Módulo 4.3

Redes y Seguridad
Gr. Ing. Informática
J.L. Vázquez Poletti

Objetivo

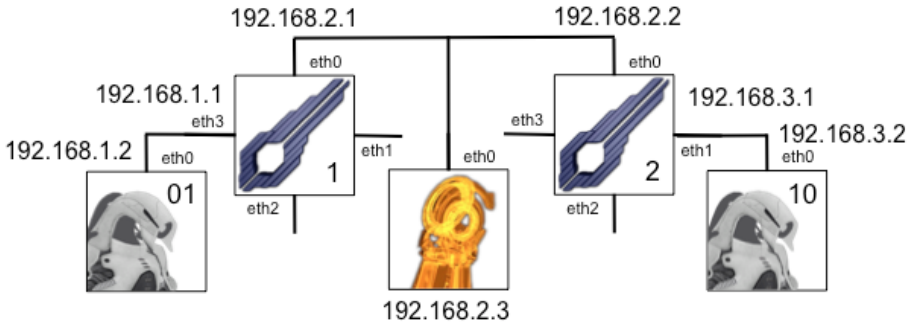
En este módulo aprenderemos a tender túneles SSH y VPNs, de tal manera que haremos más seguras las comunicaciones.

Punto de partida

Al Cuerpo de Ingenieros de la Alianza le interesaría estudiar el funcionamiento de las comunicaciones seguras entre plataformas Geth. Para ello, se ha establecido un punto de escucha en una red de interconexión entre otras dos redes.

Escenario

En esta práctica se cuenta con 2 LANs unidas por una WAN.



- El gateway por defecto de **RyS-Geth01** y **RyS-Geth10** son **RyS-Router1** y **RyS-Router2** respectivamente.
- **RyS-Backtrack** no tiene gateway por defecto.
- Hay que activar el IP forwarding en los routers.
- Antes de arrancar **RyS-Backtrack** hay que activar el **Modo promiscuo "Permitir MVs"** en las opciones avanzadas de la interfaz de la máquina en **VirtualBox**.
- Todas las tareas de configuración en los routers se realizan con el usuario **root**.

Los routers deben configurarse para utilizar la segunda versión del protocolo de intercambio de rutas (RIP) provisto por **quagga**. En cada uno:

1. Editar **/etc/quagga/daemons**. Los demonios **zebra** y **ripd** tienen que estar a **yes**.
2. Hay unas versiones de ejemplo de los ficheros de configuración de estos dos demonios en **/usr/share/doc/quagga/examples/**. Copiarlos a **/etc/quagga** dejándoles la extensión **.conf**.
3. En el fichero de configuración de **zebra**, añadir los interfaces activos usando **interface <*ethN*>**.
4. En el fichero de configuración de **rip**, añadir los interfaces activos usando **network <*ethN*>**.
5. Reiniciar **quagga** con **/etc/init.d/quagga restart**.

Máquinas Virtuales

- RyS-Backtrack (root:toor)
- RyS-Geth01 (ubuntu:reverse)
- RyS-Geth10 (ubuntu:reverse)
- RyS-Router1 (ubntu:reverse)
- RyS-Router2 (ubntu:reverse)

Más información

Backtrack: http://www.backtrack-linux.org/

Quagga: http://www.nongnu.org/quagga/

Wireshark: http://www.wireshark.org/

NCat: http://nmap.org/ncat/

OpenSSH: http://www.openssh.org/

OpenVPN: http://www.openvpn.net/

Para verificar que la configuración ha sido un éxito:

1. Con el comando **route -n** se deben ver todas las redes desde cualquier router.
2. Se debe poder hacer **ping** desde **RyS-Geth01** a **RyS-Geth10** y viceversa.

A partir de aquí, **wireshark** debe arrancarse en **RyS-Backtrack**. El uso de los filtros es importante para no perder detalle del tráfico que se desea visualizar.

Comunicaciones no seguras

A continuación se verá (nuevamente) como el contenido del tráfico sin cifrar puede ser visualizado por cualquier atacante que esté escuchando en el medio, tanto si se trata de una conexión a un servicio estándar como si no.

1. Conectarse de **RyS-Geth01** a **RyS-Geth10** por telnet. Desconectarse tras ejecutar algún comando

¿El tráfico de qué sentido se visualiza?
¿Qué diferencia hay entre esta captura y la realizada en el módulo 4.1 (suplantación del router)? ¿Por qué?

2. Crear un servidor con **ncat** en **RyS-Geth10** que escuche en el puerto **24**.
3. Conectarse usando **ncat** a **RyS-Geth10** desde **RyS-Geth01** e intercambiar algunas frases. Para desconectar, basta con pulsar **CTRL+C**.

