TP Redes Privadas Virtuales (VPNs)

Fecha de Entrega: 08/12/2020

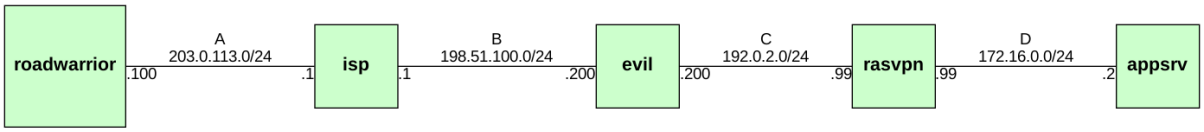
Objetivo: Conocer herramientas de uso extendido para configurar redes privadas virtuales. Reconocer los escenarios de uso habituales de VPN y los servicios de seguridad asociados. Categorías ISO: FCAP**S**.

Bibliografía

- HERTZOG, R.; MAS, R. 2015. Capítulo 10. Sección 2: "Red privada virtual". En *El manual del Administrador de Debian*. Freexian. https://debian-handbook.info/browse/es-ES/stable/sect.virtual-private-network.html
- YONAN J. 2003. *The User-Space VPN and OpenVPN*. http://52.52.21.174/papers/BLUG-talk/index.html

Experiencia de laboratorio

En esta experiencia se utilizará el laboratorio netkit-lab_openvpn.tar.gz, cuya topología se detalla a continuación. Realice los pasos necesarios para descargarlo y validar que se inicia correctamente.



Realizar ping desde el equipo RoadWarrior hasta appserver. ¿Llegan los paquetes?

Túnel punto a punto simple

Como puede ver, la última red es privada y NO se realiza NAT, por lo cual, por defecto, no debería ser posible llegar desde el equipo RoadWarrior hasta el servidor AppServer.

Como primera medida, se configurará un túnel entre el equipo RoadWarrior y el servidor de acceso remoto RASVPN utilizando el software OpenVPN que ya está instalado en ambos equipos. Para ello:

1. **En el servidor RASVPN**: Crear el extremo final del túnel y asignarle la interfaz virtual `tun0` Configurar la ip `10.9.8.1` al extremo local del túnel y `10.9.8.2` al extremo remoto.

   ```
   openvpn --dev tun0 --ifconfig 10.9.8.1 10.9.8.2 &
   ```

2. Determinar cuál es la dirección IP pública del servidor RASVPN y anotarla.

3. **En el cliente RoadWarrior**, crear el otro extremo del túnel y la interfaz virtual mediante:

   ```
   openvpn --dev tun0 --ifconfig 10.9.8.2 10.9.8.1 --remote IP_SERVIDOR_VPN &
   ```

4. Verificar que la interfaz del túnel (`tun0`) está activa mediante el comando `ip link` ¿Cuántas interfaces de red poseen los hosts? ¿Cuáles de ellas son reales/físicas?

5. Comprobar que la configuración haya sido exitosa utilizando el comando `ping` contra el extremo opuesto del túnel.

6. Realizar lo mismo desde **RoadWarrior** contra el servidor **AppSrv**. ¿Se reciben los paquetes de ida y vuelta? ¿qué es lo que sucede?

7. Ahora bien, para resolver este inconveniente vamos a aprovechar lo que ya conocemos de ruteo. ¿Qué nueva ruta hay que adicionar en **RoadWarrior** para que éste envíe los paquetes que van a la red de **AppSrv** *a través del túnel que lo conecta con* **RASVPN**?

8. **En el equipo Evil**, iniciar una captura de tráfico con el comando `sniff` o bien con:

   ```
   ngrep -q -W none -d eth
   ```

9. **En la máquina real**, ejecutar el comando `vdump A > vpn-simple-A.pcap` para iniciar una captura de tráfico sobre la red "A" y dejarla capturando.

10. En la misma máquina real, ejecutar el comando `vdump D > vpn-simple-D.pcap` para iniciar una captura de tráfico sobre la red "D" y dejarla capturando.

