

INSTALACIÓN Y CONFIGURACIÓN DE OPENVPN

Para poder realizar la instalación y configuración de la aplicación OpenVPN, antes de nada habrá que descargar los paquetes necesarios a tal efecto. En este caso, si ambas máquinas funciona bajo GNU/LINUX habrá que instalar openVPN en ambas y configurar una como servidor y otra como cliente. En el caso de que el servidor funcione bajo Linux pero el cliente sea una máquina Windows, habrá que instalarle el cliente adecuado para poder establecer la red virtual.

Comencemos con la instalación de los paquetes. Si bien con openvpn sería suficiente, en el caso de que se quiera utilizar con certificados, habrá que instalar openssh (si no está ya instalado).

```
aptitude install openvpn openssh
```

SEGURIDAD EN LA CONEXIÓN VPN

GENERACIÓN DE LA CLAVE SECRETA

Mediante la siguiente directiva se genera la clave privada. Esa clave tienen que poseerla tanto el servidor como el cliente, con lo cual, tras generarla en el servidor, hay que colocarla en el directorio correcto del mismo y a su vez copiarla al cliente mediante un medio que no comprometa la seguridad de la clave. En este caso se ha optado por copiarlo al cliente mediante ssh.

```
openvpn --genkey --secret giltza.key  
cp giltza.key /etc/openvpn  
scp giltza.key ip_del_servidor:/etc/openvpn/
```

GENERACIÓN DE CERTIFICADOS (MODO GENERAL)

En el caso de que se quiera aumentar la seguridad en la VPN, podría optarse por protegerlo mediante clave secreta y certificados. Las claves privadas tendrán que mantenerse siempre en privado, los ficheros con los certificados en cambio pueden publicarse o compartirse con total libertad.

Se seleccionará en este ejemplo el servidor como máquina generadora y gestora de las claves y los certificados.

En primer lugar hay que edita el fichero openssl.cnf (/usr/share/ssl/openssl.cnf), con las siguientes modificaciones:

- hacer que la opción dir apunte al directorio que se haya creado (habrá que crearlo) como espacio de trabajo para las claves.
- Incrementar default_days (si es que interesa) para que la VPN tenga el tiempo de validez que se estime oportuno (por defecto es de un año).

- Establecer `certificate` y `private_key` para que apunten al certificado y a la clave maestra que se generará a continuación.

Se pueden hacer más modificaciones para adaptar `openssl.cnf` a nuestras necesidades, pero no es necesario hacer más modificaciones para poder crear los certificados.

Lo primero, se generará la Autoridad de Certificación (CA), un par de certificado/clave privada para los próximos 10 años (o los que se estimen oportunos):

```
openssl req -nodes -new -x509 -keyout my-ca.key -out my-ca.crt -days 3650
```

Ahora llega el turno de crear un par certificado/clave tanto para el servidor como para el cliente.

```
openssl req -nodes new -keyout servidor.key -out servidor.csr
openssl ca -out servidor.crt -in servidor.csr
openssl req -nodes new -keyout cliente.key -out cliente.csr
openssl ca -out cliente.crt -in cliente.csr
```

Una vez generados los pares de claves tanto para el servidor como para el cliente, hay que copiar de forma segura `cliente.crt`, `cliente.key` y `my-ca.crt` al cliente.

Finalmente basta con establecer los parámetro Diffie-Hellman en el servidor con el siguiente comando:

```
openssl dhparam -out dh1024.pem 1024
```

Existen más métodos para generar claves y certificados, incluso pueden comprarse a una entidad certificadora, pero se ha utilizado esta forma a modo de explicación, con gasto cero.

GENERAR CERTIFICADOS MEDIANTE UTILIDAD EASY-RSA OPENVPN

Otro modo de generar las claves y los certificados generados en el apartado anterior es valerse de los scripts que la propia aplicación de OpenVPN incluye en su directorio `/usr/share/doc/openvpn/easy-rsa`, los cuales facilitan bastante la tarea.

Lo primero será copiar el directorio `easy-rsa` a `/etc/openvpn` y situarse en dicho directorio.

```
cp -a /usr/share/doc/openvpn/easy-rsa /etc/openvpn
cd /etc/openvpn/easy-rsa
```

Para generar la entidad certificadora:

```
./clean-all
./build-ca
```

Al ejecutar build-ca se solicitan los datos necesarios a fin de generar la entidad certificadora:

En éste último paso se nos pedirá una serie de información sobre nuestra red/empresa que debemos llenar lo más fielmente posible:

```
Generating a 1024 bit RSA private key
.....
.....++++++.....++++++
writing new private key to 'ca.key'
-----
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.
-----
Country Name (2 letter code) [KG]:ES
State or Province Name (full name) [NA]:Euskadi
Locality Name (eg, city) [BISHKEK]:Bilbo
Organization Name (eg, company) [OpenVPN-TEST]:pfc-server
Organizational Unit Name (eg, section) []:test
Common Name (eg, your name or your server's hostname) []:nagore
Email Address [me@myhost.mydomain]:nagore@abartiateam.com
```

Si bien el resto de los datos puede rellenarse como mejor convenga, la variable Common Name es muy importante, ya que cuando una máquina trate de comprobar la validez del certificado lo hará contra el FQDN de la máquina certificadora, con lo cual el Common Name tiene que coincidir con ella.

