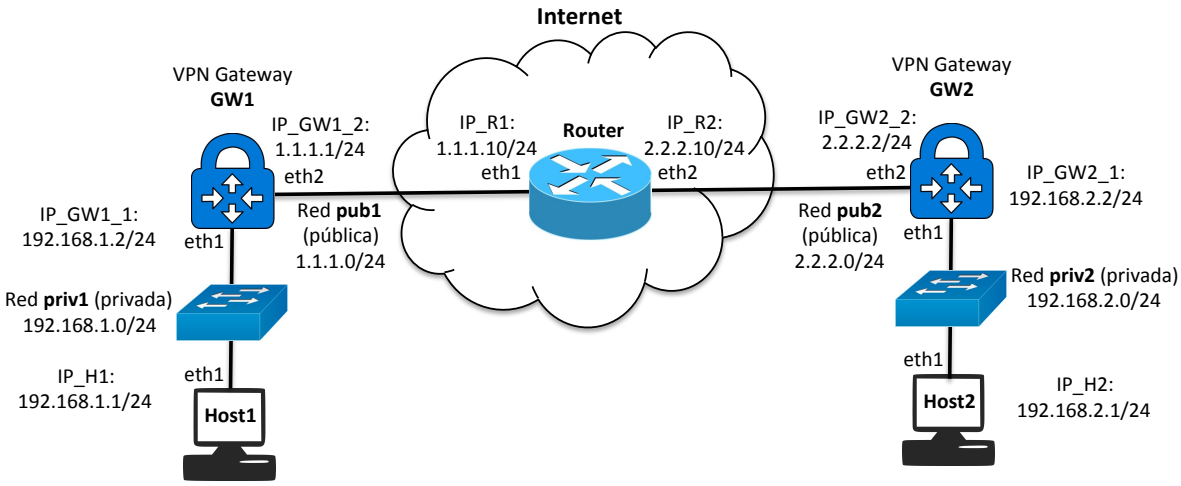
Seguridad en Redes

Práctica 3.5. IPsec

Preparación del entorno

En esta práctica vamos a implementar una VPN de tipo *site-to-site* basada en IPsec (módo túnel). Para ello, vamos a usar cinco MVs (`router`, `gw1`, `gw2`, `host1` y `host2`), dos redes internas que emulan redes privadas (`priv1` y `priv2`), y otas dos redes internas que emulan redes públicas (`pub1` y `pub2`):



Importa una MV, haz 4 clonaciones enlazadas y añade uno o dos interfaces de red a cada MV, según sea necesario, conectados a la red correspondiente.

Configura `router`:

```
sudo ifdown eth0
sudo ip link set dev eth1 up
sudo ip link set dev eth2 up
sudo ip addr add 1.1.1.10/24 broadcast + dev eth1
sudo ip addr add 2.2.2.10/24 broadcast + dev eth2
sudo sysctl -w net.ipv4.ip_forward=1
```

Configura `gw1`:

```
sudo apt-get update
sudo apt-get install ssh strongswan
sudo ifdown eth0
sudo ip link set dev eth1 up
sudo ip link set dev eth2 up
sudo ip addr add 192.168.1.2/24 broadcast + dev eth1
sudo ip addr add 1.1.1.1/24 broadcast + dev eth2
sudo ip route add default via 1.1.1.10
sudo sysctl -w net.ipv4.ip_forward=1
```

Configura `gw2`:

```
sudo apt-get update
```

```
sudo apt-get install ssh strongswan
sudo ifdown eth0
sudo ip link set dev eth1 up
sudo ip link set dev eth2 up
sudo ip addr add 192.168.2.2/24 broadcast + dev eth1
sudo ip addr add 2.2.2.2/24 broadcast + dev eth2
sudo ip route add default via 2.2.2.10
sudo sysctl -w net.ipv4.ip_forward=1
```

Configura `host1`:

```
sudo ifdown eth0
sudo ip link set dev eth1 up
sudo ip addr add 192.168.1.1/24 broadcast + dev eth1
sudo ip route add default via 192.168.1.2
```