¿Qué diferencia hay en el envío de los datos con este método y el anterior?
¿Cuál es más seguro?

Túnel SSH

Una de las formas más sencillas y rápidas de crear un canal cifrado es usar un túnel construido con el protocolo SSH. La idea es crear un túnel en el que uno de los extremos sea la máquina local y el otro, una máquina a la que se pueda acceder por SSH.

El túnel se construye de la siguiente manera:

ssh -N -L *PuertoLocal*:*HostDestino*:*PuertoDestino usuario@HostExtremo*

- La opción **-L** establece el túnel.
- La opción **-N** hace que tras la autenticación no se haga login.
- **PuertoLocal** es el puerto de la máquina local que sirve como extremo del túnel.
- **HostDestino** es la máquina a la que se quiere conectar a través del puente. Hay que indicar el **PuertoDestino** de la misma.

1. Establecer un túnel que vaya desde el puerto **24** de **RyS-Geth01** a **RyS-**

Router2 y que esté dirigido hacia el puerto **23** de **RyS-Geth10**. El proceso se puede pasar a segundo plano.

¿Qué cifrado es el empleado en un sentido y en el otro?

2. Desde **RyS-Geth01** hacer telnet a su propio puerto **24**.
3. Iniciar la sesión, ejecutar algún comando y volver a salir.

¿Cuál es el puerto de salida del tráfico cifrado desde la máquina local
¿Qué ventajas y limitaciones tiene este sistema?

4. Reiniciar el túnel (pasar el existente a primer plano o directamente matar el proceso **ssh**) pero esta vez con la opción de **compresión** activada.

¿Qué tipo de compresión es la utilizada?

5. Reiniciar el túnel pero ahora indicando que se quiere usar el algoritmo de cifrado **blowfish**.

¿Qué mejora tiene **blowfish** frente a **3des**?

VPN

A continuación se establecerá varios tipos de túnel VPN entre los routers. El proceso requiere dar de alta un nuevo interfaz virtual (dentro de la propia infraestructura virtual que ya se está usando) llamado **tun**.

En el túnel, **RyS-Router1** tendrá la dirección **10.4.0.1** y **RyS-Router2**, **10.4.0.2**.

1. Ejecutar **modprobe tun** en ambos routers para permitir esta funcionalidad.
2. En cada router hay que ejecutar lo siguiente:

   openvpn --remote <*IPremota*> --dev tun1 --ifconfig <*IPtunelLocal*> <*IPtunelRemota*> --verb 9

   - **IPremota** es la de la red 192.168.2.0 para el otro router.
   - **IPtunel** es la IP dentro del túnel.
   - **--verb** indica el nivel de detalle.

¿Se pueden ver las direcciones del túnel en el tráfico capturado?
¿Cuál es el protocolo y puerto que abre **OpenVPN**?

3. Mediante la opción "**Expression…**" de **wireshark**, filtrar el tráfico de **OpenVPN** en **RyS-Backtrack**.
4. Pasar el proceso **openvpn** a segundo plano y hacer **ping** de un extremo a otro del túnel (usando las IPs del mismo).
5. Crear un servidor con **nc** (la versión instalada en los routers de **ncat**) en **RyS-Router2** que escuche en cualquier puerto y conectarse a él desde

RyS-Router1 (siempre usando **ncat**). Intercambiar algunas frases.

¿Se puede visualizar el contenido del tráfico generado con **ping** y con **nc**? ¿Por qué?

Ahora es el momento de levantar un túnel cifrado. La primera opción consiste en crear una clave que deberá ser compartida por ambos extremos.

1. Parar el túnel en ambos routers.
2. Crear una clave en **RyS-Router1** ejecutando:

   openvpn --genkey --secret key

   Esto creará un fichero llamado **key** en el directorio en el que se ejecutó el comando.

¿De cuántos bits es la clave?

3. Copiar la clave a **RyS-Router2** usando **scp**.
4. Volver a crear los túneles pero ahora con nivel de detalle **5** y usando el parámetro **--secret key**.

¿Se pueden ver las direcciones del túnel en el tráfico capturado?

5. Volver a generar tráfico con **ping** durante unos instantes.
6. Volver a arrancar un servidor con **nc** e intercambiar algunas frases.

¿Se puede visualizar el contenido del tráfico generado con **ping** y con **nc**? ¿Por qué?
¿Cuál es el punto débil de esta opción de túnel cifrado?

La siguiente opción consiste en cifrar utilizando TLS (Transport Layer Security). En este caso, **RyS-Router1** actuará de servidor y **RyS-Router2**, de cliente.