GENERAR EL CERTIFICADO Y LA CLAVE DE ENCRIPCIÓN PARA EL SERVIDOR

Al generar el certificado del servidor volverán a solicitarse los mismos datos que para generar la entidad certificadora. Nuevamente pueden generarse como mejor convenga, pero teniendo cuidado de no escoger el FQDN utilizado para la CA. En este ejemplo se ha tomado server, como Common Name.

```
./build-key-server server
```

Este paso genera dos archivos en el directorio /etc/openvpn/easy-rsa/keys, que habrá que copiar al directorio /etc/openvpn: server.crt y server.key

El siguiente paso es generar los certificados y las claves de seguridad de cada uno de los clientes. Los archivos obtenidos con el siguiente paso tendrán que almacenarse en la carpeta /etc/openvpn de cada cliente.

Para generar el certificado y las claves privadas se ejecuta en el servidor, dentro del directorio /etc/openvpn/easy-rsa/ :

```
./build-key client1
```

En este último caso también habrá que responder a las preguntas anteriores del modo conveniente. El parámetro `cliente1` pasado como parámetro distingue el certificado del certificado de otro cliente. Podrán crearse tantos clientes como sean necesarios cambiando el nombre que se le pasa como parámetro.

GENERAR PARÁMETROS DIFFIE-HELLMAN

```
./build-dh
```

Una vez generado tanto la CA, como los certificados del servidor y los de los clientes, habrá que asegurarse de que tanto el cliente como el servidor tienen los siguientes archivos:

Archivos en el Servidor: *ca.crt, ca.key, server.key server.crt dh1024.pem*
Archivos en el Cliente: *ca.crt, client1.crt, client1.key*

CONFIGURACIÓN DE CONEXIÓN HOST TO HOST

En esta configuración “host to host” se logra que el intercambio de paquetes entre dos máquinas se realice de forma encriptada. Para ello se crea una interfaz virtual con una IP privada a cada extremo. Se tomará como ejemplo la 192.168.2.1 como la IP del servidor y la 192.168.2.2 como la IP del cliente. Cualquier paquete que viaje entre ambas direcciones lo hará encriptado.

CONFIGURACIÓN DEL SERVIDOR

```
#/etc/openvpn/server.conf  
  
# dispositivo de tunel  
dev tun  
  
# ifconfig ip_del_servidor ip_del_cliente  
ifconfig 192.168.2.1 192.168.2.2  
  
# Clave del servidor  
secret /etc/openvpn/giltza.key  
  
#puerto (por defecto el puerto de VPN es el 1194)  
port 1194  
  
#usuario bajo el cual se ejecutará la VPN  
;user nobody  
;group nobody  
  
# opciones, comprimir con lzo, ping cada 15 segs, verbose 1 (bajo)  
comp-lzo  
ping 15  
verb 1
```

CONFIGURACIÓN DEL CLIENTE

```
#/etc/openvpn/cliente1.conf  
  
# IP publica del servidor. En caso de no tener una IP pública fija, optar por opciones #como dyndns.  
remote pfc-server.dyndns.org  
  
# puerto  
port 1194  
  
# dispositivo tunel  
dev tun  
  
# ifconfig ip_del_cliente ip_del_servidor  
tun-mtu 1500  
ifconfig 192.168.2.2 192.168.2.1  
  
# clave privada, giltza.key (en este caso)  
# Una posible ruta para la key en windows está comentada.  
#secret "c:\program files\company branded vpn\config\key.txt"  
secret /etc/openvpn/giltza.key  
  
# enable LZO compression  
comp-lzo  
  
# ping cada 10 segs  
ping 10  
  
# compresión lzo  
comp-lzo  
  
# verbose moderado, callar mas de 10 mensajes iguales  
verb 4  
mute 10
```

CONFIGURACIÓN DE CONEXIÓN ROAD WARRIOR

CONFIGURACIÓN DEL SERVIDOR

```
#/etc/openvpn/server.conf  
# dispositivo tunel  
dev tun  
  
# Claves y certificados  
ca ca.cert  
cert server.crt  
key server.key  
dh dh1024.pem  
  
# Direcciones que se asignarán a los clientes. Diferentes de las de la subred del servidor. # La IP del  
servidor en la vpn será 192.168.2.1  
server 192.168.2.0 255.255.255.0  
ifconfig-pool-persist ipp.txt  
  
# Ruta para que los clientes alcancen la red local del servidor.
```

```
# Hace que un road warrior pueda "ver" la red interna del servidor  
push "route 192.168.1.0 255.255.255.0"
```

```
keepalive 10 120  
comp-lzo  
user nobody  
group nobody  
persist-key  
persist-tun  
status openvpn-status.log  
verb 4
```

CONFIGURACIÓN DEL CLIENTE

```
#/etc/openvpn/client.conf  
# Indicamos que algunas configuraciones las tomará del servidor  
client  
# Dispositivo tunel  
dev tun  
proto udp  
# Dirección real del servidor  
remote pfc-server.dyndns.org  
port 1194  
resolv-retry infinite  
# nobind --> para asegurarse de que solo actue como cliente y nunca como servidor  
nobind  
  
persist-key  
persist-tun  
  
# Claves y certificados  
ca ca.crt  
cert client1.crt  
key client1.key  
  
comp-lzo  
verb 4
```