OpenVPN - Cómo configurar un servidor Slackware y un cliente Slackware

1. Introducción

1.1. OpenVPN(1)

OpenVPN es una aplicación de software de código abierto que implementa técnicas de red privada virtual (VPN) para crear conexiones seguras punto a punto o sitio a sitio en configuraciones enrutadas o puenteadas y facilidades de acceso remoto. Utiliza un protocolo de seguridad personalizado que utiliza SSL/TLS para el intercambio de claves. Es capaz de atravesar traductores de direcciones de red (NATs) y cortafuegos. Fue escrito por James Yonan y está publicado bajo la GNU General Public License (GPL).

OpenVPN permite a los pares autenticarse entre sí utilizando una clave secreta precompartida, certificados o nombre de usuario/contraseña. Cuando se utiliza en una configuración multicliente-servidor, permite que el servidor emita un certificado de autenticación para cada cliente, utilizando la firma y la autoridad de certificación. Utiliza extensivamente la biblioteca de cifrado OpenSSL, así como el protocolo SSLv3/TLSv1, y contiene muchas funciones de seguridad y control.

2. Ámbito de aplicación y objetivo

El objetivo de este artículo es servir como un tutorial para que los lectores configuren un servidor y un cliente VPN en Slackware básicos pero funcionales a través de Internet.

El énfasis está en proporcionar un método fiable que pueda seguirse fácilmente para configurar OpenVPN en servidores y clientes Slackware. Sin embargo, el proceso no está libre de obstáculos y requiere cierta atención y determinación.

Este artículo incluye una selección de otros tutoriales similares que se encuentran en Internet, en particular (2) y (3) y los documentos contenidos en los archivos fuente descargados. Sin embargo, se han reformado para satisfacer el objetivo.

3. Instalación

OpenVPN ya está instalado en Slackware si se ha seguido una instalación por omisión. Si este no fuera el caso, entonces el paquete está disponible en el directorio “n” del DVD de Slackware. Consulte otros documentos específicos de Slackware sobre cómo realizar esta instalación.

Si desea confirmar que OpenVPN está instalado, puede comprobarlo listando el directorio /var/log/packages/:

```
# ls /var/log/packages/openvpn*
```

4. Requisitos

Se necesitarán dos ordenadores que cumplan la funciones de servidor y de cliente. Tendrían que estar conectados a Internet a dos enrutadores y con rutas de red diferentes. Para el propósito de este tutorial, se definen detalles específicos con el fin de mejorar la legibilidad. Por supuesto, es probable que tenga direcciones diferentes, por lo que tendrá que modificarlas en consecuencia.

4.1. Servidor DNS

Normalmente se utiliza una URL para dirigirse al servidor. Esto no es obligatorio y en su lugar puede utilizar sólo la IP de Internet. Sin embargo, se recomienda utilizar una URL para acceder al servidor desde Internet, especialmente si está conectado a una IP dinámica, lo que es típico en las conexiones domésticas a Internet. El autor utiliza noip2(4) ya que es gratuito al suscribirse. Un slackbuild noip2 está disponible en http://slackbuilds.org.

4.2. Detalles del servidor

```
hostname: server1
IP: 192.168.200.195/255.255.255.0
URL:  servervpn.no-ip.org
Network Interface: eth0
```

4.3. Detalles del cliente

```
hostname: client1
IP: 192.168.1.101/255.255.255.0
Network Interface: wlan0
```

4.3 Derechos de Administrador

Necesitará tener derechos de administrador para configurar OpenVPN. Esto se aplica tanto al servidor como al cliente. Para simplificar, en este tutorial, se asumirá que todas las acciones serán realizadas por el usuario root. Naturalmente, los usuarios avanzados pueden ser más exigentes.

4.4 Posibles limitaciones y soluciones para un cliente equipado con WiFi

La disponibilidad de dos enutadores puede ser un reto. Tenga en cuenta que se necesitarán sesiones interactivas tanto en el servidor como en el cliente antes de configurar la VPN. Si el cliente está equipado con una interfaz WiFi puede haber algunas soluciones fáciles que se pueden considerar:

1. Utilice la función “Portable Wi-Fi Hot Spot” de un teléfono inteligente 3G para conectar al cliente como cliente VPN. Como el ancho de banda 3G es caro, es posible que desee minimizar el tráfico. Por ejemplo, es posible que desee desactivar servicios que no son absolutamente

esenciales durante el curso de este ejercicio, como ntpd, dropbox y tor.
2. Conecte el cliente a otra conexión WiFi disponible en las proximidades del servidor. Algunas personas afortunadas viven en áreas donde los vecinos benévolos les proporcionan acceso abierto a Internet WiFi. Se recomienda solicitar permiso antes de aceptar esta solución. En caso de que no exista tal servicio abierto, puede ser conveniente solicitar una contraseña temporal a un vecino amistoso para el servicio WiFi privado encriptado.
3. Hoy en día, muchos locales gubernamentales, como bibliotecas y ayuntamientos, ofrecen servicio gratuito de WiFi. Otros lugares como restaurantes de comida rápida, pubs, cafés, etc. también ofrecen WiFi gratuito desde su ubicación a sus apreciados clientes. Puede acceder al servidor a través de un servicio disponible como SSH desde un cliente equipado con WiFi. Si se elige esta opción para esta solución, tenga en cuenta que el cliente puede tener que pasar a través de algunos cortafuegos. Además, la conexión VPN puede constituir un incumplimiento de los términos de las condiciones que deben aceptarse antes de utilizar el servicio WiFi.

5. Creación de una infraestructura de clave pública (PKI) utilizando los scripts easy-rsa

La PKI se puede crear en cualquier ordenador con una instalación VPN, pero probablemente sea más sensato hacerlo tanto en el servidor como en el cliente, ya que ambos lo necesitarían. Una forma sencilla de crear la PKI es utilizar los scripts easy-rsa. Estos pueden ser descargados de la siguiente manera:

```
# cd
# git clone http://github.com/OpenVPN/easy-rsa
```

y luego archivarlo para usos futuros:

```
# tar cvf easy-rsa.tar easy-rsa
```

5.1 Cree las claves y certificados para el servidor

Siga estos pasos en el servidor para crear las claves y certificados necesarios:

```
# cd easy-rsa/easyrsa3
```

Cree la PKI y la CA:

```
# ./easyrsa init-pki
# ./easyrsa build-ca
```

Introduzca una contraseña PEM, verifíquela y, a continuación, introduzca un nombre para el servidor. En este artículo estoy usando los nombres de host para mayor claridad (en este caso: server1), pero puedes elegir cualquier nombre.

A continuación, genere la solicitud:

```
# ./easyrsa gen-req server1
```

Se le pedirá otra contraseña de PEM para reverificarla y confirmar que el nombre de la entidad es servidor1. Ahora puede proceder a firmar esta solicitud:

```
# ./easyrsa sign-req server server1
```

Confirme la solicitud introduciendo “sí” y, a continuación, introduzca la frase de contraseña original CA PEM.

Ahora cree dos archivos de claves adicionales:

```
# cd /etc/openvpn/certs/
# openssl dhparam -out dh2048.pem 2048
# cd /etc/openvpn/keys/
# /usr/sbin/openvpn --genkey --secret ta.key
```

5.2 Cree las claves y certificados para el cliente

Siga estos pasos en el cliente para crear las claves y certificados necesarios:

Necesitará los scripts easy-rsa, para poder copiar el tarball easy-rsa del servidor al cliente y extraerlo:

```
# cd
# tar xvf easy-rsa.tar
```

Ahora cree la PKI y genere la solicitud:

```
# cd easy-rsa/easyrsa3
# ./easyrsa init-pki
# ./easyrsa gen-req client1
```

Se le pedirá otra contraseña PEM, para volver a verificarla y confirmar que el nombre de la entidad es realmente cliente1. En este artículo estoy usando los nombres de host para mayor claridad (en este caso: cliente1), pero puede elegir cualquier nombre.

Copie pki/reqs/client1.req al servidor.

5.2.1 Firme la solicitud del cliente en el servidor

Para los fines de este artículo, se asume que el archivo de solicitud del cliente (client1.req) ha sido transferido al directorio $HOME/openvpn/ del servidor. Ahora puede proceder a importar y firmar la solicitud de cliente1:

```
# cd $HOME/easy-rsa/easyrsa3
# ./easyrsa import-req $HOME/openvpn/client1.req client1
# ./easyrsa sign-req client client1
```

Cuando se le solicite, introduzca “sí” y la contraseña CA PEM del servidor1.

Copie el archivo $HOME/easy-rsa/easy-rsa3/pki/issued/client1.crt generado de vuelta al cliente.

