

# Pt1d - Instal·lació i configuració de SSH

Pt1d - Instal·lació i configuració de SSH	1
<b>Parte 1 - SSH keys</b>	<b>2</b>
1. Generar clave ssh	3
2. Copiar la clave de la barra al servidor	4
3. Inicie sesión en el servidor, luego copie la clave en el archivo authorized_keys.	5
4. Cerrar sesión y luego intentar iniciar sesión	6
a.) Explica qué archivos tiene las claves públicas y cuál las privadas en servidor y cliente.	7
c.) Simula un ataque de “Man in the middle” y muestra el mensaje que nos da el cliente.	
	9
<b>Part 2 - Execució remota</b>	<b>19</b>
En el servidor:	20
ssh dama@192.168.128.200 -S sudo systemctl stop apache2	20
<b>Part 3 - SFTP</b>	<b>21</b>
1. Navega amb el nautilus pels fitxers del servidor mitjançant el protocol SFTP.	21
2. Utilitza la comanda "scp" per transferir arxius del host al servidor i viceversa.	22
3. Cerca la manera de pujar de cop una carpeta amb subcarpetes i arxius.	23
<b>Part 4 - Configuració del servidor</b>	<b>24</b>
4.1 Canvi de port	24
4.2 - Restriccions per usuari	26
3. Después de la modificación del fichero, reiniciamos el servicio ssh.	27
4. Ahora añadimos el usuario “james” al grupo de sudo.	28
5. Vamos a verificar si la configuración nueva funciona. - ssh -p 8000 james@192.168.1.64	28
4.3 - Restriccions per IP	29

# Parte 1 - SSH keys

## **Enunciado:**

Busca un artículo que explique cómo hacer SSH entre 2 máquinas sin necesidad de introducir contraseña. El cliente (imitando un PC de casa contra un servidor remoto) se conectará al servidor sin contraseña.

Pista: se recomienda la documentación oficial de Debian o Ubuntu, son muy claras y es un tema típico.

INFORME: lista todos los archivos y pedidos implicados y explica para qué sirve cada uno. En particular:

- a.) Explica qué archivos tiene las claves públicas y cuál las privadas en servidor y cliente.
- b.) Explica qué es el ataque "Man in the middle" y explica cuál de estos archivos nos ayuda para evitarlo y cómo lo hace.
- c.) Simula un ataque de Man in the middle y muestra el mensaje que nos da el cliente.

## 1. Generar clave ssh

En la máquina local, vamos a generar un par (público y privado) de llaves de SSH con el comando: **ssh-keygen -t rsa**

**ssh-keygen:** La herramienta que genera las llaves.

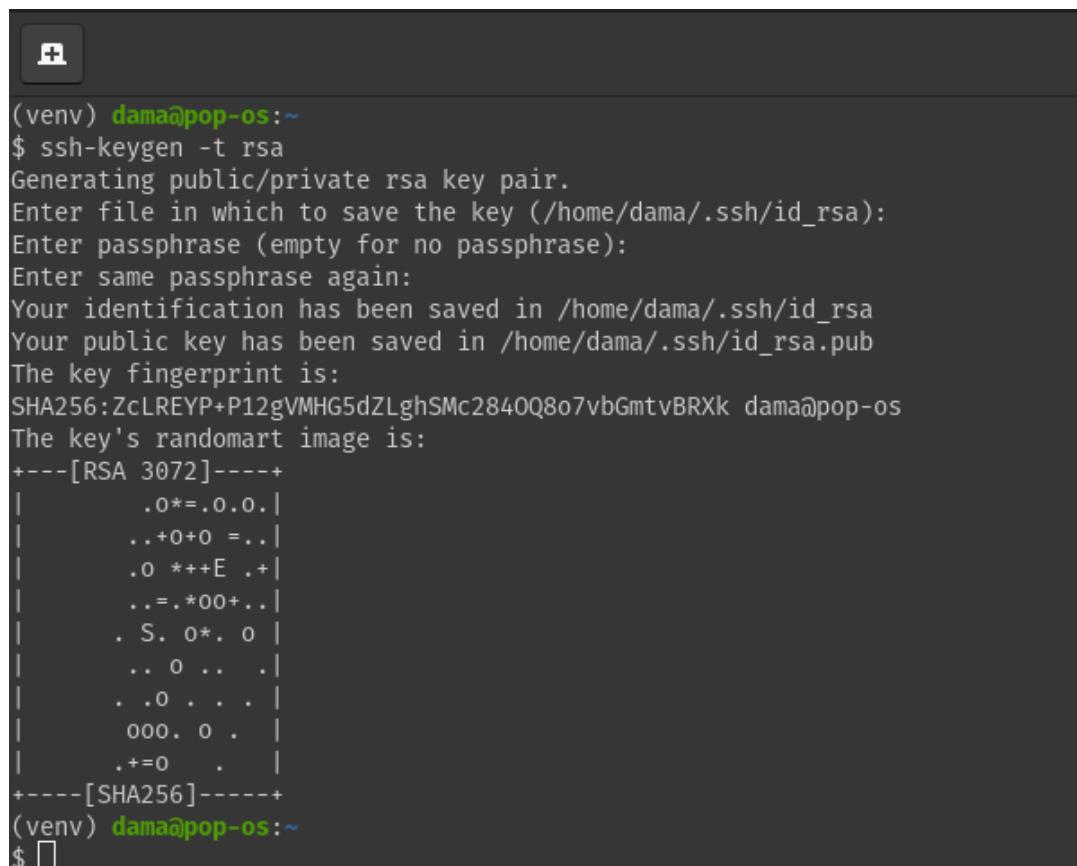
**-t:** Especifica el tipo de llave que queremos generar.

**rsa:** La especificación del tipo de encriptación de las llaves.

- Entramos las contraseñas que elegimos.
- Aceptamos los parámetros del fichero destino, que vienen por defecto.

Este proceso nos va a generar dos llaves exactamente:

- **id\_rsa:** Que es la clave privada. Eso va a quedar con nosotros en el local.
- **id\_rsa.pub:** Que es la clave pública, que vamos a enviar al destino remoto.



```
(venv) dama@pop-os:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/dama/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/dama/.ssh/id_rsa
Your public key has been saved in /home/dama/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ZcLREYP+P12gVMHG5dZLghSMc2840Q8o7vbGmtvBRXk dama@pop-os
The key's randomart image is:
+---[RSA 3072]---+
|       .0*.0.0.|
|     ..+0+0 =..|
|     .0 *++E .+|
|     ...*00+..|
|     . S. 0*. 0 |
|     .. 0 ... .|
|     . .0 . . .|
|     000. 0 . .|
|     .+=0      .|
+---[SHA256]---+
(venv) dama@pop-os:~$
```

## 2. Copiar la clave de la barra al servidor

Copiamos las llaves generadas a la máquina remota con el siguiente comando:  
**scp /home/dama/.ssh/id\_rsa.pub dama@192.168.120.200:/home/dama**

**scp:** La herramienta para copiar ficheros via ssh.

**/home/dama/.ssh/id\_rsa.pub:** La ruta absoluta de la llave **pública** de las llaves ssh en la máquina local.

**dama:** El usuario destinado en la máquina remota.

**@192.168.120.200:** Dirección IP de la máquina remota.

**:/home/dama:** La ruta remota absoluta a donde queremos enviar la llave pública.



The screenshot shows a terminal window titled "dama@pop-os: ~/.ssh". The command entered is "scp /home/dama/.ssh/id\_rsa.pub dama@192.168.120.200:/home/dama". The terminal shows the file transfer progress: "100% 565 875.2KB/s 00:00".

```
(venv) dama@pop-os: ~/.ssh
$ ll
total 24
drwxr-- 2 dama dama 4096 Oct 13 20:58 .
drwxr-xr-x 30 dama dama 4096 Oct 13 19:55 ..
-rw-r----- 1 dama dama 268 Oct 13 20:58 id_rsa
-rw-r--r-- 1 dama dama 588 Oct 13 20:58 id_rsa.pub
-rw-r----- 1 dama dama 1626 Oct 13 20:41 known_hosts
-rw-r----- 1 dama dama 790 Oct 13 20:41 known_hosts.old
(venv) dama@pop-os: ~/.ssh
$ scp /home/dama/.ssh/
/home/dama/.ssh/id_rsa          /home/dama/.ssh/id_rsa.pub      /home/dama/.ssh/known_hosts      /home/dama/.ssh/known_hosts.old
(venv) dama@pop-os: ~/.ssh
$ scp /home/dama/.ssh/id_rsa.pub dama@192.168.120.200:/home/dama
dama@192.168.120.200's password:
id_rsa.pub
(venv) dama@pop-os: ~/.ssh
$ [ ]
```

### 3. Inicie sesión en el servidor, luego copie la clave en el archivo authorized\_keys.

a., Con el comando **ssh dama@192.168.128.200** realizamos el login a la máquina remota.

b., Con el comando **cat id\_rsa.pub > .ssh/authorized\_keys** Redireccionamos el contenido de la llave publica (**id\_rsa.pub**) a fichero **authorized\_keys** que está en el directorio ocultado **.ssh**.

