Proyecto Final
Laboratorio Redes y Sistemas Operativos

Integrantes:

Apellido, Nombre	*Legajo*	*Email*
Baron, Elias	58362	eliasnbaron@gmail.com
Bracco, Bautista	59465	bautista.bracco@gmail.com

Profesor:

Jose Luis Di Biase

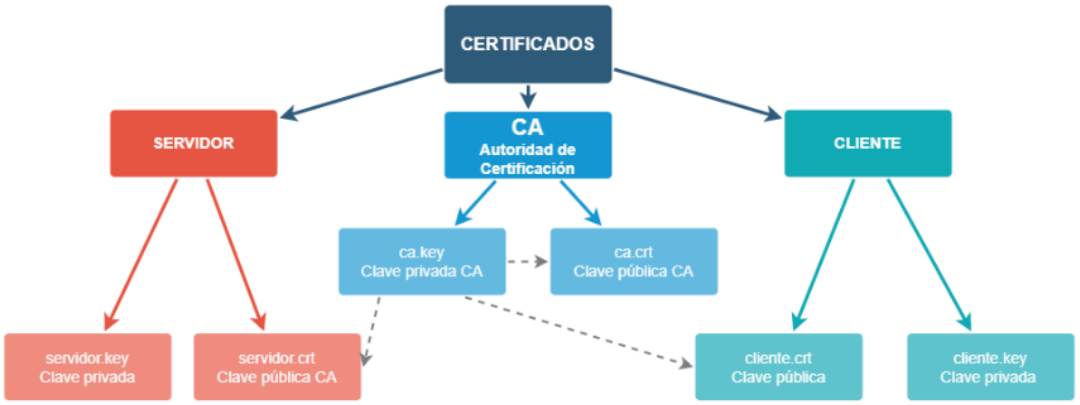
OPEN VPN

Introducción:
Para el proyecto final de la materia, decidimos implementar y probar un servidor de OpenVPN. Para ello, instalamos y configuramos desde cero el servidor, incluyendo el ruteo necesario y los archivos de configuración para el servidor y los clientes que se conectaran más adelante. Este trabajo nos permitió practicar y profundizar en temas como las redes privadas virtuales, la configuración de servidores y la seguridad en redes, pudiendo practicar también lo aprendido en la cursada.

¿Qué es una VPN?
Una VPN (Red Privada Virtual, por sus siglas en inglés) es una herramienta de seguridad que permite a los usuarios conectarse a internet a través de un túnel cifrado. Este túnel protege los datos transmitidos, asegurando la privacidad y confidencialidad de la información, además de hacer que la actividad en línea sea más segura al dificultar su monitoreo o interceptación.

¿Cómo funciona?

