Seguridad en Redes

Práctica 3.1. Exploración de redes

Creación de una máquina virtual (MV)

En todas las prácticas del Módulo 3, salvo en esta, vamos a usar una única máquina virtual (`ser.ova`). Para poder trabajar con máquinas virtuales será necesario entrar en Linux como "Usuario VMs", introduciendo tu nombre de usuario y contraseña. Una vez dentro debes seguir los siguientes pasos:

1. Abre la carpeta `Disco VMs` ubicada en el escritorio y ve al directorio `SER`.
2. Abre el fichero `SER.ova` haciendo doble *click*.
3. Pulsa en "Importar" en la ventana de VirtualBox que aparecerá.
4. Entra como usuario "usuario", con contraseña "usuario".

Podemos saltarnos todos estos pasos si al abrir VirtualBox aparece a la izquierda la máquina virtual "SER" como apagada; en este caso, simplemente se selecciona y se pulsa en *"Iniciar"* para arrancarla (o bien se hace doble clic sobre ella).

Para crear más máquinas, haremos una clonación enlazada y activaremos la casilla *"Reinicializar la dirección MAC de todas las tarjetas de red"* para que no haya colisiones.

Creación de redes virtuales

Cada MV puede tener habilitados hasta cuatro interfaces de red. Para ello en la ventana de Virtual Box seleccionamos la MV y pulsamos en *Máquina → Configuración → Red → Adaptador N*, y dentro del adaptador marcaríamos *"Habilitar adaptador de red"*.

Cada adaptador puede estar conectado a un tipo de red. Los más importantes son:

- NAT: Permite conectarse al exterior por medio de NAT (*Network Address Translation*).
- Red interna: Crea una red virtual (basada en *software*) que conecta las máquinas virtuales seleccionadas (no tiene conexión con el exterior).
- Adaptador sólo-anfitrión: Es similar a la anterior, pero también conecta el anfitrión, por medio de un interfaz de red virtual (no tiene conexión con el exterior).

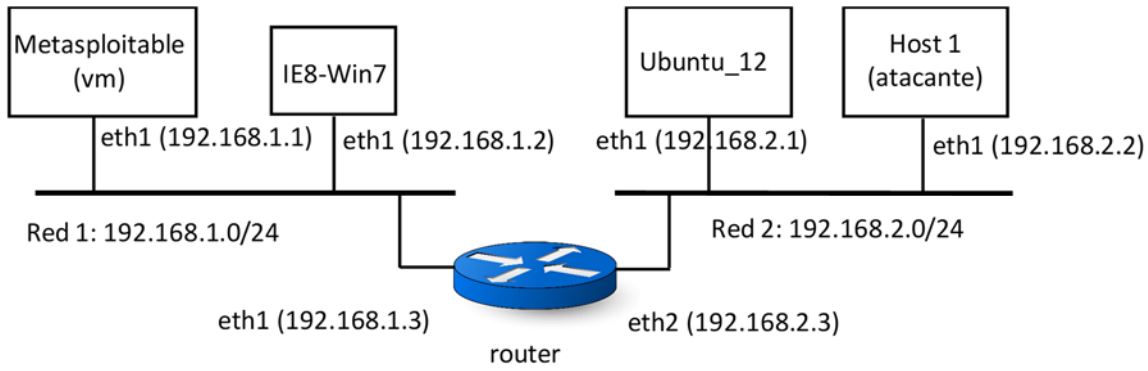
El modo NAT es útil cuando queremos conectarnos al exterior desde la MV (por ejemplo, para descargar un paquete). El modo *Red interna* es útil para aislar el entorno completamente del exterior. El modo *Adaptador sólo anfitrión* es útil cuando queremos acceder a las MVs desde la máquina anfitrión. Además, en este modo es posible configurar un servidor DHCP en el anfitrión para gestionar las direcciones IP de manera centralizada.

En el modo *Red interna*, donde no hay DHCP, hay que configurar los interfaces manualmente.

Habilitaremos el interfaz `eth0` (que se corresponde con el adaptador 1) de cada MV en modo NAT para poder instalar paquetes desde Internet. Una vez instalados los paquetes necesarios, lo desconectaremos desmarcando la casilla “Cable conectado” o bien ejecutando `sudo ifconfig eth0 down` desde un terminal dentro de la MV.

