

CONEXIONES REMOTAS

1. Métodos de conexión remota hacia un servidor local o en web

Existen métodos que nos ayudan a tener una sesión de conexión a un servidor de forma remota, unos que nos brindan mas seguridad que otros como por ejemplo:

1.1. Conexión por telnet (inseguro)

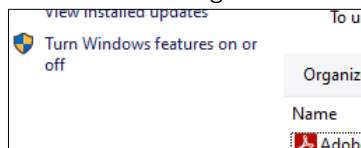
El problema de la conexión por telnet es que la conexión establecida no tiene cifrado y la conexión se vuelve insegura.

Si por ejemplo existiera un ataque MITM (Man In The Middle) el atacante inmediatamente podría capturar los datos de inicio como usuario y contraseña para poder conectarse con el servidor remoto.

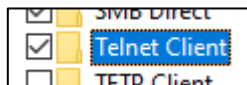
Por seguridad las versiones más actuales de Windows traen telnet deshabilitado

Habilitar telnet en Windows

- Ingresamos a **programas y características** del **panel de control**.
- una vez dentro ingresamos a **Activar o desactivar características de Windows**.



- Ahí dentro activamos la opción cliente telnet



Habilitar telnet en el servidor Linux

- Instalar telnet en el servidor (si no está instalado por defecto)
`sudo apt-get install telnetd -y`
- Reiniciar el servidor
`sudo reboot`

Conexión TELNET desde Windows al servidor

- en CMD de Windows conectamos por telnet al servidor con el siguiente comando
`telnet {DireccionIP}`

1.2. Conexión por SSH (Seguro)

SSH es una herramienta esencial para ser un administrador de servidores.

SSH, o Secure Shell, es un protocolo que se utiliza para iniciar sesión de forma segura en sistemas remotos. Es la forma más común de acceder a servidores Linux remotos.

La ventaja que tiene SSH ante Telnet es que cualquier paquete que se envía de cliente a servidor y viceversa es completamente cifrado y no legible por lo cual es un método de conexión bastante seguro.

Instalación del servicio SSH

Antes de iniciar alguna conexión hacia un servidor remoto usando el método de conexión SSH se debe tener instalado el servicio SSHD en el servidor remoto, caso contrario será imposible conectarse mediante SSH.

Para instalar el servicio ssh en un servidor remoto se debe usar el siguiente comando de acuerdo al gestor de paquetes de la distribución Linux en la cual se está trabajando

```
sudo apt install ssh -y
```

Es normal que luego de ejecutar este comando nos pida password del usuario, por que estamos pidiendo permisos de superusuario.

Para iniciar el servicio SSH

```
sudo systemctl start ssh
```

Para habilitar la ejecución automática del servicio SSH

```
sudo systemctl enable ssh
```

Para ver el estado de funcionamiento del servicio SSH

```
sudo systemctl status ssh
```

Sintaxis básica de conexión

```
ssh {ServidorRemoto}
```

En este ejemplo, ServidorRemoto es la dirección IP o el nombre de dominio al que está tratando de conectarse.

Este comando asume que su nombre de usuario en el sistema remoto es el mismo que el que usa en el sistema local.

Si su nombre de usuario es diferente en el sistema remoto, puede especificarlo usando esta otra sintaxis:

```
ssh -p 22 {UsuarioRemoto}@{ServidorRemoto}
```

Cuando se haya conectado al servidor, se le puede solicitar que verifique su identidad mediante el ingreso de una contraseña.

Para terminar la conexión podemos usar el comando

```
exit
```

1.3. Configuraciones avanzadas para SSH

1.3.1. ¿Que son las claves SSH?

La clave SSH consiste en la generación de un par de claves que proporcionan dos largas cadenas de caracteres **una pública** y **una privada**. La clave pública se instala en cualquier servidor y luego se desbloquea mediante la conexión con un cliente SSH que hace uso de la clave privada.

Si las dos claves coinciden, el servidor SSH permite el acceso sin necesidad de utilizar una contraseña.

