Implementación de Servidor SFTP/SSH con Autenticación Centralizada LDAP

Resumen Ejecutivo

Este proyecto documenta la expansión estratégica de una infraestructura de red mediante la implementación y securización de un servidor de transferencia de archivos (SFTP) y acceso remoto (SSH).
El núcleo del proyecto es el establecimiento de un entorno de acceso robusto y controlado, donde la autenticación de usuarios se gestiona de forma centralizada a través de un servidor OpenLDAP preexistente.
El diseño arquitectónico se centra en la **segregación de funciones**: los administradores del sistema obtienen acceso total vía terminal (SSH), mientras que los usuarios operativos son confinados a un entorno seguro y aislado (Chroot Jail) exclusivamente para la transferencia de archivos (SFTP).
Este enfoque no solo optimiza la operatividad, sino que también fortalece la postura de seguridad de la organización, alineándose con principios de Cero Confianza (Zero Trust) y Mínimo Privilegio (Principle of Least Privilege).

Parte 1: Fortalecimiento y Seguridad del Servidor LDAP

Antes de integrar nuevos servicios, era imperativo robustecer el núcleo de la autenticación: el servidor LDAP.
La fase inicial se centró en eliminar vulnerabilidades críticas, principalmente la exposición de datos a través de consultas no autenticadas, un vector de ataque común para el reconocimiento de la red.

1.1. Restricción de Consultas Anónimas (Anonymous Bind)

El primer paso crucial fue denegar el acceso de lectura a usuarios no autenticados (anónimos). Esta medida es fundamental para prevenir que actores maliciosos puedan enumerar usuarios, grupos y la estructura del directorio sin credenciales válidas.
Para lograrlo, se redactó un archivo de configuración en formato LDIF (LDAP Data Interchange Format). Este archivo contiene las directivas para modificar dinámicamente la configuración del motor de la base de datos **olcDatabase={1}mdb,** deshabilitando explícitamente el "**bind**" anónimo.

```
administrador-ldap@ServidorLdap:~$ touch disable-anon.ldif
administrador-ldap@ServidorLdap:~$ ls
disable-anon.ldif  samba_ldap.ldif               usuarios-Reader.ldif
Grupos             Scripts                       usuarios-recursos-humanos.ldif
Modificador        usuarios-administracion.ldif  usuarios-soporte.ldif
orgUnit            usuarios-desarrollo.ldif
Plantillas         usuariosFinales.ldif
administrador-ldap@ServidorLdap:~$ nano disable-anon.ldif
```

```
  GNU nano 2.9.3                        disable-anon.ldif

dn: cn=config
changetype: modify
add: olcDisallows
olcDisallows: bind_anon
_
add: olcRequires
olcRequires: authc_

                          [ 7 líneas escritas ]
^G Ver ayuda  ^O Guardar    ^W Buscar      ^K Cortar Text^J Justificar ^C Posición   M-U Deshacer
^X Salir      ^R Leer fich. ^\ Reemplazar  ^U Pegar txt  ^T Ortografía ^_ Ir a línea M-E Rehacer
```

Creación del archivo LDIF con la directiva para deshabilitar el acceso anónimo

Una vez creado el archivo .ldif, se utilizó la herramienta ldapmodify para aplicar la configuración. Es una práctica de seguridad estándar no editar directamente los archivos de configuración de slapd en /etc/ldap/slapd.d/, ya que son parte de una base de datos de configuración dinámica.

```
administrador-ldap@ServidorLdap:~$ sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f disable-anon.ldif
[sudo] contraseña para administrador-ldap:
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=config"

administrador-ldap@ServidorLdap:~$ _
```

Aplicación de la nueva política de seguridad usando ldapmodify.

Vulnerabilidad Mitigada: Permitir el "**bind**" anónimo expone la estructura completa del directorio, facilitando ataques de enumeración y perfilado de la organización. Deshabilitarlo es una recomendación estándar en guías de hardening como las de CIS Benchmarks

```
administrador-ldap@ServidorLdap:~$ ldapsearch -x -H ldap://192.168.1.38 -b "dc=luthor,dc=corp"
ldap_bind: Inappropriate authentication (48)
        additional info: anonymous bind disallowed
administrador-ldap@ServidorLdap:~$ _
```

Verificación: intento de consulta anónima fallido tras aplicar la configuración.

1.2. Creación del Usuario de Servicio (Service Account)

Al deshabilitar el acceso anónimo, los servicios que dependen de LDAP (como SSH/SFTP) necesitan una forma de autenticarse para validar las credenciales de los usuarios. Utilizar la cuenta de administrador principal para esta tarea representaría un riesgo de seguridad inaceptable.

La solución es crear una cuenta de servicio dedicada con privilegios estrictamente limitados. Esta cuenta solo necesita permiso de lectura **(read)** y búsqueda **(search)** sobre los atributos necesarios para la autenticación.

Para una gestión ordenada, se creó primero una Unidad Organizativa (OU) llamada "servicio" y un grupo para contener esta cuenta, estructurando el directorio de forma lógica y escalable

OU:

```
 GNU nano 2.9.3                          ouServicios.ldif

dn: ou=service,dc=luthor,dc=corp
objectClass: organizationalUnit
ou: Service

                                [ 3 líneas escritas ]
^G Ver ayuda  ^O Guardar    ^W Buscar      ^K Cortar Text^J Justificar ^C Posición   M-U Deshacer
^X Salir      ^R Leer fich. ^\ Reemplazar  ^U Pegar txt  ^T Ortografía ^_ Ir a línea M-E Rehacer
```

```
administrador-ldap@ServidorLdap:~$ ldapadd -x -D "cn=admin,dc=luthor,dc=corp" -W -f ouServicios.ldif
Enter LDAP Password:
adding new entry "ou=Service,dc=luthor,dc=corp"

administrador-ldap@ServidorLdap:~$
```

1.3. Configuración de Listas de Control de Acceso (ACL)

Con el usuario de servicio ya creado, el siguiente paso fue definir Listas de Control de Acceso (ACLs) granulares. Las ACLs son el corazón de la seguridad en LDAP, determinando con precisión qué operaciones puede realizar cada entidad sobre cada parte del directorio.

Se redactó un nuevo archivo .ldif para establecer las reglas de acceso (olcAccess) con la siguiente lógica jerárquica:

1. **Administrador (rootdn):** Acceso total de gestión (manage).
2. **Usuario de Servicio:** Permiso de lectura (read) para buscar usuarios y verificar credenciales.
3. **Usuarios autenticados (self):** Permiso para modificar su propia contraseña.
4. **Resto de entidades (incluyendo anónimos):** Solo permiso para autenticarse (auth), sin capacidad de leer o buscar datos.

```
  GNU nano 2.9.3                        Acl-servicio.ldif

dn: olcDatabase={1}hdb,cn=config
changetype: modify
replace: olcAccess
olcAccess: to attrs=userPassword
 by self write
 by dn="cn=admin,dc=luthor,dc=corp" read write
 by dn="uid=ServiceUser,ou=Service,dc=luthor,dc=corp" read
 by anonymous auth
 by * none
 olcAccess to *
 by dn="cn=admin,dc=luthor,dc=corp" read write
 by dn="uid=ServiceUser,ou=Service,dc=luthor,dc=corp" read
 by * none

                        [ 13 líneas escritas ]
^G Ver ayuda  ^O Guardar    ^W Buscar      ^K Cortar Text^J Justificar ^C Posición   M-U Deshacer
^X Salir      ^R Leer fich. ^\ Reemplazar  ^U Pegar txt  ^T Ortografía ^_ Ir a línea M-E Rehacer
```

Archivo LDIF definiendo las ACLs para el administrador y el usuario de servicio.