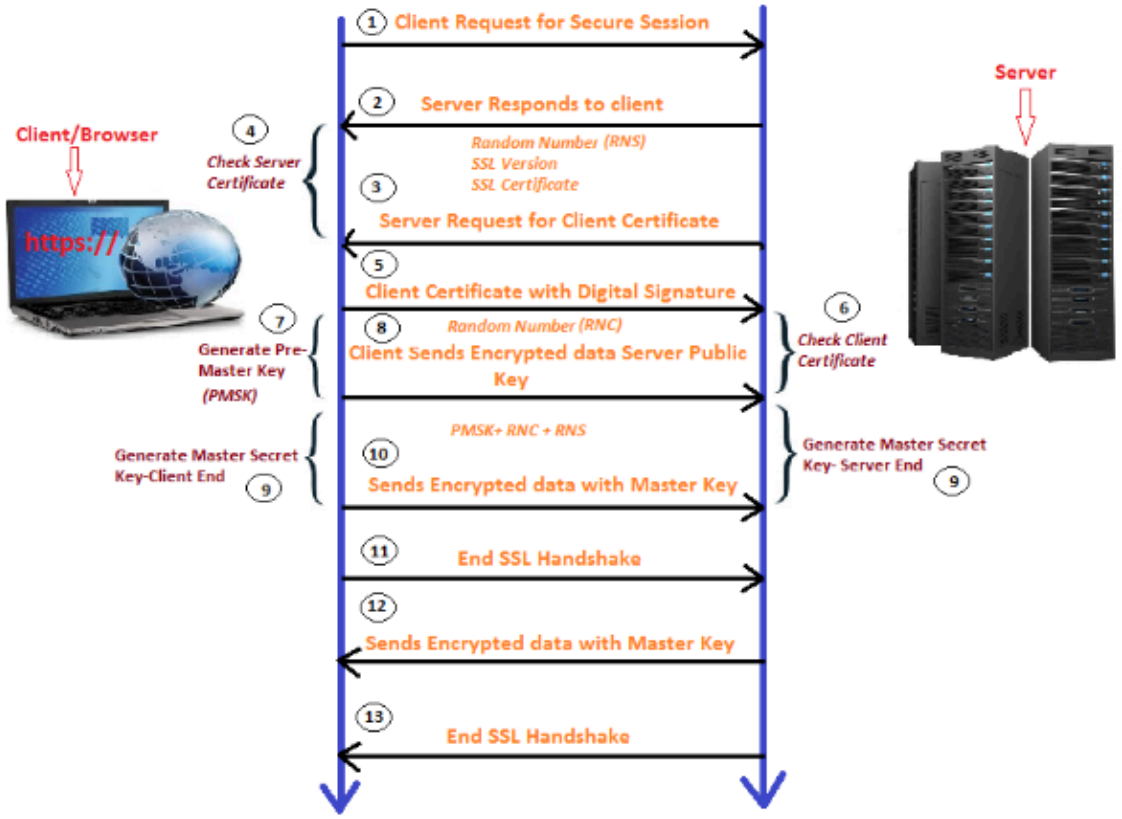
Para que una VPN proteja los datos enviados al servidor, es esencial cifrarlos. Esto se logra mediante una infraestructura de clave pública y clave privada, donde el certificado de autorización desempeña un papel crucial al garantizar la autenticidad de las claves públicas y prevenir modificaciones maliciosas. Además, se puede incorporar una capa adicional de seguridad mediante el uso de TLS (Transport Layer Security), que asegura la integridad y confidencialidad de los datos en tránsito, brindando una protección robusta frente a posibles ataques como la interceptación o la manipulación de información.



Cada equipo (clientes y servidor) debe tener su par de claves (pública y privada), el certificado de la CA (ca.crt) y la clave privada de tls-crypt (ta.key). La clave privada de la CA (ca.key) solo debe estar en el ordenador que se use para firmar y debe estar accesible únicamente para el administrador, ya que su compromiso comprometería toda la seguridad de la infraestructura.

VPN Handshake:

1. El cliente solicita al servidor iniciar una conexión VPN.
2. El servidor responde enviando su clave pública (servidor.crt).
3. El cliente valida el certificado del servidor (servidor.crt) utilizando el certificado de la autoridad certificadora (ca.crt), asegurándose de que la autoridad certificadora sea la correcta.
4. El cliente envía su propia clave pública (cliente.crt) al servidor.
5. El servidor valida el certificado del cliente (cliente.crt) utilizando también el certificado de la autoridad certificadora (ca.crt).
6. Una vez verificadas las claves, el cliente y el servidor confirman la finalización del proceso de handshake, estableciendo así una conexión segura.



Historia de OpenVPN:

OpenVPN fue creado en 2001 por James Yonan con el objetivo de desarrollar una solución de VPN de código abierto que fuera segura, flexible y fácil de implementar. En una época en la que las opciones disponibles eran costosas, complejas o vulnerables, OpenVPN se presentó como una alternativa accesible tanto para pequeñas empresas como para grandes organizaciones.

Contexto de la época:

A comienzos de los años 2000, la necesidad de establecer conexiones seguras a través de internet crecía rápidamente debido a varios factores:

- **Crecimiento del trabajo remoto**: El auge de internet como herramienta de trabajo impulsó a las empresas a buscar soluciones que permitieran a sus empleados acceder de forma remota a redes internas desde cualquier lugar.
- **Limitaciones de las soluciones existentes**: Las VPN más comunes de la época, como PPTP (Point-to-Point Tunneling Protocol) e IPsec (Internet Protocol Security), presentaban varios problemas:
    - **Compatibilidad limitada**: Funcionaban en un número reducido de sistemas operativos.
    - **Configuraciones complejas**: Especialmente en el caso de IPsec, que requería un conocimiento técnico avanzado para su implementación.
    - **Vulnerabilidades de seguridad**: Protocolos como PPTP eran propensos a ataques debido a la falta de cifrado robusto.
    - **Costos elevados**: Muchas soluciones dependían de hardware especializado o licencias costosas, lo que las hacía inaccesibles para pequeñas empresas.
- **Proliferación de redes Wi-Fi públicas**: Con el crecimiento del uso de redes inalámbricas en cafeterías, aeropuertos y otros lugares públicos, surgió la preocupación por la interceptación de datos, lo que incrementó la demanda de herramientas para proteger la información sensible.