6. Configuración del servidor

Copie los siguientes archivos generados por los scripts easy-rsa a sus respectivos directorios en el directorio /etc/openvpn/:

```
# cp $HOME/easy-rsa/easyrsa3/pki/ca.crt \
> /etc/openvpn/certs/
# cp $HOME/easy-rsa/easyrsa3/pki/issued/server1.crt \
> /etc/openvpn/certs/
# cp $HOME/easy-rsa/easyrsa3/pki/private/server1.key \
> /etc/openvpn/keys/
```

Copie el archivo server.conf de la fuente OpenVPN en el directorio de configuración de OpenVPN. La fuente de OpenVPN puede obtenerse en el DVD de Slackware o en su réplica favorita de Slackware o en http://openvpn.net. En el siguiente ejemplo, estoy descargando el código fuente de ftp.slackware.com

```
# cd /tmp/
# wget -c \
>
ftp://ftp.slackware.com/pub/slackware/slackware/source/n/openvpn/openvpn-*.t
ar.?z
# cd /usr/src/
# tar xvf /tmp/openvpn-*.tar.?z
```

Copie el archivo server.conf contenido en el código fuente en el directorio de configuración de OpenVPN:

```
# cp openvpn-*/sample/sample-config-files/server.conf \
> /etc/openvpn/
```

Edite las siguientes líneas de /etc/openvpn/server.conf

De estas líneas:

```
ca ca.crt
cert server.crt
key server.key  # Este archivo debe mantenerse en secreto


dh dh1024.pem
;tls-auth ta.key 0 # Este archivo es secreto

;user nobody
;group nobody
```

```
;log-append  openvpn.log
```

A:

```
ca /etc/openvpn/certs/ca.crt
cert /etc/openvpn/certs/server1.crt
key /etc/openvpn/keys/server1.key #Este archivo debe mantenerse en secreto

dh /etc/openvpn/certs/dh2048.pem

tls-auth /etc/openvpn/keys/ta.key 0 # Este archivo es secreto

user nobody
group nobody

log-append  /var/log/openvpn.log
```

Finalmente agregue lo siguiente a /etc/openvpn/server.conf:

```
# Seleccione un cifrado criptográfico.
# Este elemento de configuración debe ser copiado
# al archivo de configuración del cliente también.
cipher AES-256-CBC
# Si desea utilizar OpenVPN como demonio, descomente esta línea.
# En general, los servidores deberían ejecutar OpenVPN como un demonio
;daemon
```

Un lector cuidadoso puede estar tentado de descomentar la opción “demonio”. Esto no permitiría al usuario introducir la contraseña, por lo que no funcionaría hasta que la contraseña esté definida en server.conf como se describe en el Capítulo 10.

```
# cat /var/log/openvpn.log
```

Tenga en cuenta que los comentarios en server.conf pueden comenzar con # o ; Con el fin de ayudarle a introducir parámetros, los primeros se utilizan para comentar el texto, mientras que los segundos son para las líneas de configuración comentadas.

Copie el archivo rc.openvpn listado a continuación y colóquelo bajo /etc/rc.d/

```
#!/bin/sh
#
# /etc/rc.d/rc.openvpn
#
# Start/stop/restart the OpenVPN server.
#

ovpn_start() {
```

```
  if [ -x /usr/sbin/openvpn -a -r /etc/openvpn/server.conf ]; then
    echo "Starting OpenVPN:  /usr/sbin/openvpn server.conf"
    /usr/sbin/openvpn /etc/openvpn/server.conf
  fi
}

ovpn_stop() {
  killall openvpn
}

ovpn_restart() {
  ovpn_stop
  sleep 2
  ovpn_start
}

case "$1" in
'start')
  ovpn_start
  ;;
'stop')
  ovpn_stop
  ;;
'restart')
  ovpn_restart
  ;;
*)
  echo "Usage: $0 {start|stop|restart}"
esac
```

Entonces, dele permisos de ejecución:

```
# chmod 755 /etc/rc.d/rc.openvpn
```

7. Reenvío de puertos

Necesitará reenviar el tráfico desde el puerto que ha elegido para que OpenVPN sea enrutado al servidor. Para lograrlo, deberá proporcionar a su servidor una IP fija y deberá configurar su enrutador. Puede usar netconfig, network-manager o wicd para establecer la IP fija en Slackware. A continuación, también deberá consultar la documentación suministrada con el enrutador para configurar la dirección IP seleccionada reservada para el servidor y el reenvío de puertos. Para nuestra configuración predeterminada de OpenVPN, el puerto UDP sería 1194.

En caso de que haya extraviado dicha documentación, puede buscar en Internet la forma de conseguirlo. Un buen punto de partida es http://portforward.com/.

