



Homework 5: SSH Configuration

Môn học: Hệ điều hành Linux và Ứng dụng

CS11117 - 22MMT

Sinh viên: Nguyễn Hồ Đăng Duy - 22127085

Table of Contents

- [1: Disable Root SSH Access](#)
 - [2: Create SFTP-Only User \(No Shell\)](#)
 - [3: Configure SSH Key-Based Authentication](#)
-

1: Disable Root SSH Access

1.1 Edit SSH configuration file

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

Change or add the line:

```
PermitRootLogin no
```

Explanation:

- This command opens the SSH daemon configuration file. Setting `PermitRootLogin no` disables direct SSH login using the root account, reducing the attack surface and improving security.

Screenshot:

Edited `sshd_config` with `PermitRootLogin no` set.



```
denver@kali: ~
File Actions Edit View Help
GNU nano 8.3 /etc/ssh/sshd_config *

#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
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
PermitRootLogin no

[ Cancelled ]
^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^N Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo
```

1.2 Restart SSH service

```
sudo systemctl restart sshd
```

Explanation:

- Applies the changes made to the SSH config. Without restarting, modifications won't take effect.

Screenshot:

Restart ssh service.

```
(denver@kali)-[~]
$ sudo nano /etc/ssh/sshd_config
[sudo] password for denver:

(denver@kali)-[~]
$ sudo systemctl restart sshd

(denver@kali)-[~]
$
```

1.3 Result:

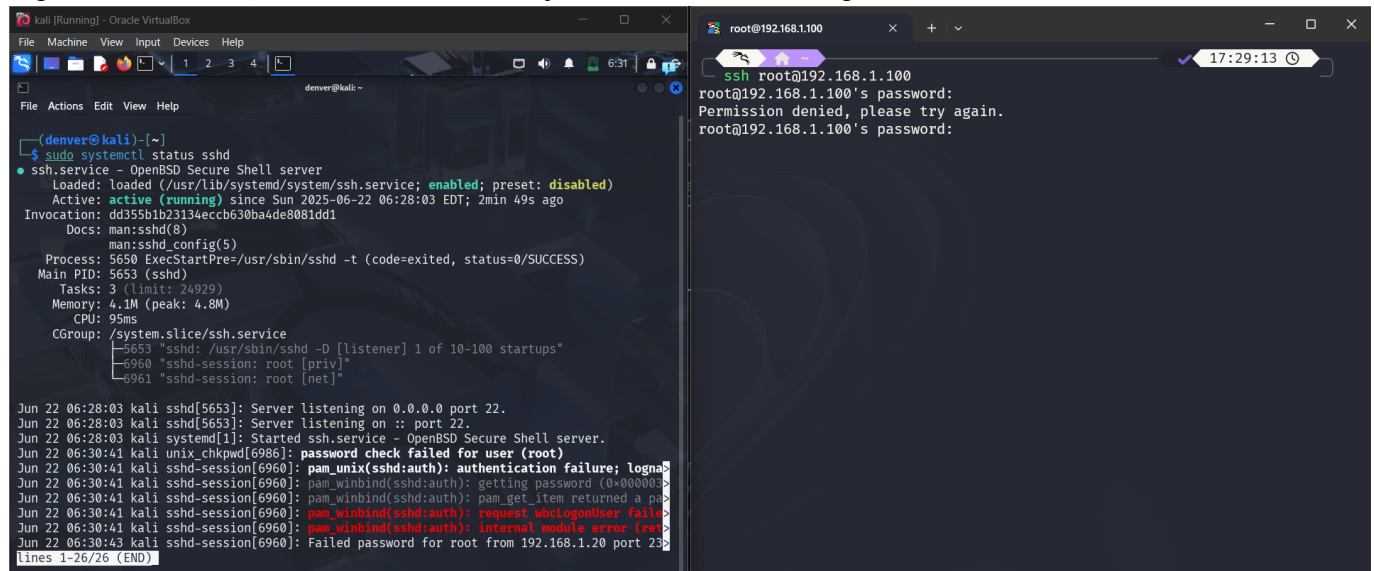
After applying the change, attempts to login via `ssh root@192.168.1.100` should be rejected with a message such as `Permission denied`. This confirms root SSH access is successfully disabled.

Output of `systemctl status sshd` showing SSH service is running.

```
(denver@kali)-[~]
$ sudo systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: disabled)
   Active: active (running) since Sun 2025-06-22 06:28:03 EDT; 1min 27s ago
  Invocation: dd355b1b23134eccb630ba4de8081dd1
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 5650 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 5653 (sshd)
    Tasks: 1 (limit: 24929)
   Memory: 1.4M (peak: 2M)
      CPU: 51ms
   CGroup: /system.slice/ssh.service
           └─5653 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jun 22 06:28:03 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Jun 22 06:28:03 kali sshd[5653]: Server listening on 0.0.0.0 port 22.
Jun 22 06:28:03 kali sshd[5653]: Server listening on :: port 22.
Jun 22 06:28:03 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
```

Login via `ssh root@192.168.1.100` was rejected and we can see log from SSH service



```

kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
denver@kali: ~
└─$ sudo systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: disabled)
   Active: active (running) since Sun 2025-06-22 06:28:03 EDT; 2min 49s ago
 Invocation: dd355b1b23134eccb630ba4de8081dd1
    Docs: man:sshd(8)
          man:sshd_config(5)
   Process: 5650 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 5653 (sshd)
    Tasks: 3 (limit: 24929)
   Memory: 4.1M (peak: 4.8M)
      CPU: 95ms
   CGroup: /system.slice/ssh.service
           └─5653 "sshd: /usr/sbin/sshd -D [listener] 1 of 10-100 startups"
             └─6960 "sshd-session: root [priv]"
               └─6961 "sshd-session: root [net]"

Jun 22 06:28:03 kali sshd[5653]: Server listening on 0.0.0.0 port 22.
Jun 22 06:28:03 kali sshd[5653]: Server listening on :: port 22.
Jun 22 06:28:03 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
Jun 22 06:30:41 kali unix_chkpwd[6986]: password check failed for user (root)
Jun 22 06:30:41 kali sshd-session[6960]: pam_unix(sshd:auth): authentication failure; logname=
Jun 22 06:30:41 kali sshd-session[6960]: pam_winbind(sshd:auth): getting password (0x00000000)
Jun 22 06:30:41 kali sshd-session[6960]: pam_winbind(sshd:auth): pam_get_item returned a pas
Jun 22 06:30:41 kali sshd-session[6960]: pam_winbind(sshd:auth): request_wbctLogonUser failu
Jun 22 06:30:41 kali sshd-session[6960]: pam_winbind(sshd:auth): internal module error (ret
Jun 22 06:30:43 kali sshd-session[6960]: Failed password for root from 192.168.1.20 port 23
lines 1-26/26 (END)

```

2: Create SFTP-Only User (No Shell)

