PRÁCTICA #2

DATOS DEL ESTUDIANTE

Nombre y apellidos	**María Victoria Maldonado Bao**
E-mail	mvictoriamb0425@uma.es
Grupo (A / B)	A
Fecha	10/2025

EJERCICIO 1. FUERZA BRUTA Y CONTRAMEDIDAS

1.1: FUERZA BRUTA Y LISTAS NEGRAS

```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -l admin -P fasttrack.txt -f 192.168.122.1 ssh
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for ill
ses (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-25 10:43:41
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 262 login tries (l:1/p:262), ~17 tries per task
[DATA] attacking ssh://192.168.122.1:22/
[STATUS] 172.00 tries/min, 172 tries in 00:01h, 94 to do in 00:01h, 12 active
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-25 10:45:16
```

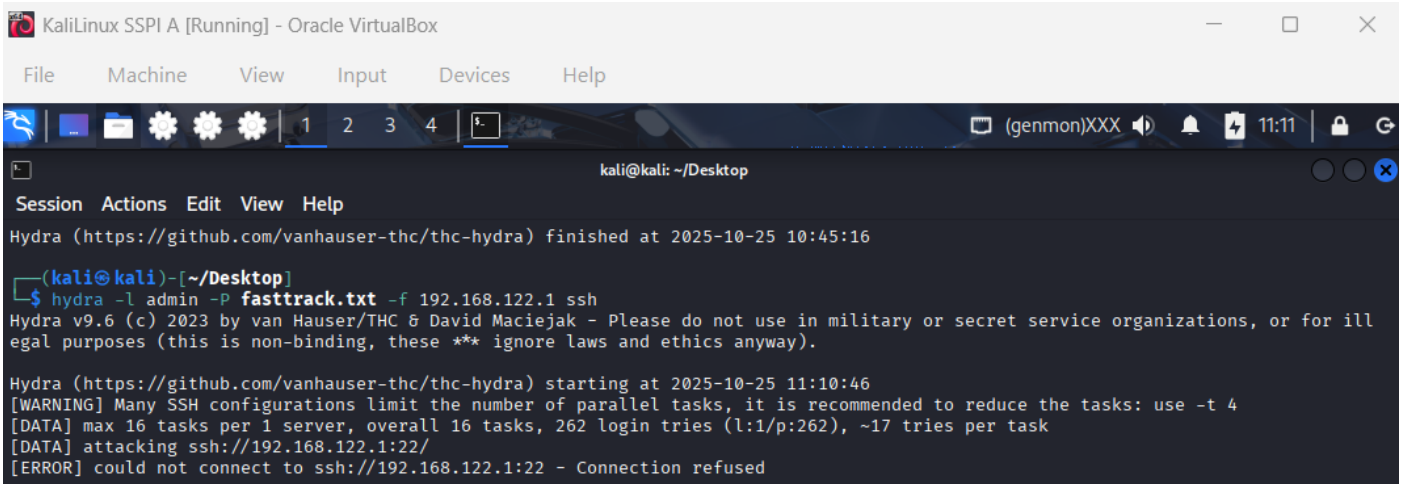
No consiguió adivinar el usuario-contraseña del router (no estaba en el archivo fasttrack.txt).

Ahora crearé 3 listas: 1-whitelist (IPs que nunca se bloquearán), 2-suspects(sospechosos pero aún no bloqueados), y 3-blocked(IPs bloqueadas sin derecho a acceso).

```
[admin@R1] > ip firewall address-list add list=whitelist address=2.2.2.254 comment="es un ordenador de confianza"
[admin@R1] > clear
bad command name clear (line 1 column 1)
[admin@R1] > ip firewall filter add chain=input protocol=tcp dst-port=22 connection-state=new src-address-list=!whitelist
 action=add-src-to-address-list address-list=suspects address-list-timeout=1h comment="SSH: Sospechosos"
[admin@R1] > ip firewall filter add chain=input src-address-list=suspects protocol=tcp dst-port=22 action=add-src-to-addr
ess-list address-list=blocked address-list-timeout=24h comment="Bloquear tras intentos repetidos"
[admin@R1] > ip firewall filter add chain=input src-address-list=blocked action=drop comment="Omitir paquetes de bloquead
os"
[admin@R1] > ip firewall address-list print
Columns: LIST, ADDRESS, CREATION-TIME
# LIST       ADDRESS    CREATION-TIME
;;; es un ordenador de confianza
0 whitelist  2.2.2.254  2025-10-25 14:59:03
[admin@R1] >
```

