Servicios en Red - Práctica 1: SSH

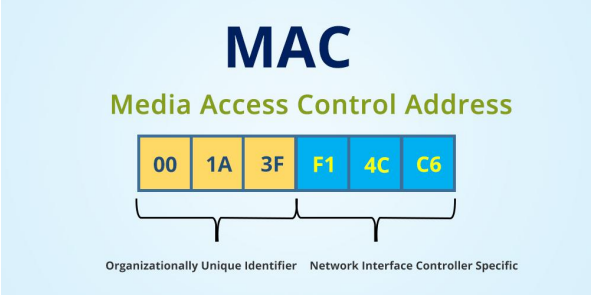
Aclaraciones

Diferencia entre un Adaptador de Red y una Interfaz de Red

Vereis que dentro de la configuración de red de VirtualBox, se hace referencia a adaptadores de red. Mientras que en la configuración de la máquina virtual, se hace referencia a interfaces de red. La diferencia principal es que el adaptador de red es el componente físico (o virtual) que permite la conexión a la red, mientras que la interfaz de red es la representación lógica de esa conexión dentro del sistema operativo.

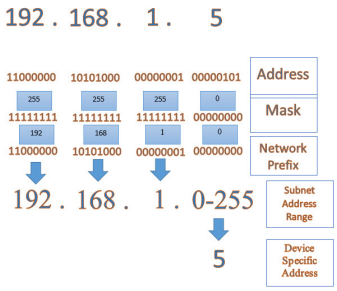
Direcciones MAC

Las direcciones MAC (Media Access Control) son identificadores únicos asignados a los adaptadores de red para su identificación en la red. Cada adaptador de red tiene una dirección MAC única que se utiliza para la comunicación en la capa de enlace de datos del modelo OSI. En VirtualBox, puedes ver y configurar las direcciones MAC de los adaptadores de red de tus máquinas virtuales. En un entorno real, las direcciones MAC son siempre únicas y se asignan de forma permanente a cada adaptador de red. Sin embargo al utilizar virtualización, a causa de clonaciones y copias de configuración, es posible que se generen direcciones MAC duplicadas. Es importante evitar y corregir esto ya que puede causar problemas de conectividad en la red al interferir con la identificación de los dispositivos.



Netmask

La máscara de red (netmask) es un número que se utiliza para dividir una dirección IP en dos partes: la parte de la red y la parte del host. Esto permite identificar qué parte de la dirección IP corresponde a la red y qué parte corresponde a los dispositivos dentro de esa red. La máscara de red se puede representar de dos formas: en notación decimal punteada (por ejemplo, 255.255.255.0) o en notación CIDR (por ejemplo, /24).



La Interfaz Loopback

La interfaz loopback, abreviada como lo, es una interfaz de red virtual que comunica a un dispositivo consigo mismo. Su dirección IP estándar es 127.0.0.1 y esta presente en todos los sistemas operativos.

Introducción

En esta práctica vamos a configurar y utilizar un servidor SSH (Secure Shell) para acceder de forma remota y segura a otros equipos. SSH es un protocolo de red que permite la **comunicación segura** entre dos dispositivos **a través de una red no segura**. Utiliza técnicas de **cifrado** para proteger la información transmitida y garantizar la autenticidad de los usuarios. SSH es ampliamente utilizado por administradores de sistemas y desarrolladores para gestionar servidores de forma remota y viene incluido en la mayoría de las distribuciones de Linux y Windows.

Preparación de las Máquinas

Para esta práctica, necesitaremos al menos dos máquinas: una actuara como cliente SSH y otra como servidor SSH. De ahora en adelante, me referire como **debian1** a la primera y **debian2** a la segunda. Mi recomendación es mantener la maquina debian instalada en clase como respaldo y hacer dos clones de esta para evitar problemas de configuración. A la hora de clonar una maquina virtual en VirtualBox es importante tener en cuenta 3 cosas; **seleccionar un nombre representativo** para facilitar la identificación de las máquinas, seleccionar el **tipo de clonación enlazada** para ahorrar espacio, y marcar la opción de "**Generar MAC aleatoria**" para evitar conflictos de red.

Creación de Nuestro Usuario

En ambas máquinas, crearemos un nuevo usuario que utilizaremos para las conexiones SSH. Ejecuta los siguientes comandos en `debian1` y `debian2`:

```
sudo adduser <usuario>
```

Donde `<usuario>` es el nombre del usuario que deseas crear, para esta práctica **debera ser vuestro nombre**. Despues de ejecutar el comando os pedira que introduzcais una contraseña para el nuevo usuario. Para estas prácticas recomiendo utilizar el propio nombre del usuario como contraseña.

Para poder utilizar el comando `sudo` con este nuevo usuario, debemos añadirlo al grupo `sudo`. Ejecuta el siguiente comando en ambas máquinas:

```
sudo usermod -aG sudo <usuario>
```

Para que el nuevo usuario pueda utilizar el comando `sudo`, es necesario cerrar la sesión y volver a iniciarla. Esto actualizará los grupos a los que pertenece el usuario. Podemos comprobarlo ejecutando el siguiente comando:

```
groups <usuario>
```

Deberías ver `sudo` en la lista de grupos a los que pertenece el usuario.

Nombrado de las Maquinas

Es recomendable asignar nombres representativos a las máquinas virtuales para facilitar su identificación. En este caso, utilizaremos los siguientes nombres:

- `debian1`: Esta será la máquina cliente SSH.
- `debian2`: Esta será la máquina servidor SSH.

Para cambiar el nombre de una maquina en Debian, edita el archivo `/etc/hostname` y reemplaza el contenido con el nuevo nombre. Luego, edita el archivo `/etc/hosts` para reflejar el cambio. Asegúrate de que la línea que contiene `127.0.1.1` apunte al nuevo nombre de la máquina.

Si cerramos la terminal y la volvemos a abrir, deberíamos ver el nuevo nombre de la máquina en el prompt.

Creación de las Interfaces de Red (Virtual Box)

Por defecto las maquinas virtuales de VirtualBox tienen configurado unicamente un adaptador de red el cual las conecta con la maquina fisica y darle acceso a internet a traves de esta. Para permitir la comunicacion entre las dos maquinas virtuales, es necesario crear y configurar un segundo adaptador de red en cada una de ellas. Esto seria equivalente a conectar ambas maquinas a un switch o con un cable directamente.

La creación de este nuevo adaptador sera identica en ambas maquinas:

1. Abre VirtualBox y selecciona la máquina virtual que deseas configurar.
2. Haz clic en "Configuración" y luego en "Red".
3. Selecciona el **adaptador de red 2** y activa la opción "**Conectado a: Red interna**".
4. Asigna un nombre a la red interna, por ejemplo "**inet**" (viene de internal network).
5. Repite estos pasos **en ambas máquinas virtuales**.

VirtualBox conecta entre si todas las máquinas virtuales que están en la misma red interna (mismo nombre de red), permitiendo la comunicación entre ellas. Esto significa que podrás hacer ping entre las máquinas y establecer conexiones SSH sin problemas una vez este todo configurado.

Es importante verificar que las direcciones MAC de las máquinas sean diferentes para evitar conflictos en la red, ya que esto causara problemas de conectividad. Al crear un nuevo adaptador de red, VirtualBox generará automáticamente una nueva dirección MAC para el adaptador. Asi que solo suele generar problemas si se nos olvida marcar la opción correcta al clonar.

Configuración de las Interfaces de Red

Ahora hemos creado un nuevo adaptador de red en cada maquina, pero aún no hemos configurado su interfaz. Para ello, seguiremos los siguientes pasos en ambas máquinas:

1. Inicia la máquina virtual y abre una terminal.
2. Ejecuta el siguiente comando para identificar las interfaces de red disponibles:

   ```
   ip a
   ```

   Deberías ver 3 interfaces de red, una interfaz loopback (`lo`), la interfaz del adaptador NAT y la nueva interfaz del adaptador de red interna. Veremos que la interfaz nueva esta **sin configurar**, por ejemplo no tiene dirección IP.

3. Configura la nueva interfaz del adaptador de red interna en ambas máquinas. Para ello, edita el archivo de configuración de red:

   ```
   sudo nano /etc/network/interfaces
   ```

   Agrega las siguientes líneas al final del archivo en `debian1`:

   ```
   auto <nueva_interfaz>
   iface <nueva_interfaz> inet static
       address 192.168.1.1
       netmask 255.255.255.0
   ```

Y en `debian2`, agrega:

```
auto <nueva_interfaz>
iface <nueva_interfaz> inet static
    address 192.168.1.2
    netmask 255.255.255.0
```

Donde `<nueva_interfaz>` es el nombre de la nueva interfaz de red que se ha creado para el nuevo adaptador (por ejemplo, `eth1`, `enp0s8`, `ens33`...).

4. Guarda los cambios y reinicia el servicio de red:

   ```
   sudo systemctl restart networking
   ```

   Esto aplicará la nueva configuración de red.

5. Verifica que la nueva interfaz de red esté configurada correctamente ejecutando:

   ```
   ip a
   ```

   Verás que la nueva interfaz tiene la dirección IP y la máscara de red que configuraste.

Nota: asegúrate de identificar correctamente qué interfaz corresponde al **adaptador 2** (el de red interna). Normalmente será `enp0s8`, pero puede variar según la versión de Debian o el tipo de adaptador usado.

Prueba de Conectividad

Para probar la conectividad entre las dos máquinas virtuales, puedes utilizar el comando `ping`. Desde `debian1`, ejecuta:

```
ping 192.168.1.2
```

Esto enviará paquetes de ping a `debian2` y deberías ver respuestas si la conexión está funcionando correctamente. Desde `debian2`, puedes hacer lo mismo para probar la conectividad con `debian1`:

```
ping 192.168.1.1
```

Resolución de Nombres

Para que las máquinas puedan resolver sus nombres entre sí, es recomendable editar el archivo `/etc/hosts` en **ambas máquinas** y agregar las siguientes líneas:

```
192.168.1.1 debian1
192.168.1.2 debian2
```

Esto hara que cuando escribamos `debian1` o `debian2` en la terminal, el sistema cambie estos nombres por la ip correspondiente especificada en el archivo `/etc/hosts`.

Podemos comprobarlo intentando hacer ping a las máquinas por su nombre:

```
ping debian1
```

```
ping debian2
```

Setup de SSH

En debian1

El cliente ssh ya viene instalado en la mayoría de las distribuciones de Linux, incluido Debian. Podemos verificarlo utilizando el comando

```
which ssh
```

`which` es un comando que se utiliza para localizar la ruta de un ejecutable en el sistema. Si el cliente ssh está instalado, este comando mostrará la ruta del ejecutable ssh. Si no está instalado, no mostrará ninguna salida.

En debian2

Por otro lado, el servidor SSH (sshd) es menos común en las instalaciones por defecto. En nuestro caso no viene instalado. Para instalarlo, podemos utilizar los siguientes comandos:

```
sudo apt update
sudo apt upgrade
sudo apt install openssh-server
```

Recuerdo que `apt` es el gestor de paquetes utilizado en Debian y sus derivados para instalar, actualizar y gestionar software. El comando `update` actualiza la base de datos de paquetes, `upgrade` actualiza los paquetes instalados a sus últimas versiones, e `install` se utiliza para instalar nuevos paquetes, en este caso `openssh-server`, que es el paquete que contiene el servidor SSH.

Podemos comprobar que el servidor SSH está instalado y funcionando correctamente utilizando el siguiente comando:

```
sudo systemctl status ssh
```

Este comando muestra el estado del servicio SSH (Servidor ssh o sshd), indicando si está activo y funcionando correctamente. Por defecto en muchos casos viene detenido y deshabilitado. En tal caso podemos habilitarlo (hacer que se inicie automaticamente con la maquina) e iniciarlo con:

```
sudo systemctl enable ssh
sudo systemctl start ssh
```

Despues podemos volver a comprobar que el servicio SSH está habilitado y ejecutandose con:

```
sudo systemctl status ssh
```

Ahora podemos proceder a la configuración del servidor SSH.

Probar conexión SSH

Para probar la conexión SSH, desde `debian1`, intenta conectarte a `debian2` utilizando el siguiente comando:

```
ssh usuario@debian2
```

Reemplaza `usuario` con el nombre de usuario que deseas utilizar en `debian2`. Si todo está configurado correctamente, deberías poder conectarte sin problemas.

Configuración del Servidor SSH

Por razones de seguridad es habitual realizar algunas configuraciones adicionales en el servidor SSH. A continuación se presentan algunas de las configuraciones más comunes:

- Cambiar el puerto por defecto (22) a otro puerto no estándar.
- Deshabilitar el acceso de root por SSH.
- Deshabilitar la autenticación por contraseña y utilizar claves asimétricas.

