Instituto Politécnico Nacional
Escuela Superior de Cómputo

PSA1. Servidor de terminal remota (SSH)

Grupo: 7CV1

Alumnos:

- **Díaz Ortiz Brandon Aldair**
- **García Cárdenas Ángel Alberto**
- **Gutiérrez Pérez Lizbeth Alejandra**
- **Nava Izquierdo César**

Unidad de Aprendizaje:

- **ADMINISTRACIÓN DE SERVICIOS EN RED**

Profesor:

- **MANUEL ALEJANDRO SOTO RAMOS**

Fecha:27/10/2024

Índice

Introducción

En base a lo aprendido en la práctica uno individual, es hora de unirse en equipo para poder desarrollar la primera práctica 1 Servidor de terminal remota (SSH) en la cual se deben cumplir los siguientes requisitos.

1. Funcionalidad requerida
    1. Conexión terminal
    2. Conexión Entorno gráfico
    3. Copia de elementos remotos al servidor (Gráfica y terminal)
    4. Tunel de aplicaciones por SSH (3 diferentes elementos)
2. Herramienta de consulta y filtrado de
    1. Resultados de conexión
    2. Operación de los usuarios en el servidor SSH

Elementos teóricos para la realización de la práctica, como se indica desde un principio
El protocolo de red conocido como SSH, cuyo nombre proviene de "Secure Shell", posibilita la comunicación de forma segura.
La comunicación segura entre dos sistemas a través de una red insegura es un desafío importante en el ámbito de la seguridad informática. El protocolo SSH es ampliamente empleado con fines de comunicación segura en entornos informáticos.
Acceder de forma segura a servidores remotos y gestionar su administración. Además de la conexión, existen otros factores a considerar.
Además de facilitar la ejecución de comandos remotos, SSH posibilita también la transferencia de archivos y la creación de túneles.
El protocolo de red SSH, posibilita la comunicación segura entre sistemas a través de redes consideradas inseguras. Esta tecnología se utiliza para establecer conexiones seguras, transferir archivos de forma segura y realizar otras formas de comunicación segura entre sistemas. El protocolo SSH se emplea ampliamente para acceder de forma segura a servidores remotos con el fin de gestionarlos. SSH, además de posibilitar la conexión mediante línea de comandos, facilita la transferencia de archivos, la tunelización de aplicaciones y otros tipos de comunicaciones seguras entre sistemas.

Desarrollo

Para el desarrollo de esta práctica se utilizó como servidor el Sistema Operativo Ubuntu 22.04 LTS instalado en una partición del computador, así mismo, para el cliente se utilizó

el Sistema Operativo Debian 12 implementado en una máquina virtual mediante la herramienta Oracle VM VirtualBox Versión 6.1.50.
Se consideró la rúbrica de criterios de evaluación publicada en la plataforma, con motivo de medir el avance obtenido.

a) Conexión terminal remota usando consola PAM:

Por la parte del servidor:

1. Actualizar los paquetes de Ubuntu e instalación OpenSSH:

```
equipo@equipo-G3-3500:~$ sudo apt update
[sudo] contraseña para equipo:
Obj:1 https://dl.google.com/linux/chrome/deb stable InRelease
Obj:2 http://mx.archive.ubuntu.com/ubuntu jammy InRelease
Obj:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Obj:4 http://mx.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:5 https://download.virtualbox.org/virtualbox/debian jammy InRelease
Obj:6 http://mx.archive.ubuntu.com/ubuntu jammy-backports InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se puede actualizar 1 paquete. Ejecute «apt list --upgradable» para verlo.
W: https://download.virtualbox.org/virtualbox/debian/dists/jammy/InRelease: Key is stored in legacy
rusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
equipo@equipo-G3-3500:~$ sudo apt install openssh-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  ncurses-term openssh-sftp-server ssh-import-id
Paquetes sugeridos:
  molly-guard monkeysphere ssh-askpass
```

2. A continuación, si todo ha salido bien, se debe activar ssh mediante el comando sudo systemctl start ssh. Podrá comprobar que el servidor se ejecuta correctamente con el comando sudo systemctl status ssh, aquí la consola deberá imprimir la leyenda “active (running)...”, esto confirma que el servidor SSH está activado correctamente.

```
equipo@equipo-G3-3500:~$ sudo systemctl start ssh
equipo@equipo-G3-3500:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-10-26 18:44:34 CST; 2s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 28627 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 28628 (sshd)
      Tasks: 1 (limit: 18850)
     Memory: 1.7M
        CPU: 17ms
     CGroup: /system.slice/ssh.service
             └─28628 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

oct 26 18:44:34 equipo-G3-3500 systemd[1]: Starting OpenBSD Secure Shell server...
oct 26 18:44:34 equipo-G3-3500 sshd[28628]: Server listening on 0.0.0.0 port 22.
oct 26 18:44:34 equipo-G3-3500 sshd[28628]: Server listening on :: port 22.
oct 26 18:44:34 equipo-G3-3500 systemd[1]: Started OpenBSD Secure Shell server.
```

3. Ahora, ingresé al archivo de configuración SSH (/etc/ssh/sshd_config), mediante el comando sudo nano /etc/ssh/sshd_config. Es posible que sudo le pida confirmación de identidad, indicando que se requiere ingresar la contraseña del equipo.

```
equipo@equipo-G3-3500:~$ sudo nano /etc/ssh/sshd_config
[sudo] contraseña para equipo:
```

```
  GNU nano 6.2                          /etc/ssh/sshd_config
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/us

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

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

#Port 22
#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

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
```

```
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
                              [ 122 líneas leídas ]
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación  M-U Deshacer
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar      ^J Justificar ^/ Ir a línea M-E Rehacer
```

4. Una vez dentro del GNU, modifique la configuración actual del archivo SSH. A continuación, se detallan las características que deben ser modificadas, junto con la razón que motivó dicho cambio:

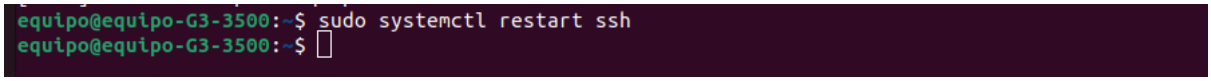
Asegurar la Autenticación por Contraseña (sin claves)

Dado que, aún no se tiene configurada la autenticación con claves, y se desea poder iniciar sesión con usuario y contraseña, usaremos la siguiente configuración: PasswordAuthentication yes.

Esto permitirá las conexiones terminales y entornos gráficos usando contraseña.

Autenticación con Claves Públicas
Ya que, posteriormente se usarán claves públicas, vamos a descomenta PubkeyAuthentication y asegurar de que esté en yes: PubkeyAuthentication yes.