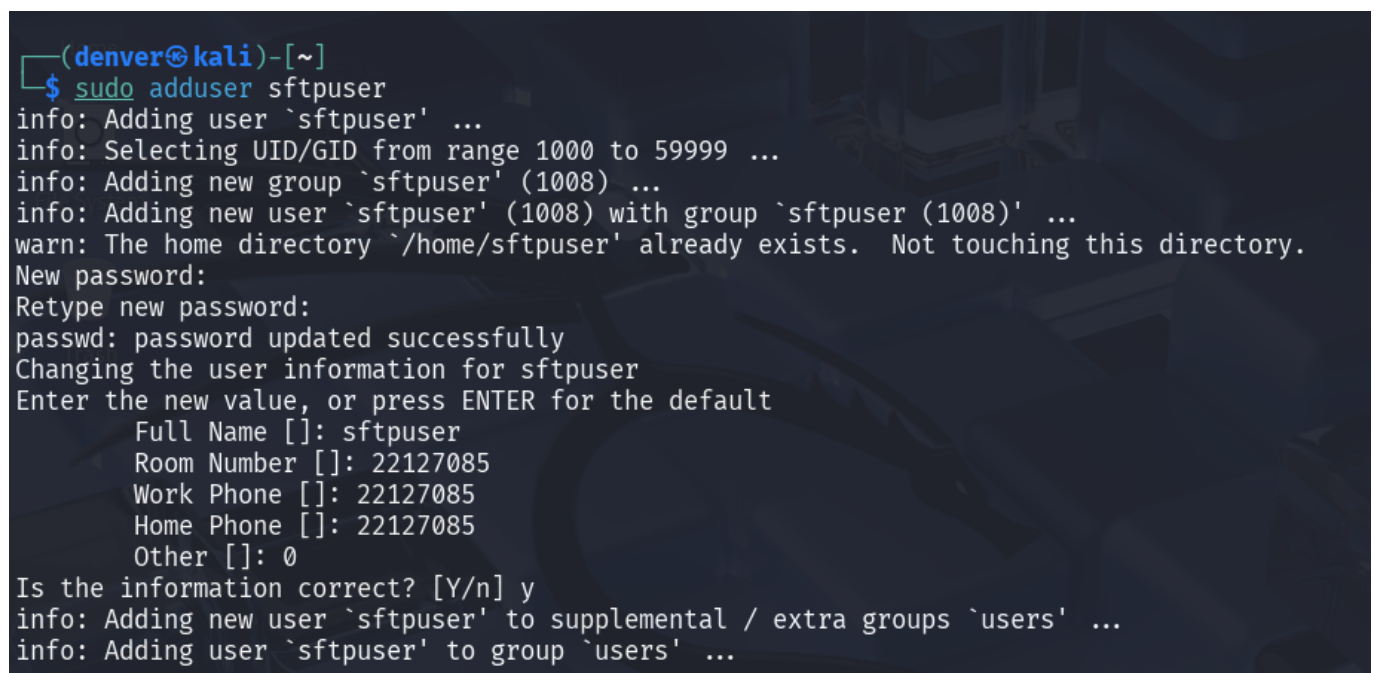
2.1 Create user and set password

```
sudo adduser sftpuser
```

Explanation:

- `adduser` creates a new account. This account will be used for SFTP only, no shell access.

Screenshot:



```

(denver@kali)-[~]
└─$ sudo adduser sftpuser
info: Adding user `sftpuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `sftpuser' (1008) ...
info: Adding new user `sftpuser' (1008) with group `sftpuser (1008)' ...
warn: The home directory `/home/sftpuser' already exists. Not touching this directory.
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for sftpuser
Enter the new value, or press ENTER for the default
  Full Name []: sftpuser
   Room Number []: 22127085
    Work Phone []: 22127085
    Home Phone []: 22127085
       Other []: 0
Is the information correct? [Y/n] y
info: Adding new user `sftpuser' to supplemental / extra groups `users' ...
info: Adding user `sftpuser' to group `users' ...