Mejor Práctica en ACLs: Las reglas de ACL en OpenLDAP se procesan en orden. Es crucial colocar las reglas más específicas primero y terminar con las más generales. La directiva olcAccess es acumulativa y permite un control de acceso muy detallado.

Finalmente, las nuevas ACLs se cargaron en la configuración activa del servidor LDAP utilizando ldapmodify con autenticación SASL/EXTERNAL, un método seguro que autentica basándose en el UID/GID del usuario del sistema que ejecuta el comando.

```
administrador-ldap@ServidorLdap:~$ sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f Acl-servicioo.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "olcDatabase={1}hdb,cn=config"

administrador-ldap@ServidorLdap:~$ _
```

Parte 2: Gestión de Usuarios y Automatización de Acceso SFTP

Una vez securizado el directorio LDAP, el enfoque se desplazó hacia la gestión eficiente y escalable de los permisos de acceso al servicio SFTP, utilizando el directorio central como única fuente de verdad (Single Source of Truth).

2.1. Creación del Grupo de Control SFTPUsers en LDAP

Para evitar la gestión de permisos usuario por usuario en el servidor SFTP, se adoptó un modelo de control de acceso basado en grupos (GBAC). Se creó un grupo específico en LDAP llamado **SFTPUsers**.
La lógica es simple y potente: para conceder acceso SFTP a un usuario, un administrador de LDAP simplemente lo añade como miembro de este grupo. Para revocar el acceso, lo elimina. Esto centraliza la administración, reduce la carga operativa y minimiza el riesgo de errores de configuración en los servidores individuales.

```
  GNU nano 2.9.3                    Grupos/SFTPUsers.ldif

dn: cn=SFTPUsers,ou=Groups,dc=luthor,dc=corp
objectClass: top
objectClass: posixGroup
cn: SFTPUsers
gidNumber: 2005

                        [ 5 líneas leídas ]
^G Ver ayuda  ^O Guardar    ^W Buscar      ^K Cortar Text^J Justificar ^C Posición   M-U Deshacer
^X Salir      ^R Leer fich. ^\ Reemplazar  ^U Pegar txt  ^T Ortografía ^_ Ir a línea M-E Rehacer
```

Definición del grupo `SFTPUsers` en LDAP para centralizar los permisos de acceso

2.2. Automatización y Sincronización (Script de Gestión)

El servidor SFTP aplica las políticas de Chroot y permisos del sistema de archivos basándose en grupos definidos en **LDAP**. Para garantizar que la información utilizada por el servicio SFTP esté siempre alineada con el directorio LDAP central, se desarrolló un script de automatización llamado *Actualizar.sh.*

Este script funciona como un agente de sincronización con las siguientes funciones clave:

- **Consulta LDAP:** Utiliza ldapsearch para obtener la lista actual de miembros del grupo SFTPUsers desde el directorio LDAP central.
- **Validación de Consistencia:** Compara los usuarios definidos en LDAP con los usuarios habilitados para acceso SFTP en el servidor.
- **Actualización Dinámica:** Ajusta la configuración del servidor SFTP para reflejar exactamente los miembros del grupo LDAP, garantizando que solo los usuarios autorizados tengan acceso.
- **Depuración de Accesos:** Revoca de forma inmediata el acceso SFTP a los usuarios que ya no pertenecen al grupo SFTPUsers en LDAP, asegurando el cumplimiento de las políticas de seguridad.

2.3. Programación de Tareas (Crontab)

Para garantizar que el estado del servidor SFTP esté siempre sincronizado con el directorio LDAP sin intervención manual, el script Actualizar.sh se programó para ejecutarse automáticamente a intervalos regulares. Se utilizó crontab, el planificador de tareas estándar de Linux.

Esta automatización es crucial para la agilidad operativa. Un administrador puede añadir o quitar un usuario del grupo en LDAP, y el sistema propagará automáticamente el cambio de permisos al servidor SFTP en el siguiente ciclo de ejecución, sin necesidad de acceder al servidor de destino.

```
administrador-ldap@ServidorLDAP:~$ sudo mv Scripts/ActualizarSFTP.sh  /usr/local/sbin/
administrador-ldap@ServidorLDAP:~$ ls /usr/local/sbin/
ActualizarSFTP.sh
administrador-ldap@ServidorLDAP:~$ sudo cronotab -e_
```

```
  GNU nano 2.9.3                  /tmp/crontab.eTvcUd/crontab                  Modificado

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 3 * * * /usr/local/sbin/ActualizarSFTP.sh

^G Ver ayuda  ^O Guardar     ^W Buscar      ^K Cortar Text^J Justificar ^C Posición   M-U Deshacer
^X Salir      ^R Leer fich.  ^\ Reemplazar  ^U Pegar txt  ^T Ortografía ^_ Ir a línea M-E Rehacer
```

Entrada en `crontab` para la ejecución periódica del script de sincronización.