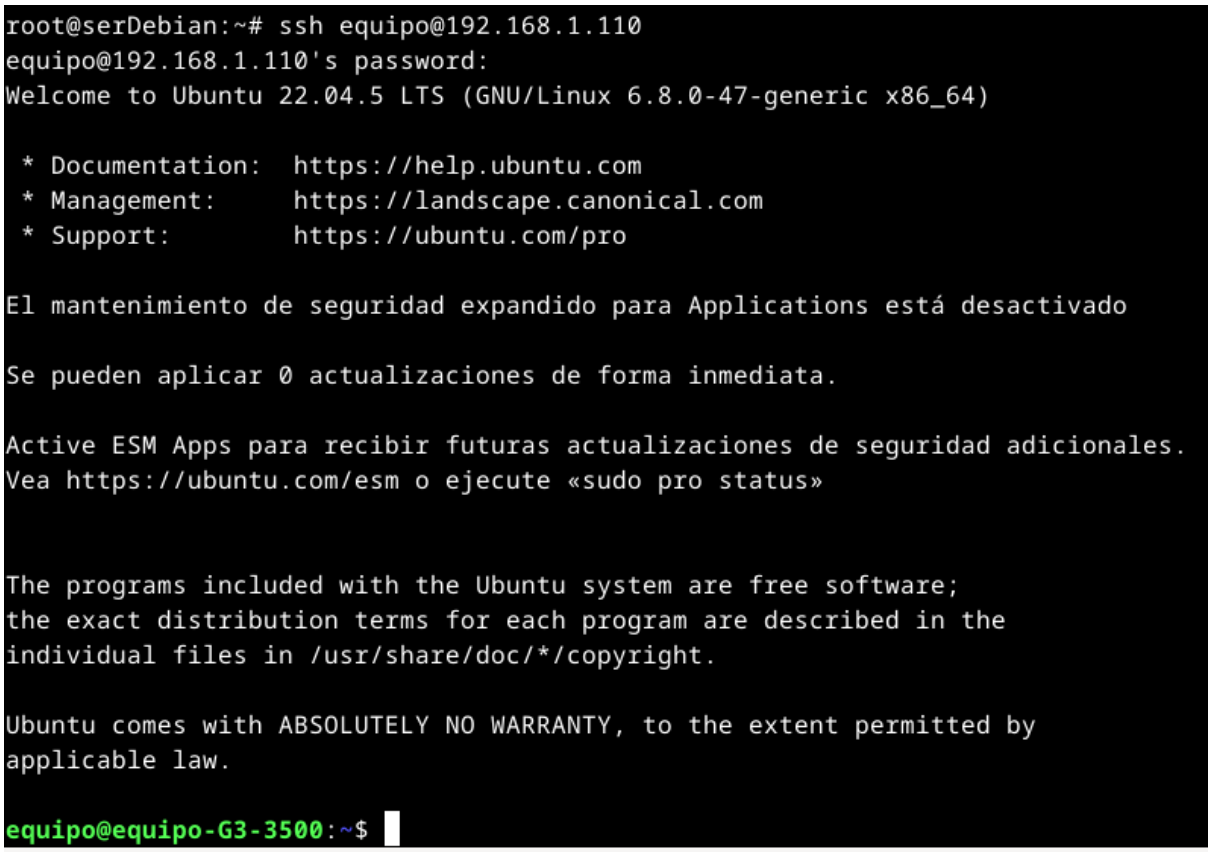
5. Después de realizar estos cambios, se reinicia el servicio SSH en el servidor para aplicar los cambios realizados, usamos el comando sudo systemctl restart ssh.



Por la parte del cliente:

1. Se inicia la máquina virtual con el Sistema Operativo Debian 12, posteriormente se inicia sesión en el mismo.

2. Abrir la consola de comando y escribir ssh nombre-usuario@id-server, en nuestro caso el comando será ssh equipo@192.168.1.110, posteriormente ingresamos la contraseña del servidor. Si todos los datos fueron ingresados correctamente se mostrará una pantalla como la siguiente:



b) Conexión terminal remota usando intercambio de llaves (sin password)

Los siguientes cambios únicamente serán ejecutados en la consola del cliente (consola de la máquina virtual Debian 12).

1. Genera un par de llaves SSH usando el comando: ssh-keygen -t rsa -b 4096

```
cerrar sesión
Connection to 192.168.1.110 closed.
root@serDebian:~# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
```

2. Acepta la ubicación por defecto presionando Enter. Si deseas, agrega una frase de seguridad para proteger la clave privada, en nuestro caso no se realizará.

```
Enter file in which to save the key (/root/.ssh/id_rsa):
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:Qxh+muCbNY17+zznkO+sS8Q8eet1a2ULuQcNVEkSr5c root@serDebian
The key's randomart image is:
+---[RSA 4096]----+
|       .      o+o.|
|      . o     .o. |
|     . o o   .  . |
|    . . B o . .. .|
|     . * S * ..+E |
|      + o o + =..o|
|     o . . + . =.+|
|        . +.=.o =.|
|         ..=B* o. |
+----[SHA256]-----+
```

3. Usando el comando ssh-copy-id equipo@192.168.1.110, se copiará, al servidor, el archivo de claves que hemos generado en el cliente.

```
|     + o o + =..o|
|    o . . + . =.+|
|       . +.=.o =.|
|        ..=B* o. |
+----[SHA256]-----+
root@serDebian:~# ssh-copy-id equipo@192.168.1.110
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa
.pub"
/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 prompt
ed now it is to install the new keys
```

4. SSH solicitará nuevamente la contraseña del servidor, otorgarla para realizar la copia sin problemas.

```
equipo@192.168.1.110's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'equipo@192.168.1.110'"
and check to make sure that only the key(s) you wanted were added.
```

5. Una vez que la copia se haya realizado exitosamente, se podrá realizar la conexión remota sin necesidad de usar la contraseña, únicamente usando el comando ssh equipo@192.168.1.110

```
root@serDebian:~# ssh equipo@192.168.1.110
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-47-generic x86_64)

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

El mantenimiento de seguridad expandido para Applications está desactivado

Se puede aplicar 1 actualización de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Oct 26 21:08:51 2024 from 192.168.1.110
equipo@equipo-G3-3500:~$
```

Creación de una jaula

Las jaulas o "chroot" en SSH son una forma de restringir el acceso de los usuarios a información sensible en un sistema. Las jaulas (chroot) son útiles para crear y mantener una copia virtual del sistema operativo en un directorio. Esto puede ser útil para pruebas y desarrollo, por ejemplo, para crear un área de prueba para un programa.

