Configuración del laboratorio virtual de seguridad

Laura Bezanilla Matellán

Índice

1. Características de las máquinas virtuales del entorno de trabajo virtual

a. Alice

Esta máquina tiene un sistema operativo de escritorio Linux, concretamente Ubuntu con una arquitectura de 32 bits. Tiene asignado un solo procesador con 512MB de memoria RAM. Para poder ver la velocidad de transmisión del adaptador de red ejecuto el siguiente comando, como se muestra en la imagen:

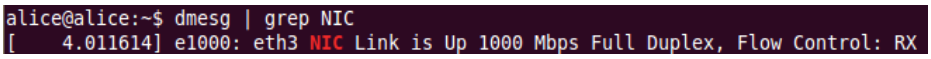

Figura 1. Velocidad de transmisión del adaptador de red de alice

Gracias a este comando podemos ver los mensajes del núcleo, generados durante el arranque de la máquina en la pantalla. A continuación, busco la cadena NIC, que es la tarjeta de interfaz de red, para poder comprobar que su velocidad es 1000 Mbps.

b. Bob

Esta máquina ejecuta un sistema operativo Debian con una arquitectura de 32 bits y está configurado como un servidor Linux. De la misma manera, tiene dedicado un procesador con 384MB de memoria RAM. La velocidad de transmisión de esta máquina es 1000 Mbps:

Figura 2. Velocidad de transmisión del adaptador de red de bob

c. Mallet

Esta máquina juega el papel del ordenador del adversario. Como *alice* ejecuta un sistema operativo de escritorio Ubuntu con una arquitectura de 32 bits. También tiene asignado un procesador con 512MB de memoria RAM. Finalmente, la velocidad de transmisión del adaptador de red es 1000 Mbps:

Figura 3. Velocidad de transmisión del adaptador de red de mallet

2. Configuración de la red del entorno de virtualización

Explicación paso a paso de como configuré las máquinas virtuales para que las tres estuvieran dentro de la red NAT 10.0.2.0/24 con el nombre “GSI”.

i. Crear la red NAT con el nombre “GSI”

Para ello utilicé las opciones de VirtualBox. Una vez abierto el programa se va a *Archivo > Preferencias > Red > Agregar nueva red NAT > Editar la red NAT seleccionada.*

En este punto, aparecerá una ventana emergente con los detalles de la red NAT seleccionada. Aquí cambiamos el nombre de “NatNetwork” por “GSI”.

ii. Meter a las tres máquinas virtuales dentro de la red NAT “GSI”

Lo primero que se debe hacer es seleccionar la máquina que se quiere meter en dicha red.

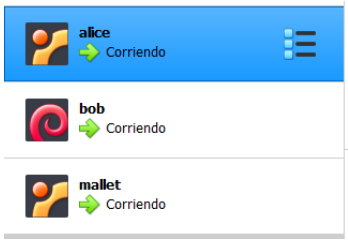

Figura 4. Seleccionar la primera máquina

El siguiente paso es hacer click en el nombre de Red interna que aparece en azul.

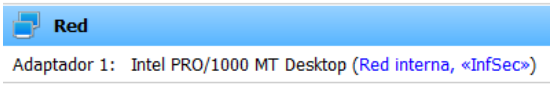

Figura 5. Seleccionar la red por defecto

El último paso sería cambiar los parámetros por defecto, que aparecen en la nueva ventana, por los que necesitamos.

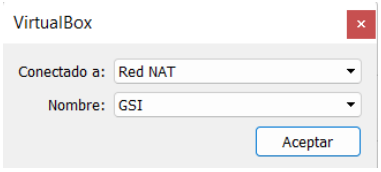

Figura 6. Conectar la máquina a la red que previamente habíamos creado

Sería necesario repetir este procedimiento con las máquinas de *bob* y *mallet*. Una vez terminado el proceso, tendría la red NAT bien configurada a nivel de hardware.

Como he configurado el entorno con una red NAT, tendremos conectadas las tres máquinas virtuales (*alice, bob* y *mallet*) y, además, habrá una cuarta que será la puerta de enlace para que se comuniquen con otras máquinas que tengan acceso a Internet.

3. Diagrama de red detallado

En la figura 7 se muestra un diagrama detallado de la red NAT 10.0.2.0/24 de nombre GSI que he creado y configurado específicamente para este entorno virtual. Para cada máquina se proporciona su dirección IPv4, su máscara de red y su dirección MAC, respectivamente de arriba abajo.

A mayores se muestra la dirección de la red en la que se encuentran las máquinas y la puerta de enlace de cada una de ellas. Como se puede ver, esta dirección es la misma ya que cuando una máquina se quiere comunicar con el exterior todos los paquetes deben de pasar obligatoriamente por el mismo router.

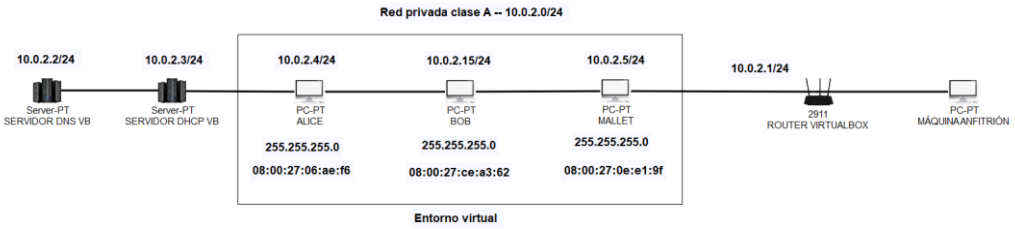

Figura 7. Diagrama de red detallado

4. Comprobaciones de que las máquinas se comunican entre sí a nivel de red

Lo primero que hice fue acceder a la máquina de *alice* con el usuario y contraseña de su mismo nombre. A continuación, le cambié la contraseña al usuario *root* para poder acceder a todos los privilegios de administrador.

Una vez que tenía privilegios, modifiqué el fichero de configuración de la interfaz de red para que esta máquina sea un cliente DHCP. Como quiero que cada vez que se encienda la máquina la interfaz de red de *alice* esté activa, pongo *auto eth3*. Acto seguido, con el comando *iface eth3 inet dhcp* le estoy diciendo que la interfaz de red *eth3* va a ser una

interfaz que va a permitir direcciones IP con versión cuatro y que además, al ser un cliente DHCP, va a tener asignada una dirección IP dinámica.

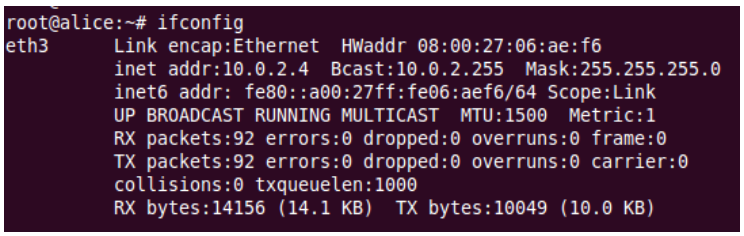

Figura 8. Comando para saber la interfaz de red

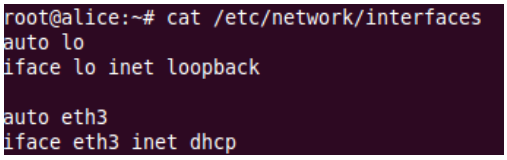

Figura 9. Contenido del fichero de configuración de la interfaz de red de alice

Para ver si realmente estos cambios han funcionado, hay que reiniciar los servicios de red, o lo que es lo mismo, pararlos y volverlos a activar. Esto lo hago con el comando */etc/init.d/networking restart*. Posteriormente, hago un *ping* a esa misma máquina para comprobar que *alice* se puede comunicar.

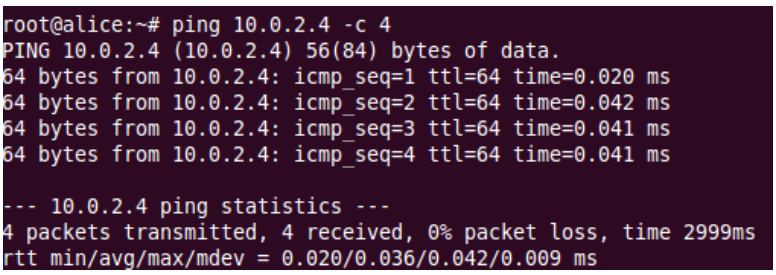

Figura 10. Comprobación de comunicación de la máquina alice

En este momento ya tengo configurada correctamente la máquina de *alice*. Voy a hacer un procedimiento similar con la máquina de *bob*.

De la misma manera que procedí con la máquina de *alice,* probé a entrar en la máquina con el nombre de *bob,* tanto en el usuario como en la contraseña. Una vez dentro, cambié la contraseña de la cuenta *root.*

En el momento en el que tengo privilegios, cambio el contenido del fichero de configuración de la interfaz de red, de la misma forma que hice con *alice*. Sin embargo, esta vez la interfaz de red es *eth0*.

Cuando ya lo tengo modificado voy a habilitar la tarjeta de red *eth0* en la máquina de *bob,* con el comando *ifup eth0*. Cuando se ejecuta el comando, lo que pasa por debajo y que nosotros no vemos, es que se va a hacer una petición al servicio DHCP y va a haber

alguien dentro de la red, en nuestro caso sería el 10.0.2.3 que es un servidor DHCP dentro de VirtualBox, que nos va a asignar dinámicamente la dirección IP 10.0.2.15 como se ve en la imagen.

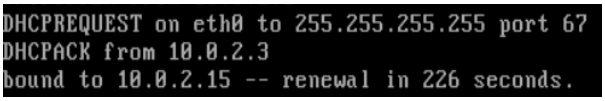

Figura 11. Salida por pantalla del comando ifup eth0

Finalmente, para comprobar que he configurado *bob* de la manera correcta, hago un *ping* a la máquina de *alice* para comprobar si se pueden comunicar entre sí.

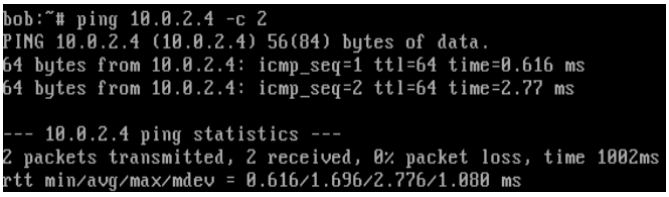

Figura 12. Ping de confirmación con dos peticiones de bob a alice

En este punto ya tengo configuradas de forma precisa con los requisitos que nos pedían las máquinas de *bob* y de *alice*, por lo que solo me falta la de *mallet*.

Lo primero que hice nada más entrar en su máquina fue comprobar si esta tenía acceso a Internet. Para ello, hice un *ping* a la dirección web de Google, pudiendo comprobar que se había traducido el nombre totalmente cualificado a su dirección IP pública. La conclusión es que la máquina de *mallet* tiene acceso a Internet.

Llegados a este punto, realizo los mismos cambios en el fichero de configuración de la interfaz de red para esta máquina. Antes de eso, ejecuto el comando *ip a* para ver cual es el alias de dicha interfaz.

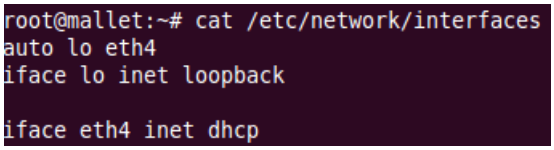

Figura 13. Contenido del fichero de configuración de la interfaz de red de mallet

Para poder comprobar que dichos cambios han funcionado, deshabilito y vuelvo a activar los servicios de red.

En vista de la situación actual, como están todas las máquinas configuradas correctamente, vamos a hacer la prueba final.

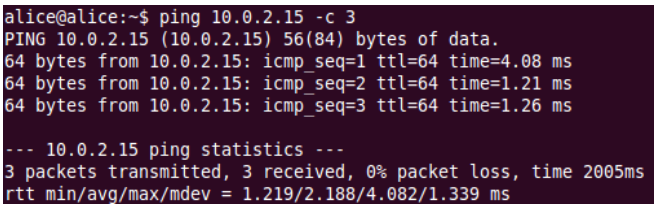

Figura 14. Ping de la máquina de alice a la de bob

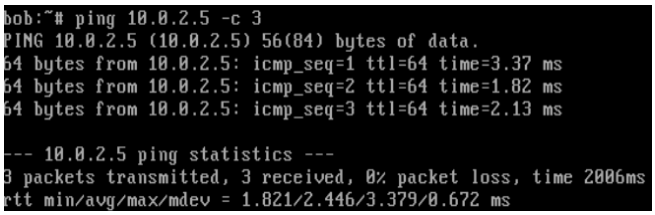

Figura 15. Ping de la máquina de bob a la de mallet

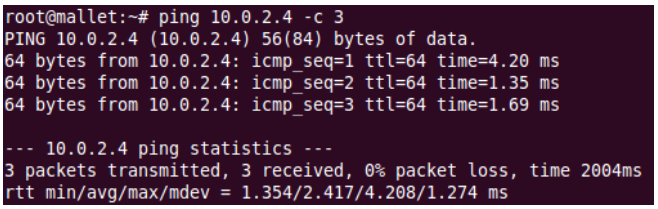

Figura 16. Ping de la máquina de mallet a la de alice

5. Inferir el sistema operativo de una máquina a través del valor del TTL

Cuando alguien hace un *ping* a otra máquina, se utiliza el protocolo ICMP que es el protocolo de control de mensajes en Internet. Este comando nos ofrece información muy interesante de la respuesta a nuestra petición, en caso de que el host remoto esté activo.

Primero hay que entender que significa el valor del TTL. Este número determina el tiempo que los datos son válidos y están disponibles en una red antes de que el router los elimine. En otras palabras, representa el número de saltos que ha dado el paquete por Internet hasta llegar a la máquina destino. En cada uno de esos saltos, el valor del TTL se decrementa en una unidad. Si en algún momento llega a cero, entonces el paquete se descarta y el host devuelve un error a la máquina que hizo la petición.

De esta manera, la mayoría de los sistemas operativos conocidos tienen un valor del TTL predeterminado. Este puede variar según la versión utilizada. Por ejemplo, en Linux sería 64 y en Windows 10 sería 128.

Finalmente, sí que se puede inferir el sistema operativo y su versión a través del valor del TTL cuando realizamos un *ping*. Sin embargo, esta técnica no es muy fiable porque los usuarios pueden modificar este valor para intentar engañar a los ciberdelincuentes.

6. Descubrimiento de los *host* en un segmento de red con la herramienta *nmap*

Para poder utilizar la herramienta *nmap* desde la máquina virtual de *mallet*, lo primero que hago es ver si la aplicación está instalada o si por el contrario la tengo que descargar.

Después de tener las tres máquinas configuradas en una red NAT creada especialmente para este entorno de trabajo, es recomendable hacer un descubrimiento de máquinas que estén conectadas a mí mismo segmento de red para tenerlas controladas.

El protocolo de red que se puede utilizar para comunicarse con máquinas y descubrirlas es ARP. Para ello necesito comprobar que realmente estoy utilizando dicho protocolo cuando hago uso de la herramienta *nmap*. Esto se puede comprobar monitorizando el tráfico de red que va a generar la máquina de *mallet* cuando se ejecute el comando *nmap*.

En este caso, he utilizado el analizador de tráfico *tcpdump* ya que estaba instalado previamente en la máquina. Para poder analizar el tráfico de red es necesario poner la tarjeta de red en modo promiscuo. Esto solo lo podríamos hacer si somos administradores.

Lo primero es saber el alias de dicha tarjeta de red con el comando *ifconfig.* El siguiente paso es decirle a *tcpdump* que ponga en modo promiscuo la tarjeta de red *eth4* para poder observar el tráfico de red que la máquina de *mallet* genere o reciba. Además, es recomendable activar la bandera para que lo haga en modo verbose.

```
mallet@mallet:~$ sudo tcpdump -i eth4 -v
tcpdump: listening on eth4, link-type EN10MB (Ethernet), capture size 96 bytes
```

Figura 17. Monitorización de la tarjeta de red eth4

Una vez que estoy preparada para escuchar, voy a ejecutar *nmap* con la bandera *-n* para no hacer nunca la resolución DNS y con *-sA* para decirle que utilice el protocolo ARP.

```
mallet@mallet:~$ sudo nmap 10.0.2.0/24 -n -sA
[sudo] password for mallet:

Starting Nmap 5.00 ( http://nmap.org ) at 2022-09-20 13:24 CEST
All 1000 scanned ports on 10.0.2.1 are unfiltered
MAC Address: 52:54:00:12:35:00 (QEMU Virtual NIC)

All 1000 scanned ports on 10.0.2.2 are unfiltered
MAC Address: 52:54:00:12:35:00 (QEMU Virtual NIC)

All 1000 scanned ports on 10.0.2.3 are filtered
MAC Address: 08:00:27:17:6D:2D (Cadmus Computer Systems)

All 1000 scanned ports on 10.0.2.4 are unfiltered
MAC Address: 08:00:27:06:AE:F6 (Cadmus Computer Systems)

All 1000 scanned ports on 10.0.2.5 are unfiltered

All 1000 scanned ports on 10.0.2.15 are unfiltered
MAC Address: 08:00:27:CE:A3:62 (Cadmus Computer Systems)

Nmap done: 256 IP addresses (6 hosts up) scanned in 3.61 seconds
```

Figura 18. Escaneo con la herramienta nmap y protocolo ARP

Como se puede ver en la imagen, la aplicación ha descubierto que existen 6 máquinas activas en nuestro segmento de red. Por la documentación aportada anteriormente se que las tres últimas máquinas son *alice*, *mallet* y *bob* respectivamente.

Gracias a *nmap*, he descubierto que hay una máquina con dirección IP 10.0.2.2 que no la tenía identificada. Esa máquina en VirtualBox es el servidor DNS.

Para demostrar que de verdad estamos escaneando con ARP me fijo en la salida de *tcpdump*.

```
mallet@mallet:~$ sudo tcpdump -i eth4 -v
[sudo] password for mallet:
tcpdump: listening on eth4, link-type EN10MB (Ethernet), capture size 96 bytes
13:24:52.640983 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 10.0.2.1 (Broadcast) te
allet.local, length 28
13:24:52.641312 ARP, Ethernet (len 6), IPv4 (len 4), Reply 10.0.2.1 is-at 52:54:00:12:35:00
 Unknown), length 46
```

Figura 19. Monitorización con el uso del protocolo ARP

Finalmente, voy a hacer otro escaneo de mi segmento de red, pero esta vez usando el protocolo ICMP para comprobar si de verdad el protocolo ARP es el más adecuado para esta función.

```
mallet@mallet:~$ sudo nmap 10.0.2.0/24 -n -sP

Starting Nmap 5.00 ( http://nmap.org ) at 2022-09-20 13:50 CEST
Host 10.0.2.1 is up (0.00028s latency).
MAC Address: 52:54:00:12:35:00 (QEMU Virtual NIC)
Host 10.0.2.2 is up (0.00026s latency).
MAC Address: 52:54:00:12:35:00 (QEMU Virtual NIC)
Host 10.0.2.3 is up (0.00026s latency).
MAC Address: 08:00:27:17:6D:2D (Cadmus Computer Systems)
Host 10.0.2.4 is up (0.00066s latency).
MAC Address: 08:00:27:06:AE:F6 (Cadmus Computer Systems)
Host 10.0.2.5 is up.
Host 10.0.2.15 is up (0.00052s latency).
MAC Address: 08:00:27:CE:A3:62 (Cadmus Computer Systems)
Nmap done: 256 IP addresses (6 hosts up) scanned in 2.23 seconds
```

Figura 20.Escaneo con la herramienta nmap y protocolo ICMP

```
mallet@mallet:~$ sudo tcpdump -i eth4 -v
tcpdump: listening on eth4, link-type EN10MB (Ethernet), capture size 96 bytes
13:50:22.736060 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 10.0.2.1 tell alice.loc
length 46
13:50:22.736084 ARP, Ethernet (len 6), IPv4 (len 4), Reply 10.0.2.1 is-at 52:54:00:12:35:00
 Unknown), length 46
```

Figura 21. Monitorización con el uso del protocolo ICMP

En conclusión, puedo decir que el protocolo más adecuado para hacer un descubrimiento de los hosts que están en el mismo segmento de red sería ARP.

El problema que existe si usara el protocolo ICMP para descubrir otros activos en la red, es que el sistema operativo Windows por defecto las solicitudes de tipo *ping* las tienen capadas, pero las de ARP no. Sin embargo, ARP también tiene sus limitaciones ya que únicamente sirve para descubrir máquinas que estén en mí mismo segmento de red.

7. Problemas que aparecieron y las soluciones

El primer problema que me encontré fue que nos dieron tres máquinas virtuales, pero no sabíamos ni el usuario ni la contraseña de ninguna de ellas. Como mucha gente hoy en día no tiene casi seguridad en sus contraseñas, probé a introducir tanto en el usuario como en la contraseña el mismo nombre que tenía asignado la máquina.

El siguiente contratiempo que apareció fue tanto no saber cuál era el nombre del fichero de configuración de la interfaz de red como el comando necesario para reiniciar los servicios de red.

Cuando entré en la máquina de *bob* vi que no tenía permisos de administrador, por lo que procedí a cambiar la contraseña de la cuenta de *root* con el comando *sudo passwd root*. Al estar trabajando con una versión de Debian un poco vieja no encontraba el comando *sudo.* Para solucionarlo entré a la cuenta de *root* gracias al comando *su,* con contraseña *bob.*

En el momento de configurar las tres máquinas para que se puedan comunicar entre sí, me encontré con el problema de que la máquina de *bob* no se encontraba en la misma red que la de *alice* impidiendo así que ambas se pudieran comunicar. Esto lo sé porque *alice* está en la red 10.0.2.0/24, mientras que *bob* se encontraba en la red 192.168.1.0/24. Para solucionarlo tuve que configurar *bob* para que estuviera en la misma red, mirando si el adaptador de red estaba conectado a la red NAT "GSI" que he creado anteriormente. Una vez confirmado que estaba enchufado correctamente, miré el contenido del fichero de configuración de la interfaz de red por si habían asignado una dirección IP estática.