Parte 3: Configuración del Servidor SSH/SFTP

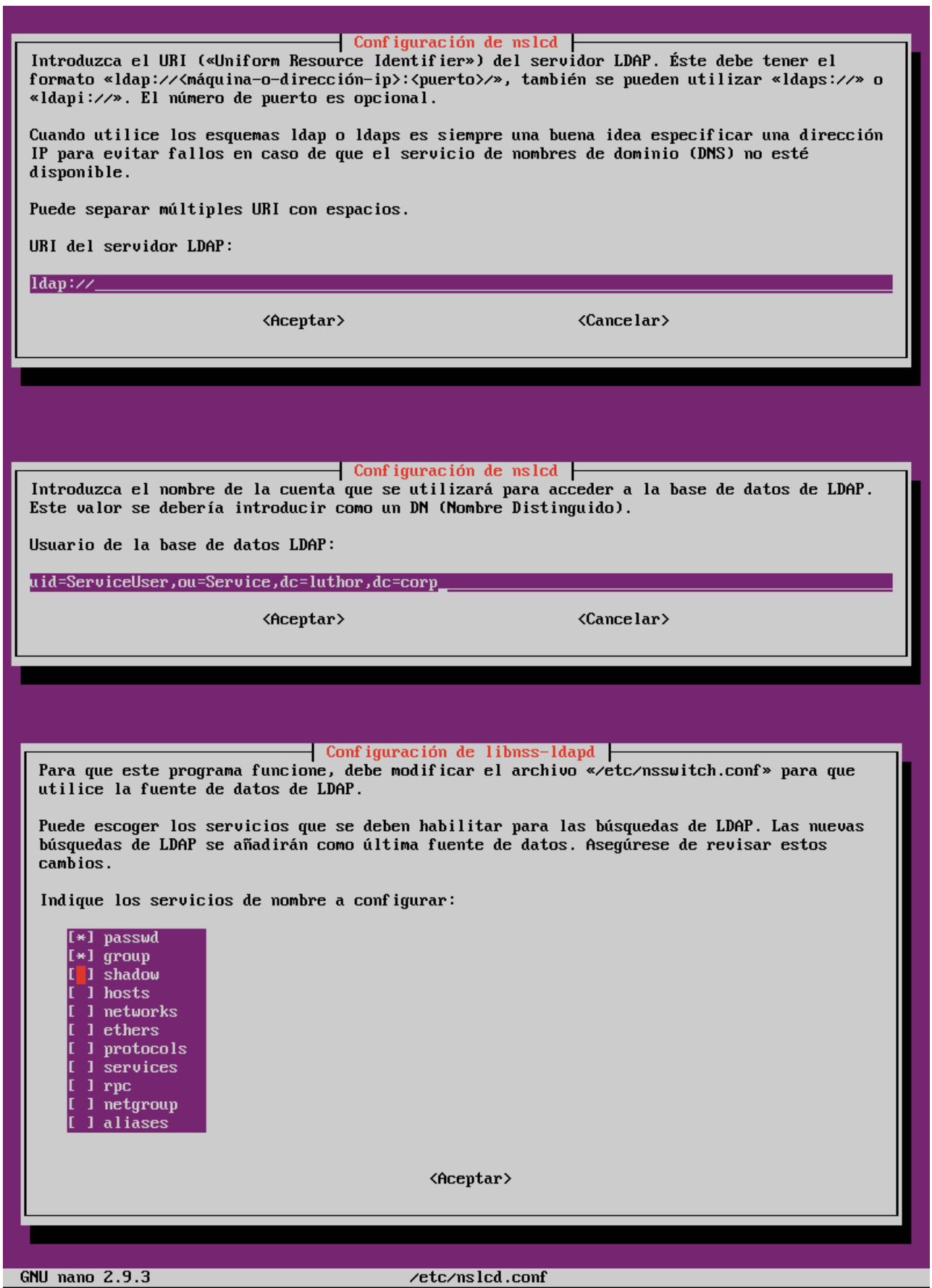
Esta fase consistió en configurar el servidor de destino para que utilizara LDAP como fuente de autenticación, integrando a los usuarios del directorio como si fueran locales.

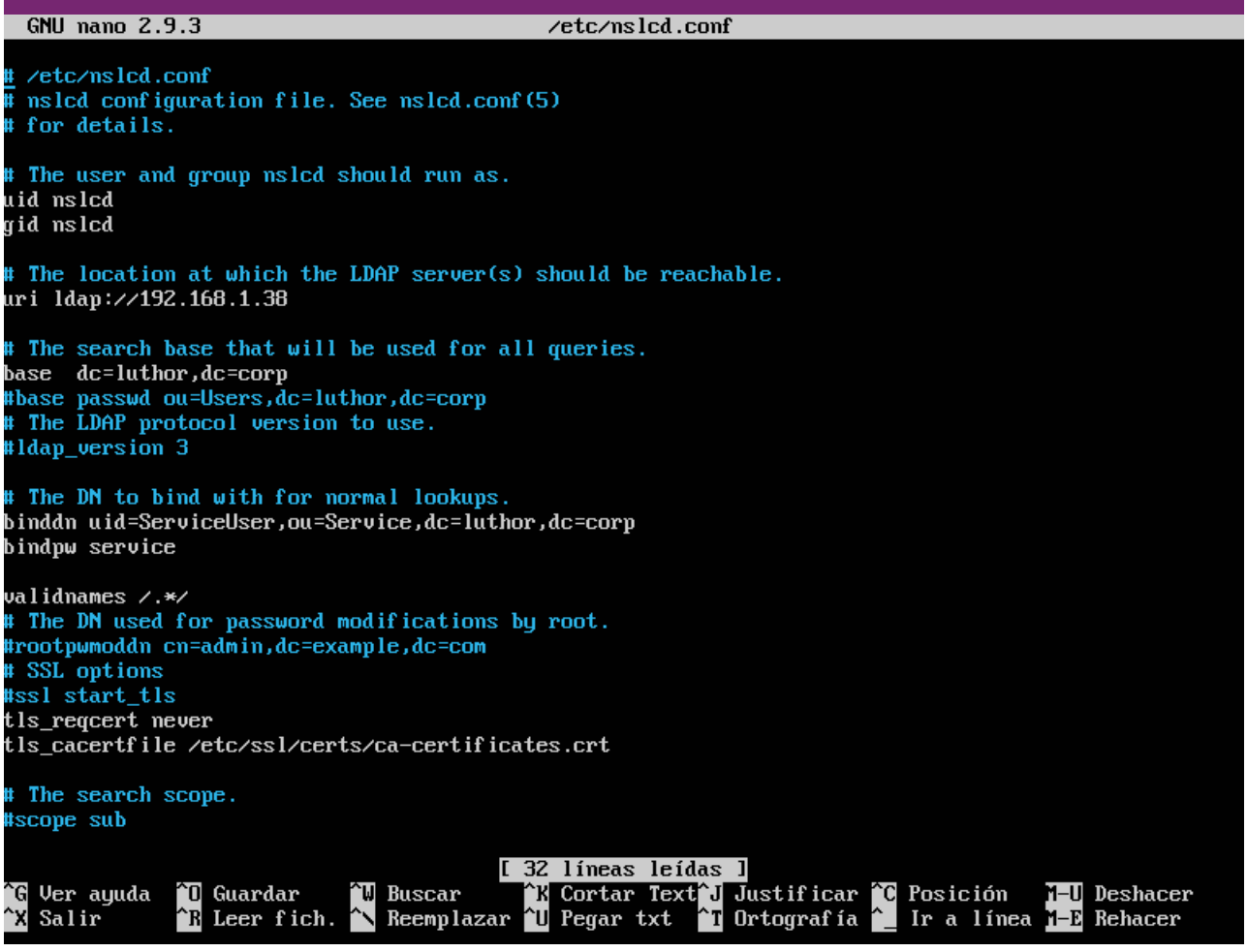
```
administradorSSH-SFTP@ServidorSSH-SFTP:~$ sudo apt update
Des:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [102 kB]
Obj:2 http://ar.archive.ubuntu.com/ubuntu bionic InRelease
Des:3 http://ar.archive.ubuntu.com/ubuntu bionic-updates InRelease [102 kB]
Des:4 http://ar.archive.ubuntu.com/ubuntu bionic-backports InRelease [102 kB]
Descargados 305 kB en 2s (133 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 3 paquetes. Ejecute «apt list --upgradable» para verlos.
administradorSSH-SFTP@ServidorSSH-SFTP:~$ sudo apt install nslcd libnss-ldapd libpam-ldapd openssh-s
erver
```