Configura `host2`:

```
sudo ifdown eth0
sudo ip link set dev eth1 up
sudo ip addr add 192.168.2.1/24 broadcast + dev eth1
sudo ip route add default via 192.168.2.2
```

Configuración manual de IPsec (sin usar IKE)

Esta primera parte consiste en configurar una VPN de tipo *site-to-site* basada en IPsec (modo túnel) entre las dos pasarelas VPN (GW1 y GW2) sin utilizar el protocolo IKE para intercambio de claves. Para ello es necesario configurar las asociaciones de seguridad (SAs) y las políticas de seguridad (SPs) de forma manual en ambas pasarelas. Igualmente, será necesario generar las claves de cifrado y de autenticación también de forma manual (ambos extremos deben usar las mismas claves).

En esta práctica se usarán claves de cifrado de 192 bits (24 bytes) y claves de autenticación para HMAC de 128 bites (16 bytes). Para generar claves aleatorias en formato hexadecimal, de 16 y 24 bytes, respectivamente, se pueden usar las siguientes órdenes:

```
$ dd if=/dev/random count=16 bs=1| xxd -ps
$ dd if=/dev/random count=24 bs=1| xxd -ps
```

Importante: como se trata de claves hexadecimales, hay que añadir el prefijo `0x` delante de dichas claves.

A continuación es necesario definir las asociaciones y políticas de seguridad en el archivo `/etc/ipsec-tools.conf` de ambas pasarelas.

Las asociaciones de seguridad (SAs) realizarán el encapsulado de seguridad ESP de IPsec con cifrado de tipo `3des-cbc` y autenticación de tipo `hmac-md5`. Será necesario definir dos asociaciones de seguridad, una por cada sentido de la comunicación. Será necesario,

por tanto, generar dos claves aleatorias de 192 bits para el cifrado y otras dos claves de 128 bits para la autenticación. Adicionalmente, cada asociación de seguridad se identificará con un SPI distinto (Security Parameters Index), en nuestro ejemplo se utilizan los valores de SPI `0x201` y `0x301`, respectivamente.

Un ejemplo de archivo de configuración para la pasarela `GW1` sería el siguiente:

```
$ cat /etc/ipsec-tools.conf
#!/usr/sbin/setkey -f
# Vaciar las SAD y SPD
flush;
spdflush;

# Definir asociacionnes de seguridad (SAs) para ESP
# realizando cifrado con claves de 192 bit (algoritmo 3des-cbc)
# autenticación empleando claves de 128 bits (algoritmo hmac-md5)

add 1.1.1.1 2.2.2.2 esp 0x201 -m tunnel
      -E 3des-cbc 0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831
      -A hmac-md5 0xc0291ff014dccdd03874d9e8e4cdf3e6;

add 2.2.2.2 1.1.1.1 esp 0x301 -m tunnel
      -E 3des-cbc 0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df
      -A hmac-md5 0x96358c90783bbfa3d7b196ceabe0536b;

# Definir politicas de seguridad (SPs)
spdadd 192.168.1.0/24 192.168.2.0/24 any -P out ipsec
      esp/tunnel/1.1.1.1-2.2.2.2/require;

spdadd 192.168.2.0/24 192.168.1.0/24 any -P in ipsec
      esp/tunnel/2.2.2.2-1.1.1.1/require;
```

El archivo de la pasarela `GW2` sería similar, pero cambiando las políticas de seguridad (*in* y *out* intercambiados):

```
$ cat /etc/ipsec-tools.conf
#!/usr/sbin/setkey -f
# Vaciar las SAD y SPD
flush;
spdflush;

# Definir asociacionnes de seguridad (SAs) para ESP
# realizando cifrado con claves de 192 bit (algoritmo 3des-cbc)
# autenticación empleando claves de 128 bits (algoritmo hmac-md5)

add 1.1.1.1 2.2.2.2 esp 0x201 -m tunnel
      -E 3des-cbc 0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831
```

```
      -A hmac-md5 0xc0291ff014dccdd03874d9e8e4cdf3e6;

add 2.2.2.2 1.1.1.1 esp 0x301 -m tunnel
      -E 3des-cbc 0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df
      -A hmac-md5 0x96358c90783bbfa3d7b196ceabe0536b;

# Definir politicas de seguridad (SPs)
spdadd 192.168.2.0/24 192.168.1.0/24 any -P out ipsec
      esp/tunnel/2.2.2.2-1.1.1.1/require;

spdadd 192.168.1.0/24 192.168.2.0/24 any -P in ipsec
      esp/tunnel/1.1.1.1-2.2.2.2/require;
```