Enfoque de OpenVPN:
James Yonan diseñó OpenVPN con una visión clara para superar estas limitaciones:

1. Accesibilidad y código abierto: Crear una solución sin costos de licencias que fuera accesible para cualquier organización, independientemente de su tamaño.

2. Seguridad mejorada: Resolver las vulnerabilidades inherentes a protocolos más antiguos como PPTP, implementando un cifrado sólido basado en TLS/SSL.
3. Facilidad de uso: Simplificar el proceso de configuración en comparación con IPsec, eliminando la necesidad de configuraciones complicadas.
4. Flexibilidad basada en software: Ofrecer una solución independiente de hardware, capaz de funcionar en múltiples sistemas operativos y adaptarse a diferentes entornos de red.

Gracias a este enfoque innovador, OpenVPN se convirtió en una de las herramientas de VPN más populares y confiables, adoptada ampliamente en entornos empresariales y personales.

Funcionalidades:
Algunas de las funcionalidades más comunes

- Acceso a contenido restringido: Permite superar restricciones geográficas para acceder a servicios o contenidos bloqueados en ciertas regiones, como plataformas de streaming o sitios web específicos.
- Anonimato: Oculta la dirección IP real del usuario, proporcionando una IP diferente para navegar de forma más privada.
- Seguridad en redes públicas: Protege los datos del usuario al usar redes Wi-Fi públicas, evitando posibles ataques como el robo de información.
- Interconexión de sucursales: Conecta oficinas o sucursales geográficamente dispersas dentro de una misma red privada, permitiendo compartir recursos de manera segura.
- Trabajo remoto seguro: Facilita el acceso seguro a redes corporativas para trabajadores remotos, protegiendo la comunicación y los recursos internos de la empresa.

Componentes principales

En la configuración de OpenVPN, es fundamental comprender los conceptos y roles de sus principales componentes:

1. **CA (Autoridad Certificadora):**
   Es responsable de generar y firmar certificados digitales que garantizan la autenticidad de los participantes en la VPN. La CA es la raíz de confianza del sistema, y su clave privada debe estar protegida para evitar compromisos de seguridad.

2. **PKI (Infraestructura de Clave Pública):**
   OpenVPN utiliza una PKI para gestionar la autenticación y el cifrado. Incluye la CA, los certificados de clientes y servidores, y las claves privadas necesarias para la comunicación segura.
3. **TLS (Transport Layer Security):**

   TLS agrega una capa adicional de seguridad mediante el cifrado de los datos en tránsito y la negociación de parámetros seguros para la conexión. Durante el proceso de handshake TLS, el cliente y el servidor intercambian certificados y claves públicas, acuerdan una clave compartida para implementar el cifrado simétrico, más eficiente para el tráfico continuo, y establecen el canal seguro utilizando dicha clave. Además, TLS protege contra ataques de tipo MITM (Man-In-The-Middle) y asegura la integridad de los datos transmitidos.

4. **Servidor VPN:**
   Es el componente principal que acepta conexiones de los clientes, valida sus certificados, y establece un canal seguro de comunicación. El servidor gestiona la configuración de red, incluyendo la asignación de direcciones IP y el reenvío de tráfico.
5. **Cliente VPN:**
   Representa a los dispositivos que se conectan al servidor para acceder a la red privada. Cada cliente debe tener un certificado y una clave privada única, además de la clave pública de la CA para verificar al servidor.
6. **Archivos de configuración:**
   Tanto el cliente como el servidor dependen de archivos de configuración que definen parámetros como las rutas, los puertos, los métodos de cifrado y las opciones de autenticación.

Tutorial

Prerrequisitos:

Para comenzar, necesitamos una máquina que esté ejecutando **Ubuntu 24.04.1**. Esta máquina será utilizada para realizar las configuraciones necesarias y funcionará como el servidor principal de nuestra VPN. Es importante asegurarse de que el sistema operativo esté actualizado y cuente con los permisos de administrador para facilitar la instalación y configuración de OpenVPN.