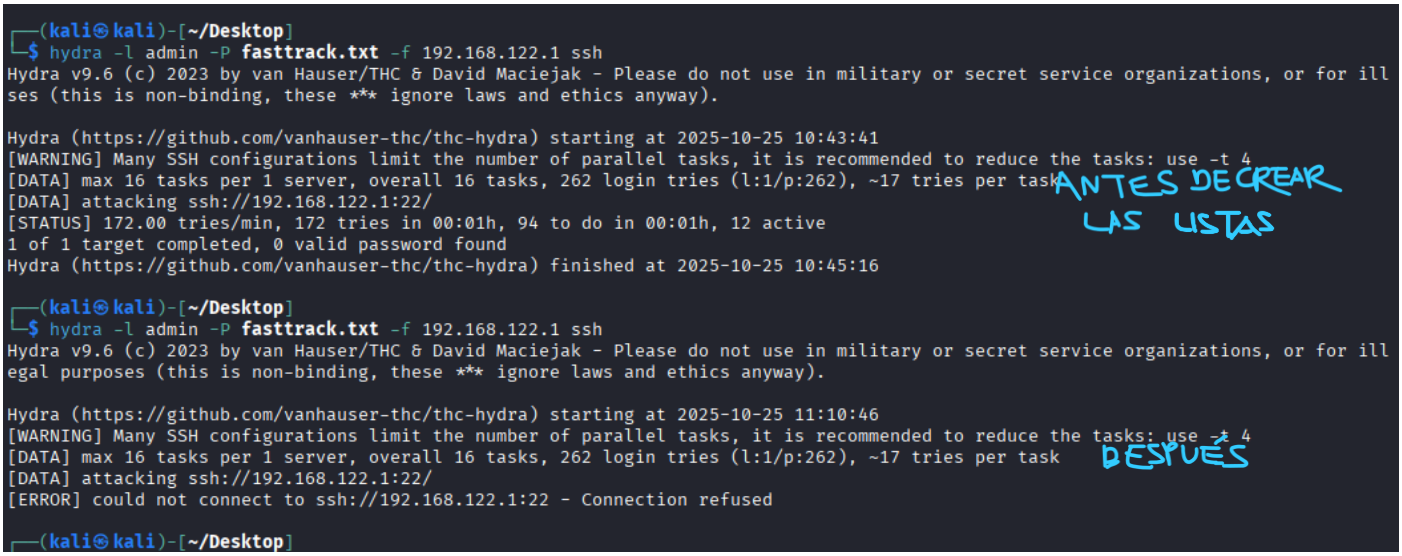
Añado el KaliLinux B (IP 2.2.2.254) en la whitelist (para comprobar luego que no bloqueará esta IP a pesar de intentar hacer luego un Brute Force Attack ya que está en la whitelist). Cualquier nuevo intento de conexión será considerado sospechoso, y tras varios intentos repetidos de inicio de sesión pasará a estar bloqueada esa IP. Los paquetes de IPs bloqueadas no serán recibidos (línea en gris).

Ahora haré el ataque desde la IP de la Whitelist y desde la otra IP que no está en la whitelist, veremos que será bloqueada por el router mientras que la IP de la whitelist seguirá pudiendo enviar paquetes.

