

IES Valle Inclán



INSTALACIÓN DE VPN



NOMBRE Y APELLIDOS DEL AUTOR

Carlos González Martín y Rocío Ceballos Mateos.

ÍNDICE

1 – Introducción

2 - Instalación OpenVPN y Entidad Certificadora

3 - Creación de claves del Servidor

4 - Clave TLS-CRYPT

5 - Creación Claves del Cliente

6 - Configuración del servidor

7 - Configuración Cliente

8 - Habilitar el Firewall

9 - Fichero ovpn

10 - Prueba desde Windows

11-Conclusión.

1. INTRODUCCIÓN

En la actualidad, la seguridad en las comunicaciones digitales es un aspecto fundamental para empresas y usuarios individuales. El crecimiento del uso de internet ha traído consigo la necesidad de proteger la privacidad y garantizar la integridad de la información transmitida a través de redes públicas.

Una de las soluciones más eficaces para este propósito es el uso de redes privadas virtuales (VPN, por sus siglas en inglés). Un servidor VPN permite establecer una conexión segura y cifrada entre dispositivos, proporcionando anonimato, acceso remoto a redes privadas y protección contra amenazas externas. Su implementación es clave en entornos empresariales, instituciones académicas y para cualquier usuario que requiera resguardar sus datos al navegar en la red.

El desarrollo y configuración de un servidor VPN requieren conocimientos técnicos sobre protocolos de comunicación, seguridad informática y administración de redes. A través de este proceso, es posible garantizar conexiones seguras y eficientes, permitiendo el acceso a recursos de manera protegida y confiable

2. INSTALACIÓN OPENVPN Y ENTIDAD CERTIFICADORA

- Iniciamos una maquina debían y le cambiamos el nombre

```
Last login: Mon Sep 23 09:03:31 CEST 2024 on tty1
root@debian-12:~# hostnamectl set-hostname debian-12-VPN
root@debian-12:~# exit_
```

```
root@debian-12-VPN: ~#
root@debian-12-VPN: ~#
root@debian-12-VPN: ~#
root@debian-12-VPN: ~#
```

- Actualizamos los repositorios.

```
root@debian-12-VPN:~# apt update
Des:1 http://security.debian.org/debian-security bookworm-security InRelease [48,0 kB]
Des:2 http://deb.debian.org/debian bookworm InRelease [151 kB]
Des:3 http://deb.debian.org/debian bookworm-updates InRelease [55,4 kB]
Des:4 http://security.debian.org/debian-security bookworm-security/main Sources [145 kB]
Des:5 http://security.debian.org/debian-security bookworm-security/main amd64 Packages [244 kB]
Des:6 http://security.debian.org/debian-security bookworm-security/main Translation-en [145 kB]
Des:7 http://deb.debian.org/debian bookworm/non-free-firmware Sources [6.436 B]
Des:8 http://deb.debian.org/debian bookworm/main Sources [9.496 kB]
Des:9 http://deb.debian.org/debian bookworm/main amd64 Packages [8.792 kB]
Des:10 http://deb.debian.org/debian bookworm/main Translation-en [6.109 kB]
Des:11 http://deb.debian.org/debian bookworm/non-free-firmware amd64 Packages [6.340 B]
```

- Procederemos a instalar el paquete "OpenVPN", que será la herramienta utilizada para establecer la conexión VPN, junto con "EasyRSA", que nos permitirá generar y gestionar la infraestructura de clave pública

```

root@debian-12-VPN:~# apt install openvpn easy-rsa
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libccid liblzo2-2 libnl-3-200 libnl-genl-3-200 libpcsc-lite1 libpkcs11-helper1 opensc opensc-pkcs11 pcscd
Paquetes sugeridos:
  pcminitools resolvconf openvpn-dco-dkms openvpn-systemd-resolved
Se instalarán los siguientes paquetes NUEVOS:
  easy-rsa libccid liblzo2-2 libnl-3-200 libnl-genl-3-200 libpcsc-lite1 libpkcs11-helper1 opensc opensc-pkcs11 openvpn pcscd
0 actualizados, 11 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 2.694 kB de archivos.
Se utilizarán 8.141 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]

```

- Vemos la versión de "openvpn" para comprobar que está instalado correctamente.

```

root@debian-12-VPN:~# openvpn --version
OpenVPN 2.6.3 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/TKINTF] [AEAD] [DCO]
library versions: OpenSSL 3.0.15 3 Sep 2024, LZO 2.10
DCO version: N/A
Originally developed by James Yonan
Copyright (C) 2002-2023 OpenVPN Inc <sales@openvpn.net>
Compile time defines: enable_async_push=no enable_comp_stub=no enable_crypto_ofb_cfb=yes enable_dco=yes enable_dco_arg=yes enable_debug=yes enable_dependency_tr
acking=no enable_dlopen=unknown enable_dlopen_self=unknown enable_dlopen_self_static=unknown enable_fast_install=needless enable_fragment=yes enable_iproute2=no
enable_libtool_lock=yes enable_lz4=yes enable_lzo=yes enable_maintainer_mode=no enable_management=yes enable_option_checking=no enable_pam_dlopen=no enable_ped
antic=no enable_pkcs11=yes enable_plugin_auth_gm=yes enable_plugin_down_root=yes enable_plugins=yes enable_port_share=yes enable_selinux=no enable_share=yes e
nable_shared_with_static_runtimes=no enable_silent_rules=no enable_small=no enable_static=yes enable_strict=no enable_strict_options=no enable_systemd=yes enabl
e_unit_tests=no enable_werror=no enable_win32_dll=yes enable_wolfssl_options_h=yes enable_x509_alt_username=yes with_aix_soname=aix with_crypto_library=openssl
with_gnu_id=yes with_mem_check=no with_openssl_engine=auto with_sysroot=no
root@debian-12-VPN:~#

