

Taller: Un Túnel Privado Rápido y Sencillo con OpenVPN

EL TÚNEL DE DATOS

Los cortafuegos a veces prohíben todo el tráfico excepto la navegación por la web, impidiendo que los usuarios puedan utilizar los servicios de IRC o los servidores de difusión, a menos que se utilice una red privada virtual como OpenVPN. **POR MIRKO DÖLLE**

Una configuración típica del cortafuegos consiste en denegar todo lo que no sea estrictamente necesario para el trabajo diario. Incluso herramientas relativamente inofensivas como una webcam o un servidor personal de IRC no funcionarán a través del cortafuegos. Aparte de rogarle al administrador que cambie las reglas, la única solución que queda es abrir un túnel privado a través del cortafuegos con OpenVPN.

En este artículo describimos cómo realizar ese túnel con una conexión VPN. Suponemos que el software de OpenVPN ya está instalado en la máquina Linux o que los lectores saben cómo obtenerlo e instalarlo. OpenVPN es una aplicación muy común que se incluye en la mayoría de las distribuciones Linux populares. Véase la documentación para saber más sobre cómo configurar OpenVPN.

Minando el Cortafuegos

OpenVPN no requiere los privilegios de root para establecer una VPN. Sólo con que el programa tenga acceso a los dispositivos virtuales TUN/TAP, es suficiente para funcionar con los privilegios del usuario. En un escenario simple, todo lo que se necesita son unos cuantos parámetros para configurar la VPN. El único fichero que se necesita es el que contiene la clave secreta, que se puede

Listado 1: openvpn-server.sh

```
01 #!/bin/bash
02
03 DEVICE="tun0"
04 PORT="1194"
05 LOCALIP="192.168.8.1"
06 REMOTEIP="192.168.8.128"
07 KEYFILE="/etc/openvpn/shared.key"
08 MAXRATE="16000"
09
10 /usr/sbin/openvpn -daemon --dev $DEVICE \
11 --proto tcp-server -port $PORT \
12 --ifconfig $LOCALIP $REMOTEIP \
13 --secret $KEYFILE --persist-tun --ping 30 \
14 --ping-restart 180 -shaper $MAXRATE \
15 --writepid /var/run/openvpn-${DEVICE}.pid
16
17 while true; do
18 if [ ! -e /var/run/openvpn-${DEVICE}.pid ]; then
19 break
20 fi
21 done
```

