Escuela técnica superior
Facultad de Ingeniería Informática

Práctica 5
SSH

Ignacio Fernández Contreras
4º Informática A

Índice

1. Parte 1 3

2. Parte 2 4

3. Parte 3 5

1. Parte 1

Supongamos un servidor para alojamiento compartido (*hosting*) en el que los usuarios sólo pueden acceder a su directorio *home* y únicamente a través de SFTP. Estos usuarios no van a poder leer ni acceder a ninguna otra parte del sistema de ficheros que no esté bajo su directorio *home*. Se pide restringir el acceso a estos usuarios (al menos dos en un grupo) a nuestro servidor SSH y realizar lo siguiente:

- Crear los usuarios asociados a un grupo en el sistema.
- Definir una estructura de carpetas *home* y conceder los permisos adecuados a los usuarios. Nota: conceder permisos de propietarios con `chown`.
- Comprobar su funcionamiento con cualquier cliente SFTP y su no denegación de acceso con SSH.

Creamos el usuario:

```
user@user:/home$ sudo useradd -m -G sftp_users nacho
user@user:/home$ sudo passwd nacho
New password:
Retype new password:
passwd: password updated successfully
```

Creamos el directorio:

```
user@user:/home$ sudo mkdir /nacho
```

Le damos el permiso correspondiente a cada usuario:

```
user@user:/home$ sudo chown root:root /nacho/
user@user:/home$ sudo chmod 755 /nacho/
user@user:/home$ sudo mkdir /nacho/sftp
user@user:/home$ sudo chown nacho:sftp_users /nacho/sftp
user@user:/home$ sudo chmod 700 /nacho/sftp
```

Modificamos la configuración de */etc/ssh/sshd/config*

```
# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
        Match Group sftp_users
#       ChrootDirectory %h
#       ForceCommand internal-sftp
        AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
```

Nos conectamos por ssh con el nuevo usuario:

```
ssh nacho@localhost -p 2222
```

Al conectarnos por ssh, recibimos error por parte del servidor, ya que lo hemos configurado para que solo se pueda conectar por sftp al directorio asignado, por tanto, vamos a conectarlo ahora por sftp:

```
sftp -P 2222 nacho@localhost
```

Dando como resueltado el siguiente log:

```
user@user:/home$ sudo journalctl -xeu ssh.service | grep "nacho"
nov 09 13:47:54 user sshd[3088]: Accepted password for nacho from 10.0.2.2 port 34584
    ssh2
nov 09 13:47:54 user sshd[3088]: pam_unix(sshd:session): session opened for user nacho
    (uid=1003) by (uid=0)
nov 09 13:47:54 user sshd[3088]: pam_unix(sshd:session): session closed for user nacho
nov 09 13:49:23 user sshd[3182]: Accepted password for nacho from 10.0.2.2 port 49424
    ssh2
nov 09 13:49:23 user sshd[3182]: pam_unix(sshd:session): session opened for user nacho
    (uid=1003) by (uid=0)
nov 09 13:49:23 user sshd[3182]: pam_unix(sshd:session): session closed for user nacho
```

2. Parte 2

Establece un mecanismo de autentificación en SSH a través de claves públicasprivadas, de forma que únicamente se pueda acceder a través de esta forma y no a través de contraseña. Se pide verificar este mecanismo con un usuario y un cliente SSH.

Generamos las claves en el cliente

```
    ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/home/in4p/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/in4p/.ssh/id_rsa
Your public key has been saved in /home/in4p/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:b1iQ6ene2Qf/wIGg3wikG6vS12IeGVaNlhumdrOJV2k in4p@archlinux
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|         *       |
|        % o      |
|       O * o . . |
|      B S E . . .|
|     o X % o . . |
|    . . * . * = .|
|     . . + + . o |
|      . . + . . o|
+----[SHA256]-----+
```

Conectamos el usuario al servidor

```
ssh -p 2222 nuevo_usuario@tu_servidor
```

En el fichero /etc/ssh/sshd$_{c}onfig$

Y reiniciamos ssh

```
sudo service ssh restart
```

Con esto, el usuario ya accede por su clave pública

3. Parte 3

Realiza un túnel Local-Port-Forwarding a uno de los servicios del Servidor Ubuntu (ej., MySQL Server). Muestra los comandos necesarios y cómo se puede acceder a este servicio de forma segura desde el SO a la máquina virtual.

Para generar el túnel:

```
ssh -L 3306:localhost:3306 -p 2222 user@localhost
    -- 3306 es el puerto redirigido
```

Accedemos a mysql a través del puerto redirigido:

```
mysql -u user -p -h localhost -P 3306
```