


SSH, Linux.

```
root@geiser-laptop:~# aptitude search openssh-client openssh-server
i  openssh-client                    - secure shell (SSH) client, fo
p  openssh-client:i386                - secure shell (SSH) client, fo
p  openssh-client-ssh1                - secure shell (SSH) client for
i  openssh-server                    - secure shell (SSH) server, fo
p  openssh-server:i386                - secure shell (SSH) server, fo
root@geiser-laptop:~#
```



Indece:

Introducción.

Contexto.

Instalación.

Seguridad.

ssh.

scp.

sftp.

sshpas.

Tunneling.

Conclusiones.

Bibliografía.

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.

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 host.deny 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 como scp y sftp. Para terminar y no menos importante vamos a hacer tunneling con conexiones a través de OpenSSH.

Instalación:

Para instalarlo sólo debemos escribir dentro de un shell.

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

Seguridad:

Debemos asegurar este servicio:

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
```

2- Quienes pueden usar conexiones ssh: Vamos a hacer algunas modificaciones en el archivo de configuración */etc/ssh/sshd_config*, para decidir quienes se conectan al server.

- Port 5022 “Cambiamos el puerto de escucha de este servicio.”
- PermitRootLogin no “Evitamos que el ususrio root haga un login atravez de este servicio.”
- AllowUsers administrador “Permitimos el acceso sólo al user administrador.”
- MaxAuthTries 4 “Intentos permitidos de autenticación 4.”
- MaxSessions 4 “Secciones permitidas 4.”

Así quedaría el *sshd_config*

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

PermitRootLogin prohibit-password

PasswordAuthentication no
ChallengeResponseAuthentication no

MaxAuthTries 4
MaxSessions 4

AllowUsers administrador

UsePAM yes

X11Forwarding yes
PrintMotd no
AcceptEnv LANG LC_*

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

SSH:

Con ssh, podemos conectarnos a un server remoto.

```
sudo ssh administrador@192.168.1.1
```

Con ssh, podemos ejecutar commands en un server remoto.

```
sudo ssh administrador@192.168.1.1 ls -l /
```

SCP:

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

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

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:

Tunneling:

Bibliografia:

<https://es.wikipedia.org/wiki/OpenSSH>

<https://phoenixnap.com/kb/linux-scp-command>

<https://www.tecmint.com/prevent-ssh-brute-force-login-attacks/>