```

- Ahora vamos a configurar tanto la "infraestructura de clave pública" (que se encargará de autenticar a las partes involucradas en la comunicación, como veremos al verificarlo) como la "autoridad certificadora" (que garantiza que el servidor es realmente quien dice ser). Para evitar que las actualizaciones sobrescriban los certificados que hemos generado, copiamos el siguiente directorio:

```

root@debian-12-VPN:~# cp -r /usr/share/easy-rsa/ /etc/openvpn/
root@debian-12-VPN:~# ls -l /etc/openvpn
total 16
drwxr-xr-x 2 root root 4096 nov 11 2023 client
drwxr-xr-x 3 root root 4096 feb 10 22:49 easy-rsa
drwxr-xr-x 2 root root 4096 nov 11 2023 server
-rwxr-xr-x 1 root root 1468 nov 11 2023 update-resolv-conf
root@debian-12-VPN:~#

```


3. CREACIÓN DE CLAVES DEL SERVIDOR

- Vamos a generar tanto la clave pública como la privada del servidor. Asignaremos un nombre, manteniendo el que usamos en el comando. Esto nos proporcionará la clave privada (private) y la clave pública (req), la cual deberá ser firmada por nuestra entidad certificadora.

```
root@debian-12-VPN:/etc/openvpn/easy-rsa# ./easyrsa gen-req sad-vpn nopass
* Notice:
Using Easy-RSA configuration from: /etc/openvpn/easy-rsa/pki/vars

* Notice:
Using SSL: openssl OpenSSL 3.0.15 3 Sep 2024 (Library: OpenSSL 3.0.15 3 Sep 2024)

.....
+-----+
|                                     |
| You are about to be asked to enter information that will be incorporated |
| into your certificate request.                                             |
| What you are about to enter is what is called a Distinguished Name or a DN. |
| There are quite a few fields but you can leave some blank.                 |
| For some fields, there will be a default value,                            |
| If you enter '.', the field will be left blank.                           |
|-----+
Common Name (eg: your user, host, or server name) [sad-vpn]: practica-vpn
* Notice:
Keypair and certificate request completed. Your files are:
req: /etc/openvpn/easy-rsa/pki/reqs/sad-vpn.req
key: /etc/openvpn/easy-rsa/pki/private/sad-vpn.key
root@debian-12-VPN:/etc/openvpn/easy-rsa#
```

- Firmamos la clave pública ("sign-req") utilizando el mismo nombre que asignamos previamente ("sad-mirasan"). Finalmente, confirmamos la operación ("yes"). Para completar el proceso, se requiere la contraseña establecida anteriormente para la entidad certificadora.

```
root@debian-12-VPN:/etc/openvpn/easy-rsa# ./easyrsa sign-req server sad-vpn
* Notice:
Using Easy-RSA configuration from: /etc/openvpn/easy-rsa/pki/vars

* Notice:
Using SSL: openssl OpenSSL 3.0.15 3 Sep 2024 (Library: OpenSSL 3.0.15 3 Sep 2024)

You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 825 days:

subject=
  commonName = practica-vpn

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes

Using configuration from /etc/openvpn/easy-rsa/pki/562e73c8/temp.7b0fabcb
Enter pass phrase for /etc/openvpn/easy-rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'practica-vpn'
Certificate is to be certified until May 16 21:56:27 2027 GMT (825 days)

Write out database with 1 new entries
Database updated

* Notice:
Certificate created at: /etc/openvpn/easy-rsa/pki/issued/sad-vpn.crt
root@debian-12-VPN:/etc/openvpn/easy-rsa#
```

- Para organizar mejor el trabajo, vamos a mover todos los certificados a un directorio común, de manera que estén reunidos en un solo lugar.

Ahora tendríamos: la pública de la entidad certificadora (“ca.crt”), la privada del servidor (“sad-vpn.crt”) y la pública del servidor (“sad-vpn.key”):

```
root@debian-12-VPN:/etc/openssl/easy-rsa# cp /etc/openssl/easy-rsa/pki/issued/sad-vpn.crt
easyrsa openssl-easyrsa.cnf pki/ vars.example x509-types/
root@debian-12-VPN:/etc/openssl/easy-rsa# cp /etc/openssl/easy-rsa/pki/issued/sad-vpn.crt /etc/openssl/server/
root@debian-12-VPN:/etc/openssl/easy-rsa# cp /etc/openssl/easy-rsa/pki/ca.crt /etc/openssl/server/
root@debian-12-VPN:/etc/openssl/easy-rsa# cp /etc/openssl/easy-rsa/pki/private/ /etc/openssl/server/
ca.key sad-vpn.key
root@debian-12-VPN:/etc/openssl/easy-rsa# cp /etc/openssl/easy-rsa/pki/private/sad-vpn.key /etc/openssl/server/
root@debian-12-VPN:/etc/openssl/easy-rsa# ls -l /etc/openssl/server/
total 16
-rw----- 1 root root 1204 feb 10 22:57 ca.crt
-rw----- 1 root root 4637 feb 10 22:57 sad-vpn.crt
-rw----- 1 root root 1704 feb 10 22:58 sad-vpn.key
root@debian-12-VPN:/etc/openssl/easy-rsa# _
```

4. Clave TLS-CRYPT

- Esto nos brinda una capa adicional de seguridad. Más adelante, la integraremos en las últimas etapas del proceso, pero por ahora solo la crearemos.

