SSH, Linux.

Indece:

1- Introducción:

Estos pequeños tips, van dirigidos a todos los informáticos con mentes inquietas, que se inicien en este mundo maravilloso. 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 unos de los primeros que usamos al instalar una distro de linux, incluso en ubuntu y debian puedes instarlo 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 con tcp_wrappers y de iptables. Estas notas han sido probadas en Ubuntu_20, Debian_10 y para finalizar lo dockerizamos en una image debian:latest. Así como vamos a hacer uso de las aplicaciones que incluye ssh, como son scp y sftp. También y no menos importante vamos a hacer tunneling con conexiones atravez 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
```

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

```
chown root:root /etc/ssh/sshd_config && chmod 700 /etc/ssh/sshd_config
```

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 ususrio root haga un login atravez 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 DROP
iptables -A INPUT -p tcp --dport 5022 -j DROP
```

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 sólo a las ips 192.168.1.1 y 1.2.

```
sudo nano /etc/hosts.allow
```

```
---
#
sshd : 192.168.1.1 192.168.1.2
---
```

5- SSH:

Para conectarnos, necesitamos tener installed el client y un remote server esperando conexiones. Usando el command ssh, podemos conectarnos.

```
sudo ssh administrador@192.168.1.1
```

Con ssh, también podemos ejecutar commands en un server remoto.

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

6- 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 remoto.

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

7- SFTP:

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

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

8- Tunneling:

Antes de iniciar una conexión bajo un ssh tunneling, debemos estar seguros de:

```
sudo nano /etc/ssh/sshd_config
```

```
---
#
AllowTcpForwarding yes
---
```

```
ssh -L 6900:localhost:5900 administrador@192.168.1.1 -p 5022
```

```
6900                                    ---> port local
localhost                               ---> system local
5900                                    ---> remote local
administrador@192.168.1.1 -p  5022      ---> system remote
```

```
!Nota:
Así podremos conectarnos al vnc_server que esta en la ip 192.168.1.1, pero usaremos open_ssh para
establecer la conexión atravez de este y todo encrypted.
```

8- DOCKER:

Para este punto debemos tener instalado docker, yo personalmente uso docker.io, así como una imagen de la distro de preferencia. !lo demas es pan comido.

Listamos las images disponibles. !vemos que está la que usaremos.

```
sudo docker image ls
---
#
debian                  latest          6f4986d78878   12 months ago   124MB
---
```

Levantamos un container de prueba.

```
sudo docker run -ti debian:latest bash
```

Desde otro shell nos aseguramos que este está running.

```
sudo docker ps
---
cff704df5add   debian:latest    "bash"  26 seconds ago   Up 23 seconds    peaceful_khorana
---
```

Crearemos un container con nombre, con un volumen para no perder los cambios en el file sshd_config, con una static ip address. !Yo he creado una red de tipo bridge con antelación.

```
sudo docker run -ti --name debian-openssh \
--net homenet --ip 172.15.0.3 -p 222:22 \
-d \
-v /home/docker/openssh/app/etc/ssh:/etc/ssh \
debian:latest
```

Ahora llegó el momento de entrar al container, hacerle un update, instalar openssh-server, darle permiso para que el root user pueda entrar.

```
sudo docker exec -it 194d7a814abc /bin/bash
apt update
apt install -y aptitude openssh-server nano
aptitude search aptitude openssh-server nano

nano /etc/ssh/sshd_config
---
PermitRootLogin yes
---

/etc/init.d/ssh restart
```

Le damos un password al root user para cuando nos lo pida al establecer una ssh conexión.

```
passwd root
---
passwd root
New password:
Retype new password:
passwd: password updated successfully
---
```

!!! probamos y establecemos conexión.

```
telnet localhost 222 o telnet 172.15.0.3 22
ssh root@localhost -p 222 o ssh root@172.15.0.3
---
The authenticity of host '[localhost]:222 ([127.0.0.1]:222)' can't be established.
ECDSA key fingerprint is SHA256:yegDBT/XDBRN6Kc6XGdvHDR6YyNOBx8hHharOf80Fi4.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
---
```