```

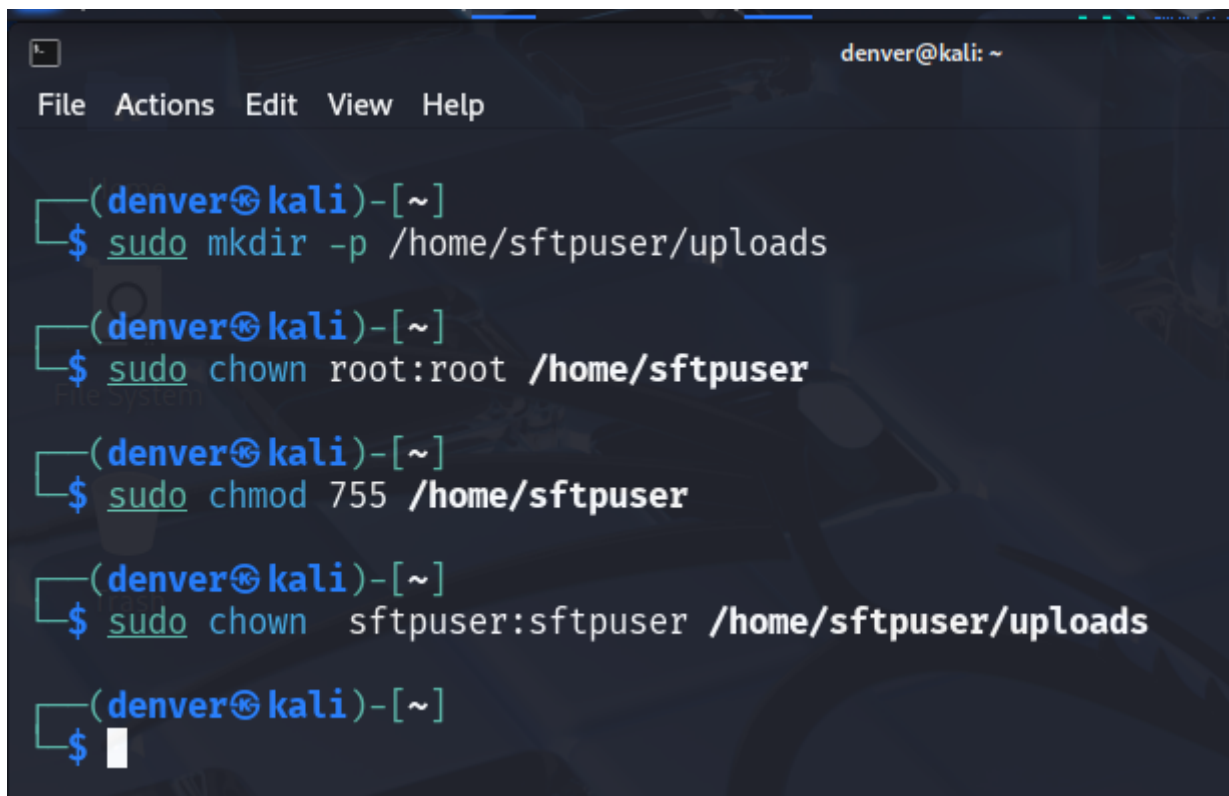
2.2 Setup chroot directory

```
sudo mkdir -p /home/sftpuser/uploads
sudo chown root:root /home/sftpuser
sudo chmod 755 /home/sftpuser
sudo chown sftpuser:sftpuser /home/sftpuser/uploads
```

Explanation:

`mkdir -p` ensures upload folder exists. `chown root:root` and `chmod 755` are required for SSH chroot jails. User must only have ownership inside the uploads folder.

Screenshot:

A screenshot of a terminal window titled 'denver@kali: ~'. The terminal shows a series of commands being executed in a shell. The prompt is '(denver@kali)-[~]'. The commands are: 'sudo mkdir -p /home/sftpuser/uploads', 'sudo chown root:root /home/sftpuser', 'sudo chmod 755 /home/sftpuser', and 'sudo chown sftpuser:sftpuser /home/sftpuser/uploads'. The terminal output shows the commands being executed successfully. The prompt is now '\$'.

2.3 Update SSH configuration

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

Add to bottom:

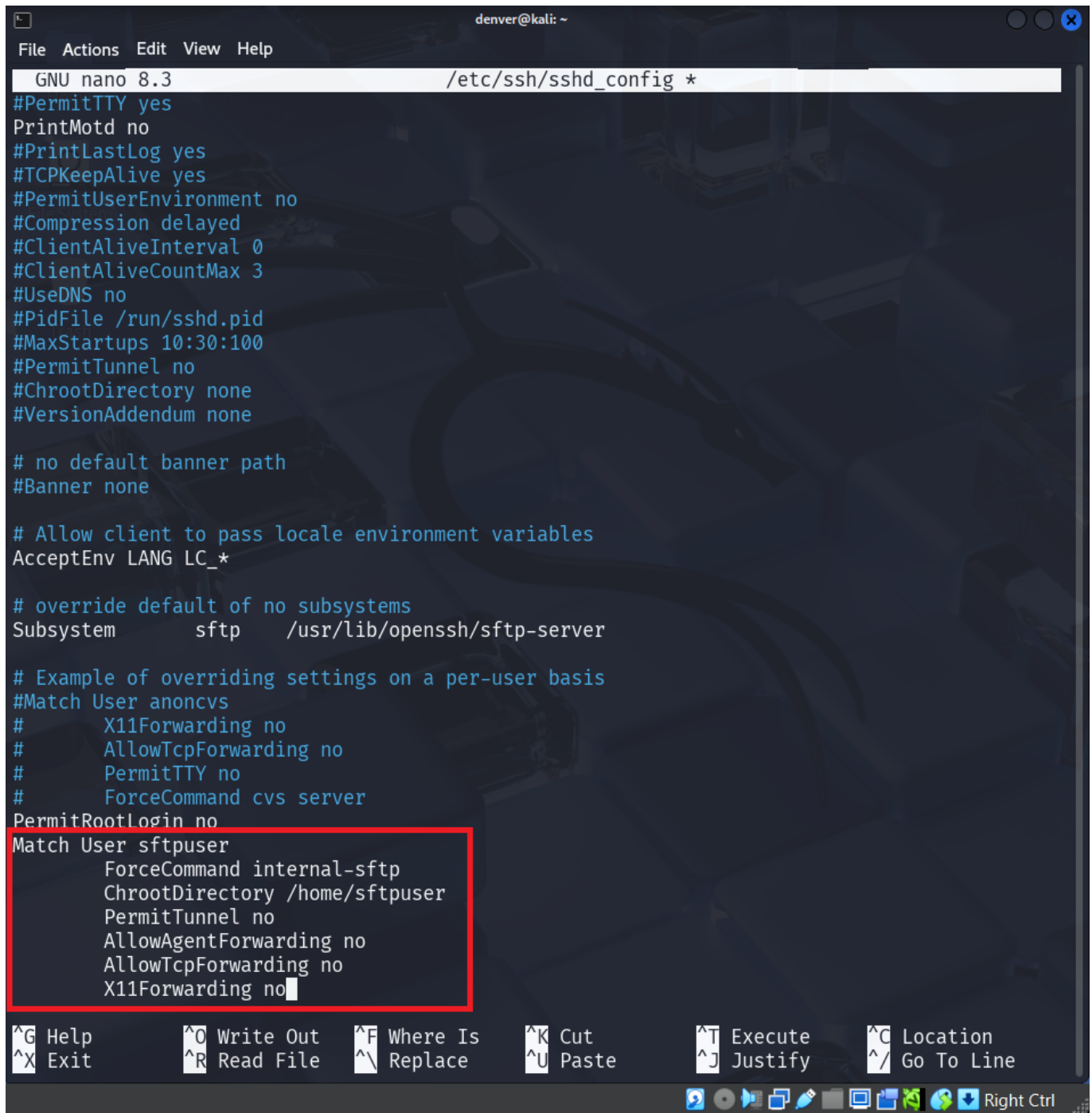
```
Match User sftpuser
    ForceCommand internal-sftp
    ChrootDirectory /home/sftpuser
    PermitTunnel no
    AllowAgentForwarding no
    AllowTcpForwarding no
    X11Forwarding no
```

Explanation:

This restricts `sftpuser` to SFTP only. `ChrootDirectory` confines the user, preventing access to other system areas:

- `Match User sftpuser`: a conditional block: only the following lines will apply to `sftpuser`.
- `ForceCommand internal-sftp`: Prevents user from executing arbitrary commands — limits to file transfer only.
- `ChrootDirectory /home/sftpuser`: Enhances security by sandboxing the user into a limited environment.
- `PermitTunnel no`: Disables port tunneling features for `sftpuser`.
- `AllowAgentForwarding no`: Security hardening — avoids abuse of forwarded authentication.
- `AllowTcpForwarding no`: Blocks `sftpuser` from using SSH to access or route other network traffic.
- `X11Forwarding no`: Disables X11 forwarding (GUI over SSH).

