terminal windows side-by-side. The left window is titled 'DK - Kali' and shows the configuration file for the 'radiusd' daemon. It contains a 'business' section and a 'huntgroup' section named 'time_of_day'. The 'time_of_day' section includes a condition where if the User-Name is 'rogerl' and the time is less than or equal to 1800, the user is granted access ('ok'). The right window is also titled 'DK - Kali' and shows the logs from the 'radiusd' daemon. The logs detail the processing of multiple requests from the IP 192.168.1.121 over port 1812. The logs show the daemon sending Access-Accept messages to the client, which then sends EAP messages. The logs indicate that the daemon is ready to process requests after handling these connections.

```

root@dk-kali:/etc/freeradius/3.0
File Actions Edit View Help
#business      NAS-IP-Address = 198.51.100.5, NAS-Port-Id
= 0
#       User-Name = rogerl,
#       User-Name = henks,
#       Group = business,
#       Group = staff
huntgroup {
    time_of_day {
        if ((User-Name == /hostapd/ && %TIME_OF_DAY% >= 0800
        && %TIME_OF_DAY% <= 1800) {
            ok
        }
    }
}
File Actions Edit View Help
(34) } # else = noop
(34) } # policy remove_reply_message_if_eap = noop
(34) if (EAP-Key-Name && $reply:EAP-Session-Id) {
(34) if (EAP-Key-Name && $reply:EAP-Session-Id) → FALSE
(34) } # post-auth = noop
(34) Sent Access-Accept Id 9 from 192.168.1.121:1812 to 192.
168.1.121:53575 length 171
(34) MS-MPPE-Recv-Key = 0x3db73a8353323f764226b69efe15cf07
e4c3c8b73eb01d304f034eb318c36ef6
(34) MS-MPPE-Send-Key = 0x43e5bea95c069c1e04ffd0a1d247e96
7b997c96746c62b686c76d4060dad127
(34) EAP-Message = 0x03f30004
(34) Message-Authenticator = 0x00000000000000000000000000000000
00000000000000000000000000000000
(34) User-Name = "bob"
(34) Framed-MTU += 994
(34) Finished request
Waking up in 4.9 seconds.
Cleaning up request packet ID 0 with timestamp +2362 due
to cleanup_delay was reached
(26) Cleaning up request packet ID 1 with timestamp +2362 du
e to cleanup_delay was reached
(27) Cleaning up request packet ID 2 with timestamp +2362 du
e to cleanup_delay was reached
(28) Cleaning up request packet ID 3 with timestamp +2362 du
e to cleanup_delay was reached
(29) Cleaning up request packet ID 4 with timestamp +2362 du
e to cleanup_delay was reached
(30) Cleaning up request packet ID 5 with timestamp +2362 du
e to cleanup_delay was reached
(31) Cleaning up request packet ID 6 with timestamp +2362 du
e to cleanup_delay was reached
(32) Cleaning up request packet ID 7 with timestamp +2362 du
e to cleanup_delay was reached
(33) Cleaning up request packet ID 8 with timestamp +2362 du
e to cleanup_delay was reached
(34) Cleaning up request packet ID 9 with timestamp +2362 du
e to cleanup_delay was reached
Ready to process requests

```

Fichier `users` : on spécifie le group auquel appartient l'utilisateur `bob`.