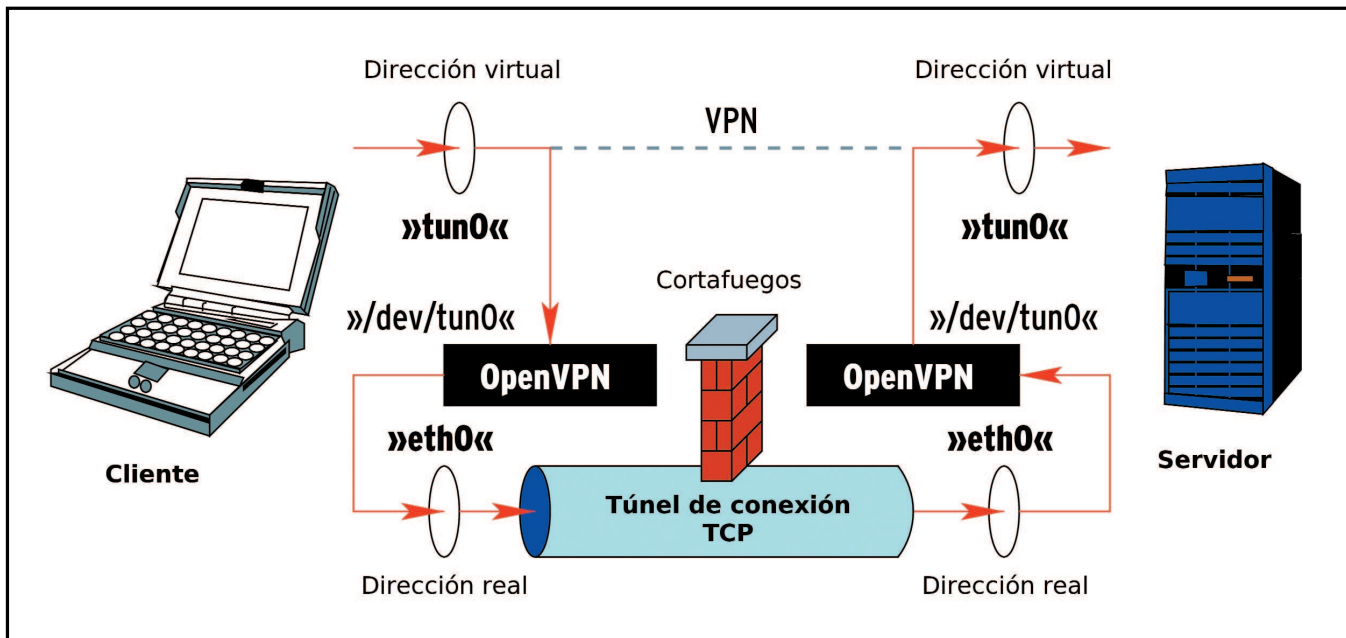


Figura 1: Utilizando TCP los clientes configuran una conexión encriptada al servidor OpenVPN y transmiten por el túnel cualquier otro protocolo hacia el servidor.

crear en la máquina cliente con el comando `openvpn --genkey --secret secret.key` para posteriormente copiarlo en el servidor.

Un protocolo no orientado a la conexión, como UDP, es la mejor opción para realizar el túnel con TCP/IP: esto evita que se dé el escenario en el que pueda producirse un timeout simultáneo entre OpenVPN y la conexión TCP, que podría ocasionar una avalancha de paquetes de reintento. Desafortunadamente, la mayoría de los cortafuegos y los routers de las empresas o de los cibercafés no permiten que los clientes puedan recibir paquetes UDP desde Internet.

La única alternativa en este caso es optar por una conexión TCP. Esto implica que el servidor, para que el cliente pueda establecer la conexión VPN, ha de estar escuchando en un puerto determinado a la espera de conexiones entrantes.

Una Configuración Simple

Los Listados 1 y 2 muestran dos scripts para el servidor y para el cliente. OpenVPN no requiere ficheros de configuración, por lo que, la mejor opción para el establecimiento de la configuración del túnel se realiza a través de la línea de comandos mediante sus parámetros. Al principio de los Listados 1 y 2 se muestran algunas opciones que los usuarios tienen que cambiar para adaptarlo a sus necesidades.

El servidor no necesita mucha información: OpenVPN requiere la dirección IP del servidor (*LOCALIP*) y de la máquina cliente (*REMOTEIP*) de las interfaces VPN, el dispositivo del túnel, el número del puerto y el fichero con la clave. La variable *MAXRATE* tan sólo es necesaria para controlar el tráfico y establecer la tasa de transferencia máxima para la transmisión de paquetes medida en bytes por segundos.

Controlando el Tráfico

Si el script del servidor se está ejecutando en un servidor raíz, por ejemplo, puede que al principio no tenga mucho sentido el control de tráfico, a menos que se sea partidario de mantener un perfil bajo y esconder el túnel en el tráfico general de la red.

La principal ventaja para el control del tráfico se encuentra en las pequeñas redes con una conexión DSL. Como el ancho de banda de subida es normalmente mucho más bajo que el ancho de banda de bajada en las líneas DSL, el resto de conexiones a Internet diferentes del túnel se pueden ver afectadas. Yo suelo usar un límite de 16 kbps, como se muestra en el Listado 1, para una conexión DSL con un ancho de banda máximo de subida de 256KBits/s, estableciendo este límite me permite disponer casi de la mitad del ancho de banda para otros servicios como Apache o simplemente para navegar por la red.

La sintaxis de OpenVPN desde las líneas 10 a la 15 hace uso de las variables definidas al principio. Los parámetros críticos aquí son `--daemon`, `--proto tcp-server` y `--persist-tun`. `--daemon` ejecuta OpenVPN en modo servicio, permitiendo al script del servidor continuar su ejecución. Esto no es importante en el Listado 1, ya que el bucle que va desde la línea 17 a la 21 impide que finalice el script antes de que finalice OpenVPN. Sin embargo, si necesita abrir el cortafuegos o cambiar los parámetros del router, las líneas desde la 15 a la 17 son el mejor lugar para realizarlo. Se puede añadir cualquier comando al final del script si se quiere ejecutar alguna acción final.

Protocolos del Cliente y del Servidor

El parámetro `--proto tcp-server` habilita OpenVPN en el servidor y le indica que se ponga a la escucha de las conexiones TCP entrantes. El servicio del cliente usa el protocolo `tcp-client`, como se muestra en la línea 13 del Listado 2.

Sin `--persist-tun`, el interfaz TUN estaría cerrado y sería reabierto cada vez que el túnel fuera interrumpido, lo que implicaría la pérdida de las entradas de enrutamiento del túnel y la pérdida de todas las conexiones.

El bucle de las líneas 17 a 21 del Listado 1 es una versión simplificada si se compara con el bucle del script del listado en [2]; carece del mecanismo de

Listado 2: `openvpn-client.sh`

```

01 #!/bin/bash
02 DEVICE="tun0"
03 REMOTE="athome.dyndns.org"
04 GATEWAY="192.168.1.254"
05 PORT="1194"
06 LOCALIP="192.168.8.128"
07 REMOTEIP="192.168.8.1"
08 REMOTENET="192.168.42.0/24"
09 KEYFILE="/etc/openvpn/shared.key"
10 MAXRATE="16000"
11
12 /usr/sbin/openvpn -daemon --dev $DEVICE \
13 --remote $REMOTE -proto tcp-client \
14 --port $PORT --ifconfig $LOCALIP $REMOTEIP \
15 --secret $KEYFILE --persist-tun --ping 30 \
16 --ping-restart 180 \
17 --writepid /var/run/openvpn-${DEVICE}.pid
18
19 for ((i=0; i<10; i=${i+1})); do
20   ifconfig $DEVICE >/dev/null 2>/dev/null
21   if [ "$?" -eq 0 ]; then
22     route add $REMOTE gw $GATEWAY
23     if [ "$?" -eq 0 ]; then
24       route del default gw $GATEWAY
25       route add -net $REMOTENET gw $REMOTEIP
26       route add default gw $REMOTEIP
27       ping -c 1 $REMOTEIP >/dev/null 2>/dev/null &
28       break
29     fi
30   fi
31   sleep 5
32 done
33
34 while true; do
35   if [ ! -e /var/run/openvpn-${DEVICE}.pid ]; then
36     break
37   fi
38 done
39
40 route del $REMOTE gw $GATEWAY
41 route add default gw $GATEWAY

```

control de fallos que verifica si OpenVPN está aún ejecutándose. En el caso de que el bucle finalice el script procesará el código final de limpieza tras la línea 21. `openvpn-server.sh` puede ser invocado fácilmente usando una entrada en `init` en `/etc/inittab`.

El script cliente para OpenVPN del Listado 2 es el doble de largo que el script del servidor, ya que tiene que configurar el entorno del router de la máquina cliente después de llamar a OpenVPN en las líneas 12 a 17. Esto supone que OpenVPN ha de establecer el dispositivo de red, que se verifica comprobando el estado de salida de `ifconfig` en las líneas 20 y 21.

Ahora que el dispositivo del túnel está operativo, la línea 22 establece una ruta estática al servidor OpenVPN.

La conexión VPN depende ahora de la ruta por defecto, así que la línea 24 del

script puede borrar dicha ruta, que posteriormente vuelve a establecerse en la línea 26 usando la puerta de enlace del servidor OpenVPN. A partir de este punto, cualquier conexión de red nueva será enrutada a través de la VPN hacia el servidor (Figura 1). OpenVPN acaba de abrir ahora una brecha en el cortafuego de la empresa y la máquina cliente dispone desde este momento de un acceso a Internet oculta y sin restricciones. El comando ping de la línea 27 abre un túnel VPN, que se ha configurado pero que aún permanece cerrado en este punto.

Enrutado Individual

Para impedir que el script cliente sea obstruido por problemas de enrutamiento, los comandos de enrutado en el bucle desde las líneas 19 a 32 se repiten diez veces como mucho.

El bucle en las líneas 34 a 38 está reducido como ocurre en el script del servidor del Listado 1, aunque esto no tiene efecto en la funcionalidad: espera que OpenVPN termine. Las líneas 40 y 41 borran después la ruta estática al servidor OpenVPN y reestablecen la puerta de enlace por defecto.

La solución para establecer un túnel OpenVPN mostrado en este artículo requiere un servidor para cada cliente VPN. Este requerimiento podría quedar fuera de control fácilmente si se necesitan soportar múltiples clientes o si la red engloba varias oficinas de la empresa. Para los usuarios que consideren requerimientos mayores, OpenVPN versión 2.0 o posterior proporcionan infraestructura de Clave Pública, CAs y soporte para múltiples clientes para escenarios más avanzados. Pero para un servidor personal de IRC o para una webcam, una configuración simple basada en la línea de comandos tal como la descrita en este artículo, es suficiente. ■



RECURSOS

- [1] OpenVPN: <http://openvpn.net>
- [2] Scripts OpenVPN: <http://www.mirko-doelle.de/linux/openvpn-server.sh> y ... / [openvpn-client.sh](#)