8. Configuración del cliente

En el equipo de cliente, siga las siguientes instrucciones para configurarlo.

Descargue el tarball de código fuente de OpenVPN y extráigalo como se explica en el Capítulo 6, luego proceda a copiar el archivo de configuración incluido para los clientes:

```
# cp /usr/src/openvpn-*/sample/sample-config-files/client.conf \
> /etc/openvpn/
```

Edite las siguientes líneas de /etc/openvpn/client.conf

```
remote my-server-1 1194

;user nobody
;group nobody

ca ca.crt
cert client.crt
key client.key

;tls-auth ta.key 1
```

a las siguientes líneas:

```
remote servervpn.no-ip.org 1194

user nobody
group nobody

ca /etc/openvpn/certs/ca.crt
cert /etc/openvpn/certs/client1.crt
key /etc/openvpn/keys/client1.key

tls-auth /etc/openvpn/keys/ta.key 1
```

Finalmente agregue lo siguiente a /etc/openvpn/client.conf:

```
# Select a cryptographic cipher.
# This config item must be copied to
# the server config file as well.
cipher AES-256-CBC
```

Tenga en cuenta que los comentarios en client.conf pueden ser # o ; Los primeros se utilizan para comentar el texto, mientras que los segundos son para las líneas de configuración comentadas. Esto le ayudará mucho en el proceso de configuración.

Necesitará este archivo generado por los scripts easy-rsa del cliente:

```
cp $HOME/easy-rsa/easyrsa3/pki/private/client1.key \
> /etc/openvpn/keys/
```

y lo siguiente de los scripts easy-rsa del servidor:

```
$HOME/easy-rsa/easyrsa3/pki/ca.crt
$HOME/easy-rsa/easyrsa3/pki/issued/client1.crt
```

y este archivo también:

```
/etc/openvpn/keys/ta.key
```

Coloque estos archivos como se indica en client.conf. Así que ca.crt y client1.crt van bajo /etc/openvpn/certs/ mientras que client1.key y ta.key van bajo /etc/openvpn/keys/

9. Ensayo de la VPN

En el servidor:

```
# /etc/rc.d/rc.openvpn start
```

Introduzca la contraseña PEM del servidor cuando se le solicite.

En el cliente:

```
# /usr/sbin/openvpn /etc/openvpn/client.conf
```

Introduzca la contraseña PEM del cliente cuando se le solicite. Para detener OpenVPN en el cliente, pulse CTRL+C

En ambos debería ver una nueva interfaz de red llamada tun0. En el servidor, obtuve lo siguiente:

```
# ifconfig tun0
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.8.0.1  netmask 255.255.255.255  destination 10.8.0.2
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen
100  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Del mismo modo, en el cliente:

```
# ifconfig tun0
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.8.0.6  netmask 255.255.255.255  destination 10.8.0.5
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen
```

```
100  (UNSPEC)
       RX packets 0  bytes 0 (0.0 B)
       RX errors 0  dropped 0  overruns 0  frame 0
       TX packets 0  bytes 0 (0.0 B)
       TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Naturalmente, puede hacer ping al servidor desde el cliente (o viceversa):

Por ejemplo, desde el cliente:

```
# ping -c 3 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_req=1 ttl=64 time=2888 ms
64 bytes from 10.8.0.1: icmp_req=2 ttl=64 time=1997 ms
64 bytes from 10.8.0.1: icmp_req=3 ttl=64 time=1324 ms

--- 10.8.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 1324.475/2070.293/2888.429/640.527 ms, pipe 3
```

10. Almacenamiento de la contraseña PEM en un archivo seguro e inicio automático del servicio tras el arranque

Para iniciar el servicio OpenVPN en el arranque, se necesita una entrada en /etc/rc.d/rc.local, pero tendría que introducir la contraseña del servidor PEM cada vez. Esto podría ser indeseable si el servidor no está localizable. Si este es el caso, cree un archivo que contenga su contraseña PEM en una ubicación segura; por ejemplo, /root/password.ovpn que contenga sólo esta contraseña. Entonces restrinja su permiso:

```
# chmod 600 /root/password.ovpn
```

En el servidor, edite /etc/openvpn/server.conf con las siguientes líneas:

```
askpass /root/password.ovpn
auth-nocache
```

También, descomente la opción 'daemon'.

Esto puede repetirse también en el cliente, simplemente edite /etc/openvpn/client.conf en lugar de /etc/openvpn/server.conf.