Actualizacion de paquetes

Para que el sistema operativo reconozca a los usuarios y grupos de LDAP, es necesario configurar dos componentes clave:

NSS (Name Service Switch): Permite que las funciones estándar de la biblioteca C (como getpwnam, getgrnam) obtengan información de fuentes alternativas además de los archivos locales (/etc/passwd, /etc/group).
nslcd (Name Service LDAP Caching Daemon): Actúa como un demonio que realiza las consultas LDAP en nombre de NSS y mantiene una caché local para mejorar el rendimiento y la resiliencia.
Se instalaron los paquetes necesarios (nslcd, libnss-ldapd, libpam-ldapd) y se configuraron para apuntar al URI del servidor LDAP, especificando el DN base de búsqueda y la cuenta de servicio creada en la Parte 1 para realizar las consultas.





Parte 4: Gestion de Administradores y Accesos Remotos

Para una administración segura y trazable, se implementó un modelo de acceso administrativo diferenciado, restringiendo el acceso SSH completo a un grupo selecto de administradores locales.

4.1. Creación de Usuarios de Administración y Grupo de Sistema

En lugar de permitir el acceso directo como root, una práctica de seguridad deficiente, se crearon cuentas de usuario nominales para cada administrador (ej. Admin1). Estas cuentas se agruparon en un grupo local del sistema llamado Admins_SSH.
Este enfoque mejora la trazabilidad y la rendición de cuentas (accountability), ya que todas las acciones administrativas quedan registradas en los logs del sistema bajo el nombre del usuario que las realizó, en lugar de un genérico "root".

Creación de Grupo de administradores:

sudo groupadd Admins_SSH

Creación de los usuarios administradores:

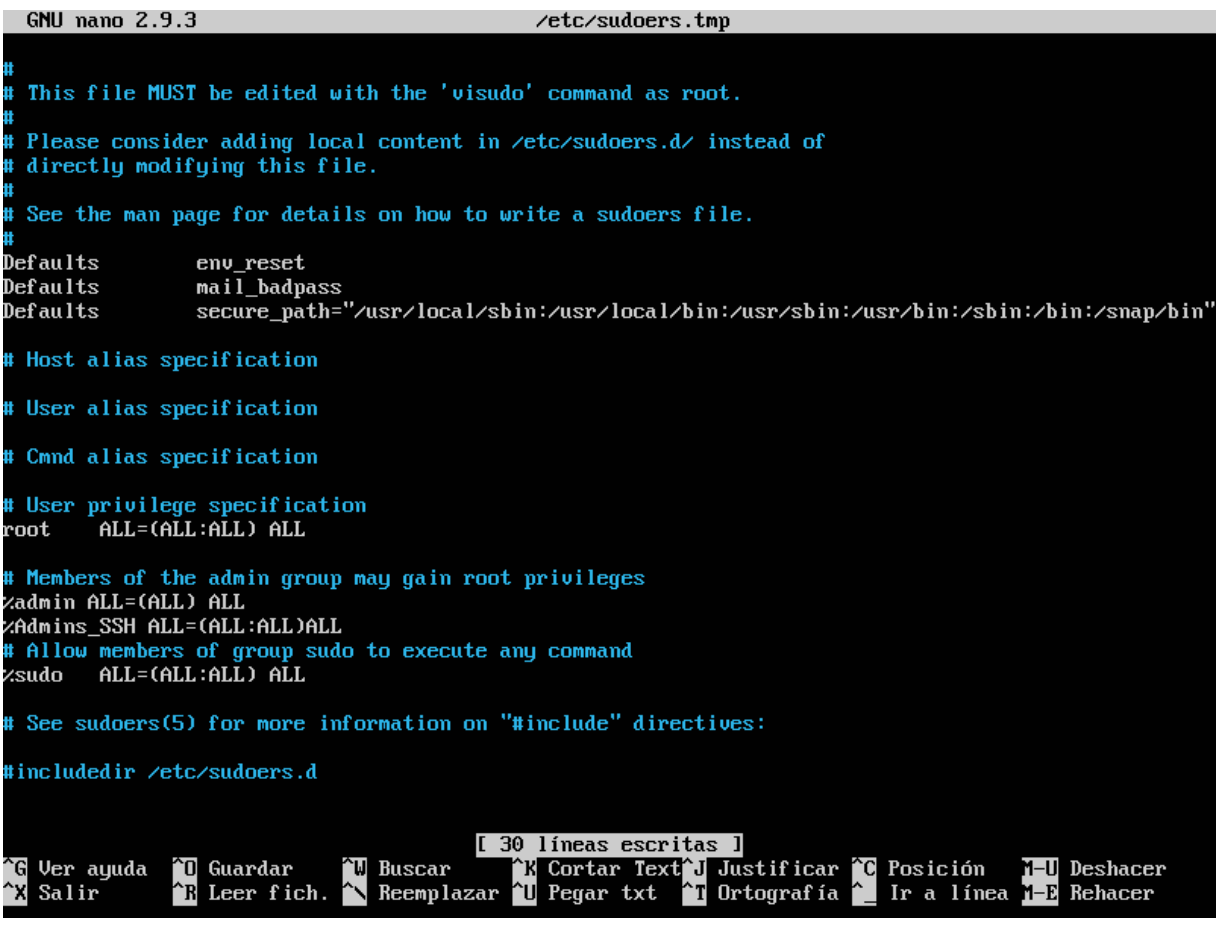
sudo useradd -m Admin1

Inclusión en el grupo de administración:

sudo usermod -aG admins admin_red

4.2. Configuración de Privilegios con `sudoers`

Para que los miembros del grupo **Admins_SSH** pudieran realizar tareas administrativas, se les otorgaron privilegios de superusuario a través de sudo. Se editó el archivo */etc/sudoers* utilizando el comando seguro ***sudo visudo***, que realiza una comprobación de sintaxis antes de guardar para evitar bloqueos del sistema.
La línea ***"%Admins_SSH ALL=(ALL:ALL) ALL "***concede a todos los miembros del grupo Admins_SSH la capacidad de ejecutar cualquier comando como cualquier usuario en cualquier host.



Edición del archivo `sudoers` para otorgar privilegios al grupo `Admins_SSH`.

Uso de `sudo` vs. `root` login:

Utilizar `sudo` es una práctica de seguridad fundamental. Limita la exposición de la contraseña de `root`, proporciona un registro de auditoría detallado de los comandos ejecutados y permite una asignación de privilegios mucho más granular que el acceso `root` total.

Parte 5:Preparación del Entorno SFTP (Chroot Jail)

La configuración del ";encarcelamiento" o **Chroot Jail** es el pilar de la seguridad para los usuarios de SFTP. Su objetivo es restringir a cada usuario a su propio directorio, impidiendo que navegue por el resto del sistema de archivos del servidor.