Nos dirigimos al directorio donde almacenamos todas las claves y certificados. Generamos la clave con el nombre “[secre]ta.key”. Es posible que aparezca un mensaje indicando que el parámetro “--secret” está en desuso, pero seguirá funcionando sin problemas.

```
root@debian-12-VPN:/etc/openssl/server# openssl --genkey --secret ta.key
2025-02-10 22:59:12 DEPRECATED OPTION: The option --secret is deprecated.
2025-02-10 22:59:12 WARNING: Using --genkey --secret filename is DEPRECATED. Use --genkey secret filename instead.
root@debian-12-VPN:/etc/openssl/server# ls -l
total 20
-rw----- 1 root root 1204 feb 10 22:57 ca.crt
-rw----- 1 root root 4637 feb 10 22:57 sad-vpn.crt
-rw----- 1 root root 1704 feb 10 22:58 sad-vpn.key
-rw----- 1 root root 636 feb 10 22:59 ta.key
root@debian-12-VPN:/etc/openssl/server# _
```

5. CREACIÓN CLAVES DEL CLIENTE

- Creamos otro directorio para trabajar de forma más cómoda como hicimos anteriormente y también quitamos los permisos tanto de usuario como de grupo:

```
root@debian-12-VPN:/etc/openssl/server# mkdir /etc/openssl/client/keys
root@debian-12-VPN:/etc/openssl/server# chmod -R 700 /etc/openssl/client
root@debian-12-VPN:/etc/openssl/server# ls -l /etc/openssl/client
total 4
drwx----- 2 root root 4096 feb 10 22:59 keys
root@debian-12-VPN:/etc/openssl/server# ls /etc/openssl/client
keys
root@debian-12-VPN:/etc/openssl/server# ls /etc/openssl/
client easy-rsa server update-resolv-conf
root@debian-12-VPN:/etc/openssl/server# ls -l /etc/openssl/
total 16
drwx----- 3 root root 4096 feb 10 22:59 client
drwxr-xr-x 4 root root 4096 feb 10 22:50 easy-rsa
drwxr-xr-x 2 root root 4096 feb 10 22:59 server
-rwxr-xr-x 1 root root 1468 nov 11 2023 update-resolv-conf
root@debian-12-VPN:/etc/openssl/server#
```


6. CONFIGURACIÓN DEL SERVIDOR

- Para configurar el archivo correctamente, primero debemos obtenerlo, ya que no se incluye por defecto. Contamos con "client.conf" para la configuración del cliente y "server.conf" para la del servidor.

```
root@debian-12-VPN:/etc/openvpn/easy-rsa# ls -l /usr/share/doc/openvpn/examples/sample-config-files/
total 64
-rw-r--r-- 1 root root 3591 abr 13 2023 client.conf
-rwxr-xr-x 1 root root 3562 abr 13 2023 firewall.sh
-rwxr-xr-x 1 root root 62 abr 13 2023 home.up
-rw-r--r-- 1 root root 11386 abr 13 2023 loopback-client
-rw-r--r-- 1 root root 694 abr 13 2023 loopback-server
-rwxr-xr-x 1 root root 62 abr 13 2023 office.up
-rwxr-xr-x 1 root root 63 abr 13 2023 openvpn-shutdown.sh
-rwxr-xr-x 1 root root 776 abr 13 2023 openvpn-startup.sh
-rw-r--r-- 1 root root 131 abr 13 2023 README
-rw-r--r-- 1 root root 10882 nov 11 2023 server.conf
-rw-r--r-- 1 root root 2005 abr 13 2023 tls-home.conf
-rw-r--r-- 1 root root 2034 abr 13 2023 tls-office.conf
root@debian-12-VPN:/etc/openvpn/easy-rsa#
```

- Lo copiamos.

```
root@debian-12-VPN:/etc/openvpn/easy-rsa# cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf /etc/openvpn/server/
root@debian-12-VPN:/etc/openvpn/easy-rsa# ls /etc/openvpn/server
ca.crt sad-vpn.crt sad-vpn.key server.conf ta.key
root@debian-12-VPN:/etc/openvpn/easy-rsa# ls -l /etc/openvpn/server
total 32
-rw----- 1 root root 1204 feb 10 22:57 ca.crt
-rw----- 1 root root 4637 feb 10 22:57 sad-vpn.crt
-rw----- 1 root root 1704 feb 10 22:58 sad-vpn.key
-rw-r--r-- 1 root root 10882 feb 11 11:42 server.conf
-rw----- 1 root root 635 feb 10 22:59 ta.key
root@debian-12-VPN:/etc/openvpn/easy-rsa#
```

- Lo editamos.
 - Ponemos el nombre
 - La directiva la desactivamos con ";" y "dh none". **Línea 86.**
 - Dejamos la red que creará la VPN como una subred: **Línea 92.**
 - La red (subred) que utilizará la VPN. **Línea 101.**
 - Que te lo redirija por la puerta de enlace del servidor, que tengan salida a internet los clientes básicamente **Línea 192.**
 - Clave adicional que creamos (la TLS). **Línea 244-245.**
 - Cambiamos el cifrado por los recomendados **Línea 253-254-255.**
 - Clientes simultáneos que vamos a admitir **Línea 270.**
 - Desactivar los permisos tanto de usuario como de grupo al activar la VPN por seguridad. **Línea 278.**
 - Aquí se almacenan las conexiones actuales. **Línea 290.**

```

GNU nano 7.2 /etc/openvpn/server/server.conf *
75 # Any X509 key management system can be used.
76 # OpenVPN can also use a PKCS #12 formatted key file
77 # (see "pkcs12" directive in man page)
78 ca ca.crt
79 cert sad-vpn.crt
80 key sad-vpn.key # This file should be kept secret
81
82 # Diffie hellman parameters.
83 # Generate your own with:
84 #   openssl dhparam -out dh2048.pem 2048
85 :dh dh2048.pem
86 dh none
87 # Network topology
88 # Should be subnet (addressing via IP)
89 # unless Windows clients v2.0.9 and lower have to
90 # be supported (then net30, i.e. a /30 per client)
91 # Defaults to net30 (not recommended)
92 topology subnet
93
94 # Configure server mode and supply a VPN subnet
95 # for OpenVPN to draw client addresses from.
96 # The server will take 10.8.0.1 for itself,
97 # the rest will be made available to clients.
98 # Each client will be able to reach the server
99 # on 10.8.0.1. Comment this line out if you are
100 # ethernet bridging. See the man page for more info.
101 server 10.8.0.0 255.255.255.0
102
103 # Maintain a record of client <-> virtual IP address