c., El comando **cat .ssh/authorized\_keys** nos enseña el contenido del fichero, así podemos chequear si el proceso ha sido realizado con éxito.

```
(venv) dama@pop-os:~/ssh
$ ssh dama@192.168.128.200
dama@192.168.128.200's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-50-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Oct 13 07:06:57 PM UTC 2022

System load:  0.0          Processes:           130
Usage of /:   18.4% of 14.38GB  Users logged in:        1
Memory usage: 2%            IPv4 address for enp0s3: 192.168.128.200
Swap usage:   0%

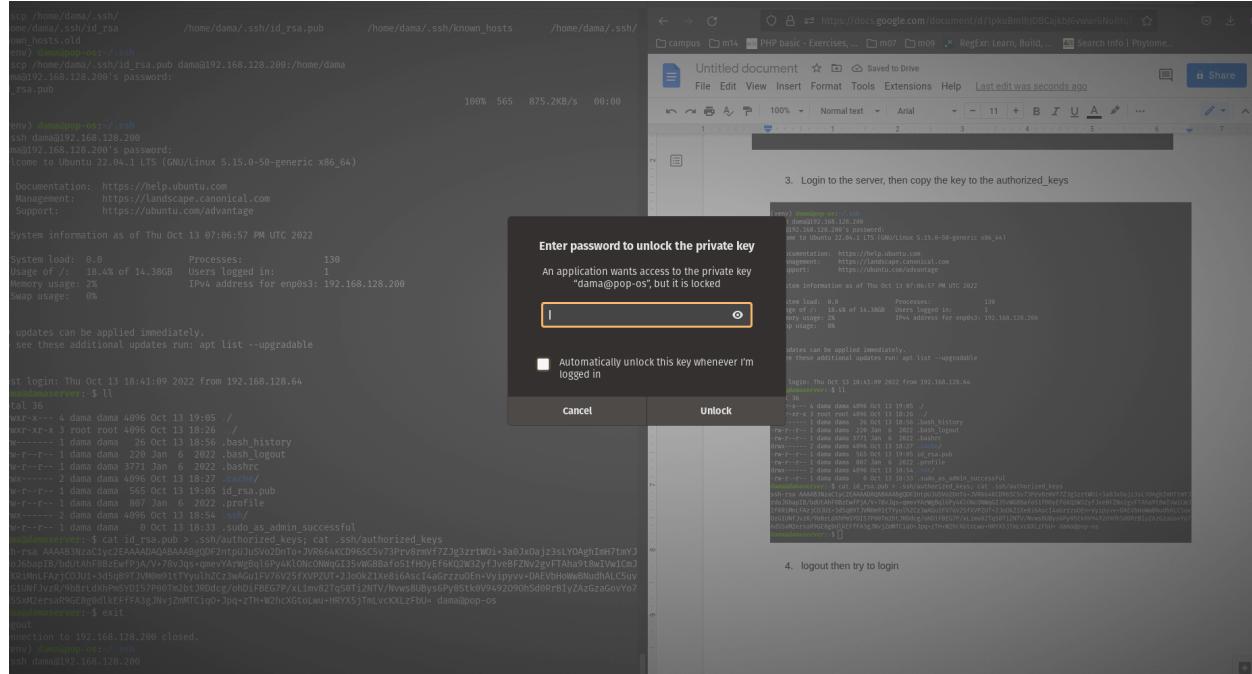
39 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Thu Oct 13 18:41:09 2022 from 192.168.128.64
dama@damaserver:~$ ll
total 36
drwxr-x--- 4 dama dama 4096 Oct 13 19:05 .
drwxr-xr-x  3 root root 4096 Oct 13 18:26 ..
-rw-----  1 dama dama   26 Oct 13 18:56 .bash_history
-rw-r--r--  1 dama dama  220 Jan  6 2022 .bash_logout
-rw-r--r--  1 dama dama 3771 Jan  6 2022 .bashrc
drwx----- 2 dama dama 4096 Oct 13 18:27 .cache/
-rw-r--r--  1 dama dama  565 Oct 13 19:05 id_rsa.pub
-rw-r--r--  1 dama dama  807 Jan  6 2022 .profile
drwx----- 2 dama dama 4096 Oct 13 18:54 .ssh/
-rw-r--r--  1 dama dama     0 Oct 13 18:33 .sudo_as_admin_successful
dama@damaserver:~$ cat id_rsa.pub > .ssh/authorized_keys; cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAABgQDF2ntpUJuSVo2DnTo+JVR664KCD96SC5v73Prv8rmVf7ZJg3zrtW0i+3a0Jx0ajz3sLYOAghImH7tmYJ
rd0J6bapIB/bdUtAhF8BzEwfPjA/V+78vJqs+qmevYArWgBql6Py4kL0nONwqG135vWGBBafo51fHoYEf6KQ2W3ZyfJveBFZnv2gvFTAh9t8wIVw1CmJ
2fKRiMnLFAzjCOJU1+3d5qB9TJVm0m91tTYu1hZCz3wAGu1FV76V25fxVPZUT+2JoOkZ1Xe8i6AscI4aGrzzuOEn+Vyipyvv+DAEVbHoWwBNudhALC5uv
0zG1UMfJvzR/9bBrLdxhPm5YDI57P00Tm2btJRDrv/ohDiFBEG7P/xL1mv82TqS0T1NTV/Nws8UBys6Py0Stk0V9492090hSd0RtBiYzAzGzaGovYo7
6d5Sxm2ersaR9GE8g0lKEFFFA3gJNvjZmMTCiQ0+Jpq+zTH+W2hcXGtoLwu+HRYX5jTmLvcKXLzFbU= dama@pop-os
dama@damaserver:~$ 
```

## 4. Cerrar sesión y luego intentar iniciar sesión

Hacemos el logout de la sesión de ssh con el comando exit (otra opción: <enter> ~.) y hacemos el login de nuevo.

Como lo podemos fijar, ya no nos pide la contraseña del usuario remoto, pero en su lugar, ya nos pide la contraseña de la clave ssh, que hemos generado en el punto 1).

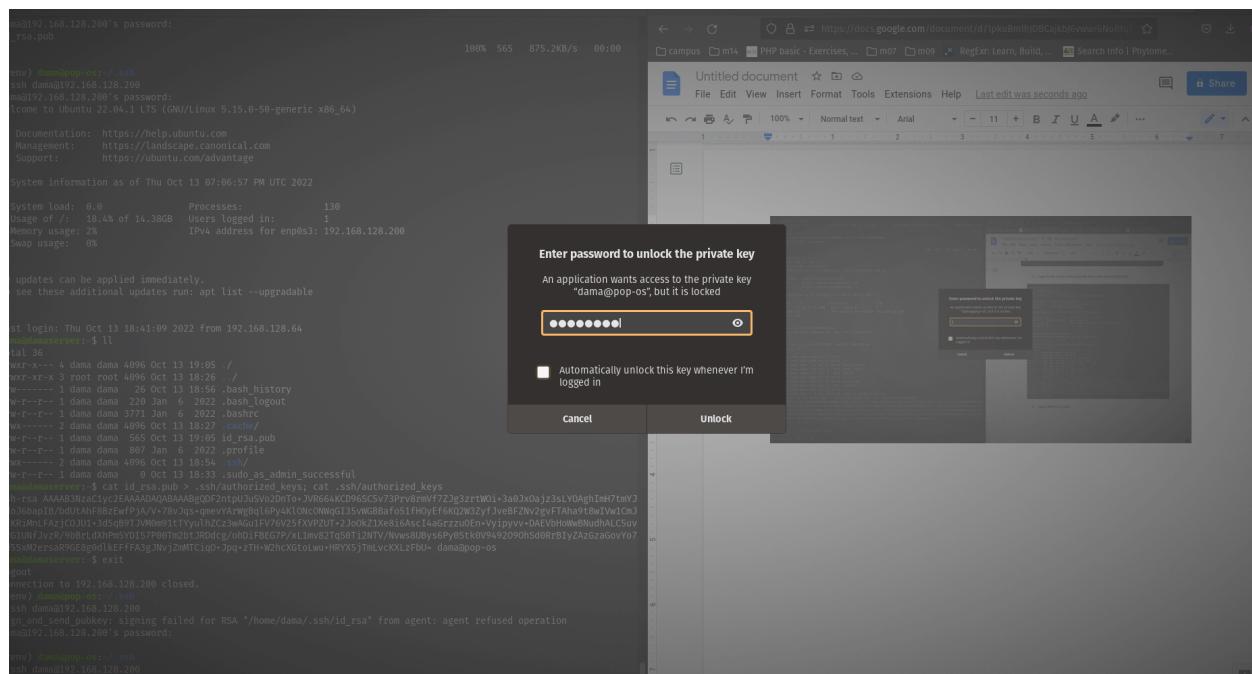


ssh /home/dama/.ssh/  
cdm\_dama/.ssh/id\_rsa.pub  
own files.old  
env) dama@pop-0s:~/ssh  
scp /home/dama/.ssh/id\_rsa.pub dama@192.168.128.200:/home/dama  
mag192.168.128.200's password:  
\_rsa.pub  
env) dama@pop-0s:~/ssh  
ssh dama@192.168.128.200  
mag192.168.128.200's password:  
!come to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-50-generic x86\_64)  
Documentation: https://help.ubuntu.com  
Management: https://landscape.canonical.com  
Support: https://ubuntu.com/advantage  
System information as of Thu Oct 13 07:06:57 PM UTC 2022  
System load: 0.0 Processes: 130  
Usage of /: 18.4% of 14.38GB Users logged in: 1  
Memory usage: 2% IPv4 address for enp0s3: 192.168.128.200  
Swap usage: 0%  
  
updates can be applied immediately.  
see these additional updates run: apt list --upgradable  
st login: Thu Oct 13 18:41:09 2022 from 192.168.128.64  
mag192.168.128.200:~\$ ll  
total 36  
wxr-x--- 4 dama dama 4096 Oct 13 19:05 /  
wxr-xr-x 3 root root 4096 Oct 13 18:26 ./  
w-r--r-- 1 dama dama 26 Oct 13 18:56 .bash\_history  
w-r--r-- 1 dama dama 220 Jan 6 2022 .bash\_logout  
w-r--r-- 1 dama dama 3771 Jan 6 2022 .bashrc  
wxr---- 2 dama dama 4096 Oct 13 18:27 .cache/  
w-r--r-- 1 dama dama 565 Oct 13 19:05 id\_rsa.pub  
w-r--r-- 1 dama dama 897 Jan 6 2022 .profile  
w-r--r-- 1 dama dama 4096 Oct 13 19:05 .ssh/  
w-r--r-- 1 dama dama 0 Oct 13 18:33 .sudo\_as\_admin\_successful  
mag192.168.128.200:~\$ cat id\_rsa.pub > .ssh/authorized\_keys cat .ssh/authorized\_keys  
n-rsa AAAA3HzxJcy1c2AAAADQABAAEoDfntUjSu20Dt+JNv664KC096SCv73Pv8vNmF723g3zrtW01:3a0Jx0ajz3sLYOAghIaH7mYJ  
oJebapTB/bduAH88ZewFpJA/V+7vJqs+meVwBq16p4k10mC0NwG135w6Bafo5ifhOyEfkoKwZ2yJv+eBZn/2gvFTAh9t8wVw1caJ  
KRJ1mFaJzrCOU1+j3dsq9tJWm09l1Tyu1hZc3wAgG1h7V62fXpZUT+2JooK2zXe16AscI4gczru0E+nVjpyvv+DAEVbHwM@luduALCSuv  
GIUNYJzvR/98rlrxNmHDY157P0Wtm2btJR0dcg/oNoIFEG7p/xLmrv82tq50t12N7//NwesBu8By56p85tk0v9492090h5d0Rb1yZaGzGoVYo7  
55aM2ersArGEG8gd1kFrffA3gJN)zMTMCiqD+Jpq+2tH-WzhCKGt0lWu+HRX5jTMLvCxKLzFBu+ dama@pop-0s  
mag192.168.128.200:~\$ . . .  
mag192.168.128.200:~\$ exit  
logout  
connection to 192.168.128.200 closed.  
env) dama@pop-0s:~/ssh  
ssh dama@192.168.128.200  
3. Login to the server, then copy the key to the authorized\_keys

Enter password to unlock the private key  
An application wants access to the private key "dama@pop-0s", but it is locked  
|  
 Automatically unlock this key whenever I'm logged in  
Cancel Unlock

4. logout then try to login

```
[env) dama@192.168.128.200:~$ ll  
total 36  
wxr-x--- 4 dama dama 4096 Oct 13 19:05 /  
wxr-xr-x 3 root root 4096 Oct 13 18:26 ./  
w-r--r-- 1 dama dama 26 Oct 13 18:56 .bash_history  
w-r--r-- 1 dama dama 220 Jan 6 2022 .bash_logout  
w-r--r-- 1 dama dama 3771 Jan 6 2022 .bashrc  
wxr---- 2 dama dama 4096 Oct 13 18:27 .cache/  
w-r--r-- 1 dama dama 565 Oct 13 19:05 id_rsa.pub  
w-r--r-- 1 dama dama 897 Jan 6 2022 .profile  
w-r--r-- 1 dama dama 4096 Oct 13 19:05 .ssh/  
w-r--r-- 1 dama dama 0 Oct 13 18:33 .sudo_as_admin_successful  
mag192.168.128.200:~$ cat id_rsa.pub > .ssh/authorized_keys cat .ssh/authorized_keys  
n-rsa AAAA3HzxJcy1c2AAAADQABAAEoDfntUjSu20Dt+JNv664KC096SCv73Pv8vNmF723g3zrtW01:3a0Jx0ajz3sLYOAghIaH7mYJ  
oJebapTB/bduAH88ZewFpJA/V+7vJqs+meVwBq16p4k10mC0NwG135w6Bafo5ifhOyEfkoKwZ2yJv+eBZn/2gvFTAh9t8wVw1caJ  
KRJ1mFaJzrCOU1+j3dsq9tJWm09l1Tyu1hZc3wAgG1h7V62fXpZUT+2JooK2zXe16AscI4gczru0E+nVjpyvv+DAEVbHwM@luduALCSuv  
GIUNYJzvR/98rlrxNmHDY157P0Wtm2btJR0dcg/oNoIFEG7p/xLmrv82tq50t12N7//NwesBu8By56p85tk0v9492090h5d0Rb1yZaGzGoVYo7  
55aM2ersArGEG8gd1kFrffA3gJN)zMTMCiqD+Jpq+2tH-WzhCKGt0lWu+HRX5jTMLvCxKLzFBu+ dama@pop-0s  
mag192.168.128.200:~$ . . .  
mag192.168.128.200:~$ exit  
logout  
connection to 192.168.128.200 closed.  
env) dama@pop-0s:~/ssh  
ssh dama@192.168.128.200  
4. logout then try to login
```