Creación de entornos virtuales para pruebas

Vamos a usar la siguiente topología de red, con 5 MVs y dos redes internas:



Para reproducir esta topología con las MV del laboratorio tendrás que seguir una serie pasos:

1. El primer paso consistirá en importar las máquinas virtuales y activar sus interfaces de red.
   - Las máquinas Host1 y router son máquinas virtuales `SER.ova`. Para configurarlas, accede a esta MV a través de la carpeta `Disco VMs` del Escritorio. Haz doble clic sobre la máquina y pulsa la opción Importar de Virtual Box. A continuación, selecciona la MV SER en la parte izquierda de Virtual Box y elige la opción *Configuración* al pulsar el botón derecho del ratón. Cambiar el nombre de la máquina a Host1. Vuelve a pulsar el botón derecho del ratón y selecciona la opción *Clonar* para hacer una clonación enlazada. Asigna a la nueva MV el nombre de router y **activa la casilla *“Reinicializar la dirección MAC de todas las tarjetas de red”*.**
   - Las MVs Metasploitable, IE8-Win7 y Ubuntu_12 son tres MV distintas que tendrás que descargar del Campus Virtual. Una vez descargadas, ir a la carpeta Descargas y proceder exactamente igual que con la máquina `SER.ova` para importar las máquinas a Virtual Box. En este caso no es necesario cambiar el nombre de las máquinas ni tampoco clonarlas.

2. Antes de arrancar las máquinas, activa (desde VirtualBox) los interfaces de red que sean necesarios.
   Todas las máquinas tendrán habilitado el Adaptador 1 (interfaz eth0) conectado a una red tipo NAT, los demás adaptadores habilitados estarán conectados a una Red interna. Utilizar Red 1 y Red 2 como nombre de la red según corresponda. De este modo, las MV tendrán los siguientes adaptadores conectados a Red interna:

- Host 1: Adaptador 2 (interfaz eth1)
- Router: Adaptador 2 (interfaz eth1) y 3 (interfaz eth2)
- Metasploitable: Adaptador 2
- IE8-Win7: Adaptador 2
- Ubuntu_12: Adaptador 2

3. Arranca las MVs. Los usuarios y contraseñas de las MVs son:

   - Host1 y router: usuario; usuario
   - Metasploitable: msfadmin; msfadmin
   - Ubuntu_12: tubunu (solo necesito contraseña)

4. Por ultimo, hay que configurar los interfaces de red conectados a una Red interna desde un terminal dentro de cada MV.

   - Configura `router`:

     ```
     sudo ifconfig eth1 192.168.1.3/24 up
     sudo ifconfig eth2 192.168.2.3/24 up
     sudo sysctl -w net.ipv4.ip_forward=1
     sudo apt-get update
     sudo apt-get install ssh telnetd wireshark
     ```

     Al instalar el wireshark no permitir que los usuarios sin privilegios de administración puedan capturar paquetes, es decir, que solo pueda funcionar si lo ejecuta el root

   - Configura `Metasploitable`:

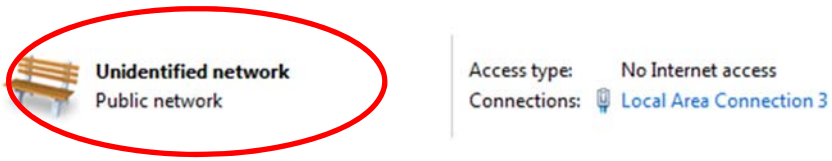
     ```
     sudo ifconfig eth1 192.168.1.1/24 up
     sudo route add –net 192.168.2.0/24 gw 192.168.1.3
     ```

   - Configura `Host1`:

     ```
     sudo ifconfig eth1 192.168.2.2/24 up
     sudo route add –net 192.168.1.0/24 gw 192.168.2.3
     sudo apt-get update
     sudo apt-get install nmap
     ```

   - Configura `IE8-Win7`:

     Ir a `Start > Control Panel > Network and Internet > View Network Status and Tasks`. Vamos a trabajar con la conexión que no tiene acceso a internet (que debería ser la correspondiente al segundo adaptador)



Pulsamos sobre `Local Area Connection 3 > Properties > Internet Protocol Version 4 (TCP/IPv4) > Properties`, aparecerá una ventana como la de la figura, rellénala con:

- IP address: 192.168.1.2
- Subnet mask: 255.255.255.0
- Default Gateway: 192.168.1.3
- Preferred DNS server y Alternate DNS server: en blanco

➢ Configura `Ubuntu_12`:

```
sudo ifconfig eth1 192.168.2.1/24 up
sudo route add –net 192.168.1.0/24 gw 192.168.2.3
```

5. Comprueba la conectividad entre las MVs con `ping`.

6. Desconecta el interfaz `eth0` de todas las MVs.

Escucha (*Sniffing*)

Wireshark

Esta herramienta se ha utilizado ampliamente en la asignatura de Redes, por lo que no nos detendremos mucho en ella. Será muy útil para analizar el tráfico generado durante los ataques que realizaremos en la siguiente sesión y para resolver cualquier problema que pueda aparecer.

- Para iniciarla, ejecuta en `router`:
  `$ sudo wireshark`

- Activa la captura de paquetes desde el interfaz eth2. A continuación, haz un `ping` desde Host 1 a Ubuntu_12 y después un `ping` desde Host1 a Metasploitable.

  ➢ **Entrega**: Explica qué tramas se observan en la máquina router y a cuál de los comandos `ping` ejecutados pertenecen. Especifica el tipo de datos contenidos en la trama (ARP o ICMP), las direcciones IP origen y destino, y las direcciones MAC origen y destino (relaciona el valor de estas direcciones con los nombres de las máquinas).

Exploración de redes (*Network scanning*)

nmap

Consulta la página de manual del comando `nmap` y la documentación en http://nmap.org/book/man.html o http://nmap.org/man/es/.

Nmap es una herramienta de exploración de redes que permite identificar cuáles son las máquinas que hay en una red, qué puertos tienen abiertos, qué sistema operativo o servicios (incluidas versiones) tienen instalados estas máquinas, si hay algún tipo de cortafuegos instalado en la máquina, … Los sistemas objetivo se pueden especificar con una secuencia

de nombres de *hosts*, una secuencia de direcciones IP, una lista (`192.168.1.1,2`), un rango (`192.168.1.1-2`), una dirección de red (`192.168.1.0/24`) o comodines (`192.168.1.*`). También se pueden limitar los puertos a explorar con la opción `-p`, seguida de una lista o rango de puertos.

En esta práctica, vamos a probar sobre la arquitectura de red de la figura distintas posibilidades de uso de nmap.

1. **Descubrimiento de IPs de una red.**
   Normalmente, cuando se explora una red, se detectan primero las IPs en uso en esa red con un escaneo poco intrusivo. Una vez detectadas estas máquinas, se hace un escaneo más profundo de cada una de ellas.

   Para el descubrimiento de IPs que vamos a hacer no se necesitan permisos de root; sin embargo, los resultados obtenidos son diferentes según ejecutemos el comando con o sin permisos de root.

   - Abre wireshark (si no lo tenías abierto) en la máquina router, reinicia la captura de paquetes (Capture → Restart) y ejecuta en Host1 los comandos,

     ```
     $ nmap –sn 192.168.2.0/24
     $ sudo nmap –sn 192.168.2.0/24
     ```

     - **Entrega**: Copia la salida de ambos sondeos. ¿Qué diferencia hay en el resultado que devuelven los dos comandos? Explica, a partir de lo capturado en el wireshark, qué comandos usa nmap para detectar las máquinas que están activas.

   - Ahora, haz un Capture → Restart en el wireshark y ejecuta en Host1:

     ```
     $ nmap –sn 192.168.1.0/24
     $ sudo nmap –sn 192.168.1.0/24
     ```

     - **Entrega**: Copia la salida de ambos sondeos. ¿Hay alguna diferencia en el resultado que devuelven los dos comandos? ¿a qué crees que es debido que ahora la ejecución del comando con `sudo` no devuelva la misma información que en la red 192.168.2.0/24? Explica, a partir de lo capturado en el wireshark, que comandos usa nmap para detectar las máquinas que están activas.

2. **Escaneo de puertos.**

   Los distintos sondeos consisten en enviar distintos mensajes TCP (activando distintas marcas o *flags*) y observar los mensajes que devuelve el sistema objetivo (o los cortafuegos que están delante de ellos). Por ejemplo, si se envía un mensaje con el *flag* SYN a un puerto abierto, el sistema responderá con un mensaje con los *flags* SYN y ACK, mientras que, si se envía a un puerto cerrado, el sistema responderá con un mensaje con el *flag* RST. Sin embargo, si se envía un mensaje con el *flag* FIN a un puerto

abierto, el sistema no responderá, mientras que, si se envía a un puerto cerrado, el sistema responderá con un mensaje con el *flag* RST.