```

```

GNU nano 7.2 /etc/openvpn/server/server.conf *
189 # (The OpenVPN server machine may need to NAT
190 # or bridge the TUN/TAP interface to the internet
191 # in order for this to work properly).
192 ;push "redirect-gateway def1 bypass-dhcp"
193
194 # Certain Windows-specific network settings
195 # can be pushed to clients, such as DNS
196 # or WINS server addresses. CAVEAT:

```

```

GNU nano 7.2 /etc/openvpn/server/server.conf *
241 # a copy of this key.
242 # The second parameter should be '0'
243 # on the server and '1' on the clients
244 :tls-auth ta.key 0 # This file is secret
245 :tls-crypt ta.key
246
247 # Select a cryptographic cipher.
248 # This config item must be copied to
249 # the client config file as well.
250 # Note that v2.4 client/server will automatically
251 # negotiate AES-256-GCM in TLS mode.
252 # See also the --comp-cipher option in the manpage
253 :cipher AES-256-CBC
254 cipher AES-256-GCM
255 auth SHA512
256
257 # Enable compression on the VPN link and push the
258 # option to the client (v2.4+ only, for earlier
259 # versions see below)
260 :compress lz4-v2
261 :push "compress lz4-v2"
262
263 # For compression compatible with older clients use comp-lzo
264 # If you enable it here, you must also
265 # enable it in the client config file.
266 :comp-lzo
267
268 # The maximum number of concurrently connected
269 # clients we want to allow.
270 max-clients 100
271
272 # It's a good idea to reduce the OpenVPN

```

```

GNU nano 7.2 /etc/openvpn/server/server.conf *
274 #
275 # You can uncomment this on non-Windows
276 # systems after creating a dedicated user.
277 user nobody
278 group nogroup
279
280 # The persist options will try to avoid
281 # accessing certain resources on restart
282 # that may no longer be accessible because
283 # of the privilege downgrade.
284 persist-key
285 persist-tun
286
287 # Output a short status file showing
288 # current connections, truncated
289 # and rewritten every minute.
290 status /var/log/openvpn/openvpn-status.log
291
292 # By default, log messages will go to the syslog (or

```

7. CONFIGURACIÓN CLIENTE.

- Seguimos los mismos pasos para indicarle que somos un cliente pero para ello copiamos el archivo de “cliente.conf”.

```
root@debian-12-VPN:/etc/openvpn/easy-rsa# cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/openvpn/client/
root@debian-12-VPN:/etc/openvpn/easy-rsa# ls -l /etc/openvpn/client/
total 8
-rw-r--r-- 1 root root 3591 feb 11 11:58 client.conf
drwx----- 2 root root 4096 feb 10 23:06 keys
root@debian-12-VPN:/etc/openvpn/easy-rsa#
```

- Lo editamos.
 - Indicarle que eres cliente **Línea 16**
 - Dirección IP y puerto donde escucha el servidor VPN: **Línea 42.**
 - Le quitamos el usuario y grupo al servicio: **Línea 61-62.**
 - Desactivamos estas después se lo vamos a pasar mediante un fichero con extensión “ovpn”.
 - **Línea 88-89-90-108.**
 - En el cifrado ponemos el mismo que en el fichero del servidor: **Línea 116-117.**

```
GNU nano 7.2 /etc/openvpn/client/client.conf *
15 # from the server.
16 client
17
18 # Use the same setting as you are using on
19 # the server.
20 # On most systems, the VPN will not function
21 # unless you partially or fully disable
22 # the firewall for the TUN/TAP interface.
23 ;dev tap
24 dev tun
25
26 # Windows needs the TAP-Win32 adapter name
27 # from the Network Connections panel
28 # if you have more than one. On XP SP2,
29 # you may need to disable the firewall
30 # for the TAP adapter.
31 ;dev-node MyTap
32
33 # Are we connecting to a TCP or
34 # UDP server? Use the same setting as
35 # on the server.
36 ;proto tcp
37 proto udp
38
39 # The hostname/IP and port of the server.
40 # You can have multiple remote entries
41 # to load balance between the servers.
42 remote 192.168.1.88 1194
43 ;remote my-server-2 1194
44
45 # Choose a random host from the remote
```

```
GNU nano 7.2 /etc/openvpn/client/client.conf
57 # a specific local port number.
58 nobind
59
60 # Downgrade privileges after initialization (non-Windows only)
61 user nobody
62 group nobody
63
```

```

GNU nano 7.2 /etc/openvpn/client/client.conf *
84 # description. It's best to use
85 # a separate .crt/.key file pair
86 # for each client. A single ca
87 # file can be used for all clients.
88 ;ca ca.crt
89 ;cert client.crt
90 ;key client.key
91
92 # Verify server certificate by checking that the
93 # certificate has the correct key usage set.
94 # This is an important precaution to protect against
95 # a potential attack discussed here:
96 # http://openvpn.net/howto.html#mitm
97 #
98 # To use this feature, you will need to generate
99 # your server certificates with the keyUsage set to
100 # digitalSignature, keyEncipherment
101 # and the extendedKeyUsage to
102 # serverAuth
103 # EasyRSA can do this for you.
104 remote-cert-tls server
105
106 # If a tls-auth key is used on the server
107 # then every client must also have the key.
108 ;tls-auth ta.key 1
109
110 # Select a cryptographic cipher.
111 # If the cipher option is used on the server
112 # then you must also specify it here.
113 # Note that v2.4 client/server will automatically
114 # negotiate AES-256-GCM in TLS mode.
115 # See also the data-ciphers option in the manpage
116 cipher AES-256-GCM
117 auth SHA512
118