11. Agregar la ruta que resuelve el problema planteado en el punto previo:

    ```
    ip route add 172.16.0.0/24 via 10.9.8.1 dev tun0
    ```

    y verificar mediante `ping` que ahora es posible llegar desde **RoadWarrior** hasta **AppSrv**.

12. Ahora sí, **desde RoadWarrior**, acceder mediante `telnet` al servidor **AppSrv**, utilizando el nombre de usuario `titopuente` y la clave `SEGURA-123`

13. Volver al equipo **Evil** y revisar la terminal. ¿Es posible ver el diálogo entre los extremos? ¿Figura la clave allí? ¿cómo es posible, si viajan dentro de un túnel?

14. Detener las capturas iniciadas en los puntos 9 y 10 y guardarlas para un análisis posterior.

Túnel punto a punto con cifrado simétrico

En este escenario se agregan algunos servicios de seguridad adicionales a la configuración OpenVPN.

1. **En RASVPN**, crear una nueva clave con:

   ```
   openvpn --genkey --secret secret.key
   ```

2. Copiar la clave generada en el servidor al equipo **RoadWarrior**. Puede utilizar para ello el directorio compartido `/hostlab`

   ```
   # en RASVPN
   cp secret.key /hostlab/secret.key
   # en RoadWarrior
   mv /hostlab/secret.key /root/secret.key
   ```

3. **En ambos extremos**, ejecutar el comando `pkill openvpn` para detener los procesos que establecen el túnel, y ubicar el archivo de clave `secret.key` en el directorio `/etc/openvpn/` siendo root el dueño del archivo.

4. **En RASVPN** crear el archivo `/etc/openvpn/tun0.conf` para configurar la interfaz y luego incorpore los siguientes parámetros:

   ```
   dev tun0
   ifconfig 10.9.8.1 10.9.8.2
   secret /etc/openvpn/secret.key
   ```

5. **En RoadWarrior**, configurar también una interfaz en el archivo `/etc/openvpn/tun0.conf` con los siguientes parámetros:

```
remote IP-DE-RASVPN
dev tun0
ifconfig 10.9.8.2 10.9.8.1
secret /etc/openvpn/secret.key
```

6. **En RASVPN** levante el túnel de configurado con:

   ```
   openvpn --config /etc/openvpn/tun0.conf &
   ```

7. **En la máquina real**, iniciar una captura de tráfico en la primera red con el comando `vdump A > vpn-simetrico.pcap` y dejarla capturando.

8. **En el equipo Evil**, iniciar una captura de tráfico con el comando `sniff` o bien mediante:

   ```
   ngrep -q -W none -d eth
   ```

9. **En RoadWarrior**, levantar el túnel configurado con el mismo comando utilizado en el punto 6 y acceder mediante telnet al servidor **AppSrv**, utilizando el nombre de usuario `titopuente` y la clave `SEGURA-123` .

10. Volviendo a la terminal del equipo **Evil**, ¿puede nuestro actor malicioso ver el tráfico entre los extremos? ¿Figura la clave allí?

11. Detener la captura iniciada en el punto 7 y guardarla para análisis posterior.

Trabajo práctico

1. Comparar las capturas `vpn-simple-A.pcap` y `vpn-simple-D.pcap` ¿En cuál de ellas se ve el tráfico telnet? ¿En cuál de ellas Wireshark indica correctamente los puertos? ¿Porqué no es posible visualizar las PDU de openVPN en la captura `vpn-simple-D.pcap` ?

2. Identifique en la captura `vpn-simple-A.pcap` , los mensajes correspondientes a PDU de OpenVPN. ¿Para qué se utiliza el campo HMAC en tales mensajes?

3. ¿Qué identifican las direcciones IP luego de la instrucción `ifconfig` ?

4. ¿Qué servicios de seguridad se proveen en la experiencia de laboratorio denominada “Túnel punto a punto simple”?

