SSH - *Secure Shell*

SSH

SSH es el nombre de un protocolo y del programa que lo implementa. Se utiliza para acceder a servidores de forma remota y permite administrarlos por medio de un intérprete de comandos.

SSH se utiliza de 2 formas:

1. **Cliente SSH:** es el equipo o dispositivo desde el cual se desea establecer una conexión remota segura.
2. **Servidor SSH:** es el equipo o servidor al que se conecta el cliente SSH.

Además de permitir la administración remota de otros dispositivos, SSH permite copiar datos de forma segura, gestionar claves RSA para no escribir claves al conectar a los dispositivos y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH.

Instalación y configuración

Generalmente, el cliente de SSH ya viene instalado en la mayoría de las distribuciones, por lo que podríamos establecer una conexión hacia servidores remotos inmediatamente al finalizar la instalación del sistema operativo. Sin embargo, SSH como servidor puede no venir instalado, para eso deberemos descargar e instalar el paquete “openssh-server” con el comando:

```
apt-get install openssh-server
```

Los archivos de configuración de OpenSSH se encuentran en /etc/ssh. Este tiene 2 archivos de configuración:

1. **sshd_config:** en este archivo se describe la configuración de SSH como servidor. En él encontraremos algunas líneas importantes a tener en cuenta para su configuración:
    - **port:** especifica el puerto TCP que se utilizará, por default viene comentado y usa el puerto 22, si se descomenta, suele ser para ponerle otro puerto de

uso. Las buenas prácticas recomiendan cambiar este puerto por otro para evitar ataques o intento de ingresos no autorizados.

- **PubkeyAuthentication:** permite la autenticación mediante clave privada.
- **AuthorizedKeysFile:** esta opción indica al servidor en donde están almacenadas las claves públicas de los usuarios. Por default es .ssh/authorized_keys (directorio y archivo que se encuentran en el home del usuario).
- **PasswordAuthentication:** permite la autenticación mediante contraseñas.
- **PermitRootLogin:** por default, y, por seguridad, este valor viene configurado con **"Prohibit-password",** pero para nuestros fines prácticos, este valor vamos a cambiarlo por "**yes"** quedando sin comentario (o sea, debemos sacarle el numeral que está delante #).

```
PermitRootLogin yes
```

2. **ssh_config:** en este archivo se describe la configuración de SSH como cliente.
    - **Host:** su valor puede ser una lista de patrones y determina que las opciones subsiguientes sean aplicadas a las conexiones realizadas a los hosts especificados.
    - **port:** es el puerto de TCP que se utiliza el servidor al que nos vamos a conectar.
    - **PubkeyAuthentication:** autenticación mediante clave pública (se recomienda dejar en "yes").
    - **IdentityFile:** archivo que contiene la clave privada.
    - **PasswordAuthentication:** sí se permite la autenticación mediante contraseña.

Forma de uso

Existen 2 formas de utilizar el comando:

1. **ssh <host>:** autentica en el servidor de destino con el mismo usuario con el que se está logueando en el origen.
2. **ssh <usuario@host>:** autentica en el servidor de destino con el usuario que se especifica a la izquierda del @.

Conexión paso a paso

Conexión inicial

El cliente se conecta al servidor en el puerto 22 (o el puerto configurado para SSH).

Intercambio de claves

El servidor envía su clave pública al cliente para identificarse.

Verificación

El cliente verifica si ya conoce la clave pública del servidor; si no, pide confirmación al usuario.

Establecimiento de una clave de sesión

El cliente genera una clave de sesión, la cifra con la clave pública del servidor, y la envía. Esta clave se usará para cifrar toda la comunicación posterior.

Autenticación

El cliente se autentica (mediante contraseña o clave pública).

Inicio de sesión

Una vez autenticado, la conexión está segura, y el usuario puede interactuar con el servidor.

Métodos de autenticación

SSH soporta diversos métodos de autenticación, entre los que se incluyen:

- Autenticación por contraseña.
- Autenticación por clave pública/privada.
- Autenticación mediante certificados.
- Autenticación con Kerberos.
- Autenticación basada en teclado interactivo o *challenge-response.*
- Autenticación por agente.

Aquí se explicarán los 2 primeros:

Autenticación por contraseña

Este es el método más básico y común. El cliente intenta conectarse al servidor SSH y se le solicita ingresar su **nombre de usuario y contraseña**. El servidor verifica si las credenciales coinciden con las almacenadas previamente en el sistema remoto. Aunque es fácil de usar, es menor seguro que otros métodos porque las contraseñas pueden ser vulnerables a ataques de fuerza bruta o robo.