```

8. HABILITAR EL FIREWALL

- IMPORTANTE, tenemos que instalar el paquete ufw (en nuestro caso en un sistema operativo debían no viene instalado).
- Añadir la regla para el puerto de “openvpn” en el servidor:

```

root@debian-12-VPN:/etc/openvpn/easy-rsa# ufw status
Status: inactive
root@debian-12-VPN:/etc/openvpn/easy-rsa# ufw allow 1194/udp
Rules updated
Rules updated (v6)
root@debian-12-VPN:/etc/openvpn/easy-rsa# ufw enable
Firewall is active and enabled on system startup
root@debian-12-VPN:/etc/openvpn/easy-rsa# ufw status
Status: active

To Action From
--
1194/udp ALLOW Anywhere
1194/udp (v6) ALLOW Anywhere (v6)

root@debian-12-VPN:/etc/openvpn/easy-rsa#

```

- Ahora es necesario habilitar el reenvío de paquetes entre interfaces, ya que el servidor añadirá una más. Para ello, accedemos al archivo “/etc/sysctl.conf” y eliminamos el comentario de la siguiente línea:

```

GNU nano 7.2 /etc/sysctl.conf *
24 # Note: This may impact IPv6 TCP sessions too
25 #net.ipv4.tcp_syncookies=1
26
27 # Uncomment the next line to enable packet forwarding for IPv4
28 net.ipv4.ip_forward=1
29

```

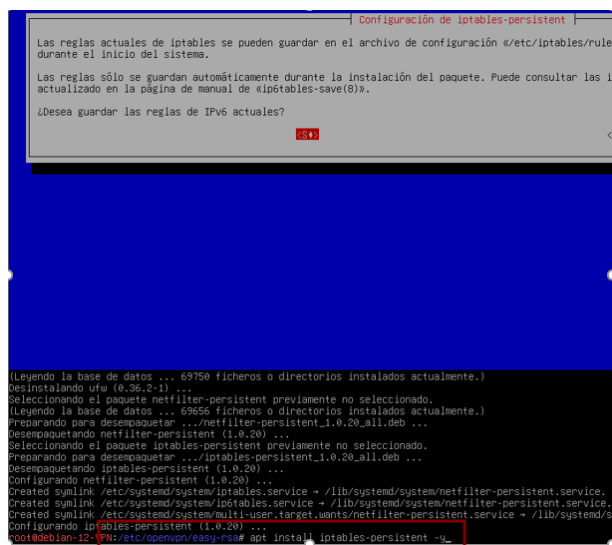
- Ponemos a “1” este fichero y con esto ya tendríamos habilitado el enrutamiento

```
root@debian-12-VPN:/etc/openvpn/easy-rsa# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
root@debian-12-VPN:/etc/openvpn/easy-rsa# cat /proc/sys
sys/      sysrq-trigger sysvipc/
root@debian-12-VPN:/etc/openvpn/easy-rsa# cat /proc/sys/net/ipv4/ip_forward
1
root@debian-12-VPN:/etc/openvpn/easy-rsa#
```

- Configuramos el cortafuegos con las siguientes reglas para habilitar el enrutamiento y el enmascaramiento.

```
valid-iptables --iptables --help
root@debian-12-VPN:/etc/openvpn/easy-rsa# iptables -I INPUT 1 -i tun0 -j ACCEPT
root@debian-12-VPN:/etc/openvpn/easy-rsa# iptables -I INPUT 1 -i enp0s3 -o tun0 -j ACCEPT
iptables v1.8.9 (nf_tables): Can't use -o with INPUT
Try `iptables -h' or 'iptables --help' for more information.
root@debian-12-VPN:/etc/openvpn/easy-rsa# iptables -I FORWARD 1 -i enp0s3 -o tun0 -j ACCEPT
root@debian-12-VPN:/etc/openvpn/easy-rsa# iptables -I FORWARD 1 -i tun0 -o enp0s3 -j ACCEPT
root@debian-12-VPN:/etc/openvpn/easy-rsa# iptables -I INPUT 1 -i enp0s3 -p udp --dport 1194 -j ACCEPT
Bad argument '1'
Try `iptables -h' or 'iptables --help' for more information.
root@debian-12-VPN:/etc/openvpn/easy-rsa# iptables -I INPUT 1 -i enp0s3 -p udp --dport 1194 -j ACCEPT
Bad argument '1'
Try `iptables -h' or 'iptables --help' for more information.
root@debian-12-VPN:/etc/openvpn/easy-rsa# iptables -I INPUT 1 -i enp0s3 -p udp --dport 1194 -j ACCEPT
root@debian-12-VPN:/etc/openvpn/easy-rsa# iptables -t nat -I POSTROUTING 1 -s 10.8.0.0/24 -o enp0s3 -j MASQUERADE
root@debian-12-VPN:/etc/openvpn/easy-rsa#
```