mag192.168.128.200:~\$ password:  
\_rsa.pub  
env) dama@pop-0s:~/ssh  
ssh dama@192.168.128.200  
mag192.168.128.200:~\$ password:  
!come to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-50-generic x86\_64)  
Documentation: https://help.ubuntu.com  
Management: https://landscape.canonical.com  
Support: https://ubuntu.com/advantage  
System information as of Thu Oct 13 07:06:57 PM UTC 2022  
System load: 0.0 Processes: 130  
Usage of /: 18.4% of 14.38GB Users logged in: 1  
Memory usage: 2% IPv4 address for enp0s3: 192.168.128.200  
Swap usage: 0%  
  
updates can be applied immediately.  
see these additional updates run: apt list --upgradable  
st login: Thu Oct 13 18:41:09 2022 from 192.168.128.64  
mag192.168.128.200:~\$ ll  
total 36  
wxr-x--- 4 dama dama 4096 Oct 13 19:05 /  
wxr-xr-x 3 root root 4096 Oct 13 18:26 ./  
w-r--r-- 1 dama dama 26 Oct 13 18:56 .bash\_history  
w-r--r-- 1 dama dama 220 Jan 6 2022 .bash\_logout  
w-r--r-- 1 dama dama 3771 Jan 6 2022 .bashrc  
wxr---- 2 dama dama 4096 Oct 13 18:27 .cache/  
w-r--r-- 1 dama dama 565 Oct 13 19:05 id\_rsa.pub  
w-r--r-- 1 dama dama 897 Jan 6 2022 .profile  
w-r--r-- 1 dama dama 4096 Oct 13 18:54 .ssh/  
w-r--r-- 1 dama dama 0 Oct 13 18:33 .sudo\_as\_admin\_successful  
mag192.168.128.200:~\$ cat id\_rsa.pub > .ssh/authorized\_keys cat .ssh/authorized\_keys  
n-rsa AAAA3HzxJcy1c2AAAADQABAAEoDfntUjSu20Dt+JNv664KC096SCv73Pv8vNmF723g3zrtW01:3a0Jx0ajz3sLYOAghIaH7mYJ  
oJebapTB/bduAH88ZewFpJA/V+7vJqs+meVwBq16p4k10mC0NwG135w6Bafo5ifhOyEfkoKwZ2yJv+eBZn/2gvFTAh9t8wVw1caJ  
KRJ1mFaJzrCOU1+j3dsq9tJWm09l1Tyu1hZc3wAgG1h7V62fXpZUT+2JooK2zXe16AscI4gczru0E+nVjpyvv+DAEVbHwM@luduALCSuv  
GIUNYJzvR/98rlrxNmHDY157P0Wtm2btJR0dcg/oNoIFEG7p/xLmrv82tq50t12N7//NwesBu8By56p85tk0v9492090h5d0Rb1yZaGzGoVYo7  
55aM2ersArGEG8gd1kFrffA3gJN)zMTMCiqD+Jpq+2tH-WzhCKGt0lWu+HRX5jTMLvCxKLzFBu+ dama@pop-0s  
mag192.168.128.200:~\$ . . .  
mag192.168.128.200:~\$ exit  
logout  
connection to 192.168.128.200 closed.  
env) dama@pop-0s:~/ssh  
ssh dama@192.168.128.200  
gn\_an\_send\_pubkey: signing failed for RSA "/home/dama/.ssh/id\_rsa" from agent: agent refused operation  
mag192.168.128.200:~\$ password:  
env) dama@pop-0s:~/ssh  
ssh dama@192.168.128.200

Enter password to unlock the private key  
An application wants access to the private key "dama@pop-0s", but it is locked  
|  
 Automatically unlock this key whenever I'm logged in  
Cancel Unlock

Entramos la contraseña y como podemos ver, funcionaba.

Utilizando llaves de ssh sube el nivel de seguridad de la conexión entre la máquina local y remota, ya que si alguien pilla la contraseña del usuario remoto, puede ganar permisos root y causar daños serios en el sistema.

```
Connection to 192.168.128.200 closed.  
(venv) dama@pop-os:~/ssh  
$ ssh dama@192.168.128.200  
sign_and_send_pubkey: signing failed for RSA "/home/dama/.ssh/id_rsa" from agent: agent refused operation  
dama@192.168.128.200's password:  
  
(venv) dama@pop-os:~/ssh  
$ ssh dama@192.168.128.200  
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-50-generic x86_64)  
  
 * Documentation: https://help.ubuntu.com  
 * Management: https://landscape.canonical.com  
 * Support: https://ubuntu.com/advantage  
  
 System information as of Thu Oct 13 07:12:40 PM UTC 2022  
  
 System load: 0.0 Processes: 130  
 Usage of /: 18.4% of 14.38GB Users logged in: 1  
 Memory usage: 2% IPv4 address for enp0s3: 192.168.128.200  
 Swap usage: 0%  
  
 39 updates can be applied immediately.  
 To see these additional updates run: apt list --upgradable  
  
Last login: Thu Oct 13 19:06:57 2022 from 192.168.128.64  
dama@damaserver:~$
```

a.) Explica qué archivos tiene las claves públicas y cuál las privadas en servidor y cliente.

En la máquina local (cliente) tenemos los siguientes archivos en el directorio “~/.ssh”:

```
(venv) dama@pop-os:~/ssh  
$ ll  
total 24  
drwx----- 2 dama dama 4096 Oct 13 20:58 ./  
drwxr-xr-x 31 dama dama 4096 Oct 18 20:49 ../  
-rw----- 1 dama dama 2655 Oct 13 20:58 id_rsa  
-rw-r--r-- 1 dama dama 565 Oct 13 20:58 id_rsa.pub  
-rw----- 1 dama dama 1768 Oct 16 19:09 known_hosts  
-rw----- 1 dama dama 790 Oct 13 20:41 known_hosts.old  
(venv) dama@pop-os:~/ssh
```

**id\_rsa:** Es una clave privada.

**id\_rsa.pub, known\_hosts y known\_hosts.old:** Son claves publicas.

En la máquina remota (servidor) tenemos los siguientes archivos en el directorio “~/ssh”:

```
Last login: Sun Oct 16 18:57:38 UTC 2022 from 192.168.1.102 on  
dama@damaserver:~$ cd .ssh  
dama@damaserver:~/ssh$ ll  
total 20  
drwx----- 2 dama dama 4096 Oct 13 18:54 ./  
drwxr-x--- 4 dama dama 4096 Oct 13 19:05 ../  
-rw----- 1 dama dama 565 Oct 13 19:08 authorized_keys  
-rw----- 1 dama dama 2655 Oct 13 18:54 id_rsa  
-rw-r--r-- 1 dama dama 569 Oct 13 18:54 id_rsa.pub  
dama@damaserver:~/ssh$ _
```

**id\_rsa:** Es una clave privada.

**id\_rsa.pub y authorized\_keys:** Son claves publicas.

Contenido truncado de unos de ellos:

```
(venv) dama@pop-os:~/ssh  
$ cat id_rsa  
----BEGIN OPENSSH PRIVATE KEY----  
b3B1bnNzaC1rZXktdjEAAAACmFlczIINi1jdHIAAAAGYmNyeXB0AAAAGAAAABDKgDfDqY  
iSSZ2Sk+IpxMGVAAAAEAAAAAEEAAAGXAAAAB3NzaC1yc2EAAAADAQABAAABgQDF2ntpUJuS  
Vo2DnTo+JVR664KCD96SC5v73Prv8rmVf7ZJg3zrtW0i+3a0Jx0ajz3sLYOaghImH7tmYJ
```

```
(venv) dama@pop-os:~/ssh  
$ cat id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAABgQDF2ntpUJuSVo2DnTo+JVR664KCD96SC5v73Prv8rmVf7ZJg3zrtW0i+3a0Jx0a  
jz3sLYOaghImH7tmYJrd0J6bapIB/bdUtAhF8BzEwfPjA/V+78vJqs+qmevYArWgBql6Py4KlONcONWqGI35vWGBBafo51fH0yEf  
6KQ2W3ZyfJveBFZNv2gvFTAh9t8wIVw1CmJ2fKRiMnLFAzjCOJU1+3d5qB9TJV0m91tTYyulhZCz3wAGu1FV76V25fXVPZUT+2  
1o0k71Xe8i6AsCT/2Cn7zvOEr+VuiowvvvDAEVbHwWwRNudhALCEuyOzC1UNf3vzP/9hRnLdyhDmEKDTcZP00Tm2bt+jPDdcg/obD
```

**b.) Explica qué es el ataque "Man in the middle" y explica cuál de estos archivos nos ayuda para evitarlo y cómo lo hace.**

Un ataque man-in-the-middle (MITM) es un término general para un atacante que se coloca en medio de una conversación entre un usuario y una aplicación, espiando o haciéndose pasar por una de las partes para que parezca un intercambio normal de se está produciendo la información.

El propósito del ataque es robar información personal, como información de inicio de sesión, información de cuenta y números de tarjetas de crédito.

En el protocolo SSH, el enfoque tradicional es usar una clave pública. La mayoría de los clientes SSH confían en la clave del servidor en la primera conexión, lo que teóricamente hace que un ataque de intermediario en la red en un momento dado sea poco probable, y para el césped proporciona la mejor compensación entre disponibilidad y seguridad. Instalación raíz.

En SSH, las claves de host protegen contra ataques humanos, pero deben administrarse correctamente, ser únicas para cada servidor y cambiarse periódicamente o cuando se vean comprometidas.

c.) Simula un ataque de “Man in the middle” y muestra el mensaje que nos da el cliente.

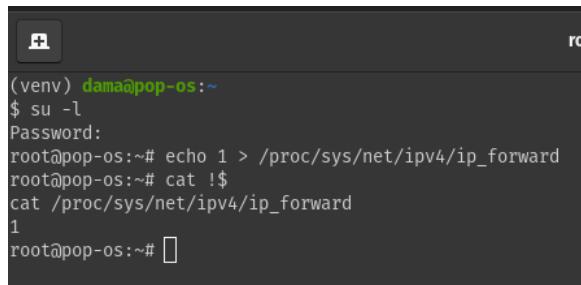
Para simular un ataque “**Man in the middle**”, necesitaremos los siguientes pasos previamente:

1. Montar una máquina virtual, que será el servidor. (Para el atacador y víctima, yo utilizo 2 portátiles en la misma red.)
2. En la máquina atacador instalar los siguientes paquetes:
  - **sudo apt install dsniff** (que tiene la herramienta “**arp spoof**”, que nos permite realizar el “**envenenamiento por arp**” en el servidor y víctima. )
  - **sudo apt install wireshark** (una herramienta con GUI, para capturar el tráfico de la máquina atacadora.)
3. En linux por defecto no tenemos activado el “port forwarding”, que nos permite el reenvío de los paquetes recibidos a la máquina.

Con el siguiente comando podemos activar esto (el **sudo** no será suficiente permiso, así tenemos que realizar el cambio como **root**):

**su -l:** Entrar como root.

**echo 1 > /proc/sys/net/ipv4/ip\_forward**

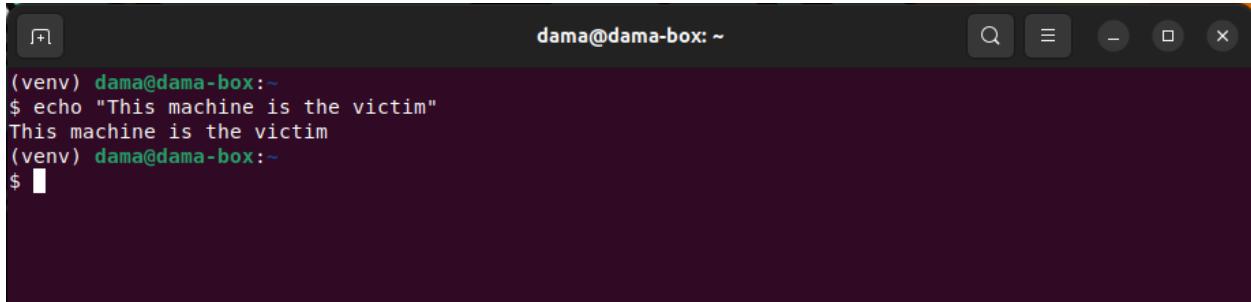


```
(venv) dama@pop-os:~$ su -l
Password:
root@pop-os:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@pop-os:~# cat !$
cat /proc/sys/net/ipv4/ip_forward
1
root@pop-os:~# 
```

4. Obtener los IP del víctima, atacador, servidor y saber como se llama el interfaz red de la máquina atacadora.

Ahora vamos a realizar el ataque “**Man in the middle**”:

**La máquina víctima:**



```
dama@dama-box:~
```

```
(venv) dama@dama-box:~
```

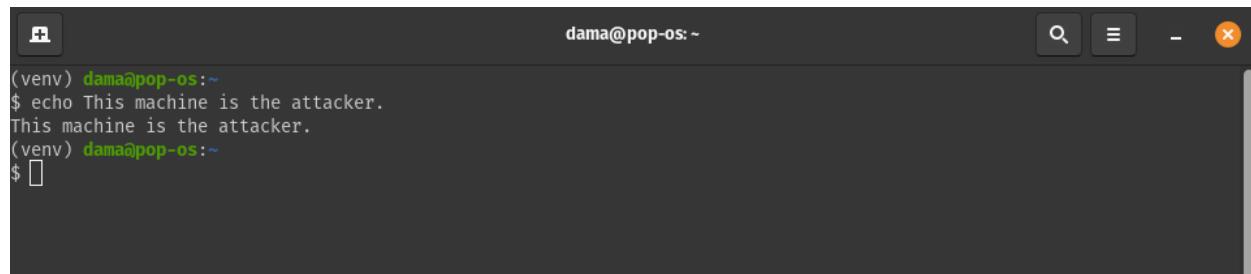
```
$ echo "This machine is the victim"
```

```
This machine is the victim
```

```
(venv) dama@dama-box:~
```

```
$ 
```

**La máquina atacador:**



```
dama@pop-os:~
```

```
(venv) dama@pop-os:~
```

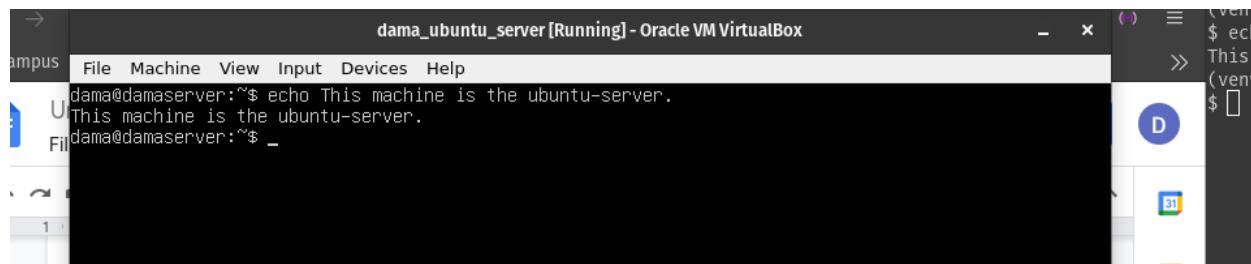
```
$ echo This machine is the attacker.
```

```
This machine is the attacker.
```

```
(venv) dama@pop-os:~
```

```
$ 
```

**El servidor ubuntu a donde vamos a conectar via SSH desde la máquina víctima:**



```
dama@damaserver:~$ echo This machine is the ubuntu-server.
```

```
This machine is the ubuntu-server.
```

```
dama@damaserver:~$ 
```

Los siguientes paquetes que necesitamos instalar en la máquina atacante:

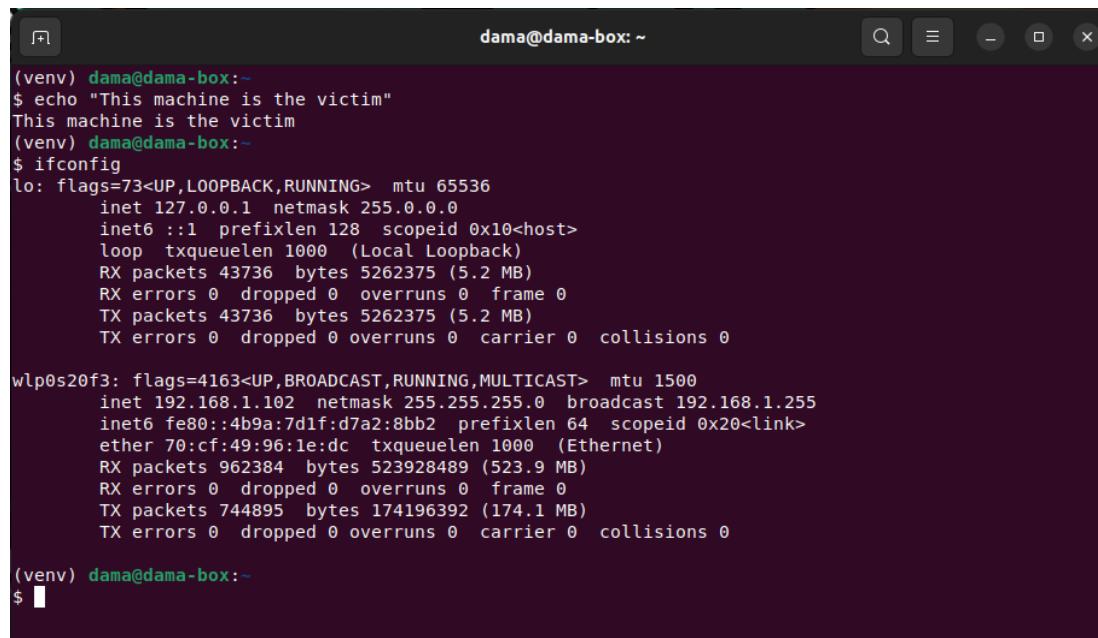
- **sudo apt install wireshark dsniff**

```
E: Unable to locate package dsniff
(venv) dama@pop-os:~
$ sudo apt install wireshark dsniff
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  dsniff wireshark
0 upgraded, 2 newly installed, 0 to remove and 244 not upgraded.
Need to get 111 kB of archives.
After this operation, 515 kB of additional disk space will be used.
Get:1 http://apt.pop-os.org/ubuntu jammy/universe amd64 wireshark amd64 3.6.2-2 [4,992 B]
Get:2 http://apt.pop-os.org/ubuntu jammy/universe amd64 dsniff amd64 2.4b1+debian-30build1 [106 kB]
Fetched 111 kB in 1s (116 kB/s)
Selecting previously unselected package wireshark.
(Reading database ... 220982 files and directories currently installed.)
Preparing to unpack .../wireshark_3.6.2-2_amd64.deb ...
Unpacking wireshark (3.6.2-2) ...
Selecting previously unselected package dsniff.
Preparing to unpack .../dsniff_2.4b1+debian-30build1_amd64.deb ...
Unpacking dsniff (2.4b1+debian-30build1) ...
Setting up wireshark (3.6.2-2) ...
Setting up dsniff (2.4b1+debian-30build1) ...
Processing triggers for man-db (2.10.2-1) ...
(venv) dama@pop-os:~
$ 
```

Ahora necesitamos recopilar la información necesaria:

En este caso solo necesitamos:

- la ip de la víctima
- la ip del servidor
- la ip del atacante
- puerto predeterminado NO NECESARIO.
- la interfaz utilizada por el atacante para conectarse a la red



The screenshot shows a terminal window titled "dama@dama-box:~". The user has run the command "ifconfig" to view the network interfaces. The output shows two interfaces: "lo" (loopback) and "wlp0s20f3" (wireless). The "lo" interface has an IP of 127.0.0.1. The "wlp0s20f3" interface has an IP of 192.168.1.102 and is connected to an Ethernet adapter with MAC address 70:cf:49:96:1e:dc.

```
(venv) dama@dama-box:~
$ echo "This machine is the victim"
This machine is the victim
(venv) dama@dama-box:~
$ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
            loop  txqueuelen 1000  (Local Loopback)
            RX packets 43736  bytes 5262375 (5.2 MB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 43736  bytes 5262375 (5.2 MB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.102  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::4b9a:7d1f:d7a2:8bb2  prefixlen 64  scopeid 0x20<link>
            ether 70:cf:49:96:1e:dc  txqueuelen 1000  (Ethernet)
            RX packets 962384  bytes 523928489 (523.9 MB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 744895  bytes 174196392 (174.1 MB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

(venv) dama@dama-box:~
$ 
```

```
dama@damaserver:$ echo This machine is the ubuntu-server.  
This machine is the ubuntu-server.  
dama@damaserver:$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
        inet 192.168.1.64 brd 192.168.1.255 netmask 255.255.255.0 broadcast 192.168.1.255  
              inet6 fe80::a00:27ff:fe26:9a25 brd fe80::ff:fe26:9a25 prefixlen 64 scopeid 0x20<link>  
        ether 08:00:27:26:9a:25 txqueuelen 1000 (Ethernet)  
          RX packets 97745 bytes 143596972 (143.5 MB)  
          RX errors 0 dropped 17 overruns 0 frame 0  
          TX packets 9867 bytes 731610 (731.6 KB)  
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
        inet 127.0.0.1 brd 127.0.0.1 netmask 255.0.0.0  
              inet6 ::1 brd ::1 prefixlen 128 scopeid 0x10<host>  
        loop txqueuelen 1000 (Local Loopback)  
          RX packets 132 bytes 10905 (10.9 KB)  
          RX errors 0 dropped 0 overruns 0 frame 0  
          TX packets 132 bytes 10905 (10.9 KB)  
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
dama@damaserver:~$
```

Dirección IP de la víctima: 192.168.1.102

Dirección IP del atacante: 192.168.1.122

Dirección IP del servidor Ubuntu: 192.168.1.64

¡Ahora tenemos que engañar a la víctima y al servidor ubuntu para que piensen que la máquina atacante es la otra usando el **ENVENENAMIENTO ARP con arpspoof!**

Dígale al servidor de ubuntu que la máquina de ataque es la máquina de la víctima.  
Y haz lo mismo a la inversa con la máquina de la víctima:

- sudo -s arpspoof -i wlp0s20f3 -t 192.168.1.64 192.168.1.102

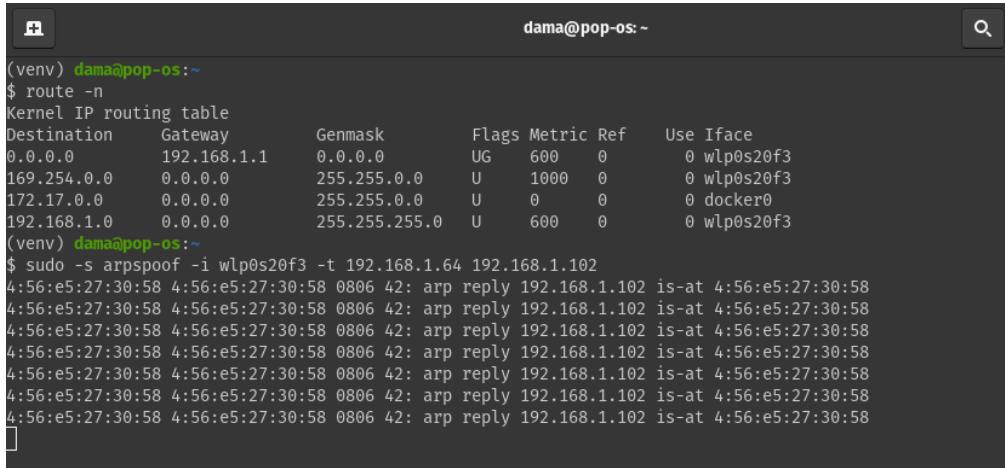
**sudo -s:** Ejecute el shell especificado por la variable de entorno SHELL si está configurado o el shell especificado por la entrada de la base de datos de contraseñas del usuario invocador.

**arpspoof:** La herramienta para envenenar las máquinas.

**-i wlp0s20f3:** Especificar el id de interfaz red de la máquina de atacador.

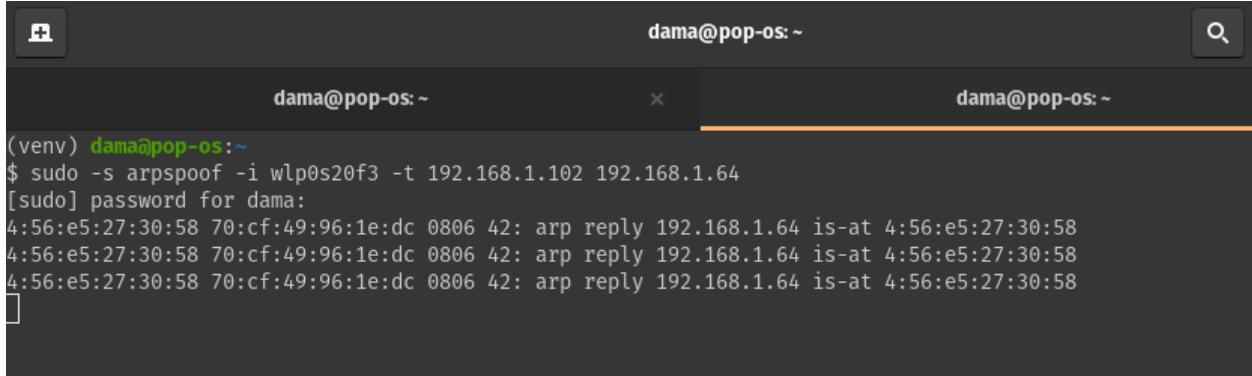
**-t 192.168.1.64 192.168.1.102:** Especificar que 1º IP piensa que nosotros somos el 2º IP.

Envenenar el servidor que piensa que nosotros somos la víctima( IP: 192.168.1.102).



```
(venv) dama@pop-os:~$ route -n
Kernel IP routing table
Destination     Gateway      Genmask      Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1  0.0.0.0      UG    600    0        0 wlp0s20f3
169.254.0.0     0.0.0.0     255.255.0.0   U     1000   0        0 wlp0s20f3
172.17.0.0      0.0.0.0     255.255.0.0   U     0       0        0 docker0
192.168.1.0     0.0.0.0     255.255.255.0  U     600    0        0 wlp0s20f3
(venv) dama@pop-os:~$ sudo -s arpspoof -i wlp0s20f3 -t 192.168.1.64 192.168.1.102
4:56:e5:27:30:58 4:56:e5:27:30:58 0806 42: arp reply 192.168.1.102 is-at 4:56:e5:27:30:58
4:56:e5:27:30:58 4:56:e5:27:30:58 0806 42: arp reply 192.168.1.102 is-at 4:56:e5:27:30:58
4:56:e5:27:30:58 4:56:e5:27:30:58 0806 42: arp reply 192.168.1.102 is-at 4:56:e5:27:30:58
4:56:e5:27:30:58 4:56:e5:27:30:58 0806 42: arp reply 192.168.1.102 is-at 4:56:e5:27:30:58
4:56:e5:27:30:58 4:56:e5:27:30:58 0806 42: arp reply 192.168.1.102 is-at 4:56:e5:27:30:58
4:56:e5:27:30:58 4:56:e5:27:30:58 0806 42: arp reply 192.168.1.102 is-at 4:56:e5:27:30:58
4:56:e5:27:30:58 4:56:e5:27:30:58 0806 42: arp reply 192.168.1.102 is-at 4:56:e5:27:30:58
```

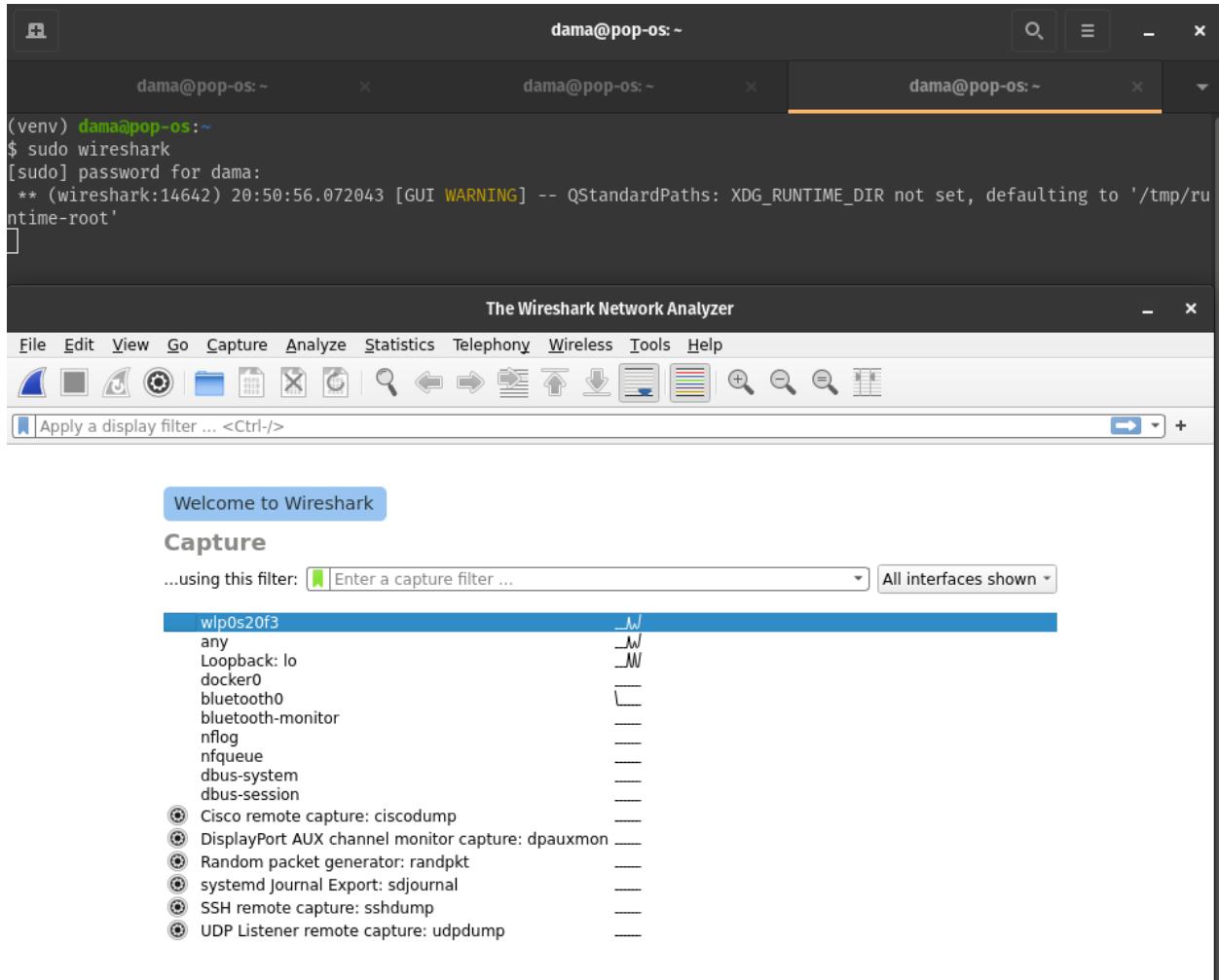
Envenenar la víctima que nosotros somos el servidor (IP: 192.168.1.64)



```
(venv) dama@pop-os:~$ sudo -s arpspoof -i wlp0s20f3 -t 192.168.1.102 192.168.1.64
[sudo] password for dama:
4:56:e5:27:30:58 70:cf:49:96:1e:dc 0806 42: arp reply 192.168.1.64 is-at 4:56:e5:27:30:58
4:56:e5:27:30:58 70:cf:49:96:1e:dc 0806 42: arp reply 192.168.1.64 is-at 4:56:e5:27:30:58
4:56:e5:27:30:58 70:cf:49:96:1e:dc 0806 42: arp reply 192.168.1.64 is-at 4:56:e5:27:30:58
```

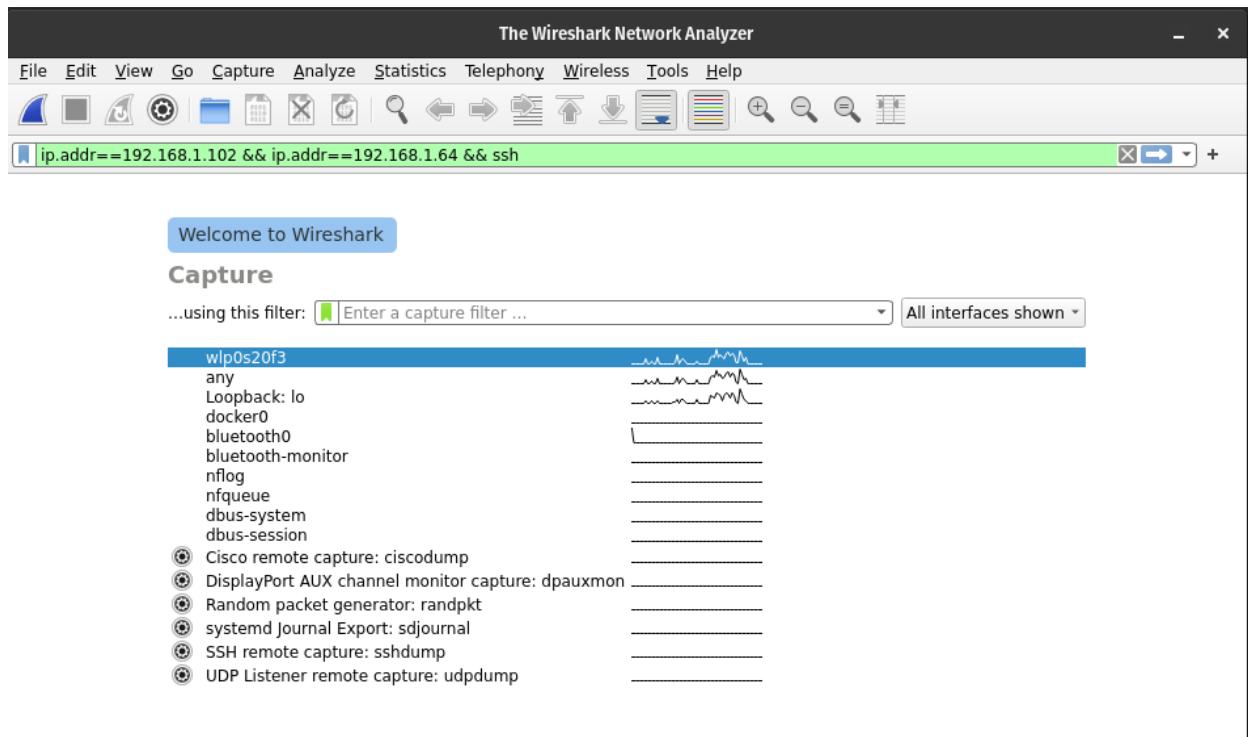
Ahora activamos wireshark (dejar corriendo el **arpspoof** ejecutándose en la máquina atacante en todo el tiempo del ataque.):

- **sudo wireshark** (necesitamos **sudo**, para tener permiso utilizar la interfaz red.)

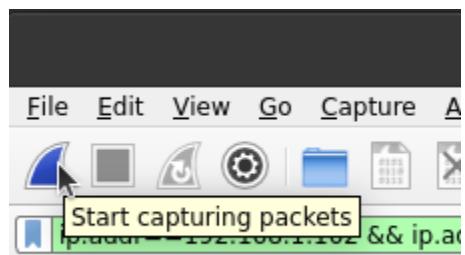


Ahora pasemos los parámetros del filtro para filtrar lo que necesitamos:

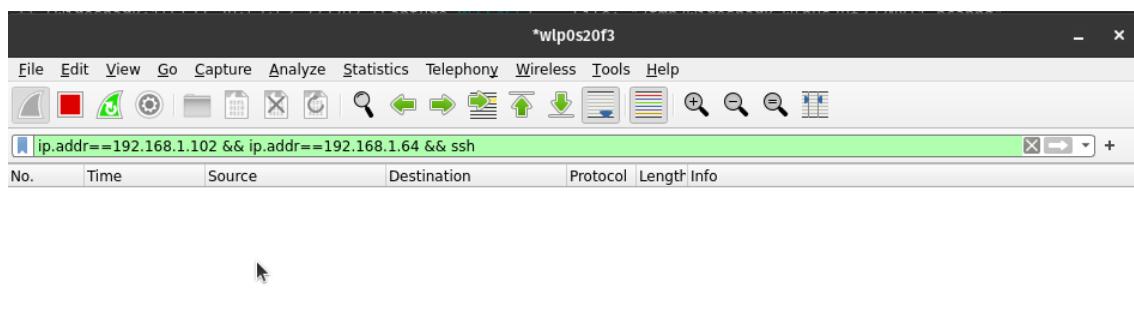
- **ip.add == 192.168.1.102 && ip.addr==192.168.1.64 && ssh**
- Solo queremos ver las capturas de los tráficos de la **victima y servidor** que ha sido realizado con el protocolo **ssh**.



Presione enter, luego comenzamos la sesión de captura haciendo clic en la pequeña cosa de la aleta de tiburón:



Como podemos ver, todavía no pasa nada porque todavía no hemos iniciado ninguna sesión SSH entre la víctima y el servidor.

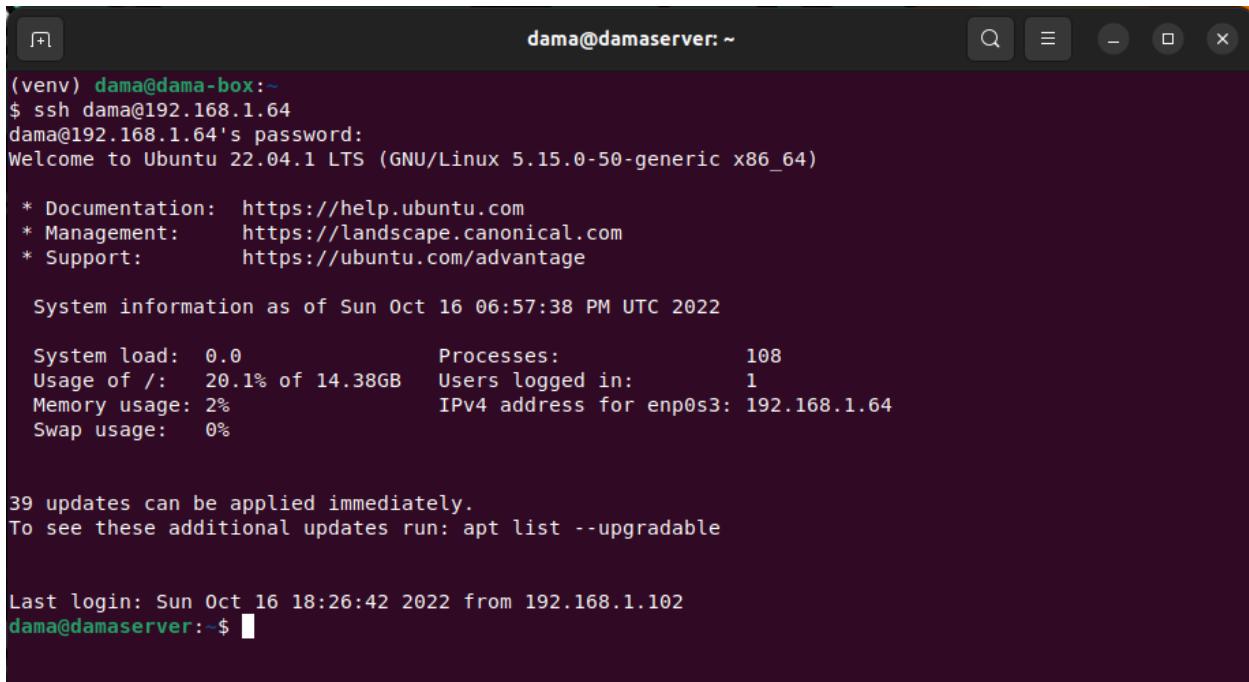


SSH desde la máquina víctima al servidor:

**ssh**: La herramienta de conexión.

**dama**: El usuario destinado en la máquina remota.

**@192.168.120.200**: Dirección IP de la máquina remota.



```
(venv) dama@dama-box:~$ ssh dama@192.168.1.64
dama@192.168.1.64's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-50-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

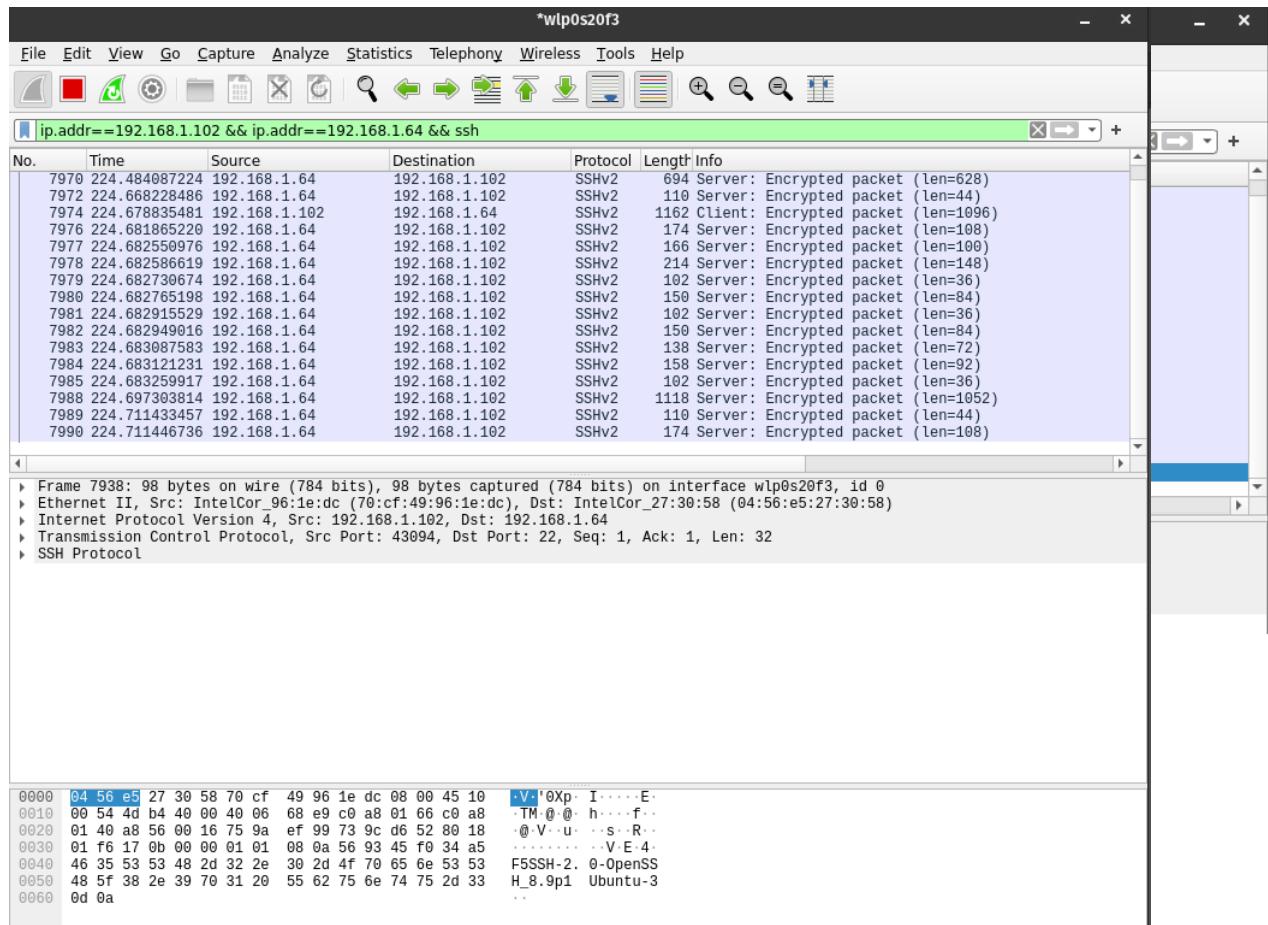
System information as of Sun Oct 16 06:57:38 PM UTC 2022

System load:  0.0          Processes:           108
Usage of /:   20.1% of 14.38GB  Users logged in:      1
Memory usage: 2%          IPv4 address for enp0s3: 192.168.1.64
Swap usage:   0%

39 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

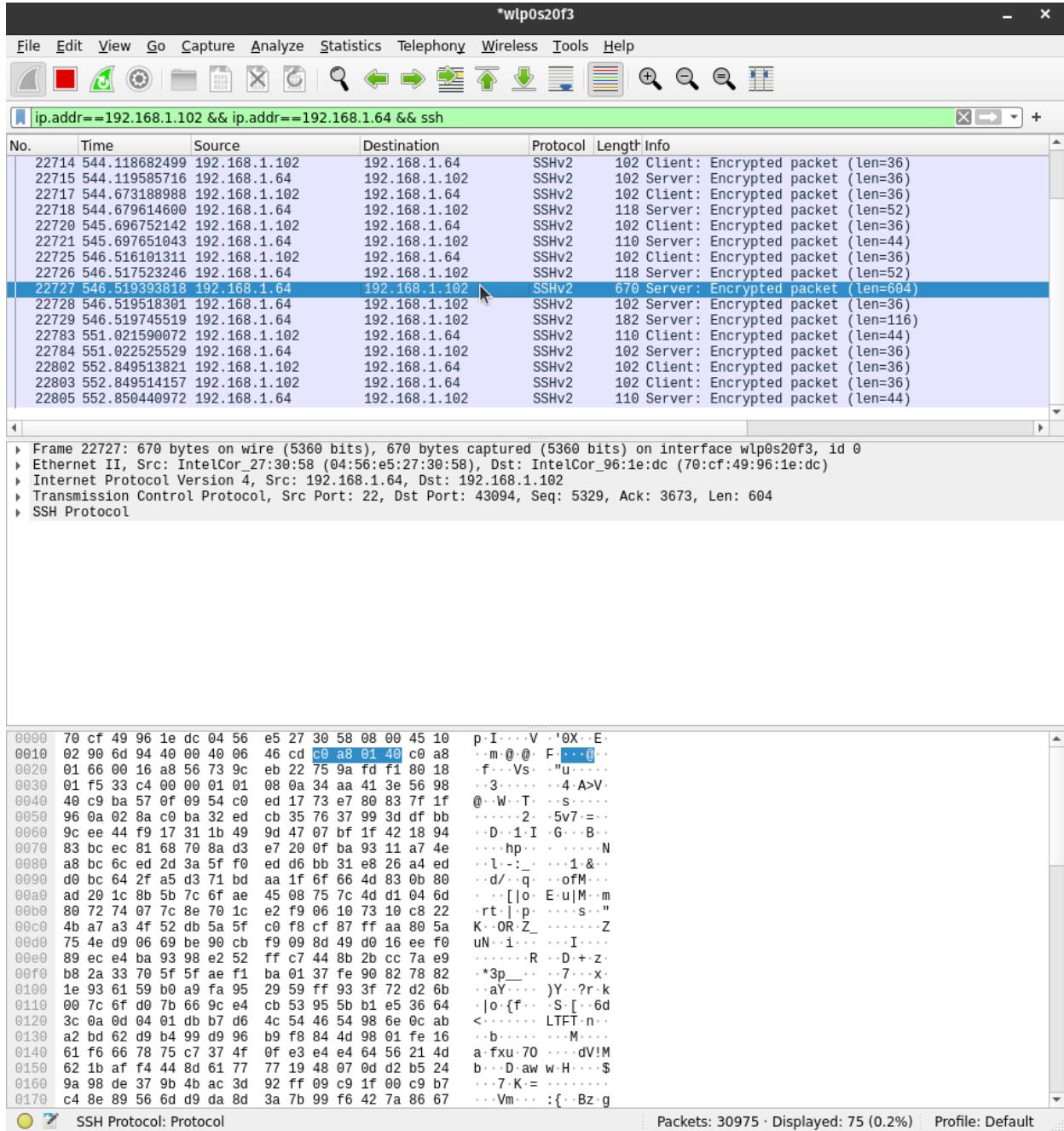
Last login: Sun Oct 16 18:26:42 2022 from 192.168.1.102
dama@damaserver:~$
```

Como podemos ver, Wireshark capturó inmediatamente el tráfico entre las dos direcciones IP (víctima y servidor). Debido a que los registros de las grabaciones se mueven rápidamente, seleccionará el último como "ancla" y luego ejecutar un comando en la máquina de la víctima para ver cómo se ve el paquete (spoiler: está encriptado, por lo que el texto será un galimatías.).



Como "podemos ver", cada vez que presionamos una tecla en la sesión del terminal ssh, lo envía como un paquete, con una longitud entre 30 ~ 60 len, pero cuando es una salida, el tamaño del paquete es mucho mayor.

Por favor, vea la grabación seleccionada en la siguiente captura de pantalla:



### Como conclusión de esta parte de la pregunta:

El cifrado es una de las partes más importantes de cualquier red. Para mantener nuestra privacidad, secretos y evitar el desorden por completo. (Por cierto, el cifrado nunca es una protección del 100 %. No confíes en las máquinas ni en las personas).

## Part 2 - Execució remota

- Busca la manera d'executar comandes remotament sense entrar en mode interactiu. És a dir, has de poder entrar en el servidor, executar la comanda i sortir, escrivint una sola comanda des del client.
- Busca la manera d'executar comandes de superusuari (root) remotament sense entrar contrasenya.
- Per exemple: engegar o parar un servidor web.

1. Utilizamos el mismo comando que usamos para conectar, pero extendemos con el comando entre comillas.

Por ejemplo:

`ssh dama@192.168.128.200 'ls -la'`

```
(venv) dama@pop-os:~$ ssh dama@192.168.128.200 'ls -la'
total 44
drwxr-x--- 4 dama dama 4096 Oct 20 18:08 .
drwxr-xr-x 3 root root 4096 Oct 13 18:26 ..
-rw----- 1 dama dama 390 Oct 20 18:09 .bash_history
-rw-r--r-- 1 dama dama 220 Jan  6 2022 .bash_logout
-rw-r--r-- 1 dama dama 3771 Jan  6 2022 .bashrc
drwx----- 2 dama dama 4096 Oct 13 18:27 .cache
-rw-r--r-- 1 dama dama 565 Oct 13 19:05 id_rsa.pub
-rw-r--r-- 1 dama dama 807 Jan  6 2022 .profile
-rwxr-xr-x 1 dama dama 32 Oct 20 18:08 remote_script.sh
drwx----- 2 dama dama 4096 Oct 13 18:54 .ssh
-rw-r--r-- 1 dama dama    0 Oct 13 18:33 .sudo_as_admin_successful
-rw----- 1 dama dama 845 Oct 20 18:08 .viminfo
(venv) dama@pop-os:~$ 
```

Estos ficheros listados no son de la máquina local, pero son de la máquina remota.

2. Ejecutamos el comando con sudo, pero añadimos un flag, de “-S”.

Ejemplo:

**ssh dama@192.168.128.200 -S sudo mkdir test\_dir**

```
(venv) dama@pop-os:~$ ssh dama@192.168.128.200 -S sudo mkdir test_dir
(venv) dama@pop-os:~$ 
[...]
-rw-r--r-- 1 dama dama 32 Oct 20 18:08 remote_script.sh*
drwx----- 2 dama dama 4096 Oct 13 18:54 .ssh/
-rw-r--r-- 1 dama dama 0 Oct 13 18:33 .sudo_as_admin_successful
drwxrwxr-x 2 dama dama 4096 Oct 20 18:18 test_dir/
-rw----- 1 dama dama 845 Oct 20 18:08 .viminfo
dama@damaserver:~$
```

En el servidor:

**ssh dama@192.168.128.200 -S sudo systemctl stop apache2**

```
(venv) dama@pop-os:~$ ssh dama@192.168.128.200 -S sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2022-10-20 18:29:20 UTC; 1min 40s ago
    Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2085 (apache2)
     Tasks: 55 (limit: 12361)
    Memory: 5.0M
       CPU: 30ms
      CGroup: /system.slice/apache2.service
              ├─2085 /usr/sbin/apache2 -k start
              ├─2087 /usr/sbin/apache2 -k start
              ├─2088 /usr/sbin/apache2 -k start

Oct 20 18:29:20 damaserver systemd[1]: Starting The Apache HTTP Server...
Oct 20 18:29:20 damaserver apachectl[2084]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 192.168.128.200. Set the 'ServerName' directive globally to suppress this message
Oct 20 18:29:20 damaserver systemd[1]: Started The Apache HTTP Server.
(venv) dama@pop-os:~$ ssh dama@192.168.128.200 -S sudo systemctl stop apache2
Failed to stop apache2.service: Interactive authentication required.
See system logs and 'systemctl status apache2.service' for details.
(venv) dama@pop-os:~$ 
$ 
```

**Bueno, el comando funcionaría si tengo configurado correctamente el apache2.**

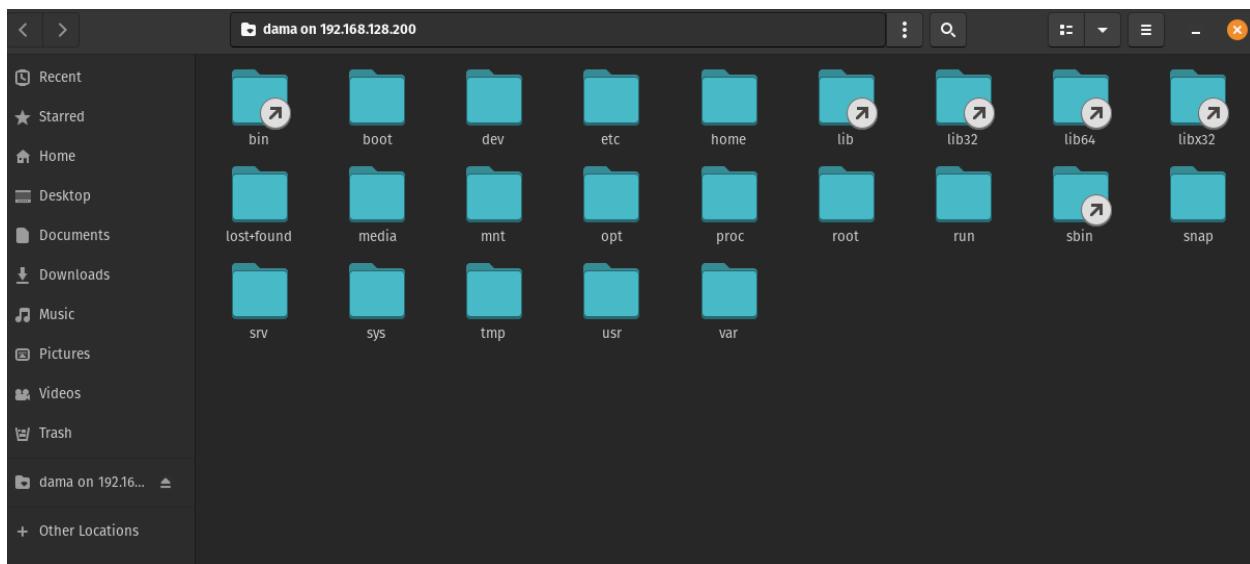
## Part 3 - SFTP

El SSH també serveix per fer transferència segura de fitxers. Realitza:

1. Navega amb el nautilus pels fitxers del servidor mitjançant el protocol SFTP.

Comando utilizado:

- nautilus sftp://dama@192.168.128.200



2. Utilitza la comanda "scp" per transferir arxius del host al servidor i viceversa.

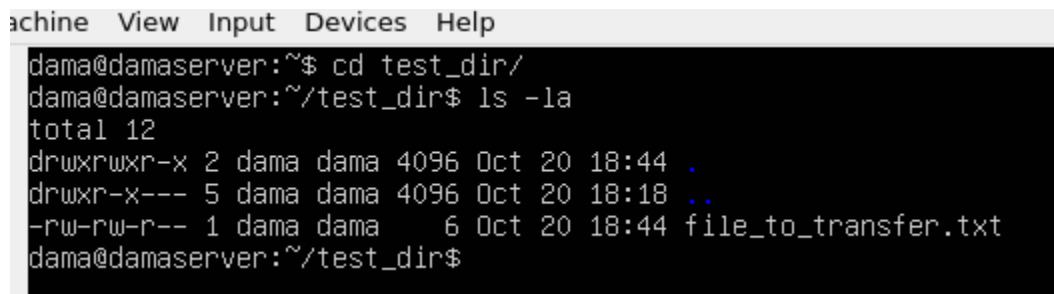
Comando utilizado:

- **scp file\_to\_transfer.txt dama@192.168.128.200:/home/dama/test\_dir**



```
(venv) dama@pop-os:~$ scp file_to_transfer.txt dama@192.168.128.200:/home/dama/test_dir
file_to_transfer.txt                                         100%   6    13.7KB/s  00:00
(venv) dama@pop-os:~$
```

Comprobar si el fichero ha sido transferido al servidor.



```
Machine View Input Devices Help
dama@damaserver:~/test_dir$ ls -la
total 12
drwxrwxr-x 2 dama dama 4096 Oct 20 18:44 .
drwxr-x--- 5 dama dama 4096 Oct 20 18:18 ..
-rw-rw-r-- 1 dama dama     6 Oct 20 18:44 file_to_transfer.txt
dama@damaserver:~/test_dir$
```

Ahora pedir un fichero del servidor desde cliente hacia al cliente.



```
(venv) dama@pop-os:~$ scp dama@192.168.128.200:/home/dama/test_dir/protid2taxid.txt.gz /home/dama
protid2taxid.txt.gz                                         100% 104MB 214.4MB/s  00:00
(venv) dama@pop-os:~$
```

### 3. Cerca la manera de pujar de cop una carpeta amb subcarpetes i arxius.

```
scp -r received_files dama@192.168.128.200:/home/dama/test_dir
```

Con el flag “-r (recursividad)” indicamos primero el directorio que queremos mandar y después y después el usuario + el ip ./“la ruta absoluta del destino”.

```
(venv) dama@pop-os:~$ scp -r
::1:          Documents/      localhost:      pop-os.localdomain:  Templates/
anaconda3/    Downloads/     m14_pt1/        protid2taxid.txt.gz  Videos/
dama@192.168.128.200  file_to_transfer.txt  Music/          Public/        VirtualBox\ VMs/
dawbio2/      intelephense/  Pictures/       public_html/    virtualenvs/
Desktop/      js_exercises/  pop-os:        received_files/
(venv) dama@pop-os:~$ scp -r received_files dama@192.168.128.200:/home/dama/test_dir
file_for_client.txt                                         100%   6   14.3KB/s  00:00
(venv) dama@pop-os:~$
```

```
file_for_client.txt                                         100%   b
dama@damaserver:~/test_dir$ ll
total 20
drwxrwxr-x 3 dama dama 4096 Oct 20 19:00 .
drwxr-x--- 5 dama dama 4096 Oct 20 19:00 ..
-rw-rw-r-- 1 dama dama   6 Oct 20 18:46 file_for_client.txt
-rw-rw-r-- 1 dama dama   6 Oct 20 18:44 file_to_transfer.txt
drwxrwxr-x 2 dama dama 4096 Oct 20 19:00 received_files/
dama@damaserver:~/test_dir$
```

# Part 4 - Configuració del servidor

## 4.1 Canvi de port

Els servidors reben molts atacs i el port 22 és el primer de la llista. Canvia el port del servidor al nº 1022.

En el fichero `/etc/ssh/sshd_config` cambiar el “Port 22” a “Port 8000”. Y después ejecutamos el comando “`systemctl restart sshd`”, para reiniciar el servicio.

```
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 8000
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none
```

```
#strictmodes yes
#MaxAuthTries 6
"/etc/ssh/sshd_config" 123L, 3265B written
dama@damaserver:~/test_dir$ systemctl restart sshd
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to restart 'ssh.service'.
Authenticating as: daniel_majer (dama)
Password:
==== AUTHENTICATION COMPLETE ===
dama@damaserver:~/test_dir$
```

- Quina comanda has de fer servir ara per connectar-te?

Con el siguiente comando: **ssh dama@192.168.128.200 -p 8000**

El flag de “-p” nos permite indicar el numero de puerto que queremos utilizar como punto de conexión (puerto 8000).

```
$ ssh dama@192.168.128.200 -p 8000
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-50-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Thu Oct 20 07:09:16 PM UTC 2022

 System load:  0.0          Processes:           113
 Usage of /:   20.5% of 14.38GB  Users logged in:      1
 Memory usage: 2%          IPv4 address for enp0s3: 192.168.128.200
 Swap usage:   0%

39 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Thu Oct 20 18:05:47 2022 from 192.168.128.122
dama@damaserver:~$ 
```

- Com faràs ara un SCP?

**Con el siguiente comando:**

- **scp -P 8000 home/dama/protid2taxid.txt.gz  
dama@192.168.128.200:/home/dama/test\_dir**

Indicando con el flag “-p” el puerto 8000 como punto de conexión y primero indicar la ruta del fichero que queremos mandar y después el usuario + el ip ./“la ruta absoluta del destino”.

```
(venv) dama@pop-os:~
$ scp -P 8000 protid2taxid.txt.gz dama@192.168.128.200:/home/dama/test_dir
protid2taxid.txt.gz                                         100% 104MB 124.7MB/s  00:00
(venv) dama@pop-os:~
$ 
```

## 4.2 - Restriccions per usuari

Restringeix el servidor per tal que l'usuari "funky" NO es pugui connectar però sí l'usuari "james" (caldrà crear-lo si no existeix).

Dóna permisos de superusuari a "james" posant-lo al grup *sudo* (en versions anteriors d'Ubuntu és el grup *admin*).

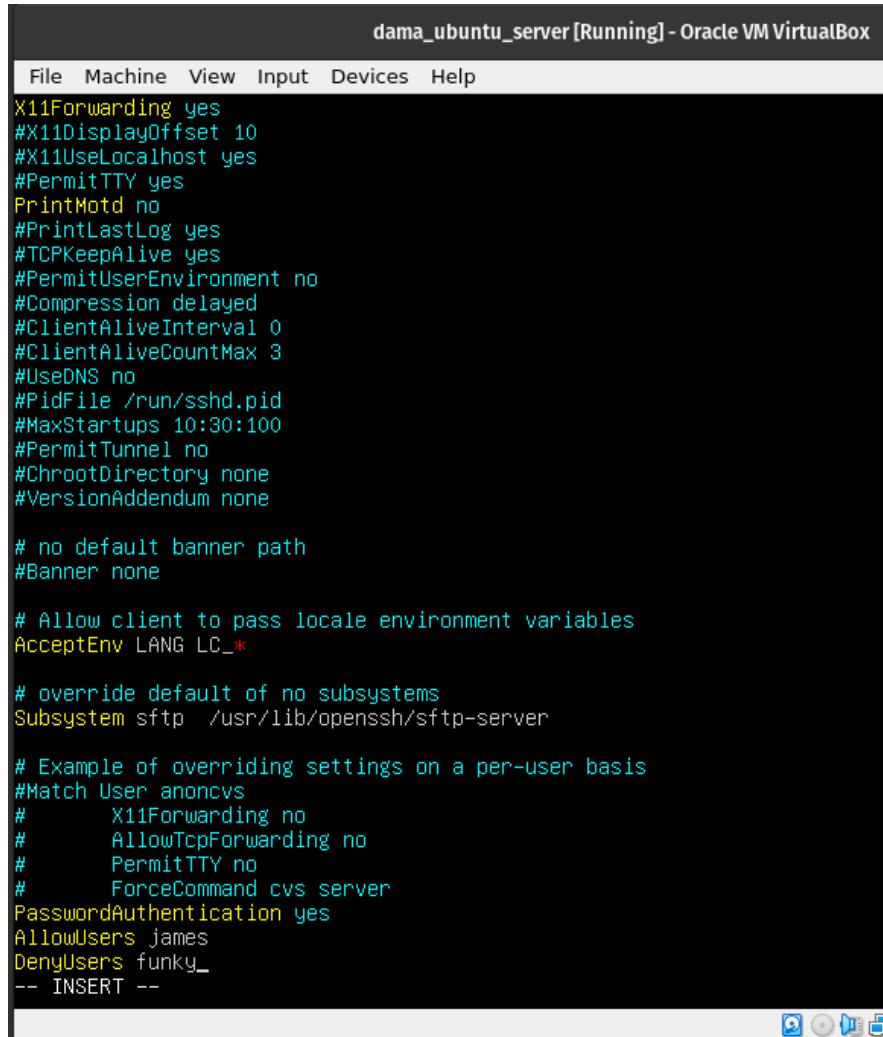
1. [Primero creamos los usuarios en el servidor:](#)

- **sudo adduser james**
- **sudo adduser funky**

```
dama_ubuntu_server [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
dama@damaserver:~$ sudo adduser james  
Adding user `james' ...  
Adding new group `james' (1001) ...  
Adding new user `james' (1001) with group `james' ...  
Creating home directory `/home/james' ...  
Copying files from `/etc/skel' ...  
New password:  
Retype new password:  
passwd: password updated successfully  
Changing the user information for james  
Enter the new value, or press ENTER for the default  
      Full Name []:  
      Room Number []:  
      Work Phone []:  
      Home Phone []:  
      Other []:  
Is the information correct? [Y/n] y  
dama@damaserver:~$ sudo adduser funky  
Adding user `funky' ...  
Adding new group `funky' (1002) ...  
Adding new user `funky' (1002) with group `funky' ...  
Creating home directory `/home/funky' ...  
Copying files from `/etc/skel' ...  
New password:  
Retype new password:  
passwd: password updated successfully  
Changing the user information for funky  
Enter the new value, or press ENTER for the default  
      Full Name []:  
      Room Number []:  
      Work Phone []:  
      Home Phone []:  
      Other []:  
Is the information correct? [Y/n] y  
dama@damaserver:~$ _  
s
```

2. En el /etc/ssh/sshd\_config fichero añadimos los siguientes parámetros de configuración:

- **AllowUsers** james
- **DenyUsers** funky



```
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
PrintMotd no
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
PasswordAuthentication yes
AllowUsers james
DenyUsers funky_
-- INSERT --
```

3. Despu s de la modificación del fichero, reiniciamos el servicio ssh.

```
DenyUsers funky
"/etc/ssh/sshd_config" 125L, 3301B written
dama@damaserver:~$ sudo systemctl restart sshd
dama@damaserver:~$
```

#### 4. Ahora añadimos el usuario “james” al grupo de sudo.

- **sudo usermod -aG sudo james**

```
dama_ubuntu_server [Running] - Oracle VM V

File Machine View Input Devices Help
ama@damaserver:/etc$ sudo usermod -aG sudo james
ama@damaserver:/etc$ groups james
ames : james sudo
ama@damaserver:/etc$ _
```

#### 5. Vamos a verificar si la configuración nueva funciona.

- **ssh -p 8000 james@192.168.1.64**

Con el usuario “james” hemos podido conectarnos.

```
(venv) dama@pop-os:~
$ ssh -p 8000 james@192.168.1.64
james@192.168.1.64's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-50-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Thu Oct 20 09:50:17 PM UTC 2022

 System load:  0.0                  Processes:           125
 Usage of /:   21.5% of 14.38GB   Users logged in:    1
 Memory usage: 5%                 IPv4 address for enp0s3: 192.168.1.64
 Swap usage:   0%

39 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

/usr/bin/xauth:  file /home/james/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

james@damaserver:~$ 
```

- `ssh -p 8000 funky@192.168.1.64`

Pero con el usuario “funky”, ya no nos deja.

```
james@damaserver:~$ exit
logout
Connection to 192.168.1.64 closed.
(venv) dama@pop-os:~
$ ssh -p 8000 funky@192.168.1.64
funky@192.168.1.64's password:
Permission denied, please try again.
funky@192.168.1.64's password:
Permission denied, please try again.
funky@192.168.1.64's password: □
```

## 4.3 - Restriccions per IP

Restringeix el servidor per tal que les màquines de la xarxa interna NO es puguin connectar. En [aquest link](#) indica diverses maneres de assegurar les connexions SSH. Utilitzeu DenyHosts per restringir per les IPs.

Si no tenies xarxa interna, crea la interfície i configura el client correctament i comprova que fa pings però no connecta per SSH i sí ho deixa fer des de la màquina amfitriona.

[Antes restringir todos los IP de la red interna, vamos a comprobar si aun podemos acceder con el usuario “james”.](#)

- `ssh -p 8000 james@192.168.128.200 (Indicando el puerto 8000 con “-p 8000”)`

```
(venv) dama@pop-os:~
$ ssh -p 8000 james@192.168.128.200
james@192.168.128.200's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-50-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Mon Oct 24 02:50:45 PM UTC 2022

 System load:  0.1123046875      Processes:           139
 Usage of /:   21.5% of 14.38GB   Users logged in:       1
 Memory usage: 2%                  IPv4 address for enp0s3: 192.168.128.200
 Swap usage:   0%

 39 updates can be applied immediately.
 To see these additional updates run: apt list --upgradable

Last login: Thu Oct 20 21:50:18 2022 from 192.168.1.102
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

james@damaserver:~$ □
```

Ahora vamos a modificar de nuevo el fichero `/etc/hosts.deny`, añadiendo la siguiente linea:

- **sshd: 192.168.128.\*** (Con el \* wildcard indicamos todos los hosts en la red interna.)
  - Después reiniciamos el servicio ssh: **sudo systemctl restart ssh**

Después de la modificación intentamos de nuevo hacer el login vía ssh al usuario "james".

- ssh -p 8000 james@192.168.128.200

```
(venv) dama@pop-os:~$ ssh -p 8000 james@192.168.128.200
kex_exchange_identification: read: Connection reset by peer
Connection reset by 192.168.128.200 port 8000
(venv) dama@pop-os:~$ ssh -p 8000 james@192.168.128.200
kex_exchange_identification: read: Connection reset by peer
Connection reset by 192.168.128.200 port 8000
(venv) dama@pop-os:~$
```

Como podemos ver el servidor ya no nos deja entrar desde la red interna.