Screenshot:



```
denver@kali: ~
File Actions Edit View Help
GNU nano 8.3 /etc/ssh/sshd_config *
#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
PermitRootLogin no
Match User sftpuser
    ForceCommand internal-sftp
    ChrootDirectory /home/sftpuser
    PermitTunnel no
    AllowAgentForwarding no
    AllowTcpForwarding no
    X11Forwarding no
^G Help      ^O Write Out ^F Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify  ^_ Go To Line
Right Ctrl
```

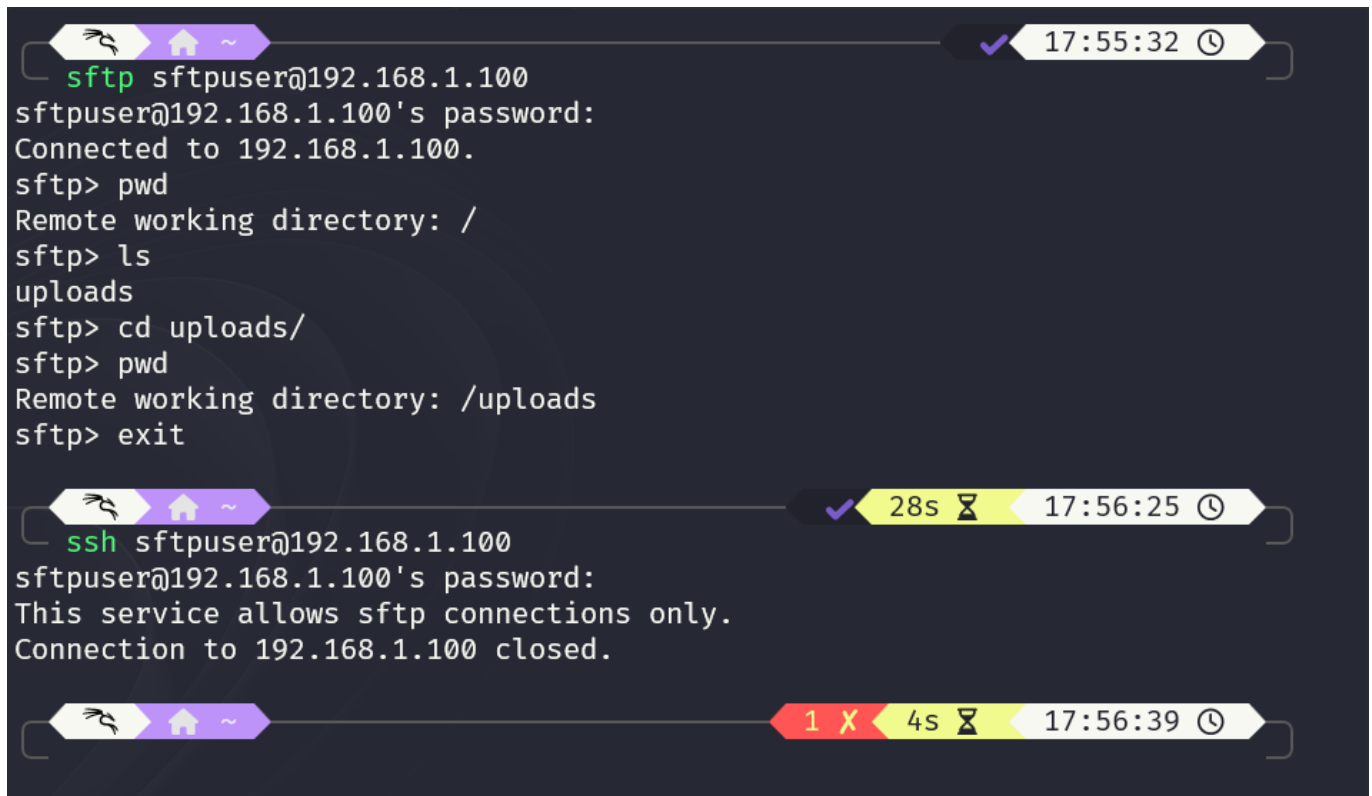
2.4 Restart SSH

```
sudo systemctl restart sshd
```