- Para asegurar que las reglas se mantengan después de un reinicio, instalamos el paquete “iptables-persistent”



```
(Leyendo la base de datos ... 69750 ficheros o directorios instalados actualmente.)
Desinstalando ufw (0.36.2-1) ...
Seleccionando el paquete netfilter-persistent previamente no seleccionado.
(Leyendo la base de datos ... 69655 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../netfilter-persistent_1.0.20_all.deb ...
Desempaquetando netfilter-persistent (1.0.20) ...
Seleccionando el paquete iptables-persistent previamente no seleccionado.
Preparando para desempaquetar .../iptables-persistent_1.0.20_all.deb ...
Desempaquetando iptables-persistent (1.0.20) ...
Configurando netfilter-persistent (1.0.20) ...
Created symlink /etc/systemd/system/iptables.service → /lib/systemd/system/netfilter-persistent.service.
Created symlink /etc/systemd/system/iptables.service → /lib/systemd/system/netfilter-persistent.service.
Created symlink /etc/systemd/system/multi-user.target.wants/netfilter-persistent.service → /lib/systemd/s
Configurando iptables-persistent (1.0.20) ...
root@debian-12-VPN:/etc/openvpn/easy-rsa# apt install iptables-persistent -y
```

- Configuramos el servicio “OpenVPN” para que se inicie automáticamente al arrancar el sistema.

```
root@debian-12-VPN:/etc/openvpn/easy-rsa# systemctl -f enable openvpn-server@server.service
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn-server@server.service → /lib/systemd/system/openvpn-server@.service.
root@debian-12-VPN:/etc/openvpn/easy-rsa# service openvpn-server status
Unit openvpn-server.service could not be found.
root@debian-12-VPN:/etc/openvpn/easy-rsa# service openvpn status
• openvpn.service - OpenVPN service
  Loaded: loaded (/lib/systemd/system/openvpn.service; enabled; preset: enabled)
  Active: active (exited) since Mon 2025-02-10 22:48:05 CET; 13h ago
  Main PID: 1202 (code=exited, status=0/SUCCESS)
  CPU: 1ms
Feb 10 22:48:05 debian-12-VPN systemd[1]: Starting openvpn.service - OpenVPN service...
Feb 10 22:48:05 debian-12-VPN systemd[1]: Finished openvpn.service - OpenVPN service.
root@debian-12-VPN:/etc/openvpn/easy-rsa#
```

- Aquí puedes comprobar que el servidor ha creado la nueva interfaz “tun0” con una dirección de red asignada según la configuración establecida.

```
root@debian-12-VPN:~# ip -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:96:ec:6a brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.88/24 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 86229sec preferred_lft 86229sec
    inet6 2a0c:5a00:5506:a000:a00:27ff:fe96:ec6a/64 scope global dynamic mngtmpaddr
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe96:ec6a/64 scope link
        valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.1/24 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::7df:7d3:b487:8695/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
root@debian-12-VPN:~#
```

9. FICHERO OVPN

- Copiamos nuevamente el archivo del cliente para utilizarlo como plantilla en el script del próximo capítulo.

```
root@debian-12-VPN:/etc/openvpn/client# ls -la
total 24
drwx----- 3 root root 4096 feb 13 19:36 .
drwxr-xr-x 5 root root 4096 feb 13 19:36 ..
-rw-r--r-- 1 root root 3604 feb 11 12:06 client.conf
drwx----- 2 root root 4096 feb 10 23:06 keys
-rw-r--r-- 1 root root 444 feb 5 08:39 make_config.sh
-rw-r--r-- 1 root root 3604 feb 13 19:30 plantilla.conf
root@debian-12-VPN:/etc/openvpn/client#
```

- Creamos script:
 - Definición de variables:
 - KEY_DIR=/etc/openvpn/client/keys → Directorio donde se almacenan las claves y certificados.
 - OUTPUT_DIR=/etc/openvpn/client/files → Directorio donde se guardarán los archivos de configuración generados.
 - BASE_CONFIG=/etc/openvpn/client/plantilla.conf → Ruta del archivo base de configuración.
 - Generación del archivo de configuración:
 - cat \${BASE_CONFIG} → Copia el contenido de la plantilla base.
 - echo -e '<ca>' → Inserta una etiqueta <ca>, que indica el inicio del certificado de la autoridad certificadora (CA).
 - \${KEY_DIR}/ca.crt → Agrega el contenido del archivo **ca.crt** dentro del archivo de configuración.
 - Similarmente, añade el certificado (.crt), la clave (.key) y el archivo de clave TLS (ta.key), cada uno con su respectiva etiqueta XML (<cert>, <key>, <tls-crypt>).
 - Finalmente, el archivo resultante se guarda en **\${OUTPUT_DIR}/\${1.ovpn}**, donde **1** es el nombre del cliente que se pasa como argumento al ejecutar el script.


```

GNU nano 7.2 make_config.sh
1 #!/bin/bash
2
3 KEY_DIR=/etc/openvpn/client/keys
4 OUTPUT_DIR=/etc/openvpn/client/files
5 BASE_CONFIG=/etc/openvpn/client/plantilla.conf
6
7 cat ${BASE_CONFIG} \
8     <(echo -e '<ca>' \
9     ${KEY_DIR}/ca.crt \
10    <(echo -e '</ca>\n<cert>' \
11    ${KEY_DIR}/${1}.crt \
12    <(echo -e '</cert>\n<key>' \
13    ${KEY_DIR}/${1}.key \
14    <(echo -e '</key>\n<tls-crypt>' \
15    ${KEY_DIR}/ta.key \
16    <(echo -e '</tls-crypt>' \
17    > ${OUTPUT_DIR}/S${1}.ovpn
18