1. Crear el directorio para la jaula.

```
equipo@equipo-G3-3500:~$ sudo mkdir -p /home/chroot
[sudo] contraseña para equipo:
```

2. Se listan los ‘character special files’ esenciales, como null, random, tty, etc., para que, posteriormente, sean replicados dentro de la jaula, asegurándose de crear cada uno con mknod y permisos adecuados. Finalmente, estos dispositivos serán movidos al subdirectorio dev dentro de /home/chroot.

```
equipo@equipo-G3-3500:~$ ls -l /dev/{null,zero,stdin,stderr,stdout,random,tty}
crw-rw-rw- 1 root root 1, 3 oct 26 01:44 /dev/null
crw-rw-rw- 1 root root 1, 8 oct 26 01:44 /dev/random
lrwxrwxrwx 1 root root   15 oct 26 01:44 /dev/stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root   15 oct 26 01:44 /dev/stdin -> /proc/self/fd/0
lrwxrwxrwx 1 root root   15 oct 26 01:44 /dev/stdout -> /proc/self/fd/1
crw-rw-rw- 1 root tty  5, 0 oct 26 22:26 /dev/tty
crw-rw-rw- 1 root root 1, 5 oct 26 01:44 /dev/zero
equipo@equipo-G3-3500:~$ cd /home/chroot
equipo@equipo-G3-3500:/home/chroot$ mkdir dev
mkdir: no se puede crear el directorio «dev»: Permiso denegado
equipo@equipo-G3-3500:/home/chroot$ sudo mkdir dev
[sudo] contraseña para equipo:
equipo@equipo-G3-3500:/home/chroot$ mknod -m 666 random c 1 8
mknod: random: Permiso denegado
equipo@equipo-G3-3500:/home/chroot$ sudo mknod -m 666 random c 1 8
equipo@equipo-G3-3500:/home/chroot$ sudo mknod -m 666 null c 1 3
equipo@equipo-G3-3500:/home/chroot$ sudo mknod -m 666 tty c 5 0
equipo@equipo-G3-3500:/home/chroot$ sudo mknod -m 666 zero c 1 5
equipo@equipo-G3-3500:/home/chroot$ chown equipo:equipo /home/chroot && chmod 0755 /home/chroot && ls
 -l /home/chroot
chown: cambiando el propietario de '/home/chroot': Operación no permitida
equipo@equipo-G3-3500:/home/chroot$ sudo chown equipo:equipo /home/chroot && chmod 0755 /home/chroot
&& ls -l /home/chroot
total 4
drwxr-xr-x 2 root root 4096 oct 26 22:28 dev
crw-rw-rw- 1 root root 1, 3 oct 26 22:29 null
crw-rw-rw- 1 root root 1, 8 oct 26 22:29 random
crw-rw-rw- 1 root root 5, 0 oct 26 22:30 tty
crw-rw-rw- 1 root root 1, 5 oct 26 22:30 zero
equipo@equipo-G3-3500:/home/chroot$ sudo mv /home/chroot/null /home/chroot/dev
equipo@equipo-G3-3500:/home/chroot$ sudo mv /home/chroot/random /home/chroot/dev
equipo@equipo-G3-3500:/home/chroot$ sudo mv /home/chroot/tty /home/chroot/dev
equipo@equipo-G3-3500:/home/chroot$ sudo mv /home/chroot/zero /home/chroot/dev
equipo@equipo-G3-3500:/home/chroot$ cd dev
equipo@equipo-G3-3500:/home/chroot/dev$ dir
null  random  tty  zero
equipo@equipo-G3-3500:/home/chroot/dev$ cd..
cd..: orden no encontrada
equipo@equipo-G3-3500:/home/chroot/dev$ cd ..
equipo@equipo-G3-3500:/home/chroot$ dir
dev
equipo@equipo-G3-3500:/home/chroot$ cd dev
equipo@equipo-G3-3500:/home/chroot/dev$ cd ..
```

3. Copiar el ejecutable /bin/bash y las bibliotecas que requiere (ld-linux-x86-64.so.2, libc.so.6, etc.) en las carpetas de la jaula (bin, lib, lib64), permitiendo que el entorno chroot tenga un intérprete de comandos funcional.

```
equipo@equipo-G3-3500:/home/chroot$ sudo mkdir bin
equipo@equipo-G3-3500:/home/chroot$ cp -v /bin/bash bin
'/bin/bash' -> 'bin/bash'
cp: no se puede crear el fichero regular 'bin/bash': Permiso denegado
equipo@equipo-G3-3500:/home/chroot$ sudo cp -v /bin/bash bin
'/bin/bash' -> 'bin/bash'
equipo@equipo-G3-3500:/home/chroot$ sudo mkdir /home/chroot/lib64
equipo@equipo-G3-3500:/home/chroot$ sudo mkdir -p /home/chroot/lib/x86_64-linux-gnu
equipo@equipo-G3-3500:/home/chroot$ sudo cp -v /lib64/ld-linux-x86-64.so.2 /home/chroot/lib64
'/lib64/ld-linux-x86-64.so.2' -> '/home/chroot/lib64/ld-linux-x86-64.so.2'
equipo@equipo-G3-3500:/home/chroot$ sudo cp -v /lib/x86_64-linux-gnu/{libtinfo.so.6,libc.so.6} /home/
chroot/lib/x86_64-linux-gnu/
'/lib/x86_64-linux-gnu/libtinfo.so.6' -> '/home/chroot/lib/x86_64-linux-gnu/libtinfo.so.6'
'/lib/x86_64-linux-gnu/libc.so.6' -> '/home/chroot/lib/x86_64-linux-gnu/libc.so.6'
```

4. Copiar los archivos básicos de configuración (/etc/passwd y /etc/group) dentro de la jaula para que las cuentas de usuario y los permisos se puedan reconocer en este entorno.

```
equipo@equipo-G3-3500:/home/chroot$ sudo mkdir /home/chroot/lib64
equipo@equipo-G3-3500:/home/chroot$ sudo mkdir -p /home/chroot/lib/x86_64-linux-gnu
equipo@equipo-G3-3500:/home/chroot$ sudo cp -v /lib64/ld-linux-x86-64.so.2 /home/chroot/lib64
'/lib64/ld-linux-x86-64.so.2' -> '/home/chroot/lib64/ld-linux-x86-64.so.2'
equipo@equipo-G3-3500:/home/chroot$ sudo cp -v /lib/x86_64-linux-gnu/{libtinfo.so.6,libc.so.6} /home/
chroot/lib/x86_64-linux-gnu/
'/lib/x86_64-linux-gnu/libtinfo.so.6' -> '/home/chroot/lib/x86_64-linux-gnu/libtinfo.so.6'
'/lib/x86_64-linux-gnu/libc.so.6' -> '/home/chroot/lib/x86_64-linux-gnu/libc.so.6'
equipo@equipo-G3-3500:/home/chroot$ sudo mkdir etc
equipo@equipo-G3-3500:/home/chroot$ sudo cp -v /etc/{passwd,group} /home/chroot/etc
'/etc/passwd' -> '/home/chroot/etc/passwd'
'/etc/group' -> '/home/chroot/etc/group'
equipo@equipo-G3-3500:/home/chroot$ cd etc
equipo@equipo-G3-3500:/home/chroot/etc$ ls
group  passwd
```