1.3.2. Creación de un par de claves SSH desde cualquier ordenador con Windows o con Linux.

- Creación de claves por tipo

```
ssh-keygen -t rsa -b 2048 -C 'CorreoElectronico'
```

Directorio de la creación de claves por defecto

Linux => /home/userName/.ssh/

Windows => C:/users/userName/.ssh/

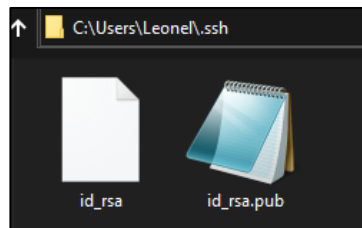
NOTA: posiblemente el directorio .ssh el Linux no este creado, deberá crearlo manualmente dentro del directorio del usuario.

- Luego de confirmar la dirección de creación de los archivos el agente creador le pedirá una frase de contraseña para proteger la clave privada (recomendado), si desea puede insertar un password, si no lo deja en blanco.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

- Se finaliza la creación del par de clave ssh, los archivos deberían de estar dentro del directorio que se indicó anteriormente, son dos archivos uno privado y uno público.



1.3.3. Cargado de la clave publica

Luego de crear el par de claves, tenemos que cargar la clave publica al servidor remoto donde queremos agregar este método de autenticación por claves ssh.

- Para poder cargar la clave publica necesitamos de un servidor con acceso remoto, para este ejemplo existe un servidor en la dirección 172.16.30.11 con nombre de usuario leonel y password leonel y usamos el siguiente comando.

```
C:\Users\Leonel\.ssh>scp ./id_dsa.pub leonel@172.16.30.11:/home/leonel
```

- Nos pedirá que confirmemos la acción tecleando yes
yes
- Luego nos pedirá introducir el password de conexión remota al servidor
leonel@172.16.30.11's password:*****

- habrá finalizado la carga de la clave publica al servidor

```
id_dsa.pub 100% 613 385.2KB/s 00:00
```

1.3.4. Configuración en el servidor para aceptar conexión mediante par de claves

La siguiente configuración se debe realizar con el usuario con el cual queremos tener acceso al servidor con el método de par de claves. el usuario root u otro usuario. para este ejemplo lo configuraremos con el usuario leonel.

- En el servidor remoto debemos asegurarnos tener un directorio .ssh en el directorio del usuario local. Ya sea root u otro usuario, si no lo tenemos lo creamos y dentro del creamos un fichero llamado authorized_keys

```
leonel@ubuntu_serv:~$ mkdir .ssh
```

```
leonel@ubuntu_serv:~$ touch /home/leonel/.ssh/authorized_keys
```
- Ahora copiamos todo el contenido del fichero id_dsa.pub al fichero authorized_keys

```
leonel@ubuntu_serv:/$ cat /home/leonel/id_rsa.pub >> /home/leonel/.ssh/authorized_keys
```
- Verificamos que se haya copiado correctamente

```
leonel@ubuntu_serv:/$ cat /home/leonel/.ssh/authorized_keys
```

