1. OBJETIVO

En esta práctica se estudia el servicio SSH *(Secure Shell)*. Se describe el servidor OpenSSH y las directivas de configuración más comunes, poniendo énfasis en la autentificación con clave pública, el intercambio seguro de archivos y el establecimiento de túneles SSH.

Usaremos la distribución Rocky Linux 9.3, aunque se recomienda hacer alguno de los apartados con Ubuntu Server 22.04.

2. PRÁCTICO

1. Comprobar que el servidor OpenSSH está instalado en el equipo (en caso contrario, hacer la instalación con el paquete openssh-server). Identificar su nombre simbólico para systemd (sshd o ssh) e iniciar o reiniciar el servicio. Dar de alta en el servidor alguna cuenta de usuario (con su correspondiente directorio *home*) y asignar contraseñas. ¿Qué comandos ha usado para estas tareas?
2. Estudiar el archivo de configuración del servidor. Realizar algunas modificaciones sobre la configuración por defecto y comprobar su efecto reiniciando el servidor: cambiar el puerto de escucha, impedir el acceso remoto al usuario root, no realizar resoluciones DNS al inicio, y permitir conexiones sólo a algunos usuarios concretos.
3. Utilizar el cliente SSH desde nuestra máquina anfitriona (ssh y/o putty) para acceder a nuestro servidor OpenSSH. Utilizar cuentas de usuario válidas en el servidor remoto y autentificación por contraseña.
4. Utilizar los comandos scp y sftp (consola Linux/Mac o PowerShell de Windows) y el Explorador de Archivos o el programa WinSCP.exe para intercambiar archivos de forma segura entre el cliente y un servidor SSH.
5. Configurar un túnel local SSH desde nuestro cliente (ssh y/o putty) a un servicio activo del propio servidor (HTTP, POP3, …). A continuación, abrir la conexión SSH y comprobar que podemos acceder al servicio remoto a través del túnel. Probar igualmente a redirigir las peticiones que entran por el túnel hacia otro servidor (por ejemplo, abrir un túnel desde el puerto cliente 2080 a través de nuestro servidor SSH y que alcance otro host (otra máquina virtual) con el servicio web arrancado en su puerto 80).
6. Repetir el apartado anterior permitiendo conexiones a nuestro túnel desde otros equipos (opción -g en SSH o *Connection → SSH → Tunnels → Local ports accept connection from other hosts* en Putty). Usar la máquina anfitriona o una nueva máquina virtual para acceder a través del túnel al servicio ofrecido.
7. Establecer un posible contexto para probar el funcionamiento de un túnel inverso SSH (*reverse*). Llevar a cabo la configuración y verificar el funcionamiento del túnel.
8. Para un usuario de prueba en el servidor SSH configurar su autentificación basada en claves pública/privada. Para generar las claves en el cliente Linux utilizar el comando ssh-keygen. En clientes Windows puede utilizar el programa puttygen.exe (ojo: se usa una clave pública RSA de 2048 bits. La clave pública obtenida en formato OpenSSH aparece en la ventana superior). Comprobar el efecto que tiene no proteger con frase de paso la clave privada. Nota: es posible cambiar la frase de paso de una clave privada con la opción –p de ssh-keygen (`ssh-keygen -p -f id_rsa`)
9. Considere que necesita crear un script para ejecutar algún comando en un equipo remoto, y que la conexión a ese equipo se realiza a través de ssh (tenemos una cuenta de usuario en el servidor). Buscar alguna solución que permita automatizar la ejecución remota sin tener que introducir la contraseña cada vez que se ejecuta el comando. Es posible usar la autentificación basada en clave pública/privada del apartado anterior (sin proteger la clave privada con frase de paso), pero también existen utilidades que nos permiten pasar el usuario y contraseña de autentificación en la llamada a la ejecución remota (buscar por *automate SSH login*).

Nota: los aspectos significativos de la configuración realizada en los apartados 1 a 9 tendrán que reflejarse en la memoria de la práctica.

3. REFERENCIAS

- http://www.openssh.com/