Una vez definidos los archivos de configuración en ambas máquinas, ejecuta ipsec en ambos extremos con la siguiente orden:

```
$ sudo /etc/init.d/setkey start
```

Entrega #1. Entrega los archivos de configuración `/etc/ipsec-tools.conf` de `gw1` y `gw2`.

Para ver que la VPN funciona, podemos establecer una conexión TCP entre las máquinas `host1` y `host2`, por ejemplo, usando `netcat` y generar tráfico entre ambas máquinas.

Ejecuta `wireshark` en el router y observa que todo el tráfico entre `host1` y `host2` que pasa por el router aparece encapsulado en paquetes ESP (observar las direcciones IP origen y destino de dichos paquetes). Analiza los paquetes ESP.

Observa las políticas y asociaciones de seguridad establecidas en `gw1` y `gw2` con las siguientes órdenes:

```
$ sudo ip xfrm policy
$ sudo ip xfrm state
```

Entrega #2. Copia y entrega las salidas de los comandos anteriores en `gw1`.

Configuración de IPsec usando IKE con clave secreta pre-compartida (psk)

En esta práctica implementaremos el mismo tipo de VPN que en el caso anterior, pero usaremos el protocolo IKE para intercambio automático de claves basado en una clave secreta pre-compartida (psk, *pre-shared key)*.

Se utilizará la implementación de IKE basada en StrongSwan. Para más información consultar la página web (https://www.strongswan.org), la página de manual del comando `ipsec`, que ofrece toda la funcionalidad de strongSwan y su archivo de configuración (`/etc/ipsec.conf`)

Configura la clave secreta, añadiendo al final del fichero `/etc/ipsec.secrets` de `gw1` la siguiente línea:

```
: PSK "Clave secreta muy segura"
```

Haz lo mismo en `gw2`:

```
: PSK "Clave secreta muy segura"
```

Normalmente, se usaría una clave generada aleatoriamente. Para ello, en lugar de una cadena, se puede indicar una secuencia de dígitos hexadecimales (comenzando con `0x`) o datos binarios codificados en Base64 (comenzando con `0s`).

En la configuración, `gw1` será el extremo izquierdo y `gw2`, el derecho. Eso permite tener la misma configuración en ambos extremos de la VPN. Sin embargo, la documentación de strongSwan sugiere denominar izquierdo al extremo local y derecho al remoto (aprovechando que, en inglés, comienzan por la misma letra).

Configura la VPN, añadiendo al fichero `/etc/ipsec.conf` de ambos extremos las siguientes líneas:

```
conn secret
     left=1.1.1.1
     leftsubnet=192.168.1.0/24
     leftauth=psk
     right=2.2.2.2
     rightsubnet=192.168.2.0/24
     rightauth=psk
      type=tunnel
     auto=start
```

Con `auto=start`, la conexión VPN se iniciaría automáticamente, lo cual es necesario si se pretende que la conexión sea permanente.

Adicionalmente, en el archivo de configuración anterior, se podrían incluir los algoritmos de cifrado específicos que usarán los protocolos ESP e IKE, por ejemplo:

```
esp=aes128-sha256
ike=aes128-sha256-modp3072
```

(si no se especifican estos parámetros, se usan los algoritmos de cifrado establecidos por defecto)

Reinicia el servicio en ambos extremos:

```
$ sudo ipsec restart
```

Ejecuta `wireshark` en el `router`.

Inicia la conexión VPN en `gw1`:

```
$ sudo ipsec up secret
```

Para ver que la VPN funciona, podemos establecer una conexión TCP entre las máquinas `host1` y `host2`, por ejemplo, usando `netcat` y generar tráfico entre ambas máquinas. Analiza los paquetes ISAKMP y ESP capturados por `wireshark` en el `router`.

