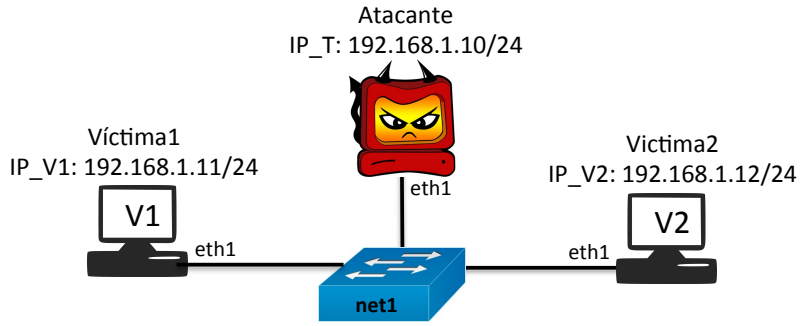
Seguridad en Redes

Práctica 4.1. Ataques Ethernet y ARP

1. Preparación del entorno

Vamos a usar la topología de red de la figura, con tres MVs.



Para ello, creamos una primera MV importando el servicio virtualizado `SER.ova` y luego creamos dos clonaciones enlazadas de dicha máquina. Cada MV tendrá dos interfaces de red:

- El adaptador 1 (`eth0`) configurado en modo NAT para instalar los paquetes necesarios (una vez instalados los paquetes necesarios, se deshabilitará la interfaz `eth0` para evitar cualquier tipo de conflicto.)
- El adaptador 2 (`eth1`) configurado en modo red interna, que se usará comunicarnos entre las distintas máquinas virtuales. Las tres interfaces `eth1` deben estar conectados a la misma red interna (por ejemplo, lo llamamos `net1`)

Configura `atacante`:

```
sudo apt-get update
sudo apt-get install dsniff nmap
sudo ifdown eth0
sudo ip link set dev eth1 up
sudo ip addr add 192.168.1.10/24 broadcast + dev eth1
```

Configura `victima1`:

```
sudo apt-get update
sudo apt-get install ssh telnet telnetd
sudo ifdown eth0
sudo ip link set dev eth1 up
sudo ip addr add 192.168.1.11/24 broadcast + dev eth1
```

Configura `victima2`:

```
sudo apt-get update
sudo apt-get install ssh telnet telnetd
sudo ifdown eth0
sudo ip link set dev eth1 up
sudo ip addr add 192.168.1.12/24 broadcast + dev eth1
```

Comprueba la conectividad entre las MVs con `ping`.

2. 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 `atacante`:

```
$ sudo wireshark
```

A continuación hacer un ping de `victima1` a `victima2`, ejecutando la siguiente orden en `victima1`:

```
$ ping -c 3 192.168.1.12
```

Comprobar si desde el atacante, hemos podido visualizar algún paquete de esta comunicación. En principio, puesto que las máquinas están conectadas a través de un switch implementado en Virtual Box, la máquina atacante no debería ver ningún paquete de la comunicación entre `victima1` y `victima2`.

Es posible cambiar la configuración de las máquinas virtuales, de manera que puedan ver todos los paquetes que intercambian otras máquinas conectadas a la misma red interna, de manera que el comportamiento de la red sea como una red de difusion (tipo hub), en lugar de una red conmutada (tipo switch). Para ello es necesario habilitar el modo promiscuo en la configuración de Virtual Box, mediante la siguiente opción del menú:

`Devices` → `Network` → `Network Settings -> Adapter 2` → `Advanced` → `Promiscous Mode` → **`Allow All`**

Activar esta opción en TODAS las máquinas y repetir el ping de `victima1` a `victima2`:

```
$ ping -c 3 192.168.1.12
```

Comprobar que ahora `atacante` sí puede ver los paquetes de esta comunicación.

telnet

Telnet es una aplicación cliente/servidor que permite abrir una sesión (*shell*) en una máquina remota. `victima1` actuará como cliente `telnet` y `victima2` como servidor servidor (`telnetd`), que utiliza el puerto TCP 23. Telnet es una aplicación no cifrada, esto significa que toda la información que intercambian el cliente y servidor, incluido el nombre de usuario y la contraseña, se transmiten en forma de texto claro (sin cifrar) a través de la red.

Arrancar el sniffer de red (`wireshark`) en el `atacante`:

```
$ sudo wireshark
```

Desde el cliente `victima1` ejecutar:

```
$ telnet 192.168.1.12
```

Introducir el login `usuario` y la contraseña `usuario`. A continuación podemos ejecutar comandos UNIX en el servidor, por ejemplo: `ls, pwd,` etc. Para salir de la sesión telnet ejecutar el comando `exit`.

Comprobar que a través del sniffer de red de `atacante` podemos ver todos los paquetes de esta comunicación. Pinchar sobre uno de los paquetes de la misma y seleccionar la siguiente opción del menú de `wireshark`:

`Analyze → Follow TCP Stream`

Esta opción nos permite visualizar en formato ASCII toda la información intercambiada entre el cliente y servidor correspondiente a esta conexión. Comprobar que podemos ver el nombre de usuario y la contraseña transmitida por la red.

ssh

ssh es una aplicación cliente/servidor que permite abrir una sesión (*shell*) en una máquina remota, pero usando comunicación cifrada, de manera que un atacante que visualice los paquetes de una conexión ssh, no podrá obtener ninguna información útil de la misma. El servidor ssh (`sshd`) utiliza el puerto TCP 22.

Arrancar el sniffer de red en el `atacante`:

```
$ sudo wireshark
```

Desde el cliente `victima1` ejecutar:

```
$ ssh usuario@192.168.1.12
```

Introducir la contraseña `usuario`. A continuación podemos ejecutar comandos UNIX en el servidor, por ejemplo: `ls, pwd,` etc. Para salir de la sesión ssh ejecutar el comando `exit`.

En el sniffer de red del atacante, pinchar sobre uno de los paquetes de la conexión ssh y seleccionar la siguiente opción del menú de `wireshark`:

`Analyze → Follow TCP Stream`

Observaremos que, en este caso, no es posible ver el contenido de los paquetes intercambiados ya que la información se transmite en forma cifrada.

Una vez finalizada esta práctica, configurar de nuevo las maquinas virtuales para que la red se comporte de nuevo como una red conmutada (tipo switch), mediante la siguiente opción del menú de Virual Box (se debe realizar en TODAS las máquinas):

`Devices → Network → Network Settings -> Adapter 2 → Advanced → Promiscous Mode →` **`Deny`**

3. ARP *poisoning*

ARP *poisoning* local

Si tenemos acceso como superusuario a una máquina, podemos alterar de forma manual la tabla ARP de la misma y provocar una denegación de servicio

A continuación hacer un ping de `victima1` a `victima2`, ejecutando la siguiente orden en `victima1`:

```
$ ping -c 3 192.168.1.12
```

La conexión entre ambas máquinas funcionará correctamente. Comprobar la tabla ARP de `victima1` mediante la siguiente orden:

```
$ sudo ip neig show
```

Veremos que `victima1` tiene almacenada en su tabla ARP la dirección IP y la dirección MAC de `victima2`.

A continuación, modificamos la tabla ARP de `victima1` para introducir una dirección MAC falsa asociada a la IP de `victima2`. Para ello ejecutamos la siguiente orden en `victima1`:

```
$ sudo ip neig change 192.168.1.12 lladdr 00:01:02:03:04:05
  nud permanent dev eth1
```

Entrega #1: Copia y entrega el contenido de la tabla arp de `victima1` (comando: `$ sudo ip neig show`).

A continuación, hacer un ping de `victima1` a `victima2`, ejecutando la siguiente orden en `victima1`:

```
$ ping -c 3 192.168.1.12
```

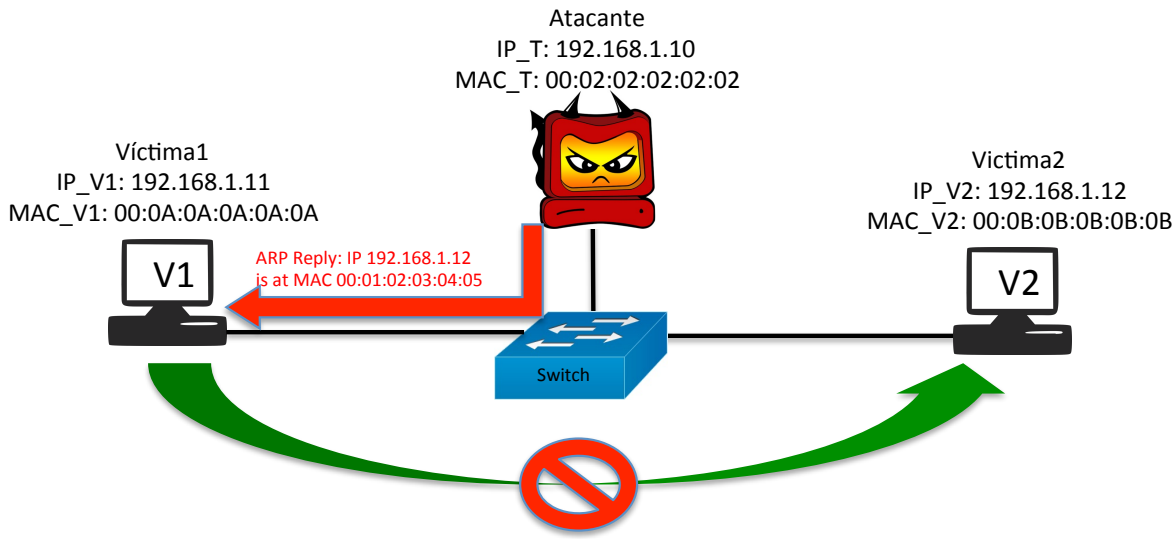
Observaremos que no hay comunicación entre `victima1` y `victima2`. Comprobar que tampoco podemos establecer una conexión telnet o ssh entre `victima1` y `victima2`.

Para eliminar la entrada falsa de la tabla ARP introducida anteriormente, ejecutar la siguiente orden en `victima1`:

```
$ sudo ip neig delete 192.168.1.12 lladdr 00:01:02:03:04:05
  nud permanent dev eth1
```

ARP *poisoning* remoto

Un atacante remoto puede introducir una entrada falsa en la tabla ARP de una máquina víctima, mediante el anuncio de mensajes ARP reply falsos. Por ejemplo, en la figura siguiente, el `atacante` anuncia a `victima1` una dirección MAC falsa asociada a la dirección IP de `victima2`. De esta forma `victima1` no podrá comunicarse con `victima2`.



Para realizar este ataque usaremos el comando `nping`, que es una de las utilidades del paquete `nmap` instalado en el atacante.

Ejecuta la siguiente orden en `atacante`:

```
$ nping --arp-type arp-reply --source-mac 00:01:02:03:04:05
  --source-ip 192.168.1.12 -c 9999 192.168.1.11
```

Los parámetros de esta orden son los siguientes:

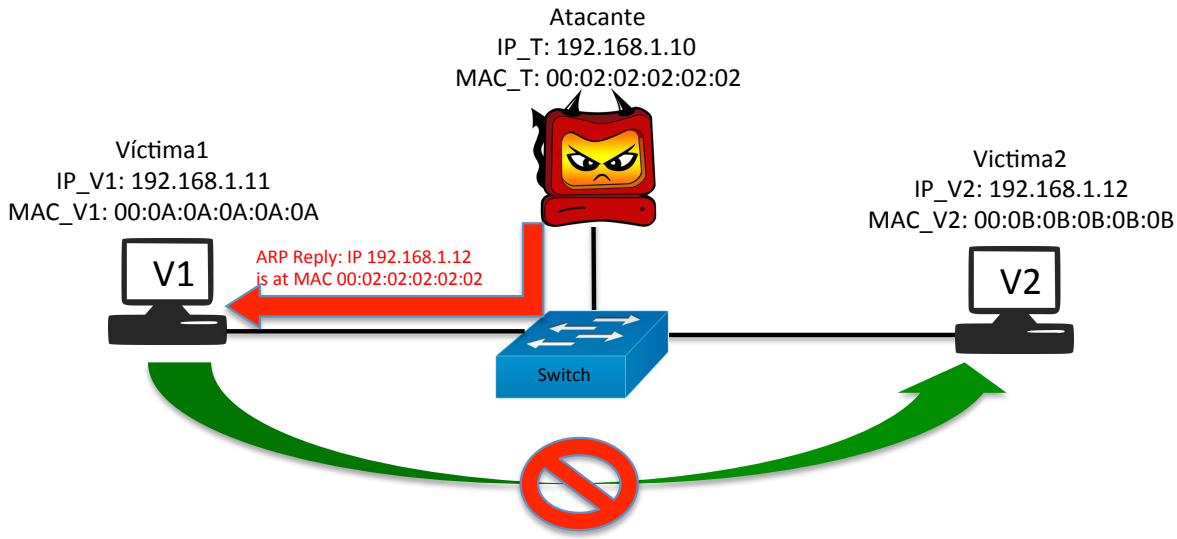
`--arp-type arp-reply`	Tipo de paquete que estamos transmitiendo (ARP reply)
`--source-mac 00:01:02:03:04:05`	Dirección MAC falsa que estamos anunciando en los mensajes ARP reply
`--source-ip 192.168.1.12`	Dirección IP fuente que estamos anunciando en los mensajes ARP reply, asociada a la MAC falsa
`-c 9999`	Nº de mensajes ARP Reply que enviamos a la red
`192.168.1.11`	Dirección IP de la máquina a la que va dirigido el ataque (target)

Comprobar que no es posible realizar un `ping` ni establecer una conexión `telnet` o `ssh` entre `victima1` y `victima2`. Abortar el comando `nping` con `^C`.

Entrega #2: Copia y entrega la salida del comando `nping` y también el contenido de la tabla arp de `victima1`

4. ARP spoofing

El ataque ARP spoofing es similar a ARP poisoning, pero el objetivo es suplantar a una máquina de la red. Para ello, mediante el anuncio de mensajes ARP reply falsos, el atacante asocia la dirección IP de la víctima a su propia dirección MAC, tal y como se muestra en la figura:



De esta forma, los paquetes que envía `victima1` a `victima2` pararán por el atacante. Si queremos que el atacante redirija los paquetes a `victima2` debemos activar el *forwarding* en el atacante.

Para realizar este ataque, podemos utilizar la herramienta `nping` usada en el ejercicio anterior, sustituyendo la MAC falsa por la MAC de `atacante`. Otra opción es utilizar la herramienta `arpspoof`, que es una de las utilidades del paquete `dsniff` instalado en el atacante.

Ejecuta la siguiente orden en `atacante`:

```
$ sudo arpspoof -i eth1 -t 192.168.1.11 192.168.1.12
```

Los parámetros de esta orden son los siguientes:

`-i eth0`	Interfaz de red por la que se realiza el ataque
`-t 192.168.1.11`	Dirección IP de la máquina a la que van dirigido el ataque (target)
`192.168.1.12`	Dirección IP fuente que estamos anunciando en los mensajes ARP reply, asociada a la MAC del atacante. Es la máquina que queremos suplantar.

Hacer un ping entre `victima1` y `victima2`. Comprobar que los mensajes `ICMP echo request` no llegan a `víctima2` pero sí llegan al `atacante` (usar `wireshark` en el `atacante` para ver estos mensajes). Si queremos que el `atacante` reenvíe los mensajes a `victima2`, debemos activar el forwarding en el `atacante`, con la siguiente orden:

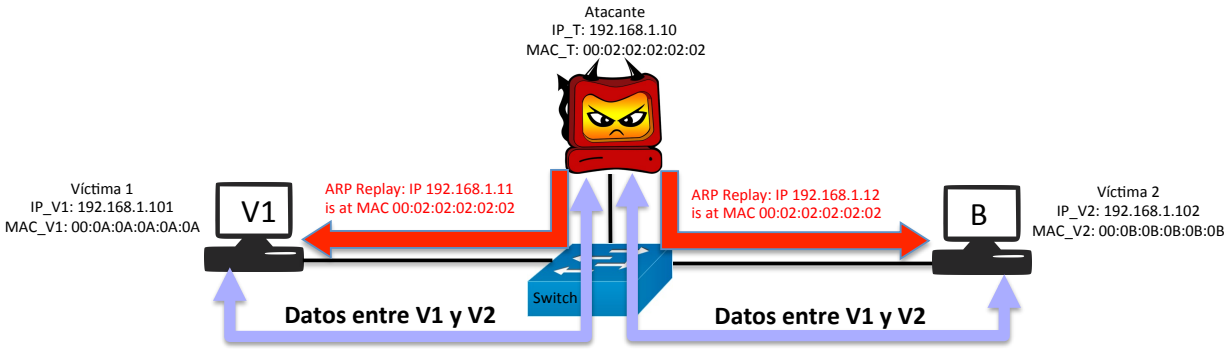
```
$ sudo sysctl –w net.ipv4.conf.all.forwarding=1
```

Abortar el comando `arpspoof` con `^C`.

Entrega #3: Copia y entrega la salida del comando `arpspoof` y también el contenido de la tabla arp de `victima1`

5. Ataque man-in-the-middle

El ataque *man-in-the-middle* es similar al anterior, pero en este caso, se asocia la MAC del atacante a las IPs de ambas víctimas, tal y como se muestra en la figura. De esta manera, todo el tráfico que intercambian las víctimas para a través del atacante. Para que el tráfico llegue de una víctima a otra, el atacante debe tener activado el *forwarding*.



Para realizar este ataque usaremos el mismo comando `arpspoof`, que en el ejercicio anterior, pero añadiendo la opción `-r`, para realizar el *ARP spoofing* sobre ambas víctimas.

Ejecuta las siguientes órdenes en `atacante`:

```
$ sudo sysctl -w net.ipv4.conf.all.forwarding=1
$ sudo arpspoof -i eth1 -t 192.168.1.12 -r 192.168.1.11
```

Comprobar que sí es posible realizar un `ping`, o establecer una conexión `telnet` entre `victima1` y `victima2`, pero observar que todo el tráfico que intercambian `victima1` y `victima2` pasa a través del `atacante` (usar `wireshark` en el `atacante` para visualizar este tráfico). Abortar el comando `arpspoof` con `^C`.

Entrega #4: Copia y entrega la salida del comando `arpspoof` y también el contenido de las tablas arp de `victima1` y `victima2`