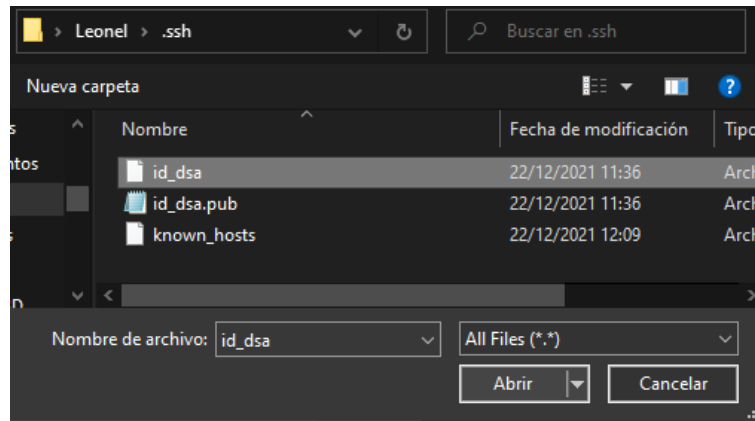
```
ssh-dss AAAAB3NzaC1kc3MAAACBAOu/ywKmkOgiWrp4ejHLeTiXmDH6nvN4DnUWw0eVvyDaotJWRHQ2nb4sfahcYhUvVaFoTgwjjNlGFpVjyvpXH8pDmt6LXBGEpOamE35JtuDrIQ7cdECpPn/ecearlqSUJfm150AK9Cg/TbnWHWY/GJVFFX+mxNziPEe7wxfLjbAAAAFQCQPddYG97rtR3LbSLEJYBIbqBnQAAAIBo0jZKzbTNmzuv8fO2i9Ab5ZwNSNk6PNZRgyjZ7iE17rDzUndMYcZiZrGAfr6f0F+axb+zQhy/VcvD2AcB9b+0p7qMJAdB7sjQch136hJtSPfHxb5ukUNNULB2RD7zL5VmpOuMYzFzbrzKs46+QENrc/zFGIu0YV18LcyRRRYggAAAAIEayesd5wIc2bYtYOTJktmCowIGicfkLzfJqe0joaykZPkeMqRs1IQP2LyOVw9pi/53fw4mNdWxiR8wbHTMx0q4pEG3MAXgZIJv4eZThgwcZ2XC7iPRwSGR+d4kVtax6mFURiMICA VzKrtKRM7klBq4lwFbGjSq82Ir3i0GERgAMU= leonel@DESKTOP-8H3FTVO
```
- Ahora configuramos los permisos necesarios para el directorio .ssh y el fichero authorized_keys para que nadie más que el usuario root y el usuario local puedan ver o editar esta clave.

```
chmod 0700 /home/leonel/.ssh
```

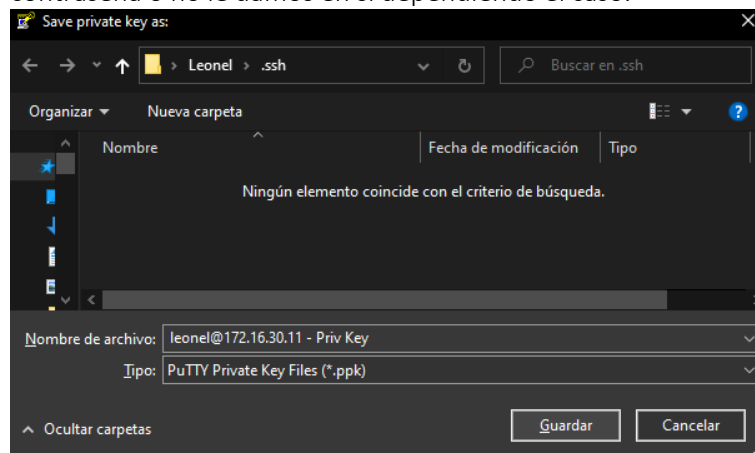
```
chmod 0644 /home/leonel/.ssh/authorized_keys
```

1.3.5. Configuración de la clave privada para tener acceso al servidor mediante el par de claves.

- Con la ayuda de la herramienta putty-gen vamos a generar un archivo.ppk
- Abrimos putty-gen y cargamos con el botón **load** la clave privada que ya habíamos creado.

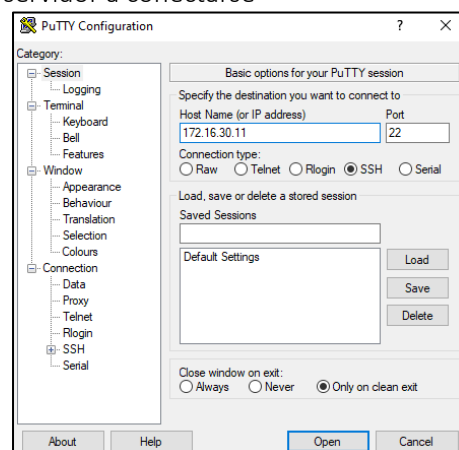


- Luego de cargar la clave privada damos clic en **Save private key** para guardar nuestra clave privada con el nombre que queramos, si pide una frase de contraseña o no le damos en si dependiendo el caso.