Toda la configuración del servidor SSH se encuentra en el archivo `/etc/ssh/sshd_config`. Si no se cambia una configuración, el valor predeterminado se aplicará. En el archivo de configuración

por defecto se encuentran comentarios aplicando el valor predeterminado a cada configuración. Para cambiar una configuración, simplemente descomenta la línea correspondiente y modifica el valor según sea necesario.

En debian2

Para cambiar el puerto por defecto, edita el archivo de configuración del servidor SSH:

`sudo nano /etc/ssh/sshd_config`

Busca la línea que dice `#Port 22` y cámbiala por `Port <nuevo_puerto>`, donde `<nuevo_puerto>` es el puerto que deseas utilizar (por ejemplo, 2222). Luego, guarda y cierra el archivo.

Para deshabilitar el acceso de root por SSH, busca la línea que dice `PermitRootLogin yes` y cámbiala por `PermitRootLogin no`.

Para que se aplique la nueva configuración, reinicia el servicio SSH:

`sudo systemctl restart ssh`

Aun no vamos a deshabilitar la autenticación por contraseña para utilizar claves asimétricas ya que necesitamos asegurarnos de que las claves públicas estén configuradas correctamente en el cliente antes de hacer este cambio o nos quedaríamos sin acceso.

En debian1

Para que nuestro cliente ssh se conecte a un servidor ssh que este escuchando en un puerto diferente al por defecto (22), debemos especificar el puerto al utilizar el comando ssh. Por ejemplo, si hemos cambiado el puerto a 2222, el comando sería:

`ssh -p 2222 <usuario>@ip_del_servidor`

Con el objetivo de iniciar sesión en el servidor SSH utilizando un sistema de claves asimétricas, primero debemos generar un par de claves en `debian1` (el cliente SSH) y luego copiar la clave pública a `debian2` (el servidor SSH). Es posible hacerlo "manualmente", pero ssh cuenta con el comando `ssh-copy-id` que facilita este proceso.

Primero generamos el par de claves en `debian1`:

`ssh-keygen -t ed25519`

El comando nos solicitara un passphrase (frase de paso) para proteger la clave privada. Esto es opcional, en este caso pulsaremos dos veces para dejarlo vacío.

Esto generará un par de claves en `~/.ssh/id_ed25519` (clave privada) y `~/.ssh/id_ed25519.pub` (clave pública). Podemos inspeccionar el contenido de estos archivos con

```
cat ~/.ssh/id_ed25519.pub
cat ~/.ssh/id_ed25519
```

Luego, copiamos la clave pública al servidor `debian2`:

`ssh-copy-id -p <puerto> -i ~/.ssh/id_ed25519.pub usuario@debian2`

Siendo `usuario` el nombre de usuario en `debian2` correspondiente a la cuenta a la que deseas acceder.

De esta manera, podrás autenticarte en el servidor `debian2` sin necesidad de ingresar una contraseña.

`ssh -p <puerto> usuario@debian2`

En debian2

Por ultimo bloquearemos el acceso por contraseña en el servidor `debian2`. Para ello, editaremos el archivo de configuración del servidor SSH:

```
sudo nano /etc/ssh/sshd_config
```

Buscaremos la línea que dice `#PasswordAuthentication yes` y la cambiaremos por `PasswordAuthentication no`. Luego, guardaremos y cerraremos el archivo.

Para que se apliquen los cambios, reiniciaremos el servicio SSH:

```
sudo systemctl restart ssh
```

Entrega

Para esta práctica, solo sera necesario que me entregueis una serie de capturas de pantalla:

1. Salida del comando `ping debian1` desde `debian2`.
2. Salida del comando `ping debian2` desde `debian1`.
3. Conexion exitosa ssh a `debian2` desde `debian1`.
4. Contenido del archivo de clave pública `~/.ssh/id_ed25519.pub` en `debian1`.
5. Contenido del archivo de clave privada `~/.ssh/id_ed25519` en `debian1`.
6. Contenido del archivo `~/.ssh/authorized_keys` en `debian2`.

Todo esto lo debereis hacer utilizando el nuevo usuario creado con vuestro nombre.