Para iniciar el servicio OpenVPN automáticamente al arrancar desde el servidor, incluya estas líneas en /etc/rc.d/rc.local

```
# Start the OpenVPN Service
if [ -x /etc/rc.d/rc.openvpn ]; then
  /etc/rc.d/rc.openvpn start
fi
```

Un método alternativo (aunque menos seguro) es eliminar la frase de contraseña del archivo server1.key por completo. No olvide establecer permisos en la clave para evitar que sea legible por todo el mundo.

```
# cd /etc/openvpn/keys
# openssl rsa -in server1.key -out tmp.key
# mv tmp.key server1.key
# chmod 600 server1.key
```

11. Enrutamiento IP

Hasta ahora hemos creado un dispositivo de túnel tanto en el servidor como en el cliente llamado tun0 que sólo es visible para estos dos equipos. Sin embargo, se necesita más trabajo para enrutar la conexión del cliente a través de tun0 y luego a la WAN que está conectada al servidor.

11.1 Configuración del servidor

Habilite el reenvío IP:

```
# chmod +x /etc/rc.d/rc.ip_forward
# /etc/rc.d/rc.ip_forward start
```

El reenvío de IP está ahora activado y lo estará también después de reiniciar.

Cree un directorio llamado ccd en /etc/openvpn

```
# mkdir /etc/openvpn/ccd/
```

Cree un fichero con el mismo nombre del cliente (en este caso cliente1) e introduzca la siguiente línea en /etc/openvpn/ccd/client1

```
iroute 192.168.1.0 255.255.255.0
```

Sustituya 192.168.1.0 255.255.255.255.0 por la ruta de red de su cliente.

Del mismo modo, edite /etc/openvpn/server.conf con las siguientes líneas:

```
push "route 192.168.200.0 255.255.255.0"

client-config-dir /etc/openvpn/ccd
route 192.168.1.0 255.255.255.0

push "redirect-gateway def1 bypass-dhcp"

push "dhcp-option DNS 208.67.222.222"
push "dhcp-option DNS 208.67.220.220"
```

Naturalmente, reemplace 192.168.200.0 255.255.255.255.0 con la ruta de red del servidor, y 192.168.1.0 255.255.255.0 con la ruta de red del cliente. 208.67.222.222 y 208.67.220.220.220 son las direcciones IP de OpenDNS.

Hasta ahora la configuración de DNS push no ha tenido éxito.

Puede utilizar los servidores DNS originales del cliente o puede reescribir /etc/resolv.conf manualmente:

```
# OpenDNS Servers
nameserver 208.67.222.222
nameserver 208.67.220.220
```

De acuerdo con su tabla de enrutamiento, sin embargo, vale la pena intentar usar los servidores DNS listados por el cliente, encuentro que generalmente todavía están disponibles, por lo que no necesitaría hacer nada. Sin embargo, tenga en cuenta las posibles fugas de DNS si está preocupado por su privacidad.

Algunos usuarios han comunicado que el administrador de red de su cliente (o cualquier otra aplicación similar) reescribió el archivo /etc/resolv.conf original después de su edición manual. Esto no ha podido ser reproducido por el autor de este artículo (todavía), pero puede considerar instalar y configurar openresolv(5) si esto le ocurre realmente. Un SlackBuild para openresolv se puede encontrar en http://slackbuilds.org. Openresolv se encuentra actualmente fuera del alcance de este artículo.

A continuación tendrá que configurar algunos reenvíos NAT de iptables en el servidor (solamente). Puede hacer esto primero limpiando las iptables:

```
# iptables -F
```

Y entonces:

```
# iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE
```

En Slackware, tal línea puede estar incluida en /etc/rc.d/rc.firewall y /etc/rc.d/rc.inet2 la ejecutará cada vez que reinicie el servidor si el primero tiene permisos ejecutables. No tiene que incluir nada en /etc/rc.d/rc.local.

Las líneas exactas que necesita incluir dependen de si ya ha introducido sus propias cadenas y reglas de filtros iptables, pero asumo que este no es el caso.