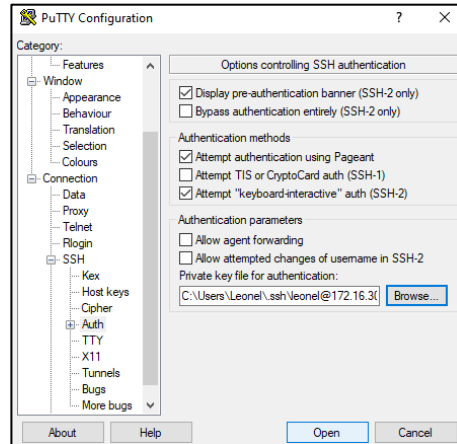


1.3.6. Probando conexión usando la aplicación putty.

- Para ellos usamos la herramienta putty, donde pondremos la dirección del servidor a conectarse



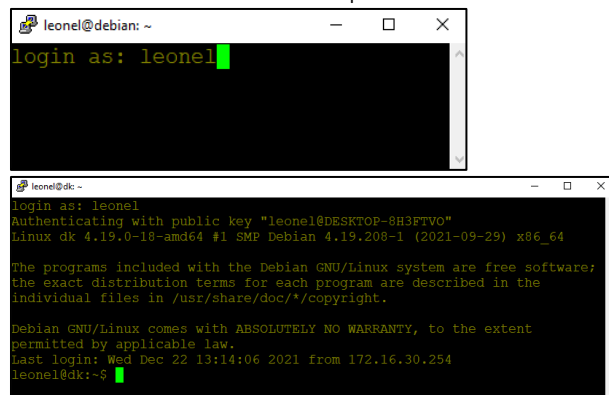
- Desplegamos SSH y en auth seleccionamos el archivo .ppk que habíamos generado y damos clic en open



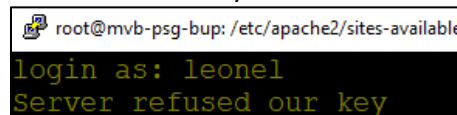
- El servidor nos responderá pidiendo el nombre del usuario, en ese momento el servidor reaccionará de dos formas

(Reacción 1 - El servidor acepta la llave para ingresar al servidor mediante SSH)

En este caso ponemos "leone1" para iniciar sesión y podremos ingresar al servidor sin necesidad de un password de acceso



(Reacción 2 - El Servidor rechaza la llave por motivos de falta de configuración dentro del servidor)



En este caso se debe recordar cómo se realizó toda la configuración anterior, con el usuario leone1 o con el usuario root ya que el modo en el que se realizó la configuración influye en gran medida ya que le agregamos permisos especiales al directorios y archivo de configuración.

Conclusión: si queremos tener acceso mediante par de claves con un usuario específico tenemos que realizar la configuración con el mismo usuario.

1.3.7. Probando conexión usando la consola de Windows o la terminal de Linux

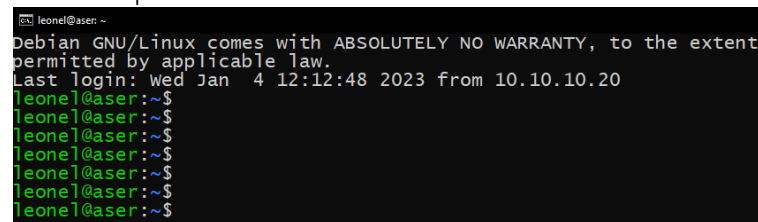
En nuestro cmd de WINDOWS desde cualquier ubicación en línea de comandos podemos usar la siguiente sintaxis para conectar al servidor remoto usando el par de llaves ssh que generamos anteriormente

```
D:\>ssh {UsuarioRemoto}@{HostRemoto} -i  
{UbicacionDeLaLLavePrivada}\{LlavePrivada}
```

Para conectarnos el servidor que tenemos de forma local usaríamos el siguiente comando

```
D:\>ssh leonel@192.168.0.171 -i C:\Users\Leonel\.ssh\id_rsa
```

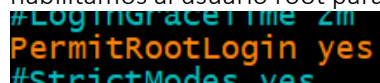
De esa forma tenemos conexión mediante el par de llaves desde cmd de Windows usando el par de claves.



```
leonel@aser: ~  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Wed Jan 4 12:12:48 2023 from 10.10.10.20  
leonel@aser:~$  
leonel@aser:~$  
leonel@aser:~$  
leonel@aser:~$  
leonel@aser:~$  
leonel@aser:~$  
leonel@aser:~$  
leonel@aser:~$
```

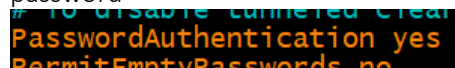
1.3.8. Habilitar la autenticación para el usuario ROOT por password (inhabilitado por defecto)

Para habilitar al usuario root que por defecto viene deshabilitado tenemos que configurar el fichero `sshd_conf` que se encuentra en `/etc/ssh/`
habilitamos al usuario root para que pueda autenticarse



```
#LoginGraceTime 2m  
PermitRootLogin yes  
#StrictModes yes
```

Habilitamos la opción para que cualquier usuario pueda autenticarse con password



```
#To disable tunneled clear  
PasswordAuthentication yes  
PermitEmptyPasswords no
```

Con estas dos modificaciones en este fichero permitimos que el usuario root pueda autenticarse en el servidor remoto solamente con nombre de usuario y password.

Luego de hacer cualquier modificación en el fichero de configuración es necesario reiniciar el servicio ssh

1.3.9. Deshabilitar la conexión remota usando password.

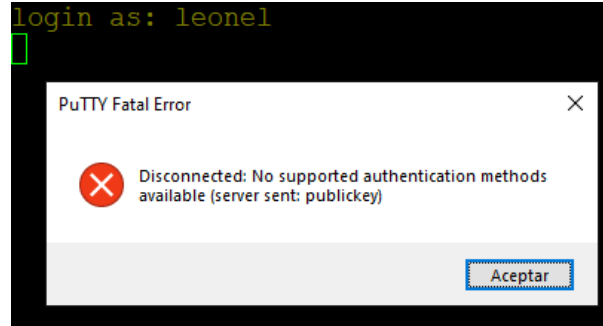
- Ingresamos al archivo de configuración SSH
sudo nano /etc/ssh/sshd_config

- Ahí dentro buscamos estas dos líneas y las modificamos de la siguiente forma

```
# To disable tunneled clear text passwords, change to no here!  
PasswordAuthentication no  
PermitEmptyPasswords no
```

- Ahora reiniciamos el servicio SSH
sudo systemctl restart ssh

- Ahora verificamos que ya no tenemos acceso por password



1.3.10. Cambiar el puerto de conexión SSH

Es muy importante cambiar el puerto 22 de una conexión SSH ya es un puerto por defecto, por tal motivo el puerto por defecto se debe cambiar a uno desconocido o uno que solo el administrador pueda conocer, también hay que tomar en cuenta que al cambiar el puerto también alguna entrada o abrir un puerto destinado en el firewall para no perder futuras conexiones.

Luego de cambiar el puerto por defecto posiblemente también abrir el nuevo puerto de conexión SSH en el firewall si lo tenemos instalado y configurado o si no lo tenemos configurado en iptables

Cambiar la configuración del puerto por defecto

sudo nano /etc/ssh/sshd_config

Cambiamos por la siguiente configuración, borrando el #

```
#Port 22  
#AddressFamily
```

```
Port 222  
#AddressFamily
```

Luego reiniciamos el servicio SSH

sudo reboot