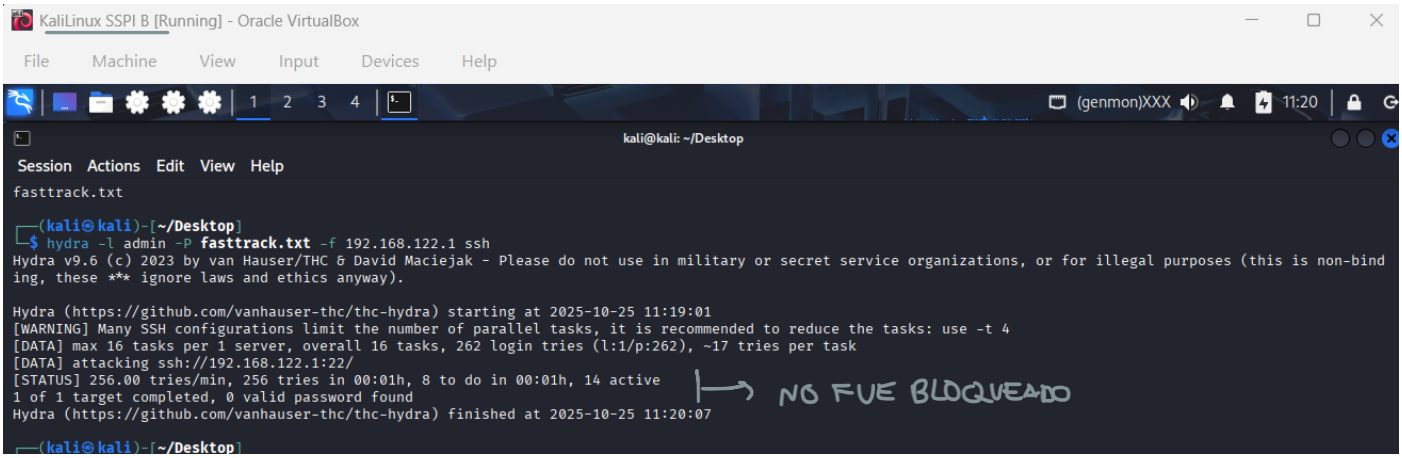


El KaliLinux A, cuya IP no estaba en la whitelist, fue bloqueada por el router y no se le permite conectarse.

--



Sin embargo, veremos que esto no afecta al KaliLinux B cuya IP está en la whitelist (no es bloqueado):



Ahora lo hacemos pero en vez de con listas con la técnica del jump-target y return (nos dará el mismo resultado):

```
[admin@R1] > ip firewall filter add chain=input action=jump jump-target=bruteforce-check comment="Saltar a la revisin de
fuerza bruta"
[admin@R1] > add chain=bruteforce-check protocol=tcp dst-port=22 connection-state=new src-address-list=!whitelist action=
add-src-to-address-list address-list=suspects address-list-timeout=1h comment="Sospechoso"
bad command name add (line 1 column 1)
[admin@R1] > ip firewall filter add chain=bruteforce-check protocol=tcp dst-port=22 connection-state=new src-address-list
=!whitelist action=add-src-to-address-list address-list=suspects address-list-timeout=1h comment="Sospechoso"
[admin@R1] > ip firewall filter add chain=bruteforce-check src-address-list=suspects action=add-src-to-address-list addre
ss-list=blocked address-list-timeout=24h comment="Bloquear si es muy sospechoso"
[admin@R1] > ip firewall filter add chain=bruteforce-check src-address-list=blocked action=drop comment="Ignorar paquetes
 de IPs bloqueadas"
[admin@R1] > ip firewall filter add chain=bruteforce-check action=return comment="Todo ok"
[admin@R1] >
```

(Los resultados tras volver a realizar ambos ataques son los mismos que con las listas anteriores).

/export de las reglas (están "duplicadas" porque están con ambos métodos):

```
/ip firewall filter
add action=add-src-to-address-list address-list=suspects address-list-timeout=1h chain=input comment=\
    "SSH: Sospechosos" connection-state=new dst-port=22 protocol=tcp src-address-list=!whitelist
add action=add-src-to-address-list address-list=blocked address-list-timeout=1d chain=input comment=\
    "Bloquear tras intentos repetidos" dst-port=22 protocol=tcp src-address-list=suspects
add action=drop chain=input comment="Omitir paquetes de bloqueados" src-address-list=blocked
add action=jump chain=input comment="Saltar a la revisin de fuerza bruta" jump-target=bruteforce-check
add action=add-src-to-address-list address-list=suspects address-list-timeout=1h chain=bruteforce-check comment=\
    Sospechoso connection-state=new dst-port=22 protocol=tcp src-address-list=!whitelist
add action=add-src-to-address-list address-list=blocked address-list-timeout=1d chain=bruteforce-check comment=\
    "Bloquear si es muy sospechoso" src-address-list=suspects
add action=drop chain=bruteforce-check comment="Ignorar paquetes de IPs bloqueadas" src-address-list=blocked
add action=return chain=bruteforce-check comment="Todo ok"
```