2.5 Result:

When logging in using `sftp sftpuser@192.168.1.100`, the user should be able to upload and download files. If attempting to SSH with `ssh sftpuser@192.168.1.100`, the session will immediately close or be denied, proving shell access is blocked.

Screenshot:



The screenshot displays three terminal sessions. The first session shows an sftp connection to 192.168.1.100, where the user 'sftpuser' enters their password, is connected, and then navigates to the '/uploads' directory. The second session shows an ssh connection to the same IP, which fails with the message 'This service allows sftp connections only.' The third session is partially visible at the bottom.

```
sftp sftpuser@192.168.1.100
sftpuser@192.168.1.100's password:
Connected to 192.168.1.100.
sftp> pwd
Remote working directory: /
sftp> ls
uploads
sftp> cd uploads/
sftp> pwd
Remote working directory: /uploads
sftp> exit
```

```
ssh sftpuser@192.168.1.100
sftpuser@192.168.1.100's password:
This service allows sftp connections only.
Connection to 192.168.1.100 closed.
```

3: Configure SSH Key-Based Authentication

3.1 Create new user

```
sudo adduser studentuser
```

Screenshot:


```
(denver@kali)-[~]
$ sudo adduser studentuser
info: Adding user `studentuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `studentuser' (1009) ...
info: Adding new user `studentuser' (1009) with group `studentuser (1009)' ...
info: Creating home directory `/home/studentuser' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for studentuser
Enter the new value, or press ENTER for the default
  Full Name []: studentuser
  Room Number []: 22127085
  Work Phone []: 22127085
  Home Phone []: 22127085
  Other []: 1
Is the information correct? [Y/n] y
info: Adding new user `studentuser' to supplemental / extra groups `users' ...
info: Adding user `studentuser' to group `users' ...

(denver@kali)-[~]
$
```

3.2 On client: generate SSH key (if not already)

```
ssh-keygen
```

Explanation:

Generates a public/private RSA key pair in `~/.ssh/id_rsa` and `id_rsa.pub`. These are used for secure login without passwords.

3.3 Copy public key to server

```
ssh-copy-id studentuser@192.168.1.100
```

Explanation:

Places the public key in the authorized list. Only users with matching private key can log in.

Screenshot:

```

ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/denver/.ssh/id_ed25519):
Enter passphrase for "/home/denver/.ssh/id_ed25519" (empty for no passphrase
):
Enter same passphrase again:
Your identification has been saved in /home/denver/.ssh/id_ed25519
Your public key has been saved in /home/denver/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:PIWqMZVH0JuXCCQxxxzWH6MUeZQaLEZpDvDrH7qdt2Y denver@LAPTOP-0DQEKJM4
The key's randomart image is:
+--[ED25519 256]--+
|      +=BXBO..      |
|      =+o@o*         |
|      o=+X.+         |
|      . =*+.         |
|      o . S..         |
|      +   + .         |
|      .   . .         |
|      o .E           |
|      . o+..         |
+---[SHA256]-----+

ssh-copy-id studentuser@192.168.1.100
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/denver/
.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to fil
ter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are pr
ompted now it is to install the new keys
studentuser@192.168.1.100's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'studentuser@192.168.1.100'"
and check to make sure that only the key(s) you wanted were added.

```

3.4 Disable password authentication (on server)

