OpenVPN

Autores: Emiliano López (emiliano.lopez@gmail.com)

Maximiliano Boscovich (maximiliano@boscovich.com.ar)

Fecha: 24/05/2018 12:16

Existen tres grandes familias de implementaciones de VPN de amplio uso: SSL, IPSec y PPTP. OpenVPN es una VPN SSL y como tal no es compatible con IPSec, L2TP o PPTP.

El protocolo IPSec está diseñado para ser implementado como una modificación del stack IP en el espacio del kernel y por esta razón, cada sistema operativo requiere su propia implementación de IPSec.

Por contraste, la implementación del OpenVPN en el espacio de usuario permite portabilidad a través de los sistemas operativos, arquitecturas de procesadores, amigable para operaciones de firewalls, soporta direcciones dinámicas y múltiples protocolos.

Existen ventajas y desventajas en ambos enfoques. La principal ventaja de OpenVPN es la portabilidad, facilidad de configuración y compatibilidad con NAT y direcciones dinámicas. La curva de aprendizaje para instalarlo y usarlo está a la par de otro cualquier otro demonio relacionado a la seguridad como por ejemplo ssh.

Historicamente, un de las ventajas de IPSec es que ha contado con soporte para diferentes equipos de hardware, pero esto está comenzando a cambiar ya que OpenVPN también viene con soporte para dispositivos de hardware dedicados.

Mientras que el protocolo PPTP tiene la ventaja de contar con el cliente pre-instalado en las plataformas Windows, análisis de expertos en criptografía han revelado vulnerabilidades de seguridad.

A TUN device can be used like a virtual point-to-point interface, like a modem or DSL link. This is called routed mode, because routes are set up to the VPN partner.

A TAP device, however, can be used like a virtual Ethernet adapter. This enables the daemon listening on the interface to capture Ethernet frames, which is not possible with TUN devices. This mode is called bridging mode because the networks are connected as if over a hardware bridge

Instalación

Instalamos el repositorio EPEL (Extra Packages for Enterprise Linux). Esto es porque OpenVPN no esta disponible en los repositorios por defecto de CentOS. EPEL es un repositorio adicional gestionado por el proyecto Fedora.

Luego instalamos el cliente-servidor `openvpn` y, `easy-rsa` un conjunto de scripts para administrar claves y certificados.

```
yum install epel-release
yum install openvpn easy-rsa
```

Vemos que la versión instalada es la 2.4.4.

Generación de certificados y claves

Copiamos los scripts de easy-rsa al directorio de openvpn:

```
cp -rf /usr/share/easy-rsa/VERSION/* /etc/openvpn/easy-rsa
```

Descargamos el archivo `vars.example` de https://github.com/OpenVPN/easy-rsa/blob/v3.0.5/easyrsa3/vars.example realizamos una copia denominada `vars` que vamos a modificar.

Una infraestructura de clave pública (PKI) se basa en la noción de confiar en una autoridad particular en autenticar un par remoto.

Para crear e inicializar una nueva PKI, **debemos** estar parados en el directorio `/etc/openvpn/` y usamos el comando:

```
./easy-rsa/easyrsa init-pki
```

Se creará una nueva estructura PKI en blanco lista para ser usada para crear una nueva CA (Autoridad de Certificación) y generar claves.

Estructura de directorios del PKI

- private/ - claves privadas generadas para el host
- reqs/ - dir with locally generated certificate requests (for a CA imported requests are stored here)

En una PKI limpia no existirá ningún archivo, solamente la estructura de directorios. Luego de llamar a los comandos se irán creando los archivos necesarios, dependiendo de la operación.

Cuando se cree una CA, una serie de archivos nuevos serán creados por una combinación de Easy-RSA e indirectamente openssl. Los **archivos importantes** del CA son:

- ca.crt - This is the CA (Certificate Authority) certificate
- index.txt - This is the "master database" of all issued certs
- serial - Stores the next serial number (serial numbers increment)
- private/ca.key - This is the CA private key (security-critical)
- certs_by_serial/ - dir with all CA-signed certs by serial number
- issued/ - dir with issued certs by commonName

Una vez creado el PKI, el próximo paso será crear una CA.

Creando la CA

Para firmar solicitudes y producir los certificados, se necesita una CA. Para crearla en el PKI que se ha creado anteriormente, ejecutar:

```
./easy-rsa/easyrsa build-ca
```

Asegúrese de usar una passphrase segura para proteger la clave privada del CA. Note que debe suministrar esta passphrase en el futuro cuando proceda a firmar certificados con su CA.

Durante el proceso de creación, además deberá seleccionar el nombre del CA, denominado Common Name (CN). Este nombre es puramente para visualización.

Una vez creada la CA debemos generar el certificado del servidor y de los clientes para ser firmados con la CA.

Certificado del servidor

Creamos el certificado:

```
./easy-rsa/easyrsa gen-req servidor-epe nopass
```

Una vez generado debemos firmarlo:

```
./easy-rsa/easyrsa sign-req server servidor-epe
```

Nos solicitará la passphrase para continuar con la firma y una serie de confirmaciones y ya hemos creado el .crt que utilizaremos posteriormente en la configuración de OpenVPN.