5. ¿Por qué usted considera que se utiliza en la experiencia de “Tunel punto a punto simétrico” un archivo de clave secreto en lugar de una cadena de texto corta?

6. ¿Qué servicios de seguridad se proveen en la experiencia de laboratorio denominada “Túnel punto a punto simétrico”?

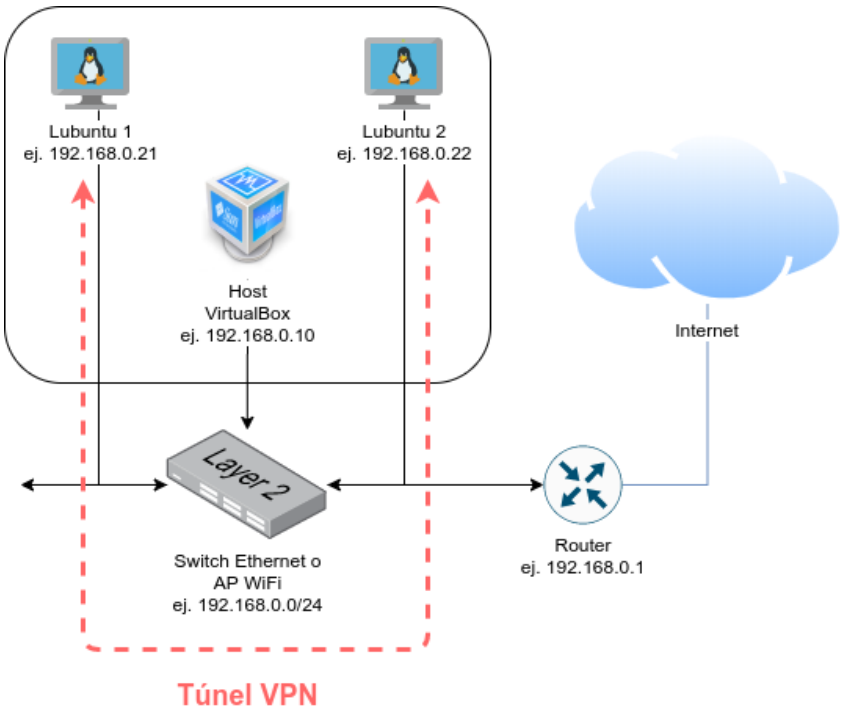
7. En la captura `vpn-simetrico.pcap` ¿Es posible ver los mensajes ICMP? ¿y las ip internas del túnel? ¿Qué datos de usuario puede extraer de la captura?

8. ¿Qué configuración debe aplicarse para convertir el túnel creado en las experiencias en una VPN tipo: a) punto a punto y b) red a red?.

9. Describa las principales diferencias entre OpenVPN y WireGuard.

Desafío: Tunel punto a punto con WireGuard y cifrado asimétrico

Si bien este escenario puede realizarse mediante OpenVPN con certificados X.509 utilizando una Infraestructura de Clave Pública (PKI) propia, en esta oportunidad utilizaremos una tecnología de reciente creación que basa su funcionamiento en cifrado asimétrico utilizando una arquitectura y un protocolo novedoso: WireGuard.



Se utilizarán dos instancias en ejecución de una misma imagen de máquina virtual corriendo sobre VirtualBox para establecer diferentes tipos de VPN entre ellas. El equipo docente proveerá una imagen base de máquina virtual con el Sistema Operativo Lubuntu funcionando, y sólo se requerirá tener instalado el software VirtualBox previamente.

Lubuntu es un “sabor” de Ubuntu que está preparado para correr en entornos con menores requerimientos de hardware. Los requisitos de hardware mínimos están en el orden de una CPU dual core, 2 GB de RAM disponibles como mínimo y 20 GB de disco libres.

