ADMINISTRACIÓN DE REDES Y SEGURIDAD

TRABAJO PRÁCTICO 4

December 6, 2018

Cátedra:
Lic: Zappellini, Bruno Damian

Integrantes:
Toledo Margalef, Pablo Adrian
UNPSJB - Trelew

SSH-KEYGEN

`ssh-keygen` es una herramienta para generar, administrar y convertir claves utilizadas por la aplicación `ssh`. También sirve para actualizar la lista de revocación de claves.

Al momento de generar claves se puede realizar con y sin contraseña. En ambos casos la generación se hace con:

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pablo/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/pablo/.ssh/id_rsa.
Your public key has been saved in /home/pablo/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Up6KjbnEV4Hgfo75YM393QdQsK3Z0aTNBz0DoirrW+c pablo@pablox
The key's randomart image is:
+---[RSA 2048]----+
|     .       ..oo..|
|    . . .   . .o.X.|
|     . . o.   ..+ B|
|    .   o.o  .+ ..|
|     ..o.S   o..   |
|    . %o=      .   |
|     @.B...      . |
|    o.=. o. . .   .|
|     .oo  E. . ..  |
+----[SHA256]-----+
```

La gran diferencia está en el momento de insertar la contraseña (o passphrase). En caso de no desear ponerle contraseña a las claves generadas se aprieta enter al momento en el que se nos pida una passphrase.

`authorized_keys` y `known_hosts`

El archivo `$HOME/.ssh/authorized_keys` contiene la lista de las claves públicas de los hosts que tienen permitido el ingreso al sistema. Y en el archivo `$HOME/.ssh/known_hosts` contiene la lista de claves de los hosts que se han conectado con el equipo.

SSH COPY ID

Con el comando `ssh_copy_id` podemos copiar a una máquina remota un archivo específico de claves para usar con ese equipo en particular.

En nuestro caso, copiamos el archivo a la máquina remota:

```
$ ssh-copy-id -i ~/.ssh/id_rsa_1_no_pass pablo@192.168.0.102
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/
    pablo/.ssh/id_rsa_1_no_pass.pub"
The authenticity of host '192.168.0.102 (192.168.0.102)' can't be
    established.
ECDSA key fingerprint is SHA256:JKiSYXpD5S8UnhArc/0
    ha9G39v09YRdrCSXcHiorlEk.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s),
    to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you
    are prompted now it is to install the new keys
pablo@192.168.0.102's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'pablo@192.168.0.102'"
and check to make sure that only the key(s) you wanted were added.
```

Y en el equipo remoto deshabilitamos la autenticación por contraseña en el archivo `/etc/ssh/sshd_config` agregando la línea `PasswordAuthentication no`.

Al intentar hacer un ingreso por ssh a la máquina remota, especificando el archivo que se copió anteriormente, no nos pedirá contraseña e ingresará.

TÚNELES SSH

Punto 4 - Ana

Para poder resolver la necesidad de Anda, que es poder redireccionar las peticiones a `localhost:3306` al servidor con la instancia real de MySQL corriendo lo que debe hacer es crear un túnel directo por ssh escribiendo:

```
$ ssh –L 3306:localhost:3306 ana@servidor
```

Lo que hace esta línea es redireccionar las peticiones a `localhost:3306` a `ana@servidor:3306` a través de una conexión ssh, que además va encriptada.

Punto 5 - Matías

Matías desea conectarse por ssh desde su computadora personal al servidor al que tiene acceso. Para lograr eso debe escribir:

```
$ ssh –R 222:localhost:22 mati@personal
```

De esta forma las conexiones al 222 del servidor (mati@peronal) van al 22 del servidor.

Así, desde la máquina personal de Matias se puede hacer `ssh server -p 222` y acceder al servidor.

Punto 6 - Túneles con NAT

Marina desea conectarse al servidor de José que está detrás de un servicio de NAT. Como ella también se encuentra tras un servicio de NAT, deben confluir en un punto accesible por ambos extremos para poder realizar la conexión por túnel.

Esto se resulve de la siguiente forma.

```
# Suponemos que 8080 es el puerto del servicio en el servidor de Jos

# Marina debe ejecutar
ssh –R 5050:localhost:8080 maquina@externa

# Jos  debe ejecutar
```

```
ssh –R 5050:localhost:8080 maquina@externa
```

Así, Marina al conectarse al puerto 8080 de su localhost, ssh estará haciendo forward al 5050 de la máquina remota, que estará haciendo el forward al puerto 8080 en la maquina de José.

SSH - OTRAS CUESTIONES

Punto 7 - X11 Forwarding

Lo que debe hacer Ernetso es:

```
$ ssh –X maquina@ip
```

De esta forma, el socket de X11 en la máquina remota hara el forward al X11 local y la ventana de, por ejemplo, Firefox se dibujará en la máquina local.

Sin embargo, se debe tener en cuenta que debe estar habilitado el forwarding del socket de X11, en el archivo de configuración de ssh.

Punto 8 - Alias

Para configurar alias en `ssh` es necesario modificar (o crear en el caso de que no exista) el archivo `/.ssh/config`. Respetando la sintaxis explicada en páginas man de `ssh_config`. En nuestro caso, tiene la siguiente forma.

```
Host servidor1
    User administrador
    HostName 192.168.1.40
    Port 22022
    PasswordAuthentication no
    IdentityFile ~/.ssh/keys/servidor1
```

Luego, al hacer `ssh servidor1`, se conectará utilizando la configuración especificada.