5.1. Estructura de Directorios y Permisos

La arquitectura del servicio SFTP se diseñó con zonas de permisos granulares. Para cada usuario, se crea una estructura de directorios específica. Por seguridad, OpenSSH impone una regla estricta: el directorio raíz de la jaula (ej. /home/usuario_sftp) debe pertenecer al usuario root y no tener permisos de escritura para otros.

Dentro de esta raíz, se crearon subdirectorios con permisos específicos:

/upload: Propiedad de root:sftpusers con permisos 1775. El sticky bit asegura que los usuarios puedan escribir y subir archivos en el directorio, pero solo puedan eliminar o modificar archivos propios, evitando interferencias entre usuarios y preservando la integridad de los contenidos gestionados por la administración.
/download: Propiedad de root:sftpusers con permisos 755 permitiendo a la administración de archivos para que el usuario los descargue.

Esta estructura evita que el usuario pueda modificar la raíz de su propia jaula, un requisito para prevenir escapes del entorno chroot.

```
administradorSSH-SFTP@ServidorSFTP:~$ sudo mkdir -p /srv/sftp
administradorSSH-SFTP@ServidorSFTP:~$ sudo mkdir /srv/sftp/download
administradorSSH-SFTP@ServidorSFTP:~$ sudo mkdir /srv/sftp/upload
administradorSSH-SFTP@ServidorSFTP:~$ sudo chown root:root /srv/sftp
administradorSSH-SFTP@ServidorSFTP:~$ sudo chown root:SFTPUsers /srv/sftp/upload
administradorSSH-SFTP@ServidorSFTP:~$ sudo chown root:SFTPUsers /srv/sftp/download
administradorSSH-SFTP@ServidorSFTP:~$ sudo chmod 755 /srv/sftp
administradorSSH-SFTP@ServidorSFTP:~$ sudo chmod 1775 /srv/sftp/upload
administradorSSH-SFTP@ServidorSFTP:~$ sudo chmod 755 /srv/sftp/download
administradorSSH-SFTP@ServidorSFTP:~$ _
```

Comandos para establecer la estructura de directorios y los permisos correctos para la jaula SFTP.

5.2. Configuracion Reglas de SSH

La configuración final se realiza en el archivo de configuración del demonio SSH, */etc/ssh/sshd_config*. Se utiliza un bloque Match Group para aplicar reglas específicas solo a los usuarios que pertenecen al grupo SFTPUsers (el grupo sincronizado en LDAP).
Match Group SFTPUsers: Aplica las siguientes directivas solo a los miembros de este grupo.

- ***ChrootDirectory :*** Define el directorio raíz de la jaula.
- ***ForceCommand internal-sftp:*** Fuerza la ejecución del subsistema SFTP interno, impidiendo que el usuario pueda solicitar una shell de comandos. Este es el control clave que limita al usuario solo a SFTP.
- ***AllowTcpForwarding no y X11Forwarding no:*** Deshabilita el reenvío de puertos y de X11, cerrando posibles túneles que podrían usarse para eludir las restricciones de red.

```
  GNU nano 2.9.3                        /etc/ssh/sshd_config

#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /var/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 internal-sftp
Port 22

Match Group Admins_SSH
        ForceCommand /bin/bash
        PermitTTY yes

Match Group SFTPUsers
        ChrootDirectory /srv/sftp
        AllowTcpForwarding no
        PasswordAuthentication yes
        X11Forwarding no
        PermitTTY no
        PermitTunnel no
        ForceCommand internal-sftp

^G Ver ayuda  ^O Guardar     ^W Buscar      ^K Cortar Text^J Justificar ^C Posición   M-U Deshacer
^X Salir      ^R Leer fich.  ^\ Reemplazar  ^U Pegar txt  ^T Ortografía ^_ Ir a línea M-E Rehacer
```

Directivas en sshd_config para activar el Chroot Jail para el grupo SFTPUsers.

Parte 6: Verificación y Pruebas desde el Cliente

La fase final del proyecto consistió en una serie de pruebas exhaustivas para validar que todos los controles de seguridad y funcionalidades operaran según lo diseñado.

6.1. Preparación del Entorno Cliente

Para las pruebas de transferencia de archivos, se utilizó **FileZilla**, un cliente SFTP/FTP de código abierto, multiplataforma y ampliamente reconocido. Su interfaz gráfica facilita la visualización del entorno de archivos restringido y la ejecución de operaciones de carga y descarga.

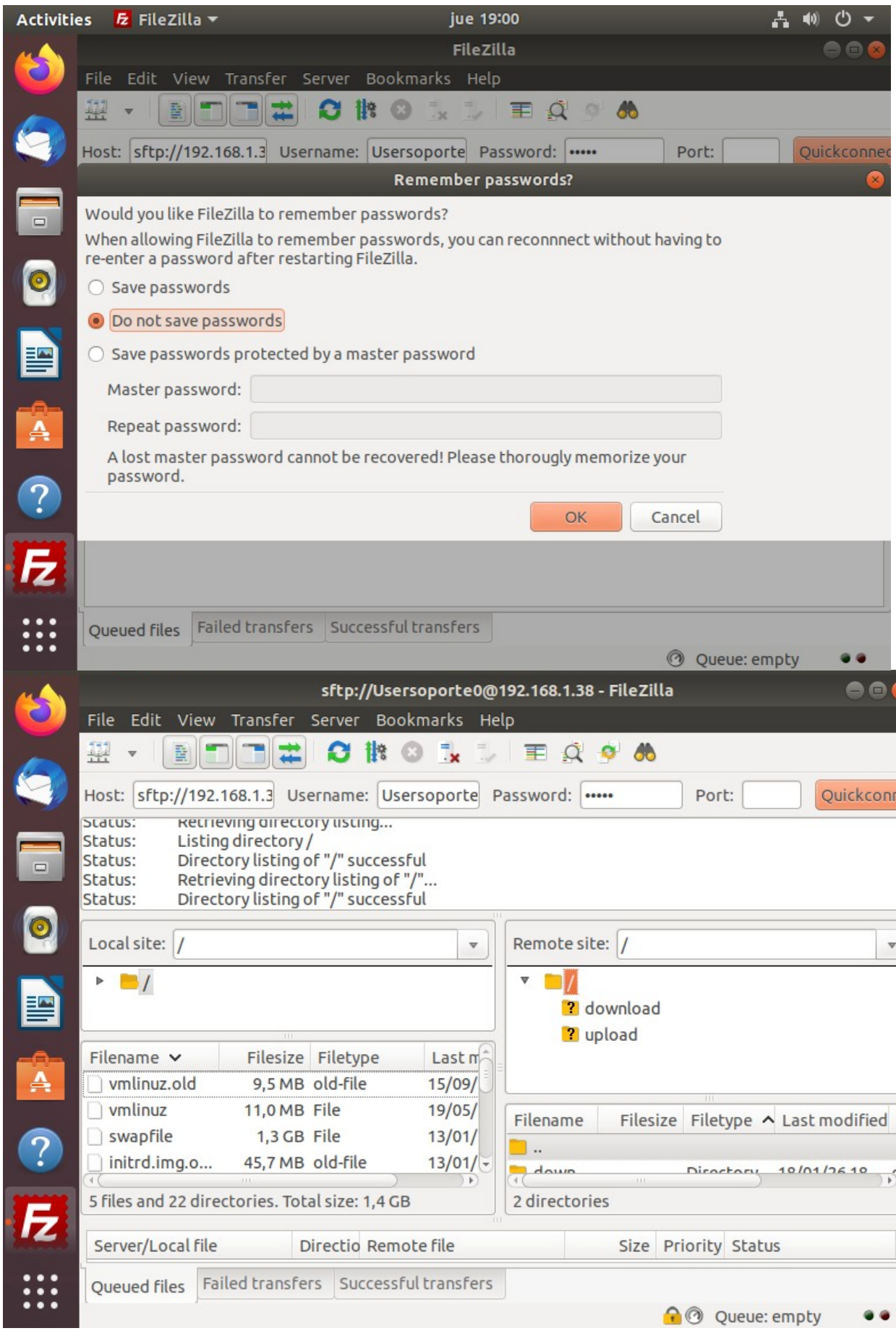
La instalación se realizó mediante el gestor de paquetes estándar en la máquina cliente.

```
sudo apt update
sudo apt install filezilla -y
```