```

- Creamos el directorio donde te va a lanzar el fichero resultante del script

```

root@debian-12-VPN:/etc/openvpn/client# ls
client.conf  keys  make_config.sh  plantilla.conf
root@debian-12-VPN:/etc/openvpn/client# mkdir files
root@debian-12-VPN:/etc/openvpn/client# ls -la
total 28
drwx----- 4 root root 4096 feb 13 19:38 .
drwxr-xr-x 5 root root 4096 feb 13 19:36 ..
-rw-r--r-- 1 root root 3604 feb 11 12:06 client.conf
drwxr-xr-x 2 root root 4096 feb 13 19:38 files
drwx----- 2 root root 4096 feb 10 23:06 keys
-rw-r--r-- 1 root root 444 feb 5 08:39 make_config.sh
-rw-r--r-- 1 root root 3604 feb 13 19:30 plantilla.conf
root@debian-12-VPN:/etc/openvpn/client# _

```

- Le damos permisos a “root” por seguridad.

```

root@debian-12-VPN:/etc/openvpn# chmod 700 client/make_config.sh
root@debian-12-VPN:/etc/openvpn# ls -l client
total 20
-rw-r--r-- 1 root root 3604 feb 11 12:06 client.conf
drwxr-xr-x 2 root root 4096 feb 13 19:38 files
drwx----- 2 root root 4096 feb 10 23:06 keys
-rwx----- 1 root root 444 feb 5 08:39 make_config.sh
-rw-r--r-- 1 root root 3604 feb 13 19:30 plantilla.conf
root@debian-12-VPN:/etc/openvpn# _

```

Lo ejecutamos:

```
root@debian-12-VPN:/etc/openvpn/client# ls keys
ca.crt prueba1.crt prueba1.key ta.key
root@debian-12-VPN:/etc/openvpn/client# ./make_config.sh prueba1
root@debian-12-VPN:/etc/openvpn/client#
```

10. PRUEBA DESDE WINDOWS

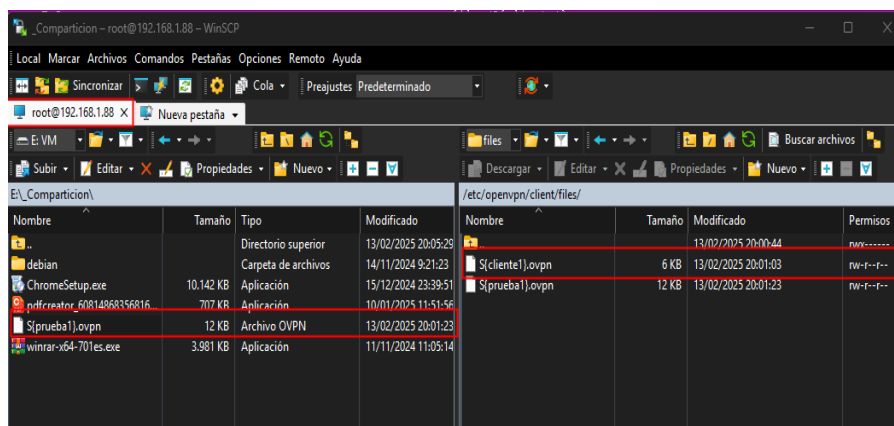
El script **make_config.sh** que creamos antes se encargó de generar automáticamente el archivo **cliente1.ovpn**, combinando:

1. **La configuración base** (del archivo plantilla **plantilla.conf**).
2. **Los certificados y claves** (ca.crt, <cliente>.crt, <cliente>.key, ta.key) dentro del archivo .ovpn, usando etiquetas <ca>, <cert>, <key>, <tls-crypt>.

Gracias a esto, el archivo **cliente1.ovpn** ya contiene todo lo necesario para conectar el cliente a OpenVPN sin necesidad de archivos adicionales. Solo necesitaremos transferir este archivo a nuestro dispositivo cliente e importarlo en OpenVPN para conectarnos

```
GNU nano 7.2 files/S(cliente1).ovpn
1 #####
2 # Sample client-side OpenVPN 2.0 config file #
3 # for connecting to multi-client server. #
4 #
5 # This configuration can be used by multiple #
6 # clients, however each client should have #
7 # its own cert and key files. #
8 #
9 # On Windows, you might want to rename this #
10 # file so it has a .ovpn extension #
11 #####
12
13 # Specify that we are a client and that we
14 # will be pulling certain config file directives
15 # from the server.
16 client
17
18 # Use the same setting as you are using on
19 # the server.
20
21 # On most systems, the VPN will not function
22 # unless you partially or fully disable
23 # the firewall for the TUN/TAP interface.
24 #
25 # On Windows, you might want to disable the
26 # Windows Firewall for the TAP-Win32 adapter
27 # before starting the client. On XP SP2,
28 # you may need to disable the firewall
29 # for the TAP adapter.
30 #
31 #dev-node MyTap
32
33 # Are we connecting to a TCP or
34 # UDP server? Use the same setting as
35 # on the server.
36 proto tcp
37 proto udp
38
39 # The hostname/IP and port of the server.
40 # You can have multiple remote entries
41 # to load balance between the servers.
```