NOTA: Esta designación cliente/servidor es en el contexto de TLS, no en el del resto de capas.

En este caso se deberá usar un sistema PKI, creando una CA y expidiendo certificados para los integrantes del túnel. Afortunadamente, **OpenVPN** viene con un conjunto de herramientas llamadas **EasyRSA**.

Antes de continuar, se deberá:

1. Bajar el interfaz **tun1** que hay levantado en ambos routers.
2. Parar el demonio **quagga** en ambos routers.

Se comenzará en **RyS-Router1**:

1. Crear el directorio **/etc/openvpn/easy-rsa/**
2. Copiar los contenidos de **/usr/share/doc/openvpn/examples/easy-rsa/2.0/** en el directorio creado anteriormente (usar la opción **-r** para la copia recursiva).

3. Copiar el fichero **openssl-1.0.0.cnf** a **openssl.cnf**.
4. Editar **/etc/openvpn/easy-rsa/vars**:

```
export KEY_COUNTRY="GE"
export KEY_PROVINCE="Tikkun"
export KEY_CITY="Rannoch"
export KEY_ORG="000001"
```

   - Cambiar todos los *changeme* de los *KEY* por "Geth".

Ahora se creará el entorno de la CA:

1. Siempre en el directorio de **easy-rsa**, ejecutar **source vars**. Ignorar el mensaje de aviso.
2. Ejecutar **./clean-all** y después **./build-ca**. Confirmar los datos de la CA.

Después se generará el certificado y la clave del servidor:

1. Ejecutar **./build-key-server server**. Confirmar los datos que ya aparecen. No es necesario introducir la contraseña ni el nombre opcional.
2. Indicar que sí se desea firmar el certificado y confirmar la decisión después (commit).

NOTA: Los certificados y las claves se generan en el directorio **/etc/openvpn/easy-rsa/keys/**.

Ahora es el turno del certificado y clave del cliente:

1. Ejecutar **./build-key client**.
2. Seguir el mismo proceso que con el servidor.

NOTA: Si se expiden más certificados y claves de cliente, hay que cambiar el **Common Name** (CN) para cada uno. En este caso solo se está usando **client**.

También se deberán generar los parámetros Diffie-Hellman usados en la negociación:

1. Ejecutar **./build-dh**.

A este punto, hay que copiar los siguientes archivos en **RyS-Router2** (en su directorio **/etc/openvpn/**). Debido a los permisos, se deberá usar **scp** para llevarlos a un directorio temporal (**/tmp/**) y de ahí, copiarlos localmente.

- **ca.crt**
- **client.crt**
- **client.key**

Es el momento de configurar el servidor en **RyS-Router1**:

1. Copiar el archivo de ejemplo **/usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz** a **/etc/openvpn/**.
2. Descomprimir el archivo.
3. Editar **server.conf**:
   - Verificar que **server** está en 10.8.0.0 255.255.255.0 (esto es la red que

se levantará en el túnel).
    - Las rutas a **ca**, **cert**, **key** y **dh** tienen que ser correctas (relativas desde **/etc/openvpn/**).
    - Agregar la ruta a la red que hay detrás de **RyS-Router2** (**route 192.168.3.0 255.255.255.0**).
    - Indicar a los clientes que se conecten la red que hay detrás de **RyS-Router1** (**push "route 192.168.1.0 255.255.255.0"**).
    - Añadir la línea **client-config-dir ccd**.
4. Crear un directorio llamado **ccd**.
5. Dentro del directorio, crear un fichero llamado **client** con el siguiente contenido:

    ```
    iroute 192.168.3.0 255.255.255.0
    ```

    Esto termina de agregar la red detrás del cliente.

Para configurar el cliente, se deberá cambiar a **RyS-Router2**:

1. Copiar el fichero de configuración de ejemplo de **/usr/share/doc/openvpn/examples/sample-config-files/client.conf** a **/etc/openvpn**.
2. Editar **client.conf**, añadiendo en **remote** la IP de **RyS-Router1**.

Es el momento de probar las configuraciones, para ello:

1. En **RyS-Router1** ejecutar **openvpn server.conf**.
2. En **RyS-Router2** ejecutar **openvpn client.conf**.
3. Verificar tanto con **wireshark** como con los propios mensajes de **openvpn** que se establece correctamente el túnel.

Ahora se consolidará el túnel y se operará normalmente:

1. Cerrar **openvpn** y arrancarlo como demonio con **/etc/init.d/openvpn restart**.
2. Verificar las tablas de rutas de los routers.
3. Hacer **ping**, **telnet** y **ssh** desde ambas LANs.

¿Qué ventajas y desventajas tiene este tipo de túnel?