- El cliente envía el nombre de usuario al servidor.
- El servidor solicita la contraseña.
- El cliente ingresa la contraseña, que se transmite de manera cifrada al servidor.
- El servidor verifica si la contraseña es correcta, y si lo es, concede acceso.

Autenticación por clave pública/privada

Este es el método más seguro y ampliamente recomendado. Aquí el cliente genera un par de claves: **una clave privada** que se guarda en su sistema y **una clave pública** que se copia en el servidor. Al intentar conectarse, el servidor usa la clave pública para cifrar un mensaje, y el cliente debe descifrarlo con su clave privada. Si la descodificación es correcta, la autenticación es exitosa.

- El cliente genera un par de claves (pública y privada).
- La clave pública se almacena en el servidor (en ~/.ssh/authorized_keys).
- Durante la conexión, el servidor cifra un desafío con la clave pública del cliente.
- El cliente utiliza su clave privada para descifrar el desafío.
- Si el cliente descifra correctamente el mensaje, la autenticación se considera válida, y no es necesario el uso de contraseñas.

Con este método, se garantiza que solo quien posee la clave privada puede autenticarse, y no requiere intercambiar o enviar contraseñas.

¿Cómo se generan las claves pública/privada?

El procedimiento incluye la generación de ambas llaves, para esto se debe ejecutar el comando:

```
ssh-keygen
```

Nota: recomendamos leer el **"man"**ual del comando, ya que puede ser útil investigar cuestiones como: el tamaño de la clave, el tipo de claves que se pueden utilizar, o el uso de contraseñas en la misma.

Al ejecutar el comando, el generador de claves nos preguntará en donde deseamos guardar las mismas, si no indicamos un lugar diferente, (lo cual recomendamos), se generarán por default en el home del usuario, dentro del directorio ".ssh", con el nombre **"id_rsa"** para la clave privada y **"id_rsa.pub"** para la clave pública.

```
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
```

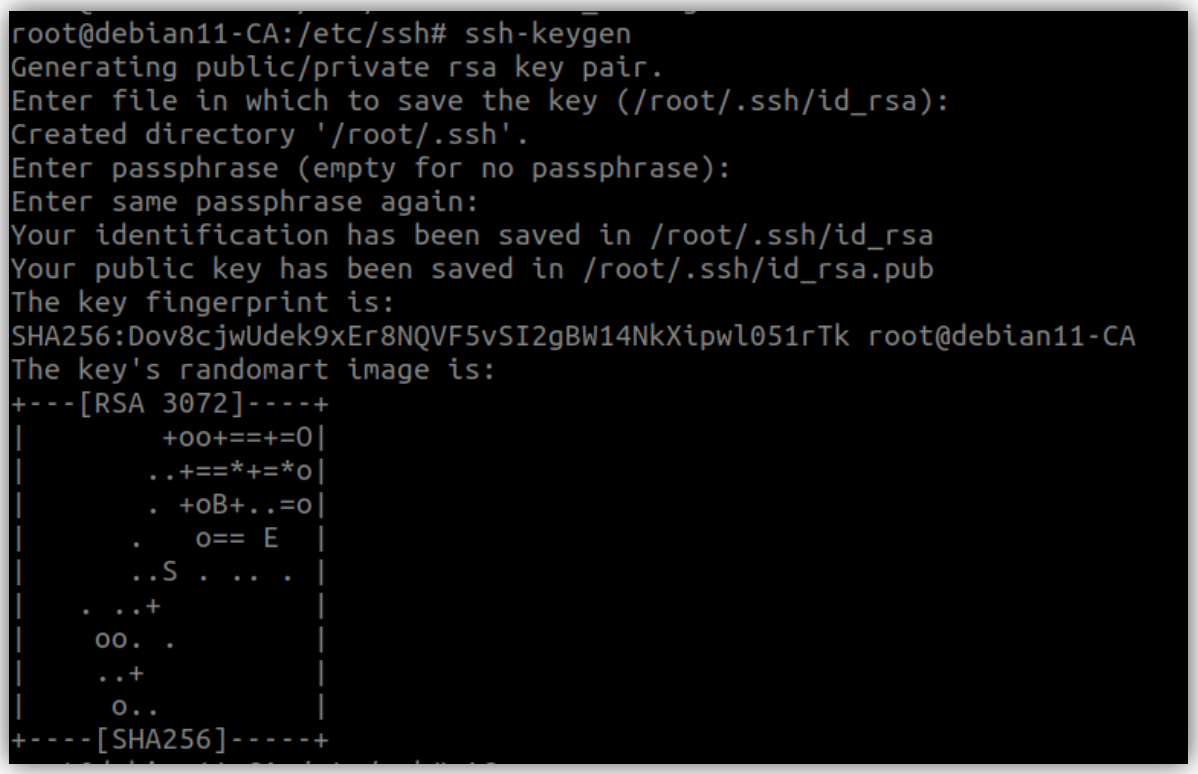
Lo siguiente que preguntara es, si queremos establecer una "frase" como contraseña de las claves. Para esta cursada no lo haremos, y deberemos dejarla en blanco, solo presionando *enter*. Vale decir que, aunque no le pongamos una "frase", dejarlo vacío, ya es una forma en sí misma de establecerle una contraseña, por lo que pedirá que lo repitamos.

