Documentació SSH

05/03/2022

Alex Garrido Hernandez
Smx 2n G1 M7

2021-2022

Índex

APARTAT 1 - Configuració de SSH

1. Explica per què serveix el servei SSH

SSH o Secure Shell, és un protocol d'administració remota que permet als usuaris controlar i modificar els seus servidors remots a través d'Internet a través d'un mecanisme d'autenticació.

2. Com es diu el servei, quin usuari l'executa, quin port fa servir

El servei SSH és executat per l'usuari **user**

```
root@debian:/home/user# ps aux | grep ssh
user        1372  0.0  0.0   5964   464 ?        Ss   18:21   0:00 /usr/bin/ssh-agent /usr/bin/im-launch x-session-manager
root        2183  0.0  0.1  13292  7332 ?        Ss   18:25   0:00 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
root        2193  0.0  0.0   6180   652 pts/0    S+   18:25   0:00 grep ssh
root@debian:/home/user#
```

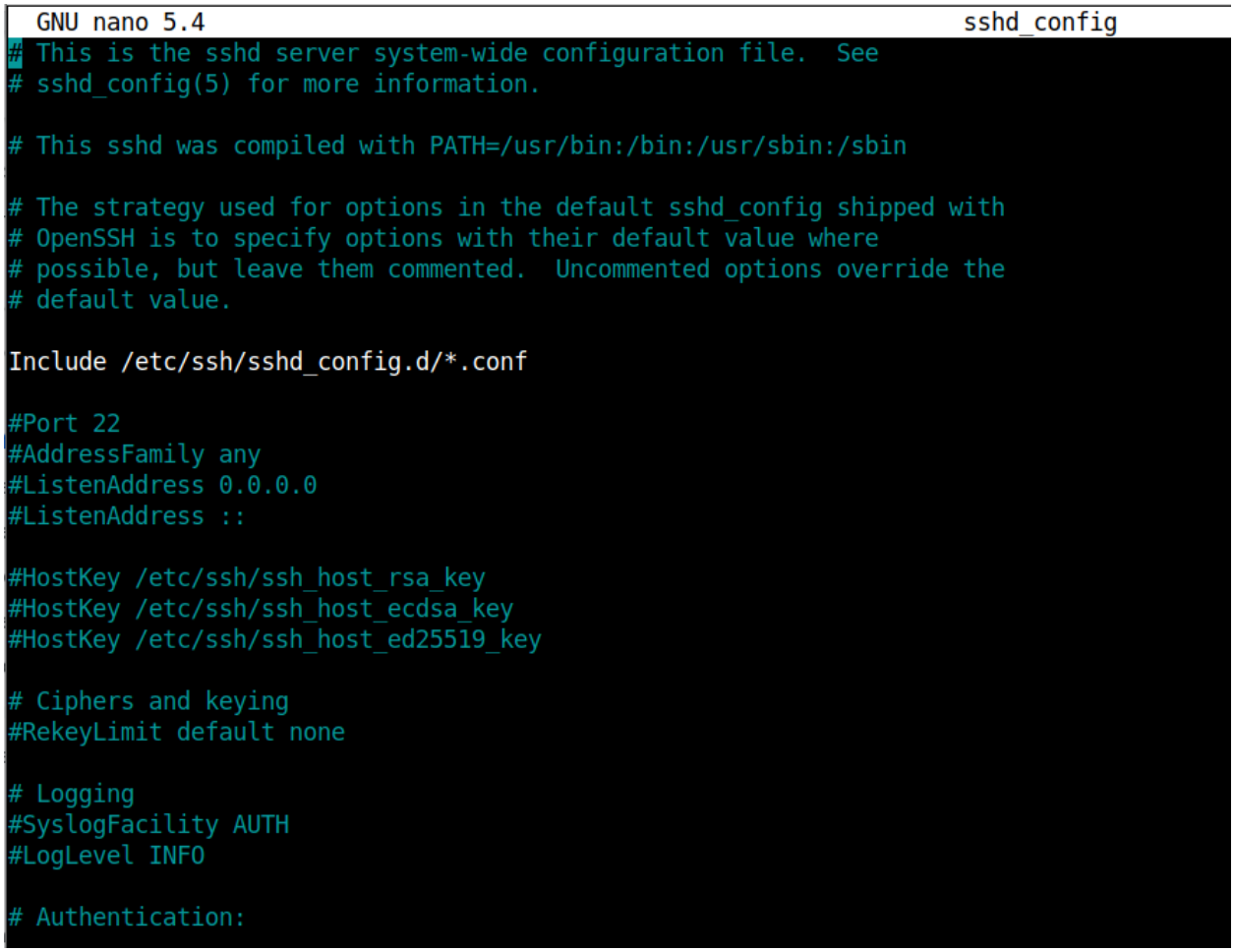
El port que fa servir el mirarem executant la comanda **nmap.** Com podem veure el servei SSH utilitza el **port 22**.

```
root@debian:/home/user# nmap localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-01 18:28 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000020s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
25/tcp open  smtp

Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds
root@debian:/home/user#
```

3. Quins fitxers fem servir per configurar el servei

Per fer la configuració del servidor SSH utilitzarem **sshd_config.** En aquest arxiu trobarem configuracions com: **el port, llistes d'adreçes, rootlogin...**



4. Instal·lació del servidor SSH

Per instal·lar el servidor SSH utilitzarem la següent comanda:

apt install openssh-server

Una vegada executem la comanda començarà a descarregar el servei, una vegada finalitzat, mirarem a la carpeta **/etc/ssh** per veure els arxius de configuració.

```
root@debian:/etc/ssh# ls
moduli      ssh_config.d  sshd_config.d       ssh_host_ecdsa_key.pub  ssh_host_ed25519_key.pub  ssh_host_rsa_key.pub
ssh_config  sshd_config   ssh_host_ecdsa_key  ssh_host_ed25519_key    ssh_host_rsa_key
root@debian:/etc/ssh#
```

5. Principals directives de configuració.

a. Configuració dels ports i targeta

Per configurar els ports del servei SSH haurem d'anar a la directiva **Ports** i posarem el port que millor ens convingui, en aquest cas és el 2022.

```
root@garrido:/home/user# ssh -p 2022 user@192.168.8.20
```

Per fer la connexió per un port específic haurem d'utilitzar l'opció -**p**.

b. Permisos i restricció d'usuaris

En la configuració del servidor de SSH tenim la directiva **AllowUsers,** aquesta directiva permet l'entrada dels usuaris que posem.

També existeix la directiva **DenyUsers,** aquesta directiva fa el contrari que l'anterior. No trobarem cap de les dues per defecte a l'arxiu de configuració, són "extres".

```
#AllowUsers joel
#DenyUsers user
```

c. Límit de temps per fer el login.

Per donar un límit de temps per iniciar sessió, haurem de modificar la directiva **LoginGraceTime.**

En aquest cas per defecte tenim 2 minuts, però podem posar **10 s** o **50 s.**

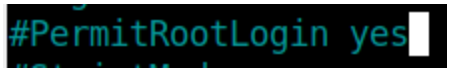
d. Nombre màxim d'intents per fer login

Serveix per donar un màxim d'intents per fer login.

Per defecte tenim **6 intents**

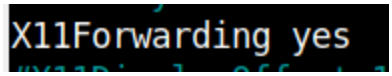
e. No permetre accessos de root

La directiva que permet l'accés de root s'anomena **PermitRootLogin**. En aquest cas, si volem que inicïin sessió amb root posarem "**yes"** o "**no".**



f. Reenviament X11

El reenviament X11 serveix per donar a la connexió SSH un entorn gràfic si l'usuari vol, aquesta directiva per defecte està activada.



g. Reenviament contrasenyes desades.

6. Connexions

a. Qui està connectat (who)

Amb la comanda **who** podem veure qui està connectat, **a quina** hora s'ha connectat, amb **quin** usuari s'ha connectat.

```
root@debian:/etc/ssh# who
user     tty7         2022-03-04 09:18 (:0)
user     tty1         2022-03-04 09:18
user     pts/1        2022-03-04 11:17 (192.168.8.18)
user     pts/2        2022-03-04 11:22 (192.168.8.18)
root@debian:/etc/ssh#
```

En aquest cas tenim al Alae connectat com a **user**

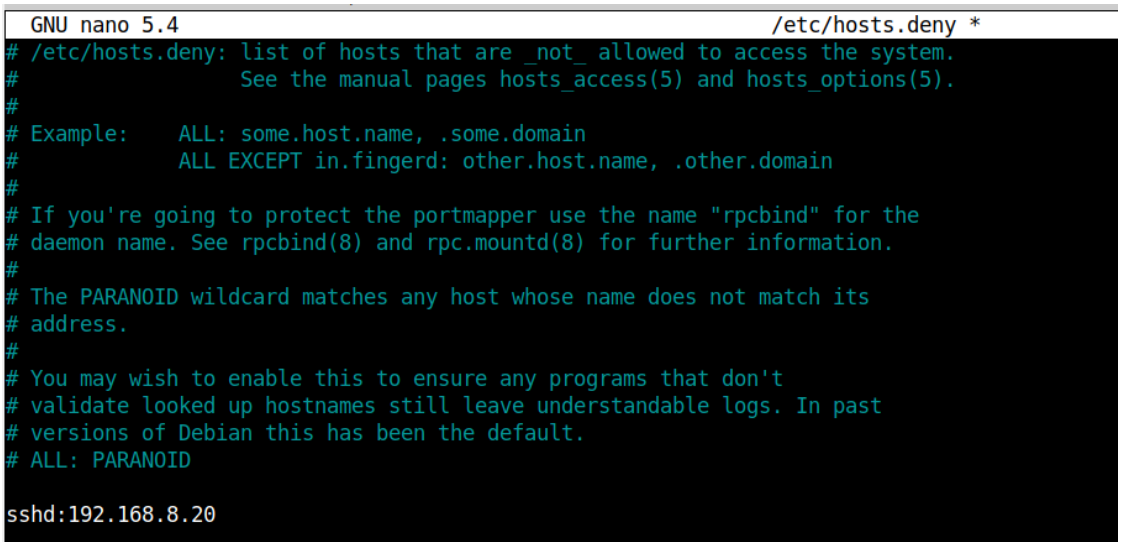
b. Quines connexions han hagut

Per veure quines connexions hi ha hagut al nostre ordinador, podrem fer servir l'arxiu **auth.log.** El trobarem a **/var/log/auth.log**.

```
 GNU nano 5.4                                    /var/log/auth.log
Mar  5 14:32:29 localhost usermod[624]: change user 'root' password
Mar  5 14:32:29 localhost groupadd[635]: group added to /etc/group: name=user, GID=1000
Mar  5 14:32:29 localhost groupadd[635]: group added to /etc/gshadow: name=user
Mar  5 14:32:29 localhost groupadd[635]: new group: name=user, GID=1000
Mar  5 14:32:29 localhost useradd[641]: new user: name=user, UID=1000, GID=1000, home=/home/user, shell=/bin/bash, from=none
Mar  5 14:32:29 localhost usermod[650]: change user 'user' password
Mar  5 14:32:29 localhost chfn[657]: changed user 'user' information
Mar  5 14:32:29 localhost usermod[665]: change user 'user' password
Mar  5 14:32:29 localhost gpasswd[673]: user user added by root to group audio
Mar  5 14:32:29 localhost gpasswd[680]: user user added by root to group cdrom
Mar  5 14:32:29 localhost gpasswd[687]: user user added by root to group dip
Mar  5 14:32:29 localhost gpasswd[694]: user user added by root to group floppy
Mar  5 14:32:29 localhost gpasswd[701]: user user added by root to group video
Mar  5 14:32:29 localhost gpasswd[708]: user user added by root to group plugdev
Mar  5 14:32:29 localhost gpasswd[715]: user user added by root to group netdev
Mar  5 14:32:29 localhost gpasswd[724]: user user added by root to group bluetooth
Mar  5 14:32:29 localhost usermod[737]: add 'user' to group 'sudo'
Mar  5 14:32:29 localhost usermod[737]: add 'user' to shadow group 'sudo'
Mar  5 14:32:29 localhost sudo:     root : PWD=/ ; USER=user ; COMMAND=/usr/bin/sh -c echo 'SU_TO_ROOT_SU=sudo' >> /home/user/.su-to->
Mar  5 14:32:29 localhost sudo: pam_unix(sudo:session): session opened for user user(uid=1000) by (uid=0)
Mar  5 14:32:29 localhost sudo: pam_unix(sudo:session): session closed for user user
Mar  5 14:32:29 localhost sudo:     root : PWD=/ ; USER=user ; COMMAND=/usr/bin/sh -c umask 0077 && mkdir -p /home/user/.kde/share/co>
Mar  5 14:32:29 localhost sudo: pam_unix(sudo:session): session opened for user user(uid=1000) by (uid=0)
Mar  5 14:32:29 localhost sudo: pam_unix(sudo:session): session closed for user user
Mar  5 14:32:29 localhost systemd-logind[912]: New seat seat0.
                                [ Read 57 lines ]
```

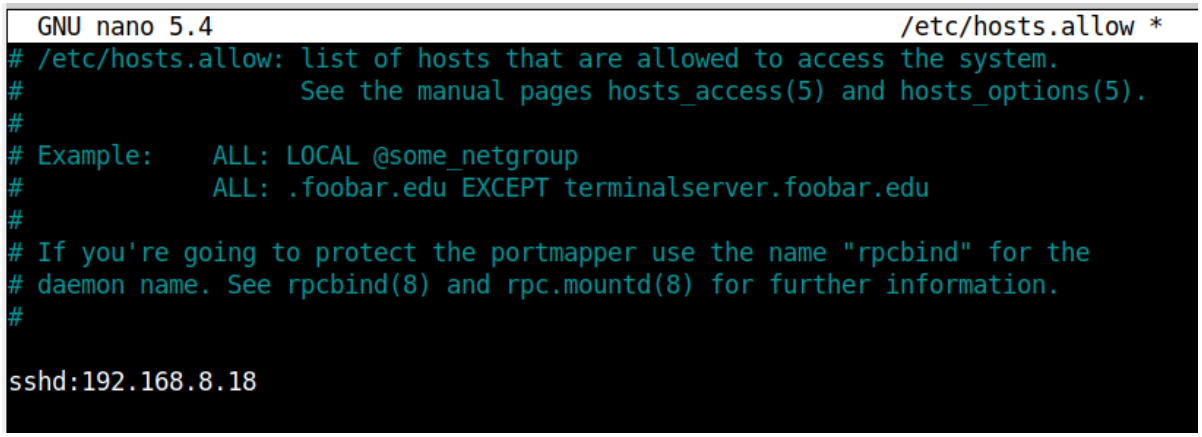
c. Permetre o bloquejar IPs per hosts

Si en aquest cas volem bloquejar/permetre IP's per hosts el que haurem de fer serà anar a l'arxiu **/etc/hosts.deny**.



Com podem veure, el que haurem de posar serà **sshd: "ip"**. En el meu cas serà la IP del meu company Joel.

En cas contrari, si volem permetre, haurem d'anar a l'arxiu **/etc/hosts.allow** i fer el mateix.



7. Mecanismes d'autenticació.

a. Usuari-contrasenya.

Com a primer mecanisme d'autenticació tenim **usuari-contrasenya**, és el més senzill, només haurem de fer un ssh normal i corrent i ens demanarà primer l'usuari i després la contrasenya.

```
root@debian:/etc/ssh# ssh user@192.168.1.140
The authenticity of host '192.168.1.140 (192.168.1.140)' can't be established.
ECDSA key fingerprint is SHA256:71ZgSgdzTT53lCn5H3VuxFlrUoZliEIyTguQR+mI/Fo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.140' (ECDSA) to the list of known hosts.
user@192.168.1.140's password:
Linux debian 5.10.0-8-amd64 #1 SMP Debian 5.10.46-4 (2021-08-03) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
user@debian:~$
```

En aquest cas ho he fet amb dues màquines virtuals a casa.

b. Clau pública/privada.

Per iniciar sessió amb les claus públiques/privades haurem de crear al "client" la nostra clau, en aquest cas no hem posat frase de pas. Es guardaran automàticament a un directori creat (ssh)

La comanda és la següent → **ssh-keygen -b 4096 -t rsa**

```
root@garridocliente:/home/user# ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:8qRFVIow+6tDsClmsuGuYipTmJIDQeEkbasvZwt0w+I root@garridocliente
The key's randomart image is:
+---[RSA 4096]----+
|o=. o   ...      |
|= o  + o .       |
| + .. . o        |
|. +  . .         |
|.B B  o S        |
|&o* o  B         |
|BE .  o .        |
|*++ ..           |
|X*....           |
+----[SHA256]-----+
root@garridocliente:/home/user#
```

A continuació haurem de passar la nostra clau pública al servidor.

Com podem veure, ens surt que s'ha **afegit 1 key.** La comanda és la següent:

ssh-copy-id -i /root/.ssh/id_rsa.pub user@192.168.*.*

```
root@garridocliente:/home/user# ssh-copy-id -i /root/.ssh/id_rsa.pub user@192.168.1.137
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '192.168.1.137 (192.168.1.137)' can't be established.
ECDSA key fingerprint is SHA256:08n+p67a1aw0Ra7bwizhv7ZIIPEfNeSREFt6ZQSBcFw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
user@192.168.1.137's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'user@192.168.1.137'"
and check to make sure that only the key(s) you wanted were added.

root@garridocliente:/home/user#
```

I com podem veure, en fer un **ssh** no ens demana cap contrasenya.

```
root@garridocliente:/home/user# ssh user@192.168.1.137
Linux garridoserver 5.10.0-8-amd64 #1 SMP Debian 5.10.46-4 (2021-08-03) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
user@garridoserver:~$
```

c. Clau pública/privada més frase de pas

La diferència amb l'anterior clau és que aquesta tindrem un passphrase, en el nostre cas serà **pelota**.

Repetirem tot el procés anterior, però a l'hora de fer el **ssh** ens demanarà aquesta **passphrase**.

```
root@garridocliente:/etc/ssh# ssh user@192.168.1.137
Enter passphrase for key '/root/.ssh/id_rsa':
Linux garridoserver 5.10.0-8-amd64 #1 SMP Debian 5.10.46-4 (2021-08-03) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Mar  5 15:10:48 2022 from 192.168.1.140
user@garridoserver:~$
```

8. Transferència de fitxers (scp)

Per fer una transferència de fitxers per **SCP** el que farem serà la següent comanda:

scp usuari@ip_remota:arxiu_remot arxiu_local → **Còpia de remot a local**

scp arxiu_local usuari@ip_remota:arxiu_remot → **Còpia de local a remot**

Existeix la posibilitat de diverses opcions a l'hora d'executar la comanda **scp**:

- **-q :** Serveix per fer una transferència en mode **quiet**. No sortirà cap procés de transferència ni el progrés.
- **-r :** Copia els arxius recursivament
- **-c :** Farà la copia amb l'arxiu comprimit

```
root@debian:/etc/ssh# scp -q user@192.168.8.20:/home/user/Desktop/fitxerprova.txt /home/user/Desktop/
root@debian:/etc/ssh#
```

9. SSH Tunneling.

a. Saltem un tallafoc.

b. Activem un sock i naveguem amb ell.

10. Errors (pot estar inclòs en cada apartat)

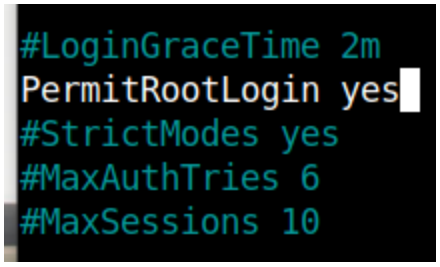
Error 1

Un error molt important és a l'hora de crear la clau privada, ja que ens pot donar error; hem de tenir molt en compte el nombre de bits. Sempre haurem de posar **4096, perquè si no podríem tenir errors.**

```
root@garridocliente:/etc/ssh# ssh-keygen -b 4096 -t rsa
```

Error 2

Per fer l'accés per clau privada/pública, haurem de tenir en compte que hem de tenir activat la directiva **PermitRootLogin** en mode **yes**.



Error 3

Un error típic per molt fàcil és, a l'hora de fer la connexió **ssh** haurem de posar el nom d'usuari correcte, per exemple **user.** Pot ser que el servidor no permet **root login** i si posem **root@192.168.*.*** no ens deixarà connectar.