The screenshot shows two terminal windows side-by-side. The left window is titled 'DK - Kali' and shows the configuration file for the 'radiusd' daemon. It includes sections for 'steve' and 'bob'. The 'bob' section specifies a 'Cleartext-Password' of 'testing', a 'Huntgroup-Name' of 'time_of_day', and a 'Reply-Message' of 'Hello, %(User-Name)'. The right window is also titled 'DK - Kali' and shows the logs from the 'radiusd' daemon. The logs detail the processing of multiple requests from the IP 192.168.1.121 over port 1812. The logs show the daemon sending Access-Accept messages to the client, which then sends EAP messages. The logs indicate that the daemon is ready to process requests after handling these connections.

```

root@dk-kali:/etc/freeradius/3.0
File Actions Edit View Help
#steve  Cleartext-Password := "testing"
#       Service-Type = Framed-User,
#       Framed-Protocol = PPP
#       Framed-IP-Address = 172.16.3.33,
#       Framed-IP-Network = 35.255.255.0,
#       Framed-Routing = Broadcast-Listen,
#       Framed-Filter-Id = "std.ppp",
#       Framed-MTU = 1500,
#       Framed-Compression = Van-Jacobsen-TCP-IP
#
# The canonical testing user which is in most of the
# examples.
#
bob   Cleartext-Password := "testing"
Huntgroup-Name = "time_of_day"
Reply-Message := "Hello, %(User-Name)"
#
# This is an entry for a user with a space in their name.
# Note the double quotes surrounding the name. If you have
# users with spaces in their names, you must also change
# the 'filter-username' policy to allow spaces.
#
# See raddb/policy.d/filter, filter_username {} section,
# "John Doe"  Cleartext-Password := "hello"
#             Reply-Message = "Hello, %(User-Name)"
#
# Dial user back and telnet to the default host for that por
t
#
#Deg   Cleartext-Password := "ge55ged"
#       Service-Type = Callback-Login-User,
#       Login-IP-Host = 0.0.0.0,
#       Callback-Number = "+9,5551212",
#       Login-Service = Telnet,
#       Login-TCP-Port = Telnet
-- INSERT --
File Actions Edit View Help
(34) } # else = noop
(34) } # policy remove_reply_message_if_eap = noop
(34) if (EAP-Key-Name && $reply:EAP-Session-Id) {
(34) if (EAP-Key-Name && $reply:EAP-Session-Id) → FALSE
(34) } # post-auth = noop
(34) Sent Access-Accept Id 9 from 192.168.1.121:1812 to 192.
168.1.121:53575 length 171
(34) MS-MPPE-Recv-Key = 0x3db73a8353323f764226b69efe15cf07
e4c3c8b73eb01d304f034eb318c36ef6
(34) MS-MPPE-Send-Key = 0x43e5bea95c069c1e04ffd0a1d247e96
7b997c96746c62b686c76d4060dad127
(34) EAP-Message = 0x03f30004
(34) Message-Authenticator = 0x00000000000000000000000000000000
00000000000000000000000000000000
(34) User-Name = "bob"
(34) Framed-MTU += 994
(34) Finished request
Waking up in 4.9 seconds.
Cleaning up request packet ID 0 with timestamp +2362 due
to cleanup_delay was reached
(26) Cleaning up request packet ID 1 with timestamp +2362 du
e to cleanup_delay was reached
(27) Cleaning up request packet ID 2 with timestamp +2362 du
e to cleanup_delay was reached
(28) Cleaning up request packet ID 3 with timestamp +2362 du
e to cleanup_delay was reached
(29) Cleaning up request packet ID 4 with timestamp +2362 du
e to cleanup_delay was reached
(30) Cleaning up request packet ID 5 with timestamp +2362 du
e to cleanup_delay was reached
(31) Cleaning up request packet ID 6 with timestamp +2362 du
e to cleanup_delay was reached
(32) Cleaning up request packet ID 7 with timestamp +2362 du
e to cleanup_delay was reached
(33) Cleaning up request packet ID 8 with timestamp +2362 du
e to cleanup_delay was reached
(34) Cleaning up request packet ID 9 with timestamp +2362 du
e to cleanup_delay was reached
Ready to process requests

```

Question 4

Après avoir modifié la version de `wpa=3` dans le fichier `hostapd.conf`, on tente à nouveau de se connecter, la connexion semble prendre sensiblement plus de temps.

Question 5

WPA3 est une meilleure sécurité Wi-Fi que WPA2. Il rend plus difficile pour les pirates de deviner les mots de passe, offre un cryptage plus fort et protège mieux les réseaux sans fil. De plus, il fonctionne bien avec les appareils plus anciens en les gardant sécurisés.