Actualizar el sistema

Para empezar, todos los programas y sistema deben estar actualizados a su versión más reciente:

```
sudo apt update
```

Instalar OpenVPN y Easy-RSA

Como el servidor que vamos a ejecutar es de openvpn, debemos instalar el programa, además de la herramienta easy-rsa para la generación de los certificados esenciales para la configuración:

```
sudo apt install openvpn easy-rsa
```

Inicializar la PKI de Easy-RSA

Una vez que Easy-RSA se encuentra instalado, primero copiamos el directorio de Easy-RSA en una dirección alternativa (Para evitar que se sobreescriba con futuras actualizaciones), luego navegamos a este y inicializamos la Infraestructura de Clave Pública (PKI) del servidor, que se utilizara en el siguiente paso:

```
sudo cp -r /usr/share/easy-rsa /etc/
```

```
cd  /etc/easy-rsa/
```

```
sudo ./easyrsa init-pki
```

Generar la autoridad certificadora

En este paso, generamos el Certificado de Autoridad (CA) y su clave.

```
sudo ./easyrsa build-ca
```

Generar los parámetros de Diffie-Hellman

Con las claves Diffie-Hellman se pueden establecer comunicaciones más seguras con una fuerte encriptación. Sin embargo, este paso se puede saltear:

```
sudo ./easyrsa gen-dh
```

Generar el certificado y la clave del servidor OpenVPN

En este paso, se genera el certificado y la clave para el servidor OpenVPN. Se puede cambiar el nombre de **server** por cualquier otro a elección (Este nombre sirve para que las conexiones de los clientes identifiquen el servidor):

```
sudo ./easyrsa build-server-full server nopass
```

La opción **nopass** desactiva la protección con contraseña.

Generar la clave HMAC

La clave de autenticación TLS/SSL precompartida agrega una capa adicional de seguridad al incluir una firma HMAC en cada paquete de saludo SSL/TLS. Esta medida ayuda a prevenir posibles ataques DoS (Denial of Service) y la sobrecarga de puertos UDP. Aunque no es obligatorio, se aconseja encarecidamente considerar este paso, especialmente si deseas protegerte contra ataques al servidor.

```
sudo openvpn --genkey secret /etc/easy-rsa/pki/ta.key
```

Copiar los certificados y las claves del servidor

En este paso, se copian los certificados y claves generadas del servidor al directorio de configuración del servidor OpenVPN.

```
sudo cp -rp
/etc/easy-rsa/pki/{ca.crt,dh.pem,ta.key,crl.pem,issued,privat
e} /etc/openvpn/server/
```

Generar los certificados y las claves del cliente OpenVPN

Acá se deben crear los certificados y claves para cada cliente que va a conectarse a nuestro servidor OpenVPN. Estos son únicos para cada uno de los clientes y son utilizados para la autenticación. En este ejemplo se realiza con **clientname** pero se debe cambiar por el nombre del cliente.

```
cd /etc/easy-rsa
```

```
sudo ./easyrsa build-client-full clientname nopass
```

Crear directorios para el cliente y copiar los archivos

Una vez generados los archivos claves de los clientes, se debe crear un directorio para cada uno cambiando **clientname** por su nombre, y allí se guardaran los archivos correspondientes al cliente. Generar directorios separados para cada cliente permite asegurar que cada cliente tenga un espacio designado para guardar sus credenciales, y hacer la administración de estas más sencilla.

```
sudo mkdir /etc/openvpn/client/clientname
```

```
sudo cp -rp
/etc/easy-rsa/pki/{issued/clientname.crt,private/clientname.k
ey} /etc/openvpn/client/clientname/
```

Generar archivos ovpn

Para permitir que los clientes se conecten desde dispositivos móviles, vamos a generar un archivo .ovpn. Este archivo combinará toda la información necesaria, como el cliente.crt, cliente.key, ca.crt y ta.key, en un solo archivo, simplificando la configuración y el uso del cliente OpenVPN.

```
cd /etc/openvpn/client
```

```
sudo nano ovpn-writer.sh
```

Y pegar en el archivo el siguiente script

```
#!/bin/sh

##
## Uso: ./ovpn-writer.sh <client.crt> <client.key> > <client.ovpn>
##

client_cert=${1?"The path to the client certificate file is required"}
client_key=${2?"The path to the client private key file is required"}

cat << EOF
client
dev tun
proto udp
remote <Ip del servidor> 1194
resolv-retry infinite
nobind
persist-key
persist-tun
verb 3
keepalive 10 120
auth SHA256
cipher AES-256-CBC
remote-cert-tls server
<ca>
EOF
cat "/etc/openvpn/server/ca.crt"
cat << EOF
</ca>
<cert>
EOF
cat ${client_cert}
cat << EOF
</cert>
<key>
EOF
cat ${client_key}
cat << EOF
</key>
```

```
<tls-crypt>
EOF
cat "/etc/openvpn/server/ta.key"
cat << EOF
</tls-crypt>
EOF
```

Configurar el servidor OpenVPN

Realizamos una copia de un modelo de archivo de configuración del servidor dentro de la carpeta de nuestro servidor OpenVPN.

```
sudo cp
/usr/share/doc/openvpn/examples/sample-config-files/server.co
nf /etc/openvpn/server/
```

Editar el archivo de configuración del servidor

Y lo editamos cambiando especialmente los parámetros marcados con verde, ingresando los datos de nuestro servidor. Los demás se pueden cambiar dependiendo de la configuración que queramos o dejarlos por defecto.

```
sudo nano /etc/openvpn/server/server.conf
```

```
port 1194
proto udp4
dev tun
ca ca.crt  # Dirección del certificado de autoridad
cert issued/server.crt # Dirección del certificado del sv
key private/server.key # Dirección de la llave del sv
dh dh.pem
topology subnet
server 172.16.20.0 255.255.255.0
ifconfig-pool-persist /var/log/openvpn/ipp.txt
push "redirect-gateway def1 bypass-dhcp"
client-to-client
keepalive 10 120
tls-crypt ta.key # Dirección de archivo ta.key
cipher AES-256-CBC
```

```
persist-key
persist-tun
status /var/log/openvpn/openvpn-status.log
log-append /var/log/openvpn/openvpn.log
verb 3
explicit-exit-notify 1
auth SHA512
max-clients 50
```

Habilitar el reenvío de IP y configurar el firewall

En este paso, habilitaremos el reenvío de IP en la configuración de **sysctl**. Esto es necesario para que el servidor pueda enrutar el tráfico entre sus interfaces, algo fundamental para el funcionamiento adecuado de la VPN.

Ejecuta el siguiente comando para modificar el archivo de configuración y habilitar el reenvío de IP:

```
sudo sed -i 's/#net.ipv4.ip_forward=1/net.ipv4.ip_forward=1/' /etc/sysctl.conf
```

```
sudo sysctl --system
```

Permite el tráfico en el puerto **UDP 1194**, que es el puerto predeterminado usado por OpenVPN:

```
sudo ufw allow 1194/udp
```

Configurar el enmascarado de IP

El enmascaramiento de IP es una técnica utilizada para enrutar tráfico de una red a otra, lo cual es esencial para que los clientes de la VPN puedan acceder a otros recursos de la red. En este paso, actualizaremos las reglas y políticas del firewall (UFW) para que el enmascaramiento funcione correctamente.

Primero con este comando se visualizará la interfaz de red que el servidor utiliza para enrutar el tráfico.

```
ip route get 8.8.8.8
```

Ejemplo de output:

```
tutorial@tutorial:/etc/easy-rsa$ ip route get 8.8.8.8
8.8.8.8 via 10.0.2.2 dev enp0s3 src 10.0.2.15 uid 1000
    cache
```

Una vez que sepamos cual es la interfaz, se debe agregar la siguiente regla en el archivo before.rules:

```
sudo nano /etc/ufw/before.rules
```

y allí se agregara la siguiente regla:

```
*nat
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s ipservidor/24 -o interfaz -j MASQUERADE
COMMIT
```

Luego, cambia la política de reenvío de "DROP" (bloquear) a "ACCEPT" (permitir) para que el tráfico reenviado por la VPN fluya correctamente.

```
sudo sed -i 's/DEFAULT_FORWARD_POLICY="DROP"/DEFAULT_FORWARD_POLICY="ACCEPT"/' /etc/default/ufw
sudo ufw reload
```