Parámetros Diffie-Hellmann y la clave tls-auth

Estos parámetros son utilizados para el intercambio de claves.

```
./easy-rsa/easyrsa gen-dh
openvpn --genkey --secret ta.key
```

Certificados para los clientes

Generamos los certificados y luego los firmamos:

```
./easy-rsa/easyrsa gen-req cliente1-epe nopass
./easy-rsa/easyrsa sign-req client cliente1-epe
```

Esto nos almacenará los archivos en las siguientes rutas:

```
/etc/openvpn/pki/issued/cliente1-epe.crt
/etc/openvpn/pki/private/cliente1-epe.key
```

Organizar los .crt y .key del servidor y clientes

Crear un directorio para los archivos del servidor y otro por cada cliente. Es **MUY** importante que tengan los siguientes archivos:

Para el servidor:

- ca.crt
- dh.pem
- servidor-epe.crt
- servidor-epe.key
- ta.key

Para el cliente:

- ca.crt
- cliente1-epe.crt
- cliente1-epe.key
- ta.key

Restará crear los archivos de configuración del servidor y del cliente (`servidor.conf` y `cliente1-epe.conf`)

Archivo de configuración del servidor

Copiamos el archivo de configuración de ejemplo:

```
cp /usr/share/doc/openvpn-VERSION/sample/sample-config-files/server.conf
/etc/openvpn
```

Para ver los protocolos de cifrado soportados podemos ejecutar `openvpn --show-ciphers`.

Parámetros

Ver la explicación del archivo de configuración de ejemplo. A continuación otro ejemplo:

```
# puerto, protocolo y tipo de interfaz
port 1194
proto udp
dev tun
# certificado del CA, del server y su clave privada
ca server/ca.crt
cert server/server-epe.crt
key server/server-epe.key
# para intercambio de claves
dh server/dh.pem
# red de los clientes
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
# rutas enviadas a clientes
#;push "route 192.168.10.0 255.255.255.0"
#;push "dhcp-option DNS 192.168.10.2"
#;push "dhcp-option DNS 192.168.10.3"
keepalive 10 120
comp-lzo
persist-key
persist-tun
status openvpn-status.log
verb 3
```

La opción server especifica la red a la que pertenecerán los clientes de VPN, es decir, a cada uno de los clientes que se conecten, se les dará una IP fija en esta subred (10.8.0.0/24). La información respecto de que IP fue asignada a que cliente vpn, es logueada en en el archivo ipp.txt definido en la opción ifconfig-pool-persist.

Los siguientes 3 parámetros son información que se envía a los clientes luego de establecer la conexión. En este caso se envía una ruta, para que los mismos puedan llegar a la subred interna (192.168.10.0/24 en este caso) utilizando como gateway al servidor de VPN (el que tendrá la ip 10.8.0.1). Además se envía información respecto de los servidores de DNS internos, para que estos puedan resolver los nombres tal y como si estuvieran dentro de la propia red interna.

Los restantes parámetros no son tan relevantes, simplemente diremos que definen el tiempo para determinar si un cliente perdió la conexión, definen que los paquetes irán comprimidos con el algoritmo lza y algunas opciones de log.

Iniciar el servidor

Deshabilitar firewalld y SELinux:

```
systemctl stop firewalld
systemctl disable firewalld
```

Editar `/etc/sysconfig/selinux` y cambiar SELINUX a `SELINUX=disabled` y reinciar el **servidor**.

Luego, `systemctl start openvpn@server.service`, para hacer el servicio permanente después del booteo use `enable` en lugar de start.

Si todo fue correctamente debería ver una nueva interfaz `tun` con la siguiente información:

```
# ip a
6: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
 state UNKNOWN group default qlen 100
    link/none
    inet 10.8.0.1/24 brd 10.8.0.255 scope global tun0
    valid_lft forever preferred_lft forever
    inet6 fe80::f19b:2f6c:3f33:c921/64 scope link flags 800
    valid_lft forever preferred_lft forever
```

Archivo de configuración del cliente

Debemos tener instalado el paquete openvpn y para su configuración nos basamos en el archivo de configuración de ejemplo para clientes:

```
cp /usr/share/doc/openvpn-VERSION/sample/sample-config-files/client.conf
/etc/openvpn/cliente1-epe.conf
```

Ahí configuramos la IP o nombre del servidor, los certificados, claves, etc.

Debemos transferir desde el servidor los 4 archivos necesarios: `ca.crt` y `ta.key` son los mismos del servidor, mientras que `cliente1-epe.crt`, `cliente1-epe.key` y `cliente1-epe.conf` son exclusivos del cliente.

Ahora, es necesario arrancar (y habilitar) OpenVPN en el inicio.

```
systemctl start openvpn-client@cliente1-epe
systemctl -f enable openvpn@server.service
```

Se debe tener en cuenta en el primer comando que luego de la `@`, es decir, `cliente1-epe`, se corresponde con el nombre del archivo de configuración, es decir `cliente1-epe.conf` por lo que **deben coincidir**.

Una vez levantado el servicio entonces deberia ver la interfaz tun creada con la ip correspondiente:

```
5: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UNKNOWN group default qlen 100 link/none
    inet 10.8.0.2/24 brd 10.8.0.255 scope global tun0
    valid_lft forever preferred_lft forever
    inet6 fe80::f154:fca4:b33d:e8dc/64 scope link stable-privacy
    valid_lft forever preferred_lft forever
```

Si sale el error debido a la imposibilidad de escribir en el openvpn-status.log se debe ejecutar:

```
ausearch -c 'openvpn' --raw | audit2allow -M my-openvpn
semodule -i my-openvpn.pp
```

Revocar certificados

ACTIVIDAD para aprobar: investigar cómo se debe revocar el certificado de un determinado cliente

Referencias

- https://github.com/OpenVPN/easy-rsa
- https://community.openvpn.net/openvpn/wiki/FAQ
- https://www.redeszone.net/redes/openvpn/
- Feilner, M. (2006). OpenVPN Building and Integrating Virtual Private Networks Learn.
- Keijser, J. J. (2011). OpenVPN 2 Cookbook. Import-01.