**Conexión con servidores remotos**

En este post aprenderemos a autenticarnos con servidores remoto, esto a través del protocolo **SSH**.

Los comandos que aquí veremos son válidos para todos los sistemas operativos basados en unix; ubuntu, fedora, red hat, mac OS etc… No importa su versión, ni su arquitectura.

**SSH.**

Si eres completamente nuevo en esto de los servidores, quizás te estes preguntando ¿Qué es SSH? Bien, deja te explico.

**SSH**, por sus siglas al español **secure shell**, es un protocolo el cual nos permite acceder, y tomar el control, de equipos remotos. Y cuando digo tomar el control me refiero al contro total. Todo esto a través de un programa con el mismo nombre **SSH**. La conexión con los equipos remotos se realiza a través del puerto 22.

Para este post será indispensable **SSH**, de igual forma será necesario que cuentes con conocimientos básicos de unix.

###Conexión a un equipo remoto.

Para que nosotros podamos realizar una conexión a un equipo remoto debemos de conocer tres cosas:

* La dirección IP del equipo al que queremos acceder.
* El nombre de usuario con el que nos autenticaremos.
* La contraseña del usuario.

Si onocemos toda esta información podemos autenticarnos remotamente.

Lo primero que debemos hacer será abrir una nueva pestaña en nuestra terminal y ejecutar el siguiente comando.

ssh [email protected]\_ip

En mi caso, quedaría de la siguiente manera. Utilizó el usuario **root**.

ssh root @159.65.27.20

usuario @ e ip deben encontrarse juntos, sin ningún espacio entre ellos.

Basta con ejecutar y completar el proceso de autenticación.

Aceptamos la advertencia y colocamos nuestra contraseña.

Si es la primera vez que se inicia sesión con el servidor, muy probable se nos pida actualizar la contraseña.

Una vez nos hayamos autenticado, tendremos control total sobre el equipo, y podremos ejecutar cualquier comando que necesitemos. 😎

Si queremos salir y regresar nuevamente a nuestro equipo local, únicamente necesitamos ejecutar exit.

##Conexión a un equipo remoto mediante llaves.

Bien, ya podemos autenticarnos con el servidor, basta con colocar, el usuario, la ip y la contraseña; Valores que debemos de tener siempre en cuenta y por su puesto, almacenados en algún lugar seguro.

De estos tres datos, el más sensible, por supuesto, es la contraseña. Colocar la contraseña siempre que necesitamos autenticarnos no suena una muy buena idea. 😲

Lo que haremos ahora para mejorar el proceso de autenticación y hacerlo más seguro, es utilizar una *llave pública*, la cual nos permita autenticarnos con el servidor sin la necesidad de colocar una contraseña.

Todos los siguientes comando los ejecutaremos en nuestro equipo local.

Abrimos una nueva pestaña en nuestra terminal y ejecutamos el siguiente comando.

cat ~/.ssh/id\_rsa.pub

Si obtenemos alguna cadena de caracteres alfanumérica, eso quiere decir que en nuestro equipo ya cuenta con la llave pública.

En caso obtengamos el mensaje No such file or directory no nos preocupemos, únicamente necesitamos generar un par de llaves (una llave pública y una llave privada).

Para todos aquellos que **NO** cuenten con la llave pública debemos de ejecutar el siguiente comando.

ssh-keygen -t rsa -b 4096

Debemos de completar el proceso de creación. Aceptamos la dirección por default donde se guardarán las llave y indicamos una contraseña.

Aquí un tutorial que nos explica, paso a paso, la creación de la llave pública.

<https://youtu.be/OAhLWoB9Ekw>

Una vez tengamos nuestra llave pública, el siguiente paso será indicarle al servidor que nos autenticaremos mediante esta llave y no usaremos más la contraseña, para ello utilizaremos el programa **ssh-copy-id**. Ejecutamos el siguiente comando.

ssh-copy-id [email protected]\_ip

Es probable que dicho programa no se encuentre instalado en nuestro equipo, si este es el caso, basta con instalarlo. El proceso de instalación es bastante sencillo, puedes apoyarte del siguiente [link](https://www.ssh.com/ssh/copy-id) para ello.

Una vez nuestra llave pública se encuentre en el servidor ya podemos autenticarnos sin la necesidad de colocar nuestra contraseña.

ssh [email protected]\_ip

**Conexión únicamente mediante llaves públicas.**

Si **NO** estamos utilizando el usuario **root** para autenticarnos, yo recomiendo indicarle al servidor que únicamente acepte conexiones con llaves públicas, esto para evitar ataques de fuerza bruta.

En caso te encuentres usando el usuario root, te recomiendo crees un nuevo usuario con permisos de administrador.

Una vez autenticados en el servidor ejecutaremos el siguiente comando.

sudo nano /etc/ssh/sshd\_config

Buscamos la línea PasswordAuthentication, eliminamos el comentario (#) y reemplazamos yespor no

Quedaría de la siguiente manera.

PasswordAuthentication no

Así mismo añadimos la siguiente instrucción.

AuthenticationMethods publickey

Guardamos los cambios y reiniciamos el servicio **ssh**.

sudo service ssh restart

**Sin salir** del servidor (muy importate esta parte), abrimos una nueva pestaña en nuestra terminal (local) y nos volvemos a autenticar.

ssh [email protected]\_ip

Ojo, es importante no cerrar la sesión con el servidor, ya que si existe algún error y no podemos autenticanos con solo la llave, al cerrar sesión no será posible ingresar nuevamente.

Si te es posible autenticare, daremos como exitosa la configuración realizada. Ya podemos salir del servidor sin preocupaciones

Esto aumenta la seguridad, sin embargo, si la llave pública por alguna razón se pierde, el acceso al servidor será imposible, por esa razón recomiendo tener en un lugar seguro la llave pública y solo implementar esta opción en el usuario **root** si es estrictamente necesario.

**Conexión a múltiples equipos.**

Ya nos olvidamos de la contraseña, sin embargo, aún nos queda recordar el usuario y la dirección ip.

Si trabajamos únicamente con un servidor es probable que no tengamos problemas para recordarlo, pero qué pasa si trabajamos con 10, 20 o más equipos. En estos casos, esos dos pequeños datos (el nombre de usuario y la dirección ip) pueden ser un gran dolor de cabeza. Afortunadamente podremos autenticarnos a los servidores mediante un alías, algo mucho más fácil de recordar.

Para auntenticarnos con un alías lo primero que debemos de hacer en editar el archivo config, archivo que se encuentra dentro de /.ssh (Si, el archivo no existe, debemos de generarlo).

cd ~/.ssh

nano config

Dentro del archivo colocamos lo siguiente:

Host nuestro\_alias

Hostname direccion\_ip

User usuario

PubKeyAuthentication yes

IdentityFile ~/.ssh/id\_rsa

El valor para el Host será el alías que queremos utilizar.

Lo interesante de todo esto es que podemos agregar la n cantidad de equipos que necesitemos, basta con seguir la estructura mencionada.

Guardamos cambios y ahora nos autenticamos vía **SSH**.

ssh nuestro\_alias

Listo! sin usuarios, dirección ip y sin contraseña, de esta forma podemos autenticarnos a servidores remotos de una mucho forma flexible y sobre todo segura. 😎