Como ya se ha explicado, como mínimo sólo necesita introducir las siguientes líneas en /etc/rc.d/rc.firewall

```
#!/bin/sh
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE
```

Si por otro lado desea un firewall mejor y tiene al menos una confianza moderada en iptables, le propongo que incluya el siguiente script en su /etc/rc.d/rc.firewall. Los comentarios en el script le

ayudarán a entender el impacto que tendrán en el servidor.

```
#!/bin/bash
# Start/stop/restart/status the firewall
IPT=/usr/sbin/iptables # Esto proporcionará cierta portabilidad
firewall_start() {
  # limpie las iptables
  echo -e "Starting the firewall ....\c"
  $IPT -F
  # directrices
  $IPT -P OUTPUT DROP
  $IPT -P INPUT DROP
  $IPT -P FORWARD DROP

  $IPT -N SERVICES # services es una cadena personalizada.

  # salida permitida
  $IPT -A OUTPUT -o lo -j ACCEPT
  $IPT -A OUTPUT -o eth0 -j ACCEPT
  $IPT -A OUTPUT -o tun0 -j ACCEPT

  # entradas permitidas
  #$IPT -A INPUT -i lo -j ACCEPT # descomentar si el host es un escritorio
  $IPT -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT #
permitir respuestas
  $IPT -A INPUT -j SERVICES # añadir la cadena de servicios a la entrada

  # reenvío permitido para OpenVPN
  $IPT -A FORWARD -i eth0 -o tun0 -m state --state ESTABLISHED,RELATED -j
ACCEPT
  $IPT -A FORWARD -s 10.8.0.0/24 -o eth0 -j ACCEPT

  # enmascarar la red OpenVPN
  $IPT -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE

  # permitir sshd en el puerto tcp por defecto 22
  #$IPT -A SERVICES -p tcp --dport 22 -j ACCEPT # Descomentar para permitir
sshd

  # permitir OpenVPN para el puerto udp 1194 por defecto
  $IPT -A SERVICES -p udp --dport 1194 -j ACCEPT
  echo "done."
}

firewall_stop() {
  echo -e "Stopping the firewall ....\c"

  # políticas (permisivas)
  $IPT -P OUTPUT ACCEPT
  $IPT -P INPUT ACCEPT
  $IPT -P FORWARD ACCEPT
```

```
  # limpie las iptables
  $IPT -F

  # eliminar la cadena personalizada de servicios
  $IPT -X SERVICES
  echo "done."
}

firewall_status() {
  $IPT -vL
}

case "$1" in
'start')
  firewall_start
  ;;
'stop')
  firewall_stop
  ;;
'restart')
  firewall_stop
  firewall_start
  ;;
'status')
  firewall_status
  ;;
*)
  echo "Usage $0 start|stop|restart|status"
esac
```

Dele al script rc del cortafuegos permiso ejecutable:

```
# chmod +x /etc/rc.d/rc.firewall
```

e inícielo:

```
# /etc/rc.d/rc.firewall start
```

Reinicie el servicio OpenVPN en el servidor:

```
# /etc/rc.d/rc.openvpn restart
```

y vuelva a conectarse con el cliente:

```
# /usr/sbin/openvpn /etc/openvpn/client.conf
```

12. Cortafuegos

En el capítulo anterior nos referimos a un firewall que puede incluir para proteger su servidor

OpenVPN. Sin embargo, este capítulo se refiere a los cortafuegos de la LAN del cliente que pueden bloquear la conexión VPN bloqueando el tráfico en el puerto UDP 1194.

Para penetrar a través del cortafuegos del cliente, puede intentar cambiar el puerto a 443 - normalmente reservado para https. El uso de TCP en lugar de UDP también ayudará. Para hacer estos cambios necesitará modificar /etc/openvpn/server.conf del servidor, desde

```
port 1194
proto udp
```

a:

```
port 443
proto tcp
```

y /etc/openvpn/client.conf del cliente, desde

```
proto udp

remote servervpn.no-ip.org 1194
```

a:

```
proto tcp

remote servervpn.no-ip.org 443
```

El script del firewall del servidor también necesitaría ser modificado. Cambie estas líneas:

```
# permite vpn en el puerto udp predeterminado 1194
$IPT -A SERVICES -p udp --dport 1194 -j ACCEPT
```

a:

```
# permite vpn en el puerto udp modificado 443
$IPT -A SERVICES -p tcp --dport 443 -j ACCEPT
```

También tiene que modificar el redireccionamiento del puerto de su enroutador al puerto TCP 443.

13. Fuentes

(1) http://en.wikipedia.org/wiki/OpenVPN

(2) https://wiki.archlinux.org/index.php/OpenVPN

(3) http://slackwiki.com/OpenVPN_smcr_2012

(4) http://www.no-ip.com

(5) http://roy.marples.name/projects/openresolv/index

- Escrito para Slackware 14.2 en Abril del 2018
- Escrito originalmente por Chris Abela
- Traducido por — *Pedro Herrero García 2019/02/17 11:11 (UTC)*

howtos, network, openvpn