

# SSH, Linux.

## Indece:

- 1- Introducción.
- 2- Contexto.
- 3- Instalación.
- 4- Seguridad.
- 5- ssh.
- 6- scp.
- 7- sftp.
- 8- sshpass.
- 9- Tunneling.
- 10- ssh y docker.

## 1- Introducción:

Estos pequeños tips, van dirigidos a todos los informáticos con mentes inquietas, que se inicien en este maravilloso mundo que es linux. Cubanos o no, que necesitan establecer conexiones seguras, cifradas entre servidores o desde su casa con otros ordenadores. El servicio ssh “secure shell”, después de configurar la red. Es uno de los primeros que usamos al instalar una distro de linux, incluso en ubuntu y debian puedes instalarlo en el mismo momento que instalas en el OS. OpenSSH tiene abundante documentación en la red, incluso en muchos idiomas.

## 2- Contexto:

Luego de instalar OpenSSH en un linux-mint\_20, vamos a personalizar su configuración, vamos a asegurarlo, vamos a establecer quienes pueden acceder a OpenSSH a través de tcp wrappers y a través de iptables. Estas notas han sido probadas en Ubuntu\_20 y Debian\_10. Así como vamos a hacer uso de las aplicaciones que incluye ssh, como son scp y sftp. Para terminar y no menos importante vamos a hacer tunneling con conexiones a través de OpenSSH.

## 3- Instalación:

Para instalarlo sólo debemos escribir dentro de un shell.

Desde el server.

```
sudo apt -y install openssh-server
```

Verificamos que esté instalado.

```
sudo apt list openssh-server
---
openssh-server/focal-updates,now 1:8.2p1-4ubuntu0.5 amd64 [installed]
---
```

Desde el client.

```
sudo apt-get -y install openssh-client
```

Verificamos que esté instalado.

```
sudo apt list openssh-server
```

```
---
openssh-server/focal-updates,now 1:8.2p1-4ubuntu0.5 amd64 [installed]
---
```

El file de configuración tanto para el server como para el client está `/etc/ssh/sshd_config`.

## 4- Seguridad:

Debemos asegurar el server que estará esperando conexiones:

4.1- Quien puede ver y modificar su configuración: El usuario root debe ser el único autorizado que pueda modificar todo el contenido dentro del folder `/etc/ssh`

```
sudo chown -Rvf root:root /etc/ssh && chmod -Rvf 644 /etc/ssh
```

4.2- Quienes pueden usar conexiones ssh: Vamos a hacer algunas modificaciones en el archivo de configuración para decidir quienes se conectan al server.

- Port 5022 “Cambiamos el puerto de escucha de este servicio.”
- PermitRootLogin no “Evitamos que el usuario root haga un login a través de este servicio.”
- AllowUsers administrador “Permitimos el acceso sólo al user administrador.”
- MaxAuthTries 3 “Intentos permitidos de autenticación 3.”
- MaxSessions 3 “Secciones permitidas 3.”

Así quedaría el `sshd_config`

```
sudo nano /etc/ssh/sshd_config
---
Port 5022
AddressFamily inet

PermitRootLogin no

ChallengeResponseAuthentication no

MaxAuthTries 3
MaxSessions 3

AllowUsers administrador

UsePAM yes

X11Forwarding yes
PrintMotd no
AcceptEnv LANG LC_*

Subsystem      sftp    /usr/lib/openssh/sftp-server
---
```

### 4.3- iptables:

El script que contiene las reglas del cortafuegos, podría permitir las conexiones SSH para una ip específica y bloquear el resto de las conexiones.

```
-A INPUT -s 192.168.1.4/32 -p tcp -m tcp --dport 5022 -j ACCEPT
-A INPUT -s 0.0.0.0/0 -p udp -m udp --dport 5022 -j DROP
```

podría ser también.

```
iptables -A INPUT -p udp --dport 5022 -j REJECT
iptables -A INPUT -p tcp --dport 5022 -j REJECT
```

#### 4.4- TCP Wrappers: host.allow y host.deny:

Con host allow y host deny podemos agregar otra capa de seguridad, le permitimos el acceso a las ips 192.168.1.1 y 1.2.

```
sudo nano /etc/hosts.allow
---
#
sshd : 192.168.1.1 192.168.1.2
---
```

#### SSH:

Necesitamos tener instalado el client y un remote server esperando conexiones. Usando el command ssh, podemos conectarnos.

```
sudo ssh administrador@192.168.1.1
```

Con ssh, podemos ejecutar commands en un server remoto.

```
ssh administrador@192.168.1.1 ls -l /
ssh administrador@192.168.1.1 cat /etc/hostname
```

#### SCP:

Con scp, podemos copiar files y folder a un ordenador remoto, su uso es similar al comando cp.

Ejemplos de copia local a remoto.

```
scp -r /path/folder administrador@192.168.1.1:/home/administrador
scp -P 5022 /path/*.txt administrador@192.168.1.1:/home/administrador
```

Ejemplos de copia remoto a local.

```
scp administrador@192.168.1.1:/home/administrador/*.txt /path_local
```

Ejemplos de copia remoto a local.

```
scp administrador@192.168.1.1:/home/administrador/*.sh root@172.16.1.53:/SALVA
```

#### SFTP:

Con sftp, podemos copiar files y folder a un ordenador remoto, su uso es similar al comando ftp.

```
sftp://administrador@192.168.1.1:5022/
```

#### sshpass:

#### Conclusiones:

#### Tunneling: