Montar un servidor casero con Raspberry Pi (Parte 7): Instalación y configuración de OpenVPN
12 junio, 2013 8:57 pm / 17 Comments / Timbleck

En esta séptima parte del tutorial veremos cómo instalar y configurar un servidor OpenVPN en Raspbian (y por consiguiente en Debian Wheezy). Crearemos una autoridad certificadora y generaremos nuestros certificados de servidor y cliente para autenticar y cifrar el contenido de nuestras comunicaciones.
Instalar y configurar servidor OpenVPN
Índice de tutoriales:

Tutoriales relacionados:

- Encendiendo ordenadores automáticamente con Wake-on-LAN y Cron.
- Copia completa de nuestra Raspberry Pi.
- Reducir el tamaño de Raspbian eliminando paquetes no usados.

¿Por qué queremos instalar un servidor OpenVPN en Raspbian?

Tener un servidor OpenVPN doméstico nos ofrecerá dos funcionalidades la mar de interesantes:

- Anonimizar conexiones.
- Acceder a recursos internos de nuestra red local desde el exterior.

El primer punto es algo básico si decidimos conectar nuestros dispositivos a una red pública. Desde que capturar tráfico en una red es algo tan trivial como tener la aplicación indicada en nuestro teléfono móvil el navegar desde cualquier red abierta sin ningún tipo de protección por nuestra parte es cuanto menos temerario. El segundo punto es una funcionalidad la mar de interesante si disponemos de algún tipo de almacenamiento compartido en la red desde el que queramos acceder desde cualquier lado.

Una vez visto lo que nos puede aportar este nuevo servicio de nuestra Raspberry Pi vamos a instalarlo y configurarlo.
Crear una autoridad certificadora y generar certificados con easy-RSA

OpenVPN se basa en OpenSSL para implementar la criptografía SSL/TLS. Para ello tenemos dos opciones:

- Configurar una clave privada compartida.
- Configurar un certificado con el estándar X.509 basado en infraestructura de llave pública.

Para este tutorial utilizaremos la segunda opción, aprovechando que OpenVPN incorpora todo lo necesario para hacerlo.

Antes que nada deberemos proceder a instalar OpenVPN en Raspbian:

$ sudo apt-get install openvpn

Después de la instalación, si nos dirigimos a la carpeta /usr/share/doc/openvpn/examples encontraremos distintas carpetas con ejemplos de configuración, claves, scripts y lo que ahora nos ocupa, con la carpeta easy-rsa, que nos ayudará a crear nuestra propia autoridad certificadora con RSA.

Así que procederemos a copiar esta carpeta de ejemplo a la ubicación que queramos para trabajar con ella:

$ sudo cp -r /usr/share/doc/openvpn/examples/easy-rsa/2.0 /etc/openvpn/

Y nos dirigimos a su nueva ubicación:

$ cd /etc/openvpn/2.0

El siguiente paso será editar el archivo vars llenándolo con nuestra información. Así que lo abrimos:

$ sudo nano /etc/openvpn/2.0/vars

Y rellenamos con nuestros datos. Aquí va un ejemplo:

```
1
2
3
4
5
6
7
8

export KEY_COUNTRY="ES"
export KEY_PROVINCE="BARCELONA"
export KEY_CITY="Barcelona"
export KEY_ORG="Sobrebits"
export KEY_EMAIL="micorreo@micorreo.com"
export KEY_CN=micommonname
export KEY_NAME=minombredeclave
export KEY_OU=IT
```

Después de esto ya estamos listos para exportar los datos del archivo vars.

```
$ sudo su
# source vars
```

Podemos comprobar que las variables se han exportado correctamente por ejemplo intentando mostrar

el valor de $KEY_CITY por terminal:

echo $KEY_CITY

Que nos debería devolver el nombre que hayamos introducido, en este caso Barcelona. Ahora ejecutaremos el script clean-all que se encargará de eliminar una posible carpeta de claves anteriormente creada.

./clean-all

Hecho esto vamos a crear el par de claves de la propia Autoridad Certificadora (CA):

./build-ca

Que nos preguntará acerca de los datos configurados anteriormente:
1
2
3
4
5
6
7
8

Country Name (2 letter code) [ES]:
State or Province Name (full name) [BARCELONA]:
Locality Name (eg, city) [Barcelona]:
Organization Name (eg, company) [SobreBits]:
Organizational Unit Name (eg, section) [IT]:
Common Name (eg, your name or your server's hostname) [micommonnamel]:
Name [minombredeclave]:
Email Address [micorreo@micorreo.com]:

Llegados aquí ya somos capaces de generar nuestros propios certificados firmados. Con ello crearemos nuestro certificado para el servidor VPN (clave privada) y los parámetros Diffie-Hellman utilizados para establecer la conexión SSL/TLS. Para el certificado del servidor:

./build-key-server raspberry.home.local

Se nos preguntarán los mismos datos que en el momento de la generación de los certificados de la CA, con lo que contestaremos con los datos deseados y seguiremos las instrucciones. Ahora vamos con los parámetros Diffie-Hellman:

./build-dh

Y por último con la clave para cada uno de los usuarios que se conecten vía VPN:

./build-key Nombreusuario

Y seguiremos los pasos rellenando los campos que se nos pregunten. Si ahora nos vamos a la carpeta ./keys (o la que hayamos asignado en el archivo vars) encontraremos un buen puñado de archivos con distintos fines. Vamos ver qué nos interesa de aquí para esta parte del tutorial:

ca.crt: Este es el certificado público de la CA, que tendremos que usar en todos los clientes y en el servidor.
raspberry.home.local.crt y raspberry.home.local.key: Certificado público y privado respectivamente del servidor. Solo los usaremos en el servidor.
dh1024.pem: Los parámetros Diffie-Hellman que se ubicarán sólo en la carpeta de OpenVPN del servidor.
Nombreusuario.crt y Nombreusuario.key: Certificados público y privado de usuario que utilizaremos en los dispositivos de dicho usuario.