Iniciar el servidor OpenVPN

Inicializamos el servicio del servidor OpenVPN:

```
sudo systemctl enable --now openvpn-server@server
```

Luego, chequeamos el estado del servicio:

```
sudo systemctl status openvpn-server@server
```

Deberíamos visualizar una salida parecida a la siguiente:

```
tutorial@tutorial:/etc/easy-rsa$ systemctl status openvpn-server@server
● openvpn-server@server.service - OpenVPN service for server
     Loaded: loaded (/lib/systemd/system/openvpn-server@.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2023-08-07 11:57:53 +0530; 5s ago
       Docs: man:openvpn(8)
             https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
             https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 437986 (openvpn)
     Status: "Initialization Sequence Completed"
      Tasks: 1 (limit: 4487)
     Memory: 1.8M
        CPU: 33ms
     CGroup: /system.slice/system-openvpn\x2dserver.slice/openvpn-server@server.service
             └─437986 /usr/sbin/openvpn --status /run/openvpn-server/status-server.log --status-versi
on 2 --suppress-timestamps --config server.conf

අගෝ 07 11:57:53 tutorial systemd[1]: Starting OpenVPN service for server...
අගෝ 07 11:57:53 tutorial systemd[1]: Started OpenVPN service for server.
```

Si todo está en orden, ¡felicitaciones! Tu servidor OpenVPN está operativo y listo para aceptar conexiones de clientes. 🎉

Conexión de los clientes al servidor OpenVPN

Una vez configurado el servidor y generado el archivo .ovpn para cada cliente, puedes usarlo para establecer la conexión desde distintos sistemas operativos. A continuación, se describe cómo conectarse en Windows, Linux y Android.

Windows

1. **Instalar el cliente OpenVPN:**
   - Descarga e instala el cliente oficial de OpenVPN desde su sitio web: OpenVPN
2. **Importar el archivo .ovpn:**
   - Copia el archivo .ovpn al directorio de configuración del cliente (normalmente C:\Program Files\OpenVPN\config\).
   - Alternativamente, puedes importarlo desde la interfaz gráfica del cliente OpenVPN.
3. **Establecer la conexión:**
   - Ejecuta el cliente OpenVPN como administrador.
   - En la bandeja del sistema, haz clic derecho sobre el icono de OpenVPN y selecciona el perfil importado.
   - Haz clic en **Conectar**.
4. **Verificar la conexión:**
   - Una vez conectado, verifica que tienes acceso a los recursos del servidor o que tu IP pública ha cambiado.

Linux

1. **Instalar OpenVPN:**
   - En distribuciones basadas en Debian/Ubuntu:
     `sudo apt install openvpn`
   - En distribuciones basadas en Red Hat/Fedora:
     `sudo dnf install openvpn`
2. **Conectarse al servidor:**
   - Coloca el archivo .ovpn en un directorio, por ejemplo, ~/openvpn/.
   - Usa el siguiente comando para iniciar la conexión:
     `sudo openvpn --config ~/openvpn/tu-archivo.ovpn`

3. **Verificar la conexión:**
    - Una vez conectado, puedes comprobar tu nueva IP pública o el acceso a los recursos de la VPN.

Android

1. **Instalar la aplicación OpenVPN:**
    - Descarga la aplicación **OpenVPN Connect** desde la Google Play Store.
2. **Importar el archivo .ovpn:**
    - Transfiere el archivo .ovpn a tu dispositivo Android.
    - Abre la aplicación OpenVPN Connect y selecciona **Importar perfil**.
    - Navega hasta el archivo .ovpn y selecciónalo.
3. **Establecer la conexión:**
    - Toca el perfil importado y selecciona **Conectar**.
    - Acepta los permisos solicitados.
4. **Verificar la conexión:**
    - Comprueba el estado en la aplicación para confirmar que la conexión está activa.

Notas adicionales:

- Asegúrate de que el archivo .ovpn contenga toda la configuración necesaria, como claves y certificados.
- En redes corporativas o restringidas, es posible que necesites configurar el cliente para usar un puerto alternativo si el puerto UDP predeterminado (1194) está bloqueado.
- Recuerda desconectar la VPN cuando ya no la necesites para evitar el consumo innecesario de recursos.