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

El proceso genera los 2 archivos que ya hemos mencionado, y que se pueden ver a continuación:

- **id_rsa**: esta clave **NO SE DEBE COMPARTIR**
- **id_rsa.pub**: esta clave es la que **SE DEBE** compartir con el servidor remoto

```
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Dov8cjwUdek9xEr8NQVF5vSI2gBW14NkXipwl051rTk root@debian11-CA
```



Fuente: software VirtualBox [captura de pantalla].

¿Cómo copiar la clave privada al servidor remoto?

Para que el servidor acepte el ingreso por medio de llaves, primero debemos copiar la clave pública que generamos dentro del archivo "authorized_keys" del servidor al que deseamos conectarnos. Para esto, utilizaremos el comando:

```
ssh-copy-id usuario@servidor_remoto
```

```
ssh-copy-id root@192.168.0.201
```

```
/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
```

```
root@192.168.0.201's password:
```

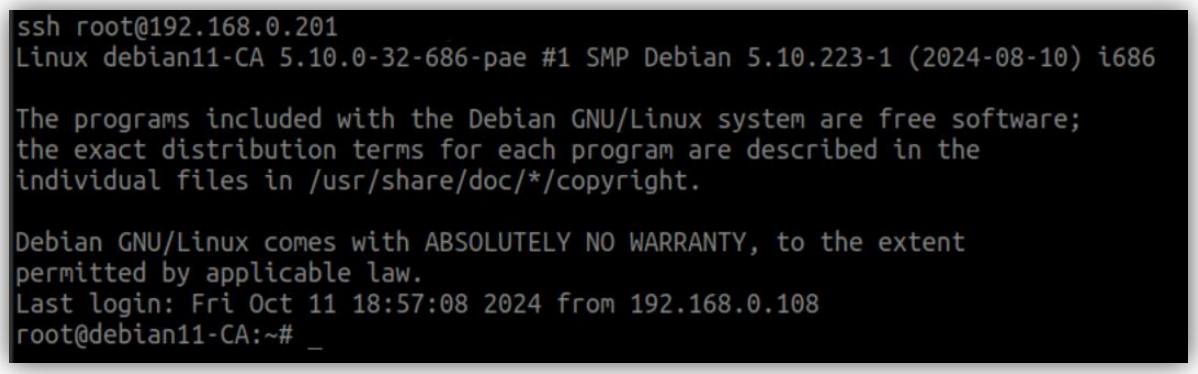
Al intentar copiar la clave como vemos, mostrará un mensaje indicando lo que queremos realizar y se quedará esperando la contraseña del usuario (en este caso, *root*), que es con el que queremos realizar la conexión.

```
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@192.168.0.201'"

and check to make sure that only the key(s) you wanted were added.
```

Luego de introducir la contraseña, se copiará la llave pública y nos invitará a volver a probar la conexión nuevamente, para comprobar que las tareas que realizamos fueron correctas. Si todo salió como esperábamos, la próxima conexión no nos debería pedir contraseña.



Fuente: software VirtualBox [captura de pantalla]

De esta forma, y sin que nos pida contraseña, ahora podemos conectarnos de acá en más por medio de las claves pública y privada.

Para desloguearnos de una conexión SSH establecida, debemos ejecutar el comando:

```
exit
```

Bibliografía

- Niklas, P. (2020). *GNU/Linux, con sabor a Debian.* Pp. 183- 208 (9.13.3).
- Virtual Box: https://www.virtualbox.org/