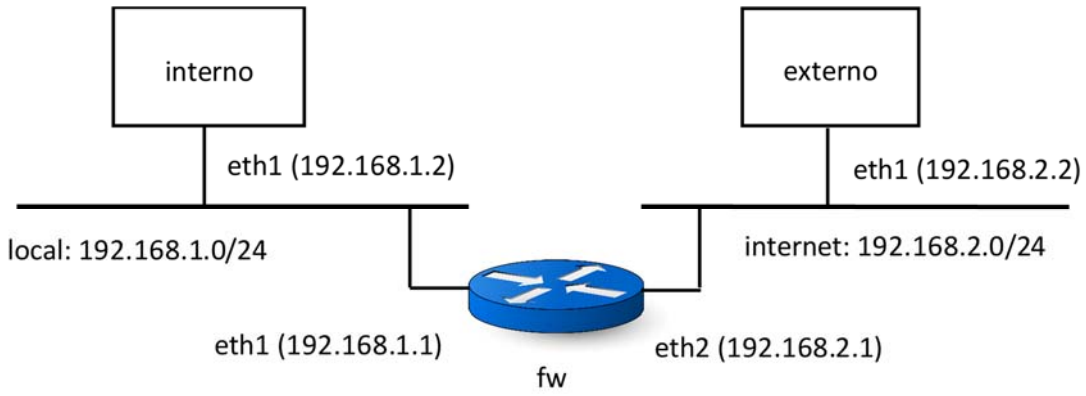
Seguridad en Redes

Práctica 3.3. Cortafuegos

Preparación del entorno

Vamos a usar dos redes internas (`internet` y `local`) y 3 MVs (`fw`, `externo` e `interno`) con la siguiente topología:



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

Configura `fw`:

```
sudo ifconfig eth1 192.168.1.1/24 up
sudo ifconfig eth2 192.168.2.1/24 up
sudo sysctl -w net.ipv4.ip_forward=1
sudo apt-get update
sudo apt-get install tinyproxy wireshark
```

Configura `externo`:

```
sudo ifconfig eth1 192.168.2.2/24 up
sudo route add –net 192.168.1.0/24 gw 192.168.2.1
sudo apt-get update
sudo apt-get install ssh hping3
```

Configura `interno`:

```
sudo ifconfig eth1 192.168.1.2/24 up
sudo route add –net 192.168.2.0/24 gw 192.168.1.1
sudo apt-get update
sudo apt-get install ssh hping3
```

Desconecta el adaptador 1 (`eth0`) de todas las MVs.

Sobre esta topología, vamos a configurar un cortafuegos de servidor protegido con

encaminador y sistema bastión combinados.

Arranca un servidor *web* simple en `externo` e `interno`:

```
$ sudo python -m SimpleHTTPServer 80
```

Filtrado de paquetes

Consulta la página de manual del comando `iptables` o, por ejemplo, https://help.ubuntu.com/community/IptablesHowTo.

- Establece la política por defecto de las tres cadenas de la tabla `FILTER` de `fw` a descartar (`DROP`).

- Permite el acceso a la interfaz *loopback*:

```
$ sudo iptables -A INPUT -i lo -j ACCEPT
$ sudo iptables -A OUTPUT -o lo -j ACCEPT
```

- Define reglas de filtrado de paquetes en `fw` para permitir conexiones a servidores externos iniciadas desde la red `local` y al servidor HTTP y SSH de `interno` iniciadas desde la red `internet`:

  - Aceptar paquetes reenviados recibidos por la interfaz interno (`eth1`).
  - Aceptar paquetes reenviados recibidos por la interfaz externo (`eth2`) con el puerto destino HTTP o SSH y la dirección IP destino de `interno`.
  - Aceptar paquetes reenviados recibidos por la interfaz externo pertenecientes a conexiones establecidas y relacionadas.
  - Registrar (`LOG`) el resto de paquetes reenviados (que serán descartados). Los paquetes descartados quedarán registrados en el fichero /var/log/syslog.

  Todas estas reglas se aplican a la cadena `FORWARD`.

- Haz *ping* desde `interno` a `externo` y al revés. ¿Se puede hacer el ping desde las dos máquinas? Comprueba lo registrado en el fichero /var/log/syslog.

  ```
  $ sudo tail /var/log/syslog
  ```

- Accede desde `interno` al servidor *web* de `externo` (con `wget` o con el navegador) y al revés. Accede desde `interno` a `externo` con SSH y al revés. ¿Se pueden hacer las conexiones HTTP y SSH a las dos máquinas? No cierres la conexión SSH en ninguna de las dos máquinas.

- Observa la tabla de estado de conexiones en el fichero `/proc/net/nf_conntrack`. Ya puedes cerrar la conexión SSH.

➢ **Entrega:** Copia la sintaxis de las reglas de iptables que has ido introduciendo. Copia, también, la salida de `iptables -L -v` en `fw`. Responde a las preguntas formuladas y explica por qué unas conexiones se pueden realizar y otras no. Copia el contenido de LOG que haga referencia a los *ping* realizados y el contenido de la tabla de estado.

OPCIONAL

- Prueba a hacer un ataque TCP SYN desde `interno` a `externo` utilizando direcciones IP origen aleatorias. No te olvides de desactivar las *syncookies* en `externo`.
  - **Entrega:** Indica los comandos que has tenido que ejecutar en cada máquina implicada para llevar a cabo el ataque y explica por qué el ataque ha tenido éxito.