Así que vamos a distribuir los distintos archivos en su sitio dentro del servidor:

```
# cp ca.crt /etc/ssl/certs/Sobrebits_CA.crt
# cp raspberry.home.local.crt /etc/ssl/certs/
# cp raspberry.home.local.key /etc/ssl/private/
# cp dh1024.pem /etc/openvpn/
```

NOTA: La ubicación de los certificados se puede configurar a mano en el archivo de configuración correspondiente de OpenVPN, pero los he copiado a los directorios donde se almacenan estos archivos en Raspbian para mantener cierta coherencia en el sistema.
Configuración del servidor OpenVPN en Raspbian

Con el tema de los certificados solucionado ahora toca afrontar la configuración del servidor OpenVPN en si. OpenVPN al arrancar, por defecto, intenta iniciar todas las conexiones VPN configuradas en los archivos *.conf dentro de su directorio /etc/openvpn/. Con su instalación, al igual que trae easy-RSA para no complicarnos mucho la vida con el tema de los certificados, también trae una configuración por defecto sobre la que podremos trabajar. Lo primero que haremos será copiar esta configuración a nuestro directorio de OpenVPN para trabajar sobre el:

```
$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn/
```

Y descomprimimos el contenido del archivo:

```
$ cd /etc/openvpn
$ gunzip server.conf.gz
```

Esto nos dejará ya el archivo server.conf editable:

```
$ sudo nano server.conf
```

El archivo es bastante extenso pero está muy bien documentado, además incorpora una configuración “out of the box” funcional, por lo cual me centraré en las opciones que yo he tocado en un inicio para que todo funcione. No dudéis en indagar un poco más al respecto, yo desde luego lo haré:
1
2
3

4
5
6
7
8
9
10
11

Los certificados anteriormente creados y ubicados.
ca /etc/ssl/certs/Sobrebits_CA.crt
cert /etc/ssl/certs/raspberry.home.local.crt
key /etc/ssl/private/raspberry.home.local.key # This file should be kept secret

Parámetros Diffie-Hellman
dh /etc/openvpn/dh1024.pem

Habilitamos a los clientes conectados para que puedan acceder a la subnet
interna
push "route 192.168.66.0 255.255.255.0"

La parte de configuración propiamente dicha del servidor OpenVPN ya la tenemos completa, ahora solo nos falta reiniciar el servicio OpenVPN, lo que cargará todos los archivos *.conf de la carpeta /etc/openvpn, con lo cual quedará creada nuestra interfaz tun0.

$ sudo /etc/init.d/openvpn restart

Configurar IP forwarding en la Raspberry Pi

En este punto del tutorial ya tenemos la configuración del servidor OpenVPN de nuestra Raspberry Pi lista. Sin embargo, si nos conectamos ahora vía OpenVPN poco o nada podremos hacer ya que cuando nuestro ordenador cliente intente alcanzar cualquier red fuera de la que le hemos configurado (por defecto la 10.8.0.0/24) los paquetes llegarán a la interfaz virtual tun0 de la Raspberry Pi y no saltarán a ninguna otra red porque por defecto ese comportamiento no viene configurado en Raspbian.

Al comportamiento descrito anteriormente se le llama IP forwarding, que es lo que vamos a configurar ahora. Primero habilitamos el forwarding de paquetes en el sistema:

$ sudo su
# echo 1 > /proc/sys/net/ipv4/ip_forward

Esta configuración se perderá con el reinicio del sistema. Si queremos que el cambio sea persistente deberemos editar el archivo /etc/sysctl.conf y descomentar la siguiente línea:
1

net.ipv4.ip_forward = 1

Lo siguiente que debemos hacer es configurar Iptables con las reglas necesarias para que el tráfico sea enrutado correctamente:

```
$ sudo iptables -A FORWARD -i eth0 -o tun0 -m state --state ESTABLISHED,RELATED -j ACCEPT
$ sudo iptables -A FORWARD -s 10.8.0.0/24 -o eth0 -j ACCEPT
$ sudo iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE
```

Al igual que con el IP forwarding estas reglas se perderán en el reinicio del sistema. Para hacer estas reglas persistentes de una forma fácil podemos valernos del paquete iptables-persistent que guardará la configuración actual de Iptables y la volverá a cargar después de cada reinicio. Para ello como siempre vamos a tirar de apt-get:

```
$ sudo apt-get install iptables-persistent
```

En la instalación se nos preguntará si queremos salvar las actuales reglas relacionadas con IPv4 y las relacionadas con IPv6. Nos interesa decirle que sí a las reglas de IPv4.

Iptables-persistent

APUNTE: Para guardar posibles futuros cambios en nuestras iptables manualmente después de la instalación podemos ejecutar:

```
$ sudo /etc/init.d/iptables-persistent save
```

Conclusión

Llegados aquí ya tenemos el servidor OpenVPN en Raspbian completamente funcional con el que poder cifrar nuestras conexiones. Lo siguiente será configurar los clientes con los certificados anteriormente generados. Para GNU/Linux aquí tenéis el tutorial, y aquí el correspondiente para Windows.

Soy consciente de que el capítulo ha quedado un poco espeso, y que sin unos conocimientos mínimos puede ser difícil de seguir. He intentado llenar el artículo de enlaces a explicaciones de los distintos conceptos tocados para facilitar la comprensión de lo que se está haciendo en cada momento.