1.2: PORT-KNOCKING (GOLPEO DE PUERTO)

```
[admin@R1] > ip firewall filter add chain=input protocol=tcp dst-port=831 action=add-src-to-address-list address-list=kno
ck1 address-list-timeout=15s comment="portknock1"
[admin@R1] > ip firewall filter add chain=input protocol=tcp dst-port=841 action=add-src-to-address-list address-list=kno
ck2 address-list-timeout=15s comment="portknock2"
[admin@R1] > ip firewall filter add chain=input protocol=tcp dst-port=851 action=add-src-to-address-list address-list=ssh
_access address-list-timeout=1m comment="Permitir SSH tras portknock"
[admin@R1] > ip firewall filter add chain=input protocol=tcp dst-port=22 action=drop comment="drop por defecto"
```

En la primera línea establecemos el primer paso de la secuencia del portknock (831->841->851), que si luego haces knock al segundo puerto correctamente (841) pasa a la siguiente línea que si haces knock al puerto 851 ya te permite acceso a SSH. Si fallas el portknock o haces knock al puerto ssh sin el portknock, se "droppea" el paquete por defecto.

```
┌──(kali㉿kali)-[~]
└─$ knock 192.168.122.1 831 841 851

┌──(kali㉿kali)-[~]
└─$ ssh admin@192.168.122.1
```

Esto es lo que pasa cuando falla el portknocking:

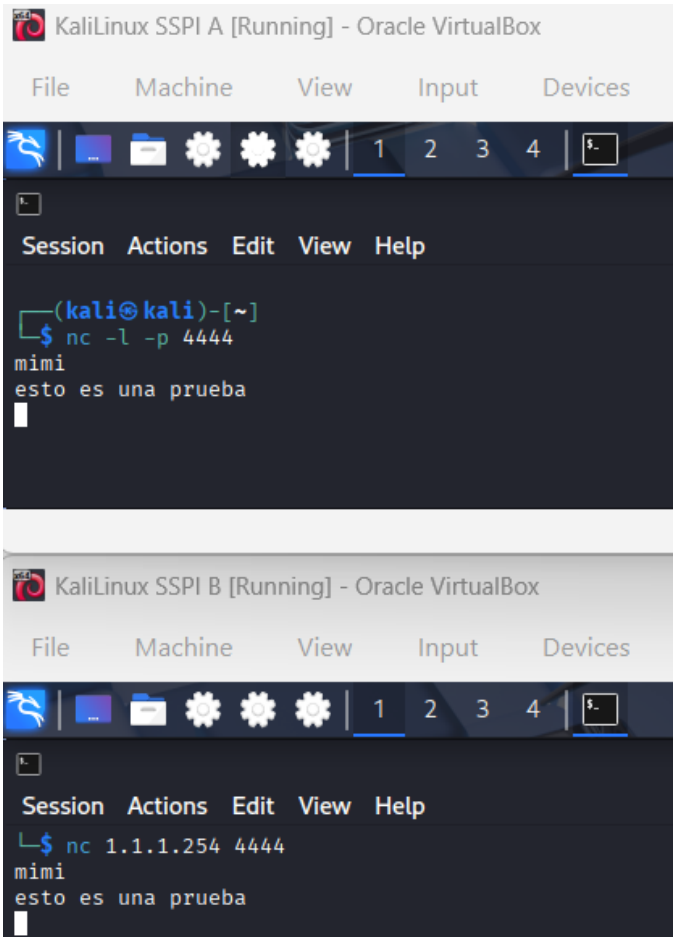
```
┌──(kali㉿kali)-[~]
└─$ ssh usuario@192.168.122.1
ssh: connect to host 192.168.122.1 port 22: Connection refused
```

/export de las reglas del firewall:

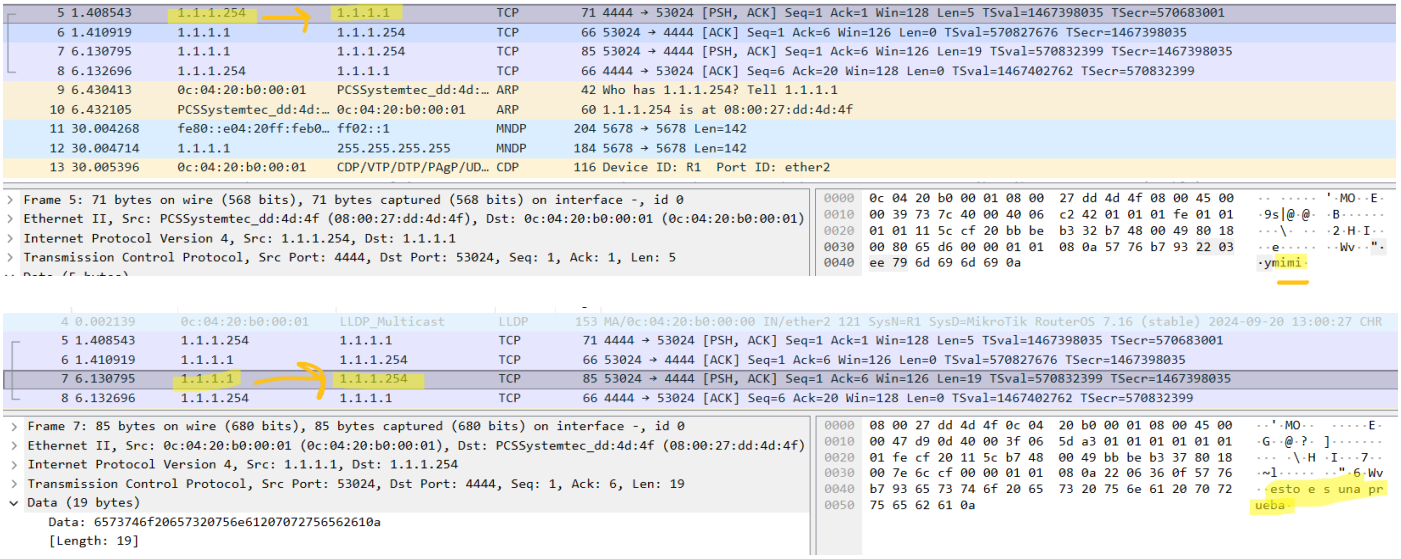
```
/ip firewall filter
add action=add-src-to-address-list address-list=knock1 address-list-timeout=15s chain=input comment=portknock1 \
    dst-port=831 protocol=tcp
add action=add-src-to-address-list address-list=knock2 address-list-timeout=15s chain=input comment=portknock2 \
    dst-port=841 protocol=tcp
add action=add-src-to-address-list address-list=ssh_access address-list-timeout=1m chain=input comment=\
    "Permitir SSH tras portknock" dst-port=851 protocol=tcp
add action=drop chain=input comment="drop por defecto" dst-port=22 protocol=tcp
/ip firewall nat
```

EJERCICIO 2. OPENVPN

Establecemos una comunicación entre ambos en el puerto 4444:

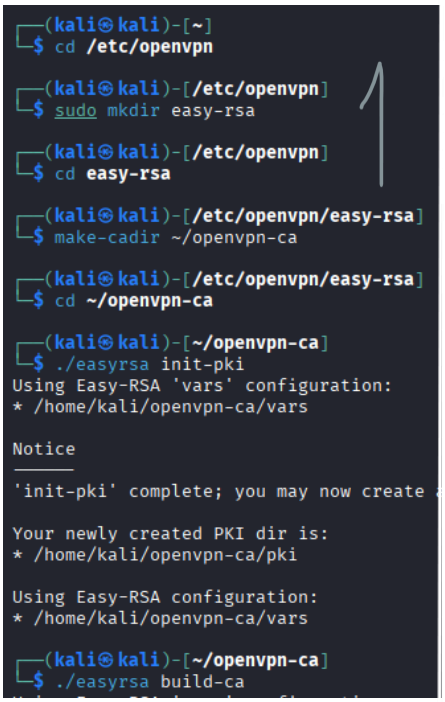


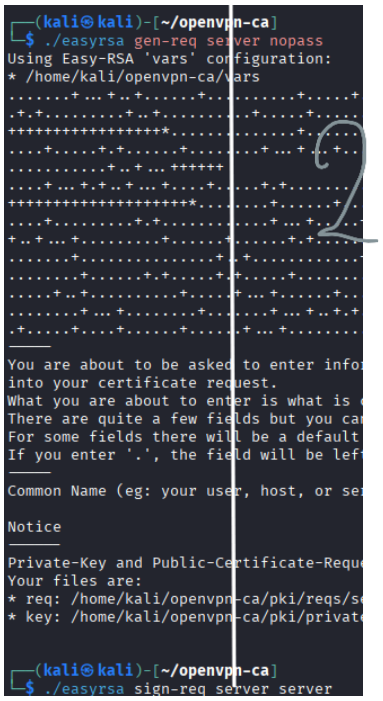
Vemos que estos mensajes se transmiten en claro, sin cifrado (subrayo el mensaje en amarillo, se observa abajo a la izquierda en los paquetes desde Wireshark):

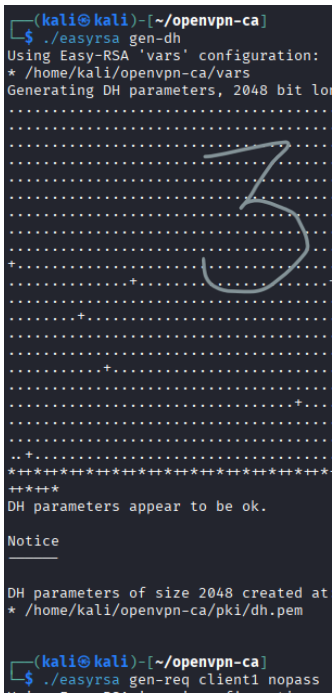


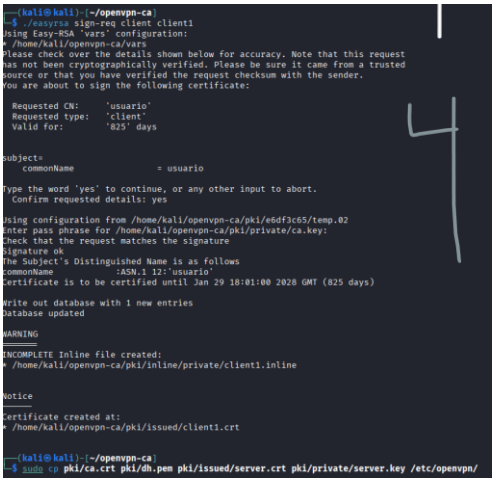
Observamos que sin VPN, los mensajes se envían sin cifrar.

Ahora configuramos la VPN primero en el servidor (KaliLinux A):









Observamos el estado del servidor VPN recién montado:

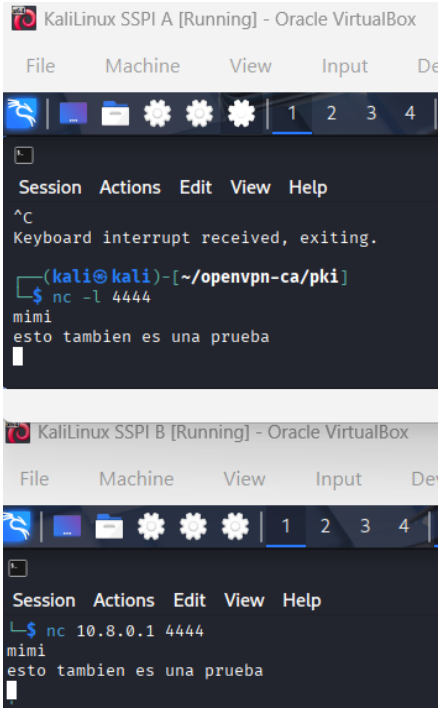
```
┌──(kali㉿kali)-[~/openvpn-ca]
└─$ sudo systemctl start openvpn@server

┌──(kali㉿kali)-[~/openvpn-ca]
└─$ sudo systemctl enable openvpn@server
^[[D^[[DCreated symlink '/etc/systemd/system/multi-user.target.wants/openvpn@server.service' → '/usr/lib/systemd/system/openvpn@.service'.

┌──(kali㉿kali)-[~/openvpn-ca]
└─$ sudo systemctl status openvpn@server
● openvpn@server.service - OpenVPN connection to server
     Loaded: loaded (/usr/lib/systemd/system/openvpn@.service; enabled; preset: disabled)
     Active: active (running) since Sun 2025-10-26 14:05:36 EDT; 14s ago
 Invocation: c9e111d309564b39936c8d377f484a33
       Docs: man:openvpn(8)
             https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
             https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 3338 (openvpn)
     Status: "Initialization Sequence Completed"
      Tasks: 1 (limit: 10)
     Memory: 3.4M (peak: 3.6M)
        CPU: 28ms
     CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
             └─3338 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.status 10 --cd /etc/openvpn --config /etc/openvpn/server.conf --w>

Oct 26 14:05:36 kali ovpn-server[3338]: net_addr_ptp_v4_add: 10.8.0.1 peer 10.8.0.2 dev tun0
Oct 26 14:05:36 kali ovpn-server[3338]: net_route_v4_add: 10.8.0.0/24 via 10.8.0.2 dev [NULL] table 0 metric -1
Oct 26 14:05:36 kali ovpn-server[3338]: Could not determine IPv4/IPv6 protocol. Using AF_INET
Oct 26 14:05:36 kali ovpn-server[3338]: Socket Buffers: R=[212992→212992] S=[212992→212992]
Oct 26 14:05:36 kali ovpn-server[3338]: UDPv4 link local (bound): [AF_INET][undef]:1194
Oct 26 14:05:36 kali ovpn-server[3338]: UDPv4 link remote: [AF_UNSPEC]
Oct 26 14:05:36 kali ovpn-server[3338]: MULTI: multi_init called, r=256 v=256
Oct 26 14:05:36 kali ovpn-server[3338]: IFCONFIG POOL IPv4: base=10.8.0.4 size=62
Oct 26 14:05:36 kali ovpn-server[3338]: IFCONFIG POOL LIST
Oct 26 14:05:36 kali ovpn-server[3338]: Initialization Sequence Completed
[212992→212992]
ndef]:1194


e=62
```

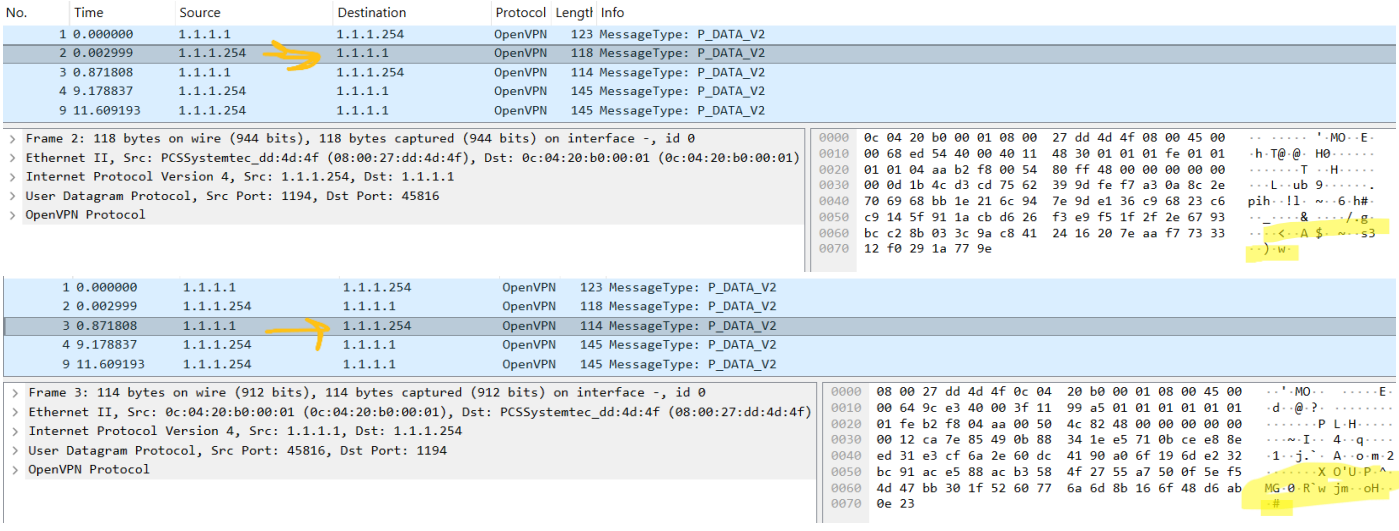
Ahora configuramos la VPN en el cliente (KaliLinux B):

```
┌──(kali㉿kali)-[~]
└─$ sudo chown root:root /etc/openvpn/ca.crt /etc/openvpn/client1.crt /etc/openvpn/client1.key

┌──(kali㉿kali)-[~]
└─$ sudo chmod 600 /etc/openvpn/client1.key

┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/openvpn/client.conf

┌──(kali㉿kali)-[~]
└─$ sudo systemctl enable --now openvpn@client
Created symlink '/etc/systemd/system/multi-user.target.wants/openvpn@client.service' → '/usr/lib/systemd/system/openvpn@.service'.

┌──(kali㉿kali)-[~]
└─$ sudo systemctl status openvpn@client --no-pager
● openvpn@client.service - OpenVPN connection to client
     Loaded: loaded (/usr/lib/systemd/system/openvpn@.service; enabled; preset: disabled)
     Active: active (running) since Sun 2025-10-26 14:23:50 EDT; 13s ago
 Invocation: 966ee0a196254a00880c64e01e8c9249
       Docs: man:openvpn(8)
             https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
             https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 3489 (openvpn)
     Status: "Initialization Sequence Completed"
      Tasks: 1 (limit: 10)
     Memory: 4M (peak: 4.3M)
        CPU: 30ms
     CGroup: /system.slice/system-openvpn.slice/openvpn@client.service
             └─3489 /usr/sbin/openvpn --daemon ovpn-client --status /run/openvpn/client.status 10 --cd /etc/openvpn --config /etc/openvpn/client.conf --writepid…

Oct 26 14:23:58 kali ovpn-client[3489]: ROUTE_GATEWAY 2.2.2.1/255.255.255.0 IFACE=eth1 HWADDR=08:00:27:71:91:b7
Oct 26 14:23:58 kali ovpn-client[3489]: TUN/TAP device tun0 opened
Oct 26 14:23:58 kali ovpn-client[3489]: net_iface_mtu_set: mtu 1500 for tun0
Oct 26 14:23:58 kali ovpn-client[3489]: net_iface_up: set tun0 up
Oct 26 14:23:58 kali ovpn-client[3489]: net_addr_ptp_v4_add: 10.8.0.6 peer 10.8.0.5 dev tun0
Oct 26 14:23:58 kali ovpn-client[3489]: net_route_v4_add: 10.8.0.1/32 via 10.8.0.5 dev [NULL] table 0 metric -1
Oct 26 14:23:58 kali ovpn-client[3489]: Initialization Sequence Completed
Oct 26 14:23:58 kali ovpn-client[3489]: Data Channel: cipher 'AES-256-GCM', peer-id: 0
Oct 26 14:23:58 kali ovpn-client[3489]: Timers: ping 10, ping-restart 120
Oct 26 14:23:58 kali ovpn-client[3489]: Protocol options: protocol-flags cc-exit tls-ekm dyn-tls-crypt
```

```
┌──(kali㉿kali)-[~]
└─$ ip -brief a show tun0
tun0             UNKNOWN        10.8.0.6 peer 10.8.0.5/32 fe80::3975:7ae8:4ab:db8d/64

┌──(kali㉿kali)-[~]
└─$ ip addre show tun0
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.6 peer 10.8.0.5/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::3975:7ae8:4ab:db8d/64 scope link stable-privacy proto kernel_ll
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ ip addr show tun0
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.6 peer 10.8.0.5/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::3975:7ae8:4ab:db8d/64 scope link stable-privacy proto kernel_ll
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
```

Y esto es lo que pasa si intentamos ver los paquetes al activar la VPN mientras se comunican, que esta vez la comunicación está cifrada:





Ya no son observables los mensajes sin cifrar desde Wireshark, como hemos podido comprobar.