Revisa el fichero de registro `/var/log/daemon.log`.

Entrega #3. Copia y entrega los nuevos registros del archivo `/var/log/daemon.log` de `gw1`.

Observa los detalles de la conexión en `gw1` y `gw2` con:

```
$ sudo ipsec status
$ sudo ipsec statusall
```

Observar las políticas y asociaciones de seguridad en `gw1` y `gw2` con:

```
$ sudo ip xfrm policy
$ sudo ip xfrm state
```

Entrega #4. Copia y entrega las salidas de los dos comandos anteriores en `gw1`.

Configura Wireshark para que descifre los paquetes ESP y compruebe su autenticidad. Para ello, en las preferencias del protocolo ESP (Edit → Preferences… → Protocols → ESP), activa todas las casillas y añade los parámetros de las asociaciones de seguridad.

Configuración de IPsec usando IKE con certificados autofirmados

Repetiremos la misma configuración que en la práctica anterior, epro en este caso, en lugar de usar una clave secreta pre-compartida, usaremos certificados autofirmados para generar la clave.

Crea una clave RSA y un certificado autofirmado en `gw1`:

```
$ sudo sh -c "ipsec pki --gen > /etc/ipsec.d/private/gw1-key.der"
$ sudo sh -c "ipsec pki --self --in /etc/ipsec.d/private/gw1-key.der --dn "CN=gw1" > /etc/ipsec.d/certs/gw1-cert.der"
```

Haz lo mismo en `gw2`:

```
$ sudo sh -c "ipsec pki --gen > /etc/ipsec.d/private/gw2-key.der"
$ sudo sh -c "ipsec pki --self --in /etc/ipsec.d/private/gw2-key.der --dn "CN=gw2" > /etc/ipsec.d/certs/gw2-cert.der"
```

Dado que son certificados autofirmados, deben estar accesibles localmente, ya que no se confiará en ningún certificado de este tipo intercambiado por la red. Ejecuta el siguiente comando en `gw1` para copiar el certificado de `gw2`:

```
$ sudo scp usuario@2.2.2.2:/etc/ipsec.d/certs/gw2-cert.der /etc/ipsec.d/certs/
```

Haz lo mismo en `gw2`:

```
$ sudo scp usuario@1.1.1.1:/etc/ipsec.d/certs/gw1-cert.der
/etc/ipsec.d/certs/
```

Configura las claves privadas, añadiendo al final del fichero `/etc/ipsec.secrets` de `gw1` la siguiente línea:

```
: RSA gw1-key.der
```

Haz lo mismo en `gw2`:

```
: RSA gw2-key.der
```

Configura la VPN, añadiendo al fichero `/etc/ipsec.conf` de ambos extremos las siguientes líneas:

```
conn sscert
      left=1.1.1.1
      leftsubnet=192.168.1.0/24
      leftcert=gw1-cert.der
      leftid="CN=gw1"
      right=2.2.2.2
      rightsubnet=192.168.2.0/24
      rightcert=gw2-cert.der
      rightid="CN=gw2"
      type=tunnel
      auto=start
```

Reinicia el servicio en ambos extremos:

```
$ sudo ipsec restart
```

Ejecuta `wireshark` en el `router`.

Inicia la conexión VPN en uno de los extremos:

```
$ sudo ipsec up sscert
```

Para ver que la VPN funciona, podemos establecer una conexión TCP entre las máquinas `host1` y `host2`, por ejemplo, usando `netcat` y generar tráfico entre ambas máquinas. Analiza los paquetes ISAKMP y ESP capturados por `wireshark` en el `router`.

Revisa el fichero de registro `/var/log/daemon.log` de `gw1` y `gw2`.

Entrega #5. Copia y entrega los nuevos registros del archivo `/var/log/daemon.log` de `gw1`.

Observa los detalles de la conexión en `gw1` y `gw2` con:

```
$ sudo ipsec status
$ sudo ipsec statusall
```

Observa las políticas y asociaciones de seguridad en `gw1` y `gw2` con:

```
$ sudo ip xfrm policy
$ sudo ip xfrm state
```

Entrega #6. Copia y entrega las salidas de los dos comandos anteriores en `gw1`.