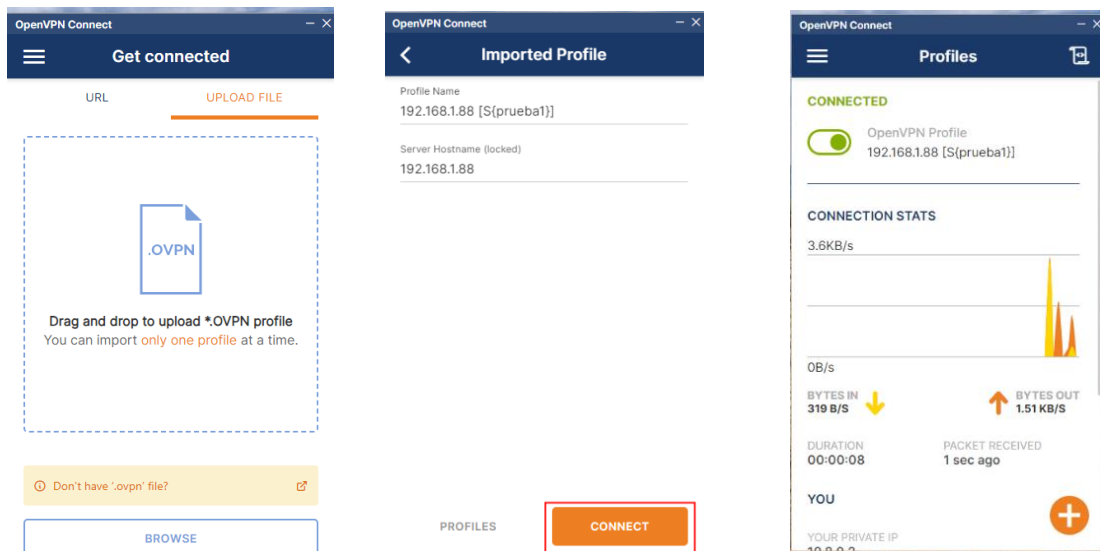
- Usaremos WINSCP para poder enviar los archivos al cliente.



- Instalamos la aplicación de “OpenVPN” en la máquina Windows



- Cargamos el fichero y comprobamos que hay conexión.
 - **Importación del perfil .ovpn (IMAGEN 1)**
 - Vemos la opción para subir un archivo .ovpn.
 - Este archivo .ovpn contiene la configuración del cliente y los certificados necesarios para conectarse al servidor VPN.
 - **Perfil importado correctamente (IMAGEN 2)**
 - Mostramos que el perfil ha sido importado correctamente.
 - Vemos nombre del perfil (192.168.1.88 [S{prueba1}]), que corresponde a la dirección IP del servidor OpenVPN.
 - **Conexión al servidor VPN (IMAGEN 2)**
 - Pulsamos el botón "CONNECT" para establecer la conexión con el servidor OpenVPN.
 - El cliente intentará conectarse usando las credenciales y claves del archivo .ovpn.
 - **Conexión establecida exitosamente (IMAGEN 3)**
 - Mostramos que la conexión está activa.
 - Podemos ver estadísticas de conexión, como velocidad de transferencia de datos, bytes enviados y recibidos, y la duración de la conexión.



- Hacemos una comprobación desde la casa de Rocío.
- La línea 42 **remote masterpruebas.zapto.org 1194** nos indica que el cliente intentará conectarse al servidor **masterpruebas.zapto.org** en el puerto **1194**.
- En la línea 43 descomentada (**;remote my-server-2 1194**), sugiere que hay un servidor alternativo que podría usarse.

```
GNU nano 7.2 files/S{cliente1}.ovpn
38
39 # The hostname/IP and port of the server.
40 # You can have multiple remote entries
41 # to load balance between the servers.
42 remote masterpruebas.zapto.org 1194
43 ;remote my-server-2 1194
44
45 # Choose a random host from the remote
46 # list for load-balancing. Otherwise
47 # try hosts in the order specified.
48 ;remote-random
49
50 # Keep trying indefinitely to resolve the
51 # host name of the OpenVPN server. Very useful
```

- **Configuración del Router:**

- Se muestra la configuración de **OpenVPN en un router**.
- **Protocolo:** UDP
- **Dirección LAN del host (Servidor OpenVPN):** 192.168.1.88
- **Puertos abiertos:** 1194 (para la conexión VPN)
- **WAN Host:** 0.0.0.0 (cualquier IP externa puede conectarse)

Esta configuración permite que el servidor VPN en la IP local 192.168.1.88 sea accesible desde Internet en el puerto 1194.

openvpn ☒ Encendido ☐ Apagado

Tus datos han sido guardados!

Nombre: openvpn

Protocolo: UDP

Conexión WAN: Auto

Dirección IP WAN Host: 0.0.0.0 ~ 0.0.0.0

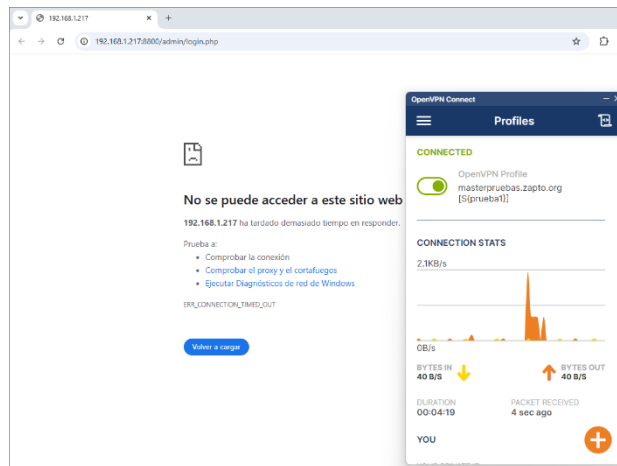
LAN Host: 192.168.1.88

Puerto WAN: 1194 ~ 1194

Puerto de LAN Host: 1194 ~ 1194

- **Prueba de conexión VPN de Rocio y problemas de navegación (Parte inferior derecha)**

- Se muestra que el cliente **OpenVPN** está **conectado** exitosamente.
- Se pueden ver estadísticas de tráfico (bytes enviados y recibidos).



11.CONCLUSIÓN

La implementación de un servidor **VPN** permite crear una conexión segura y privada entre dispositivos remotos y una red interna. Su correcta configuración incluye la instalación del servicio, la generación de certificados, la creación de reglas de firewall y la configuración del cliente. Además, es fundamental verificar el enrutamiento y la conectividad para garantizar su funcionamiento. Una VPN bien configurada mejora la seguridad, protege la privacidad y facilita el acceso remoto seguro a los recursos de una red.