- Para evitar que un ataque de este tipo tenga éxito, añade reglas de filtrado de paquetes en `fw` para impedir la suplantación de direcciones IP:
  - Elimina la primera regla de la cadena FORWARD y añade otra regla que acepte paquetes reenviados recibidos por la interfaz interno, pero solo de direcciones origen pertenecientes a la red `local` (192.168.1.0/24). Insertar esta regla en la posición 1 de la cadena (usar opción –I).

OPCIONAL

- Comprueba que ahora el ataque TCP SYN desde `interno` a `externo` no funciona.

  - Elimina las reglas de la cadena FORWARD que permiten el paso de paquetes HTTP y SSH por `eth2` y añade otras que acepten paquetes reenviados recibidos por la interfaz externo con puerto destino HTTP o SSH, dirección IP destino de `interno` y direcciones origen distintas a las de la red `local` (usa el operador negación `!`). Insertarlas en las posiciones 2 y 3 de la cadena (usar opción –I).

  ➢ **Entrega:** Copia la sintaxis de las últimas reglas de iptables que has introducido. Copia, también, la salida de `iptables -L -v` en `fw`.

OPCIONAL

- Clona una nueva máquina virtual y ponle de nombre `Host1` y configura su interfaz de red para que sea una máquina de la red `local` (IP 192.168.1.3). Comprueba que permitiendo que desde la red `internet` lleguen paquetes con una IP origen perteneciente a la red `local`, si lanzamos un ataque TCP SYN desde `externo` a `interno` haciéndonos pasar por `HOST1`, el ataque tiene éxito. Si, por el contrario, añadimos las reglas que evitan que entren por `eth2` paquetes con una IP de la red `local`, este ataque no funciona.

Proxy **HTTP**

Tinyproxy (https://tinyproxy.github.io/) es un *proxy* HTTP diseñado para ser rápido y ligero. Proporciona filtrado de URLs, monitorización de conexiones, control de acceso, inserción y borrado de cabeceras HTTP...

Consulta la página de manual y revisa su configuración (en `/etc/tinyproxy.conf`). Comprobar en la sección *ALLOW/DENY* si la red local está permitida, si no lo está añádela con *Allow*.

Arranca el tinyproxy con `$ sudo /etc/init.d/tinyproxy start`

Arranca el wireshark en `fw` capturando paquetes por la interfaz `eth1`.

- Modifica las reglas de filtrado del ejercicio anterior para que las conexiones HTTP iniciadas desde la red `local` pasen por el *proxy*:

  - Aceptar paquetes entrantes recibidos por la interfaz interno con puerto destino 8888 (donde escucha el *proxy*).
  - Aceptar paquetes salientes enviados por la interfaz externo con puerto destino HTTP.
  - Aceptar paquetes entrantes de conexiones establecidas.
  - Aceptar paquetes salientes de conexiones establecidas.
  - Descartar paquetes reenviados recibidos por la interfaz `eth1` con puerto destino HTTP.

  Para los cuatro primeros puntos, hay que definir reglas en las cadenas `INPUT` o `OUTPUT` para los paquetes entrantes o salientes, respectivamente. **Para el último punto, hay que insertar una regla en la cadena `FORWARD` antes de la que permite todos los paquetes reenviados recibidos por la interfaz `eth1`** (usa iptables –I).

Para acceder al servidor *web*, hay que configurar los clientes *web* de la red `local` para que accedan a través del *proxy*:

```
$ env http_proxy="http://192.168.1.1:8888" wget 192.168.2.2
```

O bien:

```
$ chromium --proxy-server="http://192.168.1.1:8888"
```

Comprueba el fichero de registro en `fw`:

```
$ sudo tail -f /var/log/tinyproxy/tinyproxy.log
```

Vuelve a acceder al servidor, pero capturando los paquetes en el wireshark de `fw` por la interfaz `eth2`. Fíjate que direcciones IP origen y destino aparecen tanto en los paquetes capturados en la interfaz `eth1` como en la `eth2`.

Para no tener que configurar los clientes, se puede usar un *proxy* transparente. Esto se haría con la siguiente regla en la tabla NAT del `fw`, que redirige las peticiones *web* al *proxy*:

```
$ sudo iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80
-j REDIRECT --to-port 8888
```

Abre el navegador y conéctate a http://192.168.2.2. Captura de nuevo la conexión con el wireshark comprobando las IPs de los paquetes que circulan por las dos interfaces.

- **Entrega:** Copia la sintaxis de las reglas de iptables que has ido introduciendo. También copia la salida de `iptables -L -v` y lo que se ha visualizado del fichero `tinyproxy.log` en `fw`. Indica cuáles son las IP origen y destino de los paquetes capturados en cada interfaz al conectarnos a través del *proxy* (configurando los clientes o con un *proxy* transparente).

Opcional: Configuración de un *proxy* inverso

Configura otro *proxy* (utilizar apache2) en `fw` y modifica las reglas de filtrado para que las conexiones HTTP desde la red `internet` vayan a través de este *proxy* (*proxy* inverso). Este *proxy* estaría protegiendo al servidor `externo`, ocultando su IP.

- **Entrega:** Explica, indicando los comandos usados, cómo has configurado el nuevo *proxy*.