Dicha imagen se deberá descargar e importar en VirtualBox en una máquina virtual que podría llamarse “Lubuntu1”, y luego se debe clonar la misma como otra instancia que podría llamarse “Lubuntu2”. Dado que la configuración de la interfaz de red de cada instancia se deberá encontrar establecida en “Modo Bridge”, ambas instancias al ejecutarse accederán a la red de la misma manera que lo hace el host. Se sugiere contar con un servidor DHCP (como por ejemplo el de un router de acceso a Internet) para que asigne una direción en la LAN del host a ambas VMs y entre otras cosas, las VMs accedan puedan acceder a Internet.

Dentro de la imagen, el usuario administrador del equipo es “alumno” y su contraseña es “jpostel”. La distribución del teclado está en español (de todas maneras es posible modificarla).

Para todo el TP, se utilizarán las direcciones IP de la topología ejemplo, aunque puede variar según la configuración de red local de cada alumno. Una vez en ejecución las VMs, deberá comprobarse (y anotarse) la dirección IP de cada una y que se puedan comunicar entre ellas.

Se utilizará “L1” y “L2” como abreviatura de Lubuntu1 (o rol “servidor”) y Lubuntu2 (o “cliente”), respectivamente.

El link para descargar la VM para VirtualBox es el siguiente: https://tinyurl.com/AyGR-VM2020-Q2

Hash SHA256: `f5787f2b2eb77214060c8c2e781aaef1e13395ec939e4d7feacf574468f7662e`

No olvidar comprobar que los hashes coincidan antes de continuar.

1. Instale la herramienta (válido para *buntu 20.04+):

   ```
   sudo apt-get install wireguard
   ```

   Para otros sistemas operativos ver: https://www.wireguard.com/install/

2. **En ambos equipos extremos de la vpn** genere su par de claves públicas y privadas

   ```
   wg genkey > private.key
   cat private.key
   wg pubkey < private.key > public.key
   cat public.key
   ```

3. **En el servidor** crear una nueva interfaz de red llamada wg0 y asignarle una IP

   ```
   sudo ip link add wg0 type wireguard
   sudo ip addr add 10.9.8.1/24 dev wg0
   ```

4. **En el cliente** crear una nueva interfaz de red llamada wg0 y asignarle una IP

   ```
   sudo ip link add wg0 type wireguard
   sudo ip addr add 10.9.8.2/24 dev wg0
   ```

5. **En ambos equipos** asignar la clave privada a la interfaz de red y levantar la interfaz

   ```
   sudo wg set wg0 private-key ./private.key
   ip -4 addr show
   sudo ip link set wg0 up
   ```

6. **En ambos equipos**, indicar la dirección IP del extremo y su respectiva clave pública.

   ```
   sudo wg set wg0 listen-port PUERTO_LOCAL peer CLAVE_PUB \
   allowed-ips IPVIRTUAL_EXTREMO/32 endpoint IPREAL_EXTREMO:PUERTO_EXTREMO
   ```

Las variables son las siguientes:

PUERTO_LOCAL es el puerto en el cual Wireguard va a escuchar en ese host. CLAVE_PUB es la clave pública del otro extremo.

IPVIRTUAL_EXTREMO es la IP ‘virtual’, la que pertenece a la red conformada por la VPN. En este caso se viene utilizando 10.9.8.1 para el server y 10.9.8.2 para el cliente.

IPREAL_EXTREMO es la IP en la LAN del otro extremo.

PUERTO_EXTREMO es el puerto en el cual está escuchando el servicio en el otro extremo (ya funciona conque el ‘cliente’ especifique el ‘PUERTO_LOCAL’ del servidor, pero pueden ser iguales y recíprocos).

7. Pueden comprobar el estado de la VPN wiregard en ambos lados con los comandos:

```
sudo wg show
sudo wg showconf wg0
```

8. Iniciar captura sobre la interfaz REAL del equipo.
9. Visualizar la tabla de rutas y probar el enlace VPN

```
sudo ip route show
ping 10.9.8.X
```

10. Detener la captura y determinar con ella el protocolo de transporte utilizado, los puertos origen y destino, e indicar si Wireshark interpreta apropiadamente las PDUs del protocolo de “aplicación” WireGuard.