Los sondeos típicos para detectar puertos TCP abiertos utilizan las opciones:

-sS: Escaneo TCP SYN, envía un mensaje TCP con el *flag* SYN activo
-sT: Escaneo TCP Connect, solicita al SO establecer una conexión TCP (muy similar al TCP SYN)
-sF: Escaneo TCP FIN, envía un mensaje TCP con el *flag* FIN activo
-sA: Escaneo TCP ACK, envía un mensaje TCP con el *flag* ACK activo

Si no se indica el tipo de sondeo, se realiza un sondeo TCP SYN o TCP *Connect* en función de si se dispone o no de privilegios.

Es interesante usar las opciones `--reason`, que indica la razón por la que se considera que el puerto está abierto, cerrado o filtrado, y `--packet-trace`, que muestra una descripción de los paquetes intercambiados.

- Realiza los siguientes sondeos desde `Host1` (antes de cada sondeo se recomienda borrar la ventana del wireshark de la máquina router con Capture → Restart):

```
$ sudo nmap -sS 192.168.2.3 --reason
$ sudo nmap -sS 192.168.1.2
$ sudo nmap -sS 192.168.1.1
```

  - **Entrega**: Copia la salida de cada sondeo. Explica a partir de las capturas de wireshark qué tipo de mensaje devuelve cada máquina cuando el puerto está abierto y qué tipo devuelve cuando el puerto está cerrado. ¿Se pueden detectar con esta opción puertos UDP que estén abiertos? Justifica la respuesta.

- A continuación, vamos a introducir en la **máquina Metasploitable** una regla de un cortafuego que no deje pasar los paquetes TCP con el flag SYN activo dirigidos al puerto 80,

```
$ sudo iptables –A INPUT –p tcp –tcp-flags ALL SYN –dport 80 –j DROP
```

- Ejecuta de nuevo los escaneos –sS y –sF pero solo sobre este puerto (opción –p).

  - **Entrega**: Copia la salida de cada sondeo.

- Realiza el escaneo anterior sobre la máquina router con la opción `--packet-trace`, observa que la información que devuelve es similar a la capturada con el wireshark.

- Explora los puertos TCP abiertos de las máquinas Ubuntu_12 y Metasploitable con TCP FIN y TCP ACK y la opción --reason en ambos casos.

- **Entrega**: Copia la salida de cada sondeo. En el caso de la máquina Metasploitable, ¿qué diferencias observas en el resultado que nos devuelven cada uno de los sondeos?

- Explora los puertos UDP del 1 al 200 en la máquina Metasploitable.

  - **Entrega**: Copia el comando que has ejecutado para realizar el sondeo y la salida obtenida tras la ejecución del mismo. Explica a partir de lo capturado en el wireshark qué mensajes envía Host1 para realizar el sondeo y cómo detecta que el puerto está abierto o cerrado.

3. Detectar servicios y versiones

Detecta servicios y versiones con la opción -sV:

```
$ nmap -sV direccion_IP
```

Más información en http://nmap.org/book/vscan.html.

- **Entrega**: Copia la salida obtenida al tratar de detectar los servicios y versiones asociados a cada puerto en la máquina Metasploitable (solo la parte en la que se especifica número de puerto, estado, servicio y versión).

4. Detectar sistema operativo

Nmap trata de detectar el sistema operativo de un sistema con la opción -O:

```
$ sudo nmap -O dirección_IP
```

A partir de varias pruebas, se crea una huella (*fingerprint*) del sistema que se compara con las del fichero `/usr/share/nmap/nmap-os-db`. Más información en http://nmap.org/book/osdetect.html.

- **Entrega**: Copia la salida obtenida al tratar de detectar el sistema operativo de las máquinas router, Ubuntu_12, IE8_WIN7 y Metasploitable.

Puede ocurrir que el sistema a explorar tenga instalado un cortafuego, nmap cuenta con opciones que permiten eludir cortafuegos y sistemas de detección de intrusos. Más información en http://nmap.org/book/man-bypass-firewalls-ids.html.