FileZilla en la estación de trabajo de pruebas.

6.2. Prueba de Acceso de Usuario SFTP (Chroot Jail)

Objetivo: Verificar que un usuario del grupo SFTPUsers de LDAP puede autenticarse y es correctamente "encarcelado" en su directorio chroot.

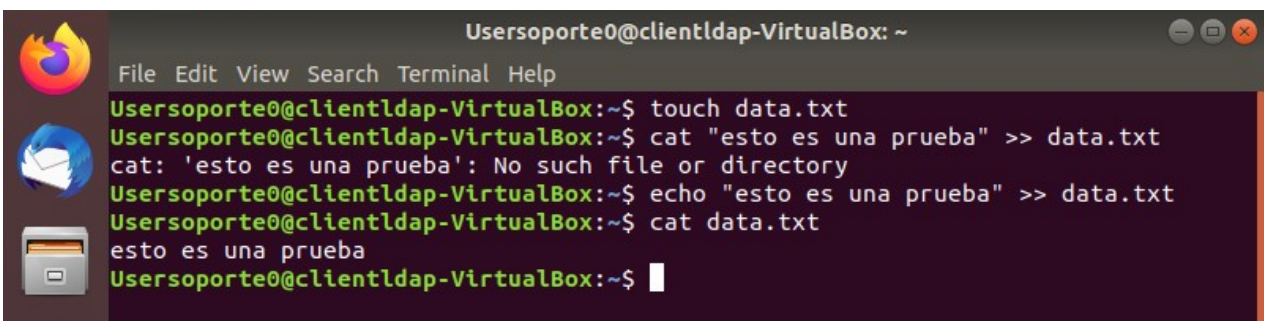


Conexión exitosa de un usuario SFTP, mostrando el entorno Chroot.

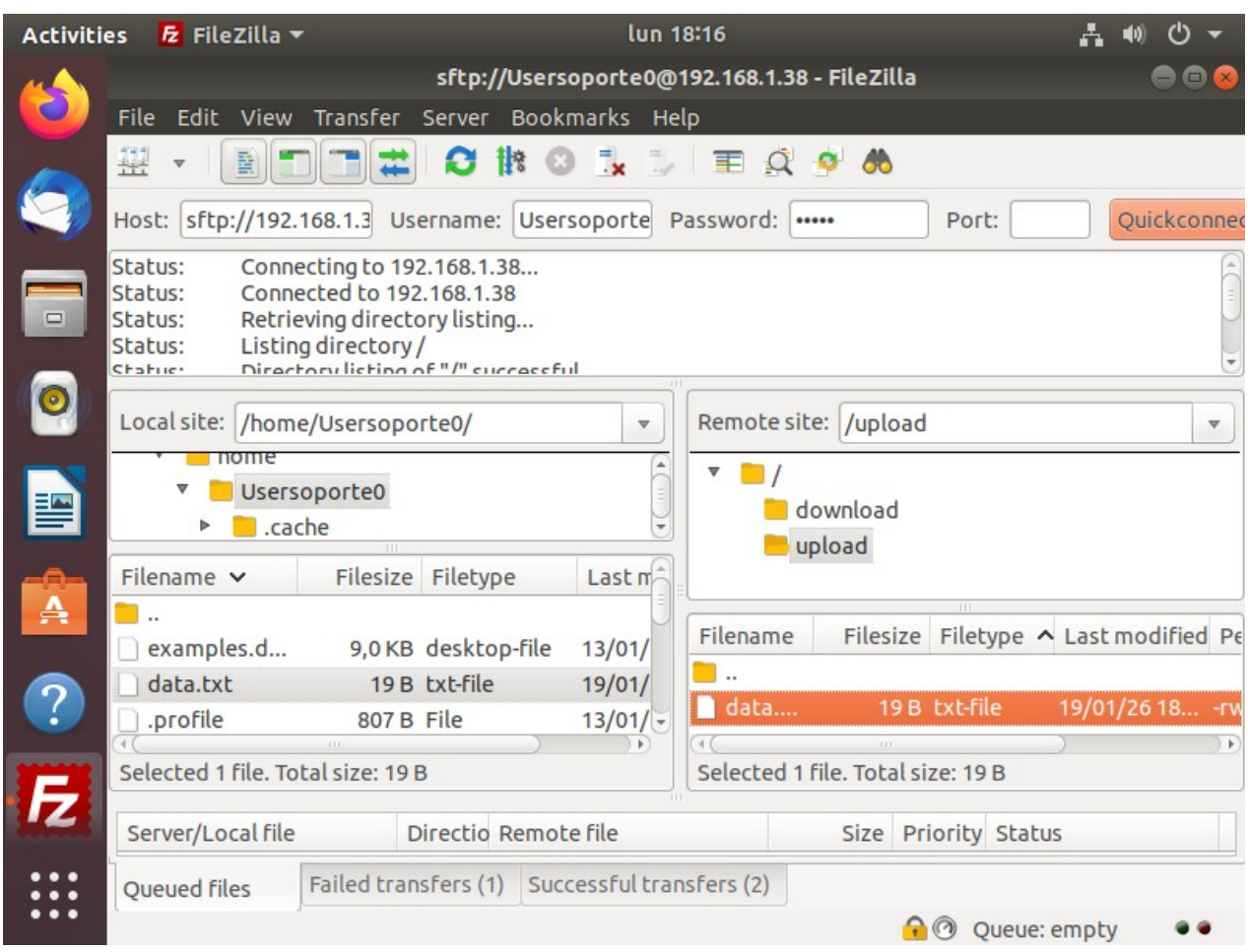
Resultado: La conexión fue exitosa. Como se observa en la captura, el cliente FileZilla muestra que la raíz del sistema de archivos para el usuario es su directorio asignado (/), y solo puede ver los subdirectorios upload y download. No tiene visibilidad del sistema de archivos real del servidor.
Práctica de Seguridad del Cliente: Se deshabilitó la opción de guardar contraseñas en FileZilla. Esto obliga a una re-autenticación en cada sesión, mitigando el riesgo de acceso no autorizado desde una estación de trabajo comprometida.

6.3. Verificación de Carga de Archivos (Upload)

Objetivo: Confirmar que el usuario tiene permisos de escritura en el directorio /upload.



Creacion del archivo a subir

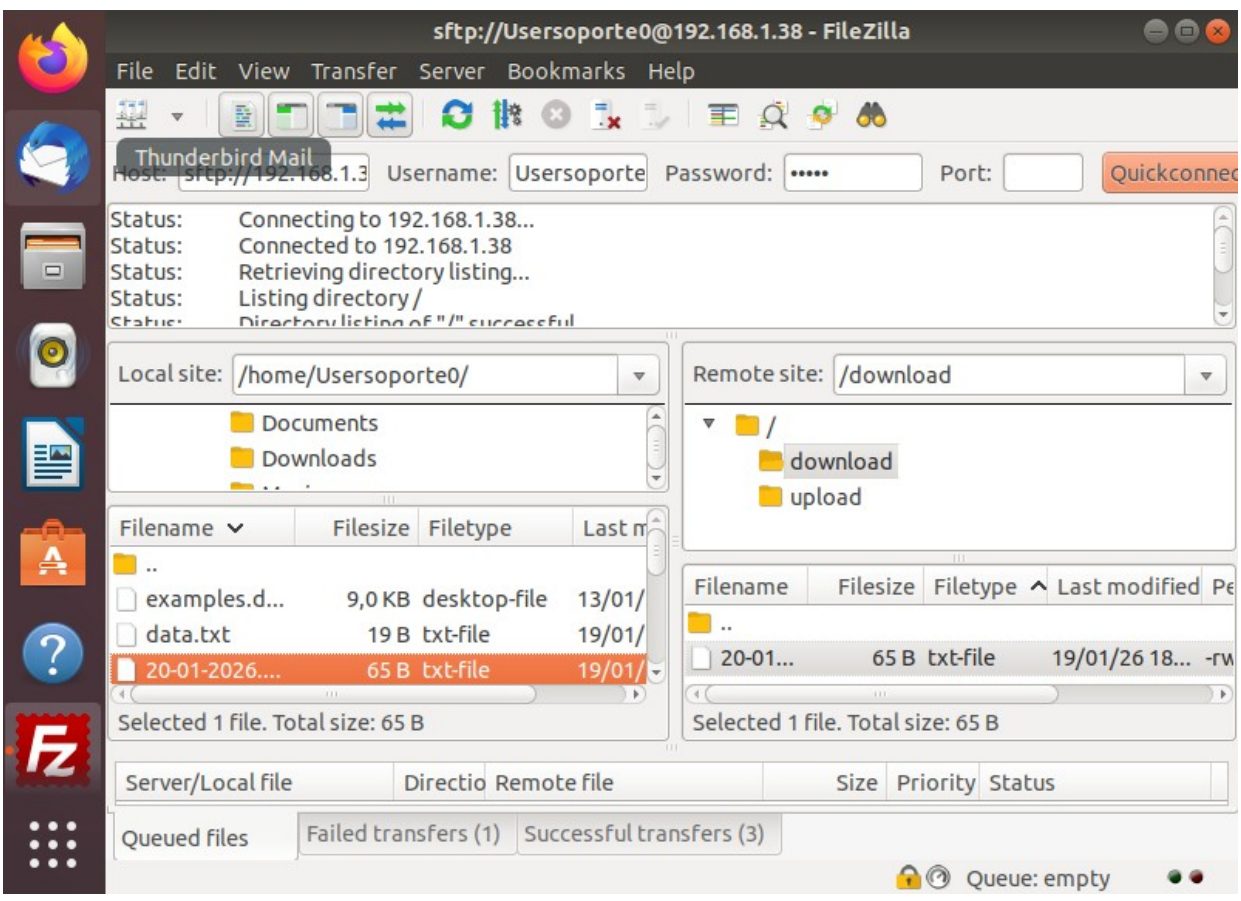


Transferencia exitosa de un archivo al directorio de subida asignado.

Resultado: Se transfirió un archivo desde el cliente al directorio ***/upload*** del servidor. La transferencia se completó exitosamente, validando que los permisos de propiedad y grupo están correctamente aplicados, permitiendo la ingesta de datos por parte del usuario.

6.5. Verificación de Descarga de Archivos (Download)

Objetivo: Asegurar que el usuario puede recuperar archivos depositados por la administración en el directorio /download.

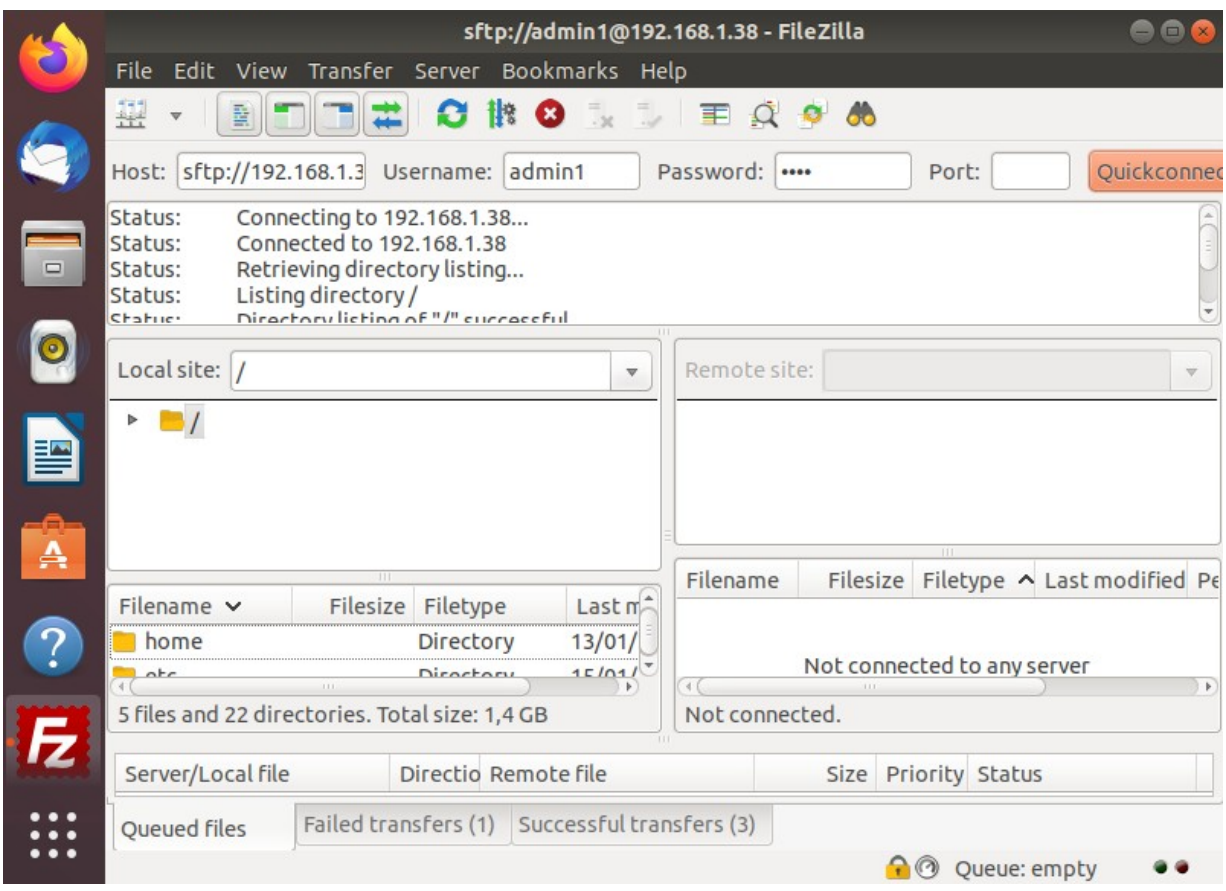


Descarga exitosa de un archivo desde la zona compartida de descarga.

Resultado: Se seleccionó un archivo previamente colocado en la carpeta ***/download*** del servidor y se inició una transferencia hacia el cliente. La operación se completó con éxito, confirmando que los permisos de lectura para el grupo operativo son funcionales.

6.6. Prueba de Seguridad: Intento de Acceso No Autorizado (Negative Testing)

Objetivo: Verificar que un usuario existente en LDAP pero que no pertenece al grupo SFTPUsers no puede acceder al servicio.

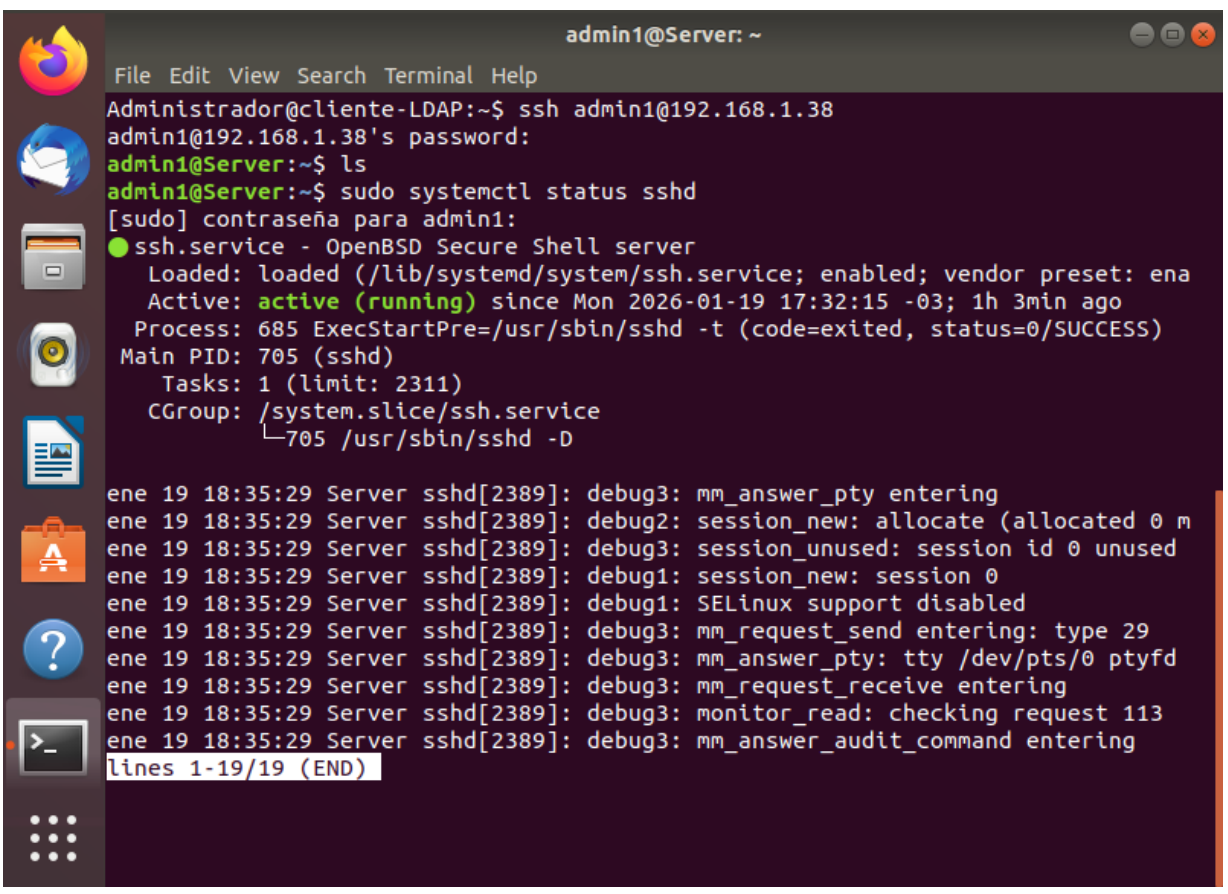


El servidor rechaza correctamente la conexión de un usuario no perteneciente al grupo SFTP.

Resultado: Se intentó una conexión con las credenciales de un usuario válido pero no autorizado. El servidor rechazó la conexión inmediatamente con un error de ";Permission denied" o "Authentication failed". Esta prueba es crucial, ya que demuestra la efectividad del filtro Match Group en sshd_config, garantizando que solo el personal explícitamente autorizado pueda interactuar con el servicio SFTP.

6.7. Verificación de Acceso Remoto Administrativo (SSH)

Objetivo: Validar que un usuario del grupo local Admins_SSH puede establecer una sesión de shell interactiva completa a través de SSH y ejecutar comandos con privilegios elevados.



Conexión SSH exitosa de un administrador, con acceso a una shell interactiva y privilegios sudo.

Resultado: Se realizó una conexión SSH con la cuenta de un administrador. A diferencia de los usuarios de SFTP, el sistema otorgó una shell de comandos completa. Se verificó la capacidad de ejecutar comandos con sudo (ej. sudo systemctl status sshd), confirmando el acceso administrativo total.

Conclusión y Justificación

Se optó por una implementación puramente sobre Linux para el servicio SFTP. Aunque Windows permite el uso de OpenSSH, la gestión de permisos granulares (ACLs), el aislamiento de usuarios (Chroot) y la eficiencia de recursos hacen de Linux el estándar de la industria para este protocolo. La implementación en Windows se considera una solución de nicho con una sobrecarga administrativa innecesaria para este flujo de trabajo.