5. Mediante el comando sudo nano /home/chroot/etc/paswd, editar el archivo para dejarlo de la siguiente manera:

```
  GNU nano 6.2                                  passwd
root:x:0:0:root:/root:/bin/bash
equipo:x:1000:1000:Equipo,,,:/home/equipo:/bin/bash
```

6. Editar el archivo de configuración de SSH (/etc/ssh/sshd_config) para agregar la condición de la jaula y el permiso permitTTY 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
Match User equipo
        ChrootDirectory /home/chroot
```

7. Recargar el SSH

```
equipo@equipo-G3-3500:/home/chroot/etc$ sudo systemctl restart ssh
equipo@equipo-G3-3500:/home/chroot/etc$
```

8. Al probar una conexión con un cliente que se encuentra dentro de la jaula el acceso deberá estar limitado de acuerdo a cómo configuramos la jaula. Así lo podemos ver en la siguiente imagen:

```
root@serDebian:~# ssh equipo@192.168.1.110
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-47-generic x86_64)

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

El mantenimiento de seguridad expandido para Applications está desactivado

Se puede aplicar 1 actualización de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Oct 26 23:19:25 2024 from 192.168.1.110
-bash-5.1$ ls
-bash: ls: command not found
-bash-5.1$
```

9. Para crea un grupo específico para los usuarios que están enjaulados y añadir un usuarios a ese grupo, debemos ejecutar el siguiente comando:

sudo groupadd jailgroup

sudo useradd -m -d /home/chroot/userjail -s /bin/bash -G jailgroup userjail

Posteriormente, debemos editar el archivo de configuración SSH para restringir el acceso a los usuarios enjaulados, para ello, se tiene que añadir estas líneas al final del archivo para configurar la jaula:

Match Group jailgroup

ChrootDirectory /home/chroot

ForceCommand internal-sftp

Conexión con retorno de Entorno gráfico aplicaciónes

En el servidor:

Aquí tienes una breve descripción de los comandos ejecutados:

1. Actualiza la lista de paquetes y versiones disponibles en los repositorios.

```
equipo@equipo-G3-3500:/$ sudo apt update
Obj:1 http://mx.archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 https://dl.google.com/linux/chrome/deb stable InRelease
Des:3 http://mx.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Des:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Des:5 http://mx.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Obj:6 https://download.virtualbox.org/virtualbox/debian jammy InRelease
Des:7 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2 113 kB]
Des:8 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [712 kB]
Des:9 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [103 kB]
Des:10 http://mx.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 DEP-11 Metadata [212 B]
Des:11 http://mx.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1 133 kB]
Des:12 http://mx.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [356 kB]
Des:13 http://mx.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Des:14 http://mx.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [5 320 B]
Des:15 http://mx.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [212 B]
Des:16 http://mx.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [23.2 kB]
Des:17 http://mx.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]
Des:18 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [43.3 kB]
Des:19 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 DEP-11 Metadata [208 B]
Des:20 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [126 kB]
Des:21 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 DEP-11 Metadata [208 B]
Descargados 5 001 kB en 2s (2 749 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se puede actualizar 1 paquete. Ejecute «apt list --upgradable» para verlo.
W: https://download.virtualbox.org/virtualbox/debian/dists/jammy/InRelease: Key is stored in legacy t
rusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
```

2. Con el comando “sudo apt install openssh-server xauth -y”: Instala el servidor SSH y “xauth”, confirmando automáticamente cualquier pregunta durante la instalación. Ambos paquetes ya están en su versión más reciente.

```
equipo@equipo-G3-3500:/$ sudo apt install openssh-server xauth -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
xauth ya está en su versión más reciente (1:1.1-1build2).
openssh-server ya está en su versión más reciente (1:8.9p1-3ubuntu0.10).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
```

3. “sudo nano /etc/ssh/sshd_config”: Abre el archivo de configuración del servidor SSH para editarlo, aquí se debe verificar que las siguientes características están habilitadas.

```
#AllowAgentForwarding yes
AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
X11DisplayOffset 10
X11UseLocalhost yes
PermitTTY yes
PrintMotd no
#PrintLastLog yes
```

4. “sudo systemctl restart ssh”: Reinicia el servicio SSH para aplicar cambios en la configuración.

En el cliente:

1. Actualiza la lista de paquetes y versiones disponibles en los repositorios.

```
root@serDebian:~# sudo apt update
Obj:1 http://deb.debian.org/debian bookworm InRelease
Obj:2 http://security.debian.org/debian-security bookworm-security InRelease
Obj:3 http://deb.debian.org/debian bookworm-updates InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
```

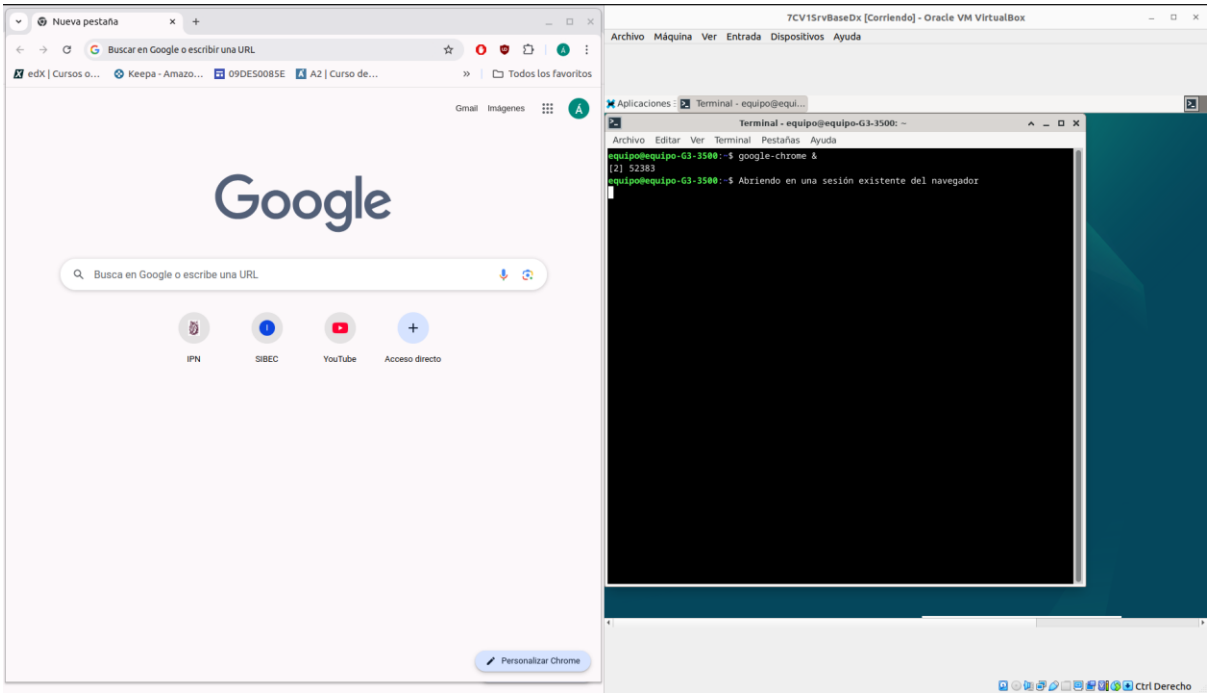
2. Instala el paquete x11-apps para probar aplicaciones gráficas.

```
root@serDebian:~# sudo apt install ssh x11-apps -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
ssh ya está en su versión más reciente (1:9.2p1-2+deb12u3).
x11-apps ya está en su versión más reciente (7.7+9).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

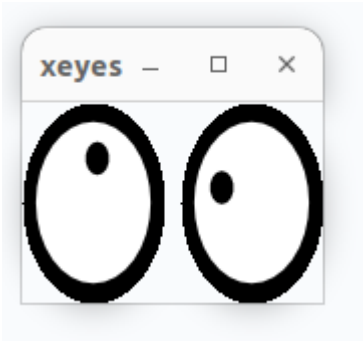
3.Inicia sesión SSH con reenvío X11 en el servidor desde la terminal del cliente.

```
root@serDebian:~# ssh -X equipo@192.168.1.110
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-47-generic x86_64)

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

El mantenimiento de seguridad expandido para Applications está desactivado

Se puede aplicar 1 actualización de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Oct 27 01:46:35 2024 from 192.168.1.110
```

4. Se ingresa el nombre de la aplicación gráfica que se desea abrir.




```
equipo@equipo-G3-3500:~$ xeyes &
[1] 53269
```




```
equipo@equipo-G3-3500:~$ gnome-calculator &
[1] 53270
```

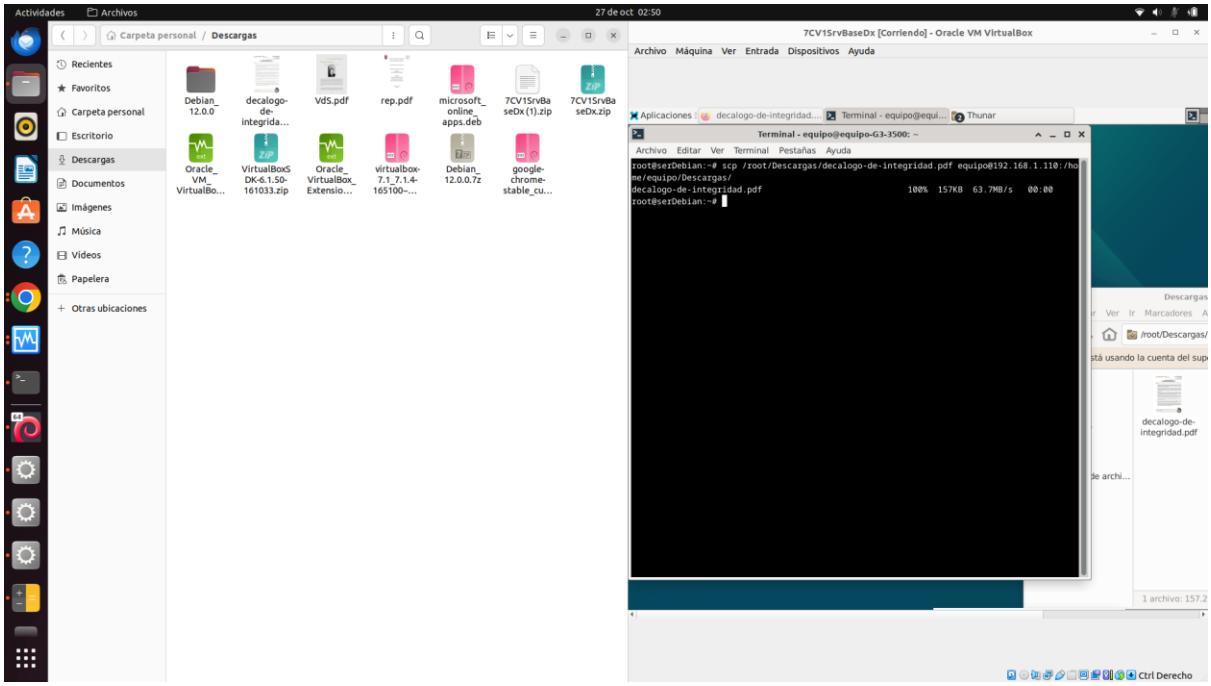
a) Copia de elementos remotos al servidor utilizando terminal

1. Para copiar elementos del cliente al servidor, se utiliza el comando scp (Secure Copy Protocol) para copiar archivos y directorios de manera segura..

   Sintaxis básica de scp:
   scp [opciones] [origen] [usuario@servidor:destino]

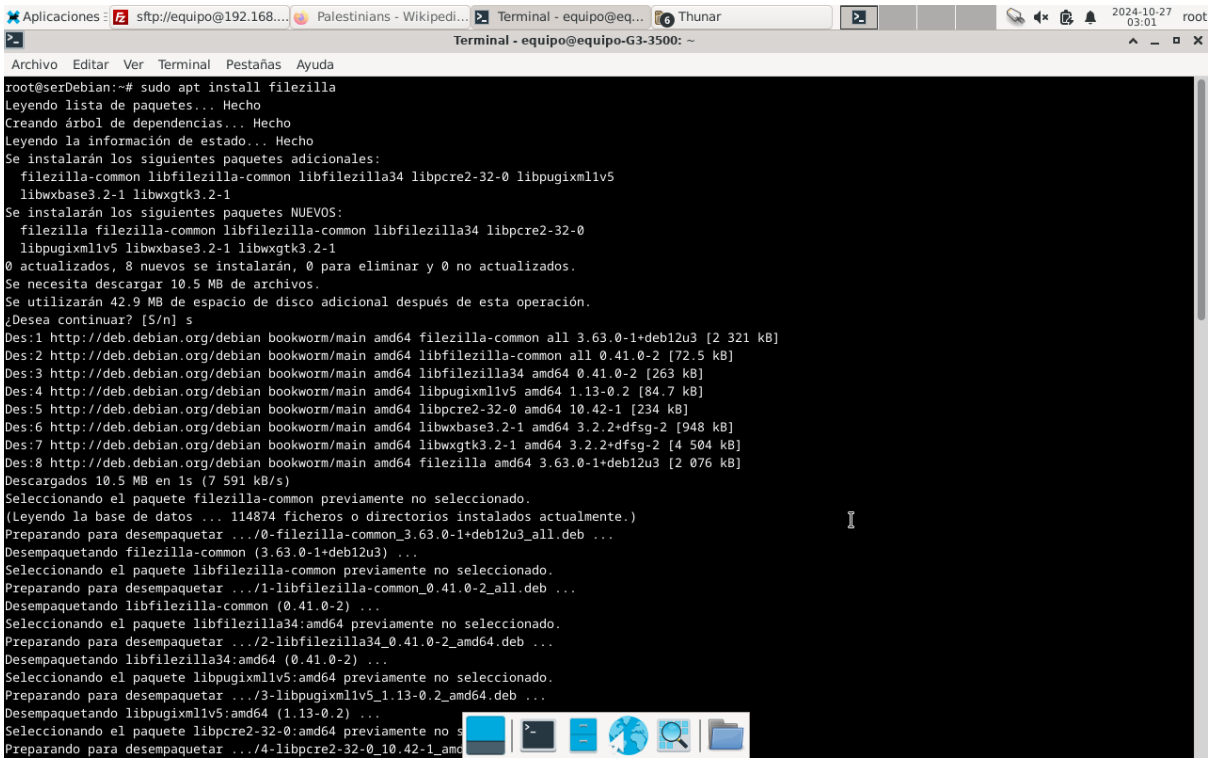
   Ejemplo: Copiar un archivo de Debian a Ubuntu

scp root/Descargas/decalogo-de-integridad.pdf root@192.168.1.110:/home/equipo/Descargas



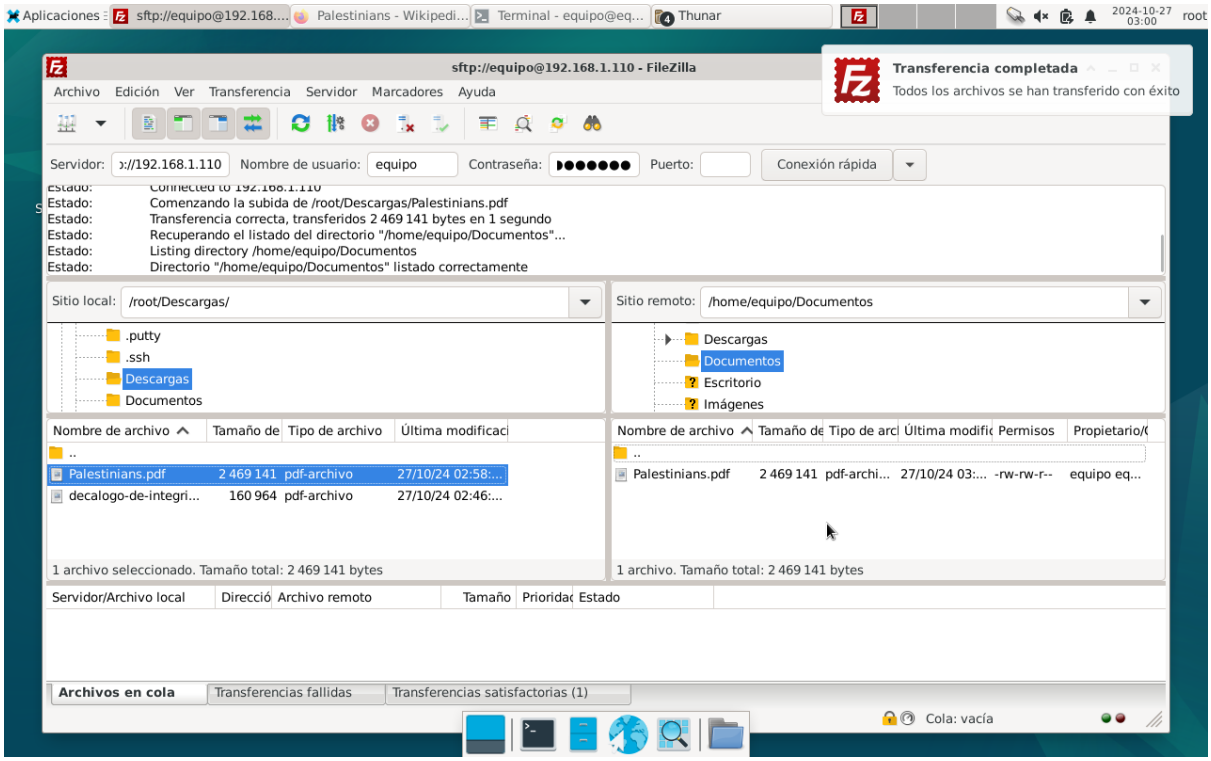
b) Copia de elementos remotos al servidor utilizando entorno gráfico remoto

1.Se utiliza SFTP (SSH File Transfer Protocol), para copiar elementos mediante un entorno gráfico. SFTP se puede usar instalando filezilla con el siguiente comando: sudo apt install filezilla



2. Abrir FileZilla y configurar la conexión:
   - Host: sftp://192.168.1.110.
   - Nombre de usuario: equipo.
   - Contraseña: ********.

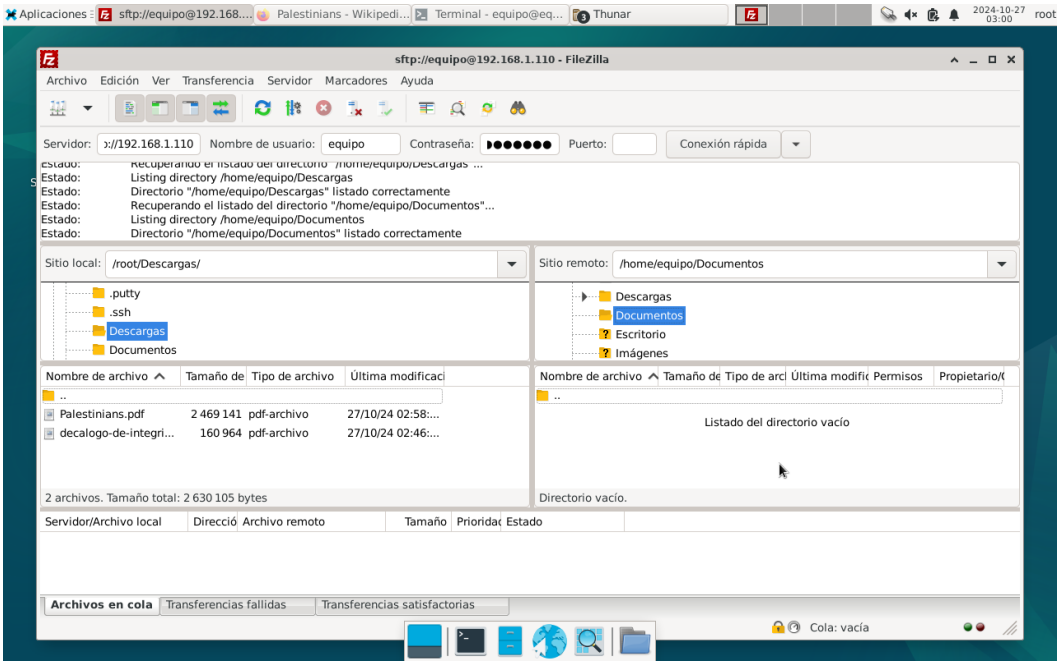
- Puerto: 22 (por defecto para SSH/SFTP).

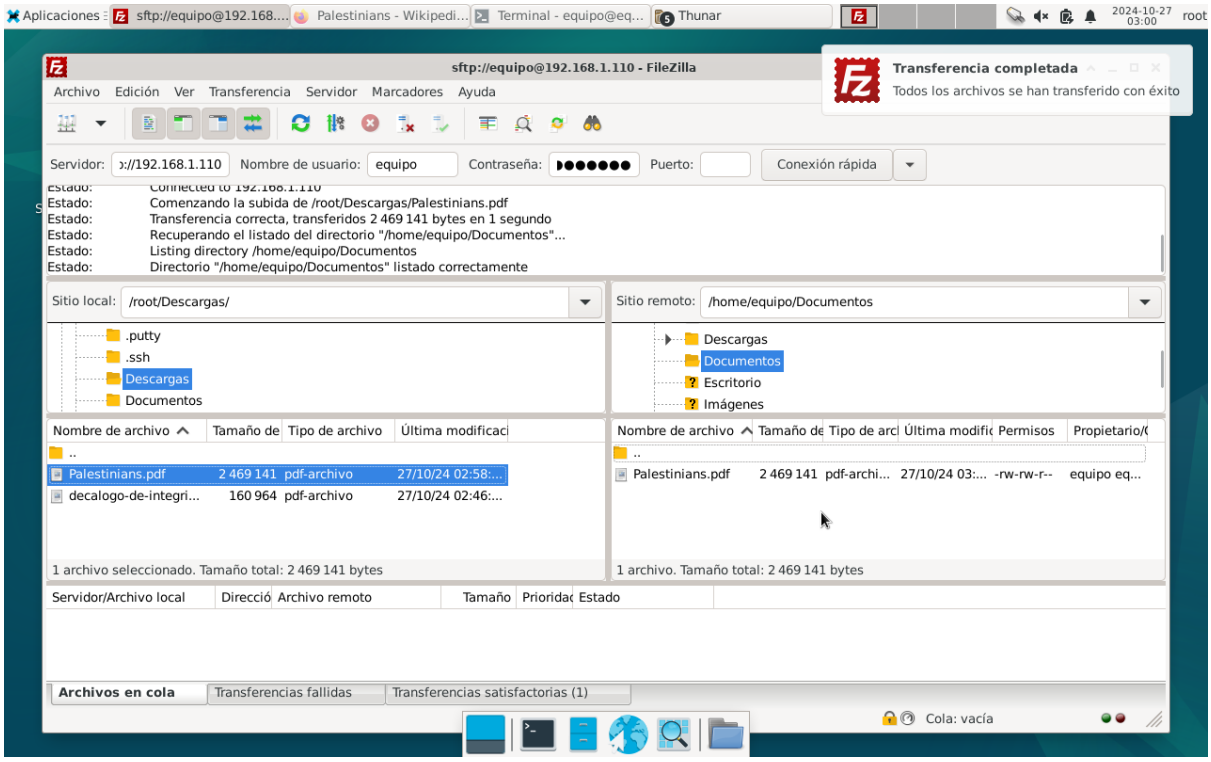


3. Conectar al servidor. Una vez conectado, verás dos paneles: uno para tu máquina local y otro para el servidor.

Copiar archivos:

- Navega a la ubicación del archivo en tu máquina local (panel izquierdo).
- Arrastra y suelta el archivo o directorio en el panel derecho donde deseas copiarlo en el servidor.





Tunel de aplicaciones por SSH (3 diferentes elementos)

1. Crear el túnel SSH para HTTP/HTTPS

Para acceder a un servidor web que se está ejecutando en el puerto 80 (HTTP) o 443 (HTTPS) en el servidor, es posible usar el siguiente comando en la consola del cliente:

ssh -L 8080:localhost:80 usuario@ip_del_servidor

```
root@serDebian:~# ssh -L 8080:localhost:80 equipo@192.168.1.110
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-47-generic x86_64)

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

El mantenimiento de seguridad expandido para Applications está desactivado

Se puede aplicar 1 actualización de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Oct 27 03:16:16 2024 from 192.168.1.110
equipo@equipo-G3-3500:~$
```

```
root@serDebian:~# ss -tnlp | grep ssh
LISTEN 0      128        127.0.0.1:8080      0.0.0.0:*    users:(("ssh",pid=2964
1,fd=5))
LISTEN 0      128          0.0.0.0:22        0.0.0.0:*    users:(("sshd",pid=262
34,fd=3))
LISTEN 0      128            [::1]:8080         [::]:*    users:(("ssh",pid=2964
1,fd=4))
LISTEN 0      128             [::]:22           [::]:*    users:(("sshd",pid=262
34,fd=4))
root@serDebian:~#
```

2. Crear el túnel SSH para SMTP

Para un servidor de correo en el puerto 25 (SMTP), se establece el túnel SSH de la siguiente manera:

ssh -L 2525:localhost:25 usuario@ip_del_servidor

Luego, podrá enviar correos utilizando un cliente de correo configurado para usar el puerto 2525 en la máquina del cliente.

```
root@serDebian:~# ssh -L 2525:localhost:25 equipo@192.168.1.110
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-47-generic x86_64)

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

El mantenimiento de seguridad expandido para Applications está desactivado

Se puede aplicar 1 actualización de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Oct 27 04:36:57 2024 from 192.168.1.110
equipo@equipo-G3-3500:~$
```

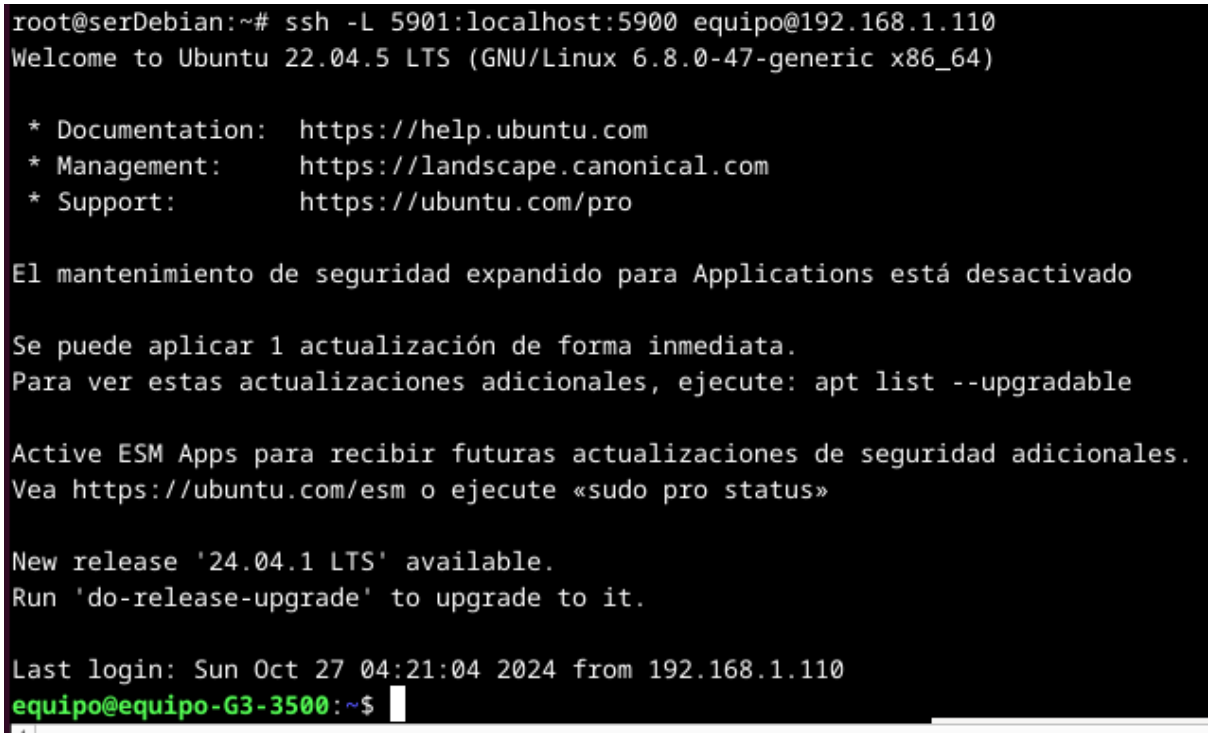
```
root@serDebian:~# ss -tnlp | grep ssh
LISTEN 0      128          0.0.0.0:22        0.0.0.0:*    users:(("sshd",pid=262
34,fd=3))
LISTEN 0      128        127.0.0.1:2525      0.0.0.0:*    users:(("ssh",pid=3015
8,fd=5))
LISTEN 0      128            [::1]:2525         [::]:*    users:(("ssh",pid=3015
8,fd=4))
LISTEN 0      128             [::]:22           [::]:*    users:(("sshd",pid=262
34,fd=4))
root@serDebian:~#
```

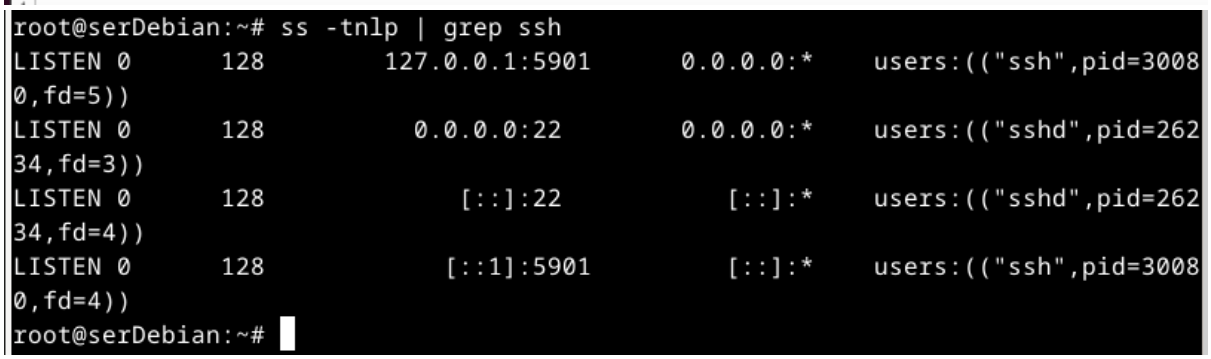
3. Túnel SSH para acceso remoto al Escritorio (VNC)

Si tiene configurado un servidor VNC en el servidor (por ejemplo, en el puerto 5900), puedes redirigir la conexión con el siguiente comando:

ssh -L 5901:localhost:5900 usuario@ip_del_servidor

Luego, en el cliente Debian, puede usar un cliente VNC (como vinagre o Remmina) para conectarse a localhost:5901 y acceder al escritorio remoto.





Herramienta de consulta y filtrado de Total de usuarios/conexión (Resumen de operación de los usuarios en el servidor SSH)

1. Consultar los usuarios conectados

Para listar los usuarios que están actualmente conectados al servidor SSH, se usa el comando “who”

Este comando mostrará una lista de usuarios conectados, junto con la hora y la terminal desde la que se conectaron.

```
equipo@equipo-G3-3500:~$ who
equipo   :0           2024-10-26 01:44 (:0)
equipo   pts/1        2024-10-26 17:06
equipo   pts/2        2024-10-26 18:38
equipo   pts/3        2024-10-26 18:40
equipo   pts/4        2024-10-26 18:41
equipo   pts/5        2024-10-26 18:44
equipo   pts/6        2024-10-27 03:12 (192.168.1.110)
```

```
equipo@equipo-G3-3500:~$ who -a
          arranque del sistema 2024-10-26 01:44
equipo   ? :0           2024-10-26 01:44   ?          1337 (:0)
          `run-level' 5 2024-10-26 01:44
equipo   + pts/1        2024-10-26 17:06 10:06       25783
equipo   + pts/2        2024-10-26 18:38 08:35       27930
equipo   + pts/3        2024-10-26 18:40 08:33       27973
equipo   + pts/4        2024-10-26 18:41 08:32       27985
equipo   + pts/5        2024-10-26 18:44 08:29       28618
equipo   + pts/6        2024-10-27 03:12 00:01       54538 (192.168.1.110)
            pts/7        2024-10-27 01:41             49345 id=ts/7  term=0 salida=4
            pts/8        2024-10-27 02:26             53121 id=ts/8  term=15 salida=0
```

2. Filtrar usuarios conectados

Si desea filtrar usuarios específicos, puede usar grep.

who | grep 192.168.1.110

```
equipo@equipo-G3-3500:~$ who | grep 192.168.1.110
equipo   pts/6        2024-10-27 03:12 (192.168.1.110)
```

3. Resumen de conexiones SSH

Para obtener un resumen de las conexiones SSH, puede usar el comando last, que muestra el historial de conexiones. Para filtrar solo las conexiones SSH, puedes hacer:
last -a | grep ssh
Esto mostrará un historial de las conexiones SSH, incluyendo la dirección IP desde la que se conectó cada usuario.

```
equipo@equipo-G3-3500:~$ last -a | grep ssh
```

4. Contar usuarios y conexiones

Para contar el total de usuarios conectados, se usar el comando “who | wc -l”.

```
equipo@equipo-G3-3500:~$ who | wc -l
7
equipo@equipo-G3-3500:~$ last -a | grep ssh | wc -l
0
```