SSH

Tarea 1: Iniciar una conexión de SSH a un servidor

1. Abre una terminal en tu máquina virtual.
2. Utiliza el comando **ssh** seguido del nombre de usuario y la dirección IP o nombre de dominio del servidor al que deseas conectarte. Por ejemplo:

   #ssh usuario@maquina04
3. Si es la primera vez que te conectas a este servidor, es posible que recibas un aviso de seguridad indicando que la autenticidad del servidor no se puede verificar. Esto sucede porque no tienes la clave pública del servidor en tu lista de **known_hosts**.

   The authenticity of host 'maquina-53-04.ars.lab.te.upm.es (10.49.12.84)' can't be established. ED25519 key fingerprint is SHA256:DrizbnXDJI53CyvtXz6FZ/KsUQ2UE/QNihLXoXYlxz8. This key is not known by any other names. Are you sure you want to continue connecting (yes/no/[fingerprint])?
4. El aviso te pedirá confirmar si deseas continuar la conexión. Si confías en el servidor, puedes escribir **yes** y presionar Enter para continuar.
5. Luego, se te pedirá ingresar la contraseña del usuario en el servidor. Ingrésala y presiona Enter.
6. Una vez que ingreses la contraseña correctamente, la conexión SSH se establecerá y estarás conectado al servidor.

   Warning: Permanently added 'maquina-53-04.ars.lab.te.upm.es' (ED25519) to the list of known hosts.

Tarea 2: Verificación automática de la clave del servidor

1. Si has aceptado la clave pública del servidor en la tarea anterior, al realizar una nueva conexión al mismo servidor, ya no recibirás el aviso de seguridad sobre la clave pública.
2. Esto se debe a que la clave pública del servidor se ha guardado en tu archivo **~/.ssh/known_hosts**.
3. El cliente SSH verificará automáticamente la clave pública del servidor con la que está almacenada en tu archivo **known_hosts** y establecerá la conexión si coincide.

Tarea 3: Intercambiar las claves de algunos servidores

1. **Abre el archivo ~/.ssh/known_hosts en un editor de texto:**
    - Puedes utilizar el editor de tu preferencia, como **nano**, **vi**, **gedit**, etc. Por ejemplo:

#nano ~/.ssh/known_hosts

    - Este archivo contiene las claves públicas de los servidores a los que te has conectado anteriormente.
2. **Localiza las líneas correspondientes a los servidores cuyas claves deseas intercambiar:**
    - Cada línea en este archivo representa la clave pública de un servidor.
    - Las líneas suelen tener el formato **[nombre_del_servidor] clave_publica**.
3. **Cambia las claves entre sí:**
    - Copia la línea que contiene la clave pública de un servidor y pégalala en lugar de la línea que contiene la clave del otro servidor.
    - Asegúrate de mantener el formato correcto y no cambiar ningún otro detalle del archivo.

ssh usuario@direccion_ip

4. **Observa el mensaje de advertencia:**
    - Al intentar conectarte al servidor, es probable que recibas un mensaje de advertencia que indica un posible problema de seguridad.
    - Esto sucede porque la clave pública del servidor no coincide con la que está almacenada en tu archivo **known_hosts**.
5. **Comprende posibles situaciones legítimas para este tipo de advertencia:**
    - A veces, las claves públicas de los servidores pueden cambiar por razones legítimas, como una actualización de seguridad o una reconstrucción del servidor.

Tarea 4: Generar una pareja de claves en la cuenta ana@alfa

1. Abre una terminal en la máquina de **ana@alfa**.
2. Ejecuta el siguiente comando para generar la pareja de claves:

bashCopy code

ssh-keygen

3. Sigue las instrucciones en pantalla y presiona Enter para aceptar los valores predeterminados.
4. Verifica que la pareja de claves se haya guardado en el directorio **~/.ssh** de **ana**.

ls ~/.ssh

Deberías ver dos archivos: uno con extensión **.pub** (la clave pública) y otro sin extensión (la clave privada).

Tarea 5: Añadir la clave pública de ana@alfa al archivo ~/.ssh/authorized_keys de bea@beta

1. Accede a la máquina **beta**.
2. Si no existe, crea el directorio **~/.ssh** en la cuenta de **bea**.

   mkdir -p ~/.ssh
3. Copia la clave pública de **ana@alfa** al archivo **~/.ssh/authorized_keys** de **bea**.

   desde la maquina con cat o nano.

Tarea 6: Abrir una sesión de SSH desde ana@alfa a bea@beta

1. ssh bea@beta
2. La sesión debería abrirse sin pedir la contraseña de **bea** ya que **ana** ha sido autorizada con su clave pública.

- **Ventajas**:
    - Elimina la necesidad de recordar contraseñas complejas.
    - Mejora la seguridad al evitar el intercambio de contraseñas en texto plano.
    - Facilita la administración de claves de acceso a través de múltiples sistemas.
- **Inconvenientes**:
    - Requiere un proceso adicional de configuración para autorizar claves públicas.
    - Si la clave privada se ve comprometida, podría permitir el acceso no autorizado a los sistemas.
    - Si se pierde la clave privada y no se cuenta con un respaldo, el acceso podría perderse irreversiblemente.

Esta forma de autenticación con clave pública permite a **ana** autenticarse y autorizarse en múltiples servidores sin necesidad de introducir contraseñas, siempre y cuando haya compartido su clave pública previamente con esos servidores. Además, puede copiar su pareja de claves a múltiples sistemas para usarla como cliente desde cualquiera de ellos.

Tarea 7

1. Abre una terminal en tu sistema local.
2. Utiliza el siguiente comando **scp** para copiar un archivo a la máquina remota:

scp ruta/del/archivo usuario@maquina-remota:ruta/de/destino

Tarea 10

1. Abre una terminal en tu sistema local.
2. Utiliza el siguiente comando SSH para ejecutar un comando remoto en el servidor:

ssh usuario@servidor "comando"

Túneles SSH

a) Abra en una ventana una sesión en el servidor (una jaula del laboratorio) y ponga a nc a escuchar en un puerto cualquiera: nc -lkv 3306

b) Abra en otra ventana una sesión en el cliente (otra jaula del laboratorio) y use telnet o nc para enviar mensajes: telnet/nc servidor 3306.

Debería ver que las líneas que escribe aparecen en la ventana del servidor. El tráfico de la prueba anterior circula en claro

Tarea 12

En el cliente:

1. Abra una conexión SSH al servidor estableciendo un túnel local utilizando el puerto 12345 en el cliente y redirigiéndolo al puerto 3306 en localhost del servidor. Ejecute el siguiente comando:

ssh -L 12345:localhost:3306 -nN ana@servidor &

Este comando establece un túnel SSH local desde el puerto 12345 del cliente al puerto 3306 en localhost del servidor. La opción **-n** evita la apertura de un terminal interactivo y **-N** indica que no se deben ejecutar comandos remotos.

En el cliente:

1. Ahora que el túnel SSH está establecido, puede enviar tráfico a través del túnel. Para hacerlo, ejecute **nc** en el cliente y conéctelo al puerto local que se especificó en el túnel (12345 en este caso). Ejecute el siguiente comando:

nc localhost 12345

Este comando conecta **nc** en el cliente al puerto local 12345, que está redirigido a través del túnel SSH al puerto 3306 en localhost del servidor

Tarea 13: Establecimiento de un túnel SSH y acceso a un servicio en una jaula del laboratorio desde casa:

- **Resumen:**
    - En esta tarea, se configura una sesión SSH desde casa a una jaula en el laboratorio y se establece un túnel SSH para acceder a un servicio **nc** en la jaula del laboratorio.
    - Luego, se verifica la conexión utilizando **telnet** o **nc** desde el ordenador de casa para conectarse al servidor de **nc** en la jaula del laboratorio.
- **Conceptos clave:**
    - Túnel SSH: Permite el acceso seguro a servicios remotos a través de una conexión SSH.
    - **nc** (Netcat): Herramienta de red para leer y escribir datos a través de conexiones de red utilizando el protocolo TCP o UDP.
    - **telnet**: Protocolo de red que permite la comunicación bidireccional entre dispositivos a través de una conexión de terminal.

Tarea 14: Establecimiento de un túnel SSH inverso y acceso a un servicio en casa desde una jaula del laboratorio:

- **Resumen:**
    - En esta tarea, se establece un túnel SSH inverso desde una jaula del laboratorio hacia un servicio **nc** en casa.
    - Luego, se verifica la conexión utilizando **telnet** o **nc** desde la jaula del laboratorio para escribir líneas que deben aparecer en el servicio **nc** de casa.
- **Conceptos clave:**
    - Túnel SSH inverso: Permite el acceso a servicios locales desde un servidor remoto utilizando SSH.
    - **telnet** / **nc**: Herramientas utilizadas para la comunicación de red a través de TCP o UDP.

Agente de SSH (opcional):

- **Resumen:**
    - El agente de SSH es una utilidad que ayuda a gestionar claves privadas cifradas para evitar la necesidad de introducir la contraseña cada vez que se requiere la clave.
    - Se puede utilizar **ssh-agent** para cargar y gestionar claves privadas cifradas, evitando la necesidad de introducir la contraseña cada vez que se utilizan.
- **Conceptos clave:**

- Agente de SSH: Utilidad que gestiona las claves privadas cifradas para su uso en conexiones SSH.
- **ssh-add**: Comando utilizado para agregar claves privadas al agente de SSH.

Tarea 15: Generación y autorización de claves SSH:

- **Resumen:**
  - En esta tarea, se crea un par de claves SSH y se autoriza la clave pública en un sistema remoto para permitir la autenticación sin contraseña.
- **Conceptos clave:**
  - Claves SSH: Permiten la autenticación segura en conexiones SSH.
  - **ssh-keygen**: Utilidad para generar pares de claves SSH.

Tarea 16: Gestión de claves SSH con ssh-agent:

- **Resumen:**
  - En esta tarea, se ejecuta **ssh-agent** para gestionar claves privadas cifradas, permitiendo su uso sin necesidad de introducir la contraseña cada vez.
- **Conceptos clave:**
  - **ssh-agent**: Utilidad que gestiona las claves privadas cifradas para su uso en conexiones SSH.
  - Variables de entorno **SSH_AUTH_SOCK** y **SSH_AGENT_PID**: Utilizadas para localizar el agente de SSH.

Tarea 17: Reenvío de solicitudes de autenticación al agente de SSH:

- **Resumen:**
  - En esta tarea, se habilita el reenvío de solicitudes de autenticación al agente de SSH para permitir la autenticación sin contraseña en sistemas remotos.
- **Conceptos clave:**
  - Reenvío de autenticación: Permite que un sistema remoto utilice claves privadas gestionadas por el agente de SSH en el sistema local.
  - Opción **-o ForwardAgent=yes**: Utilizada para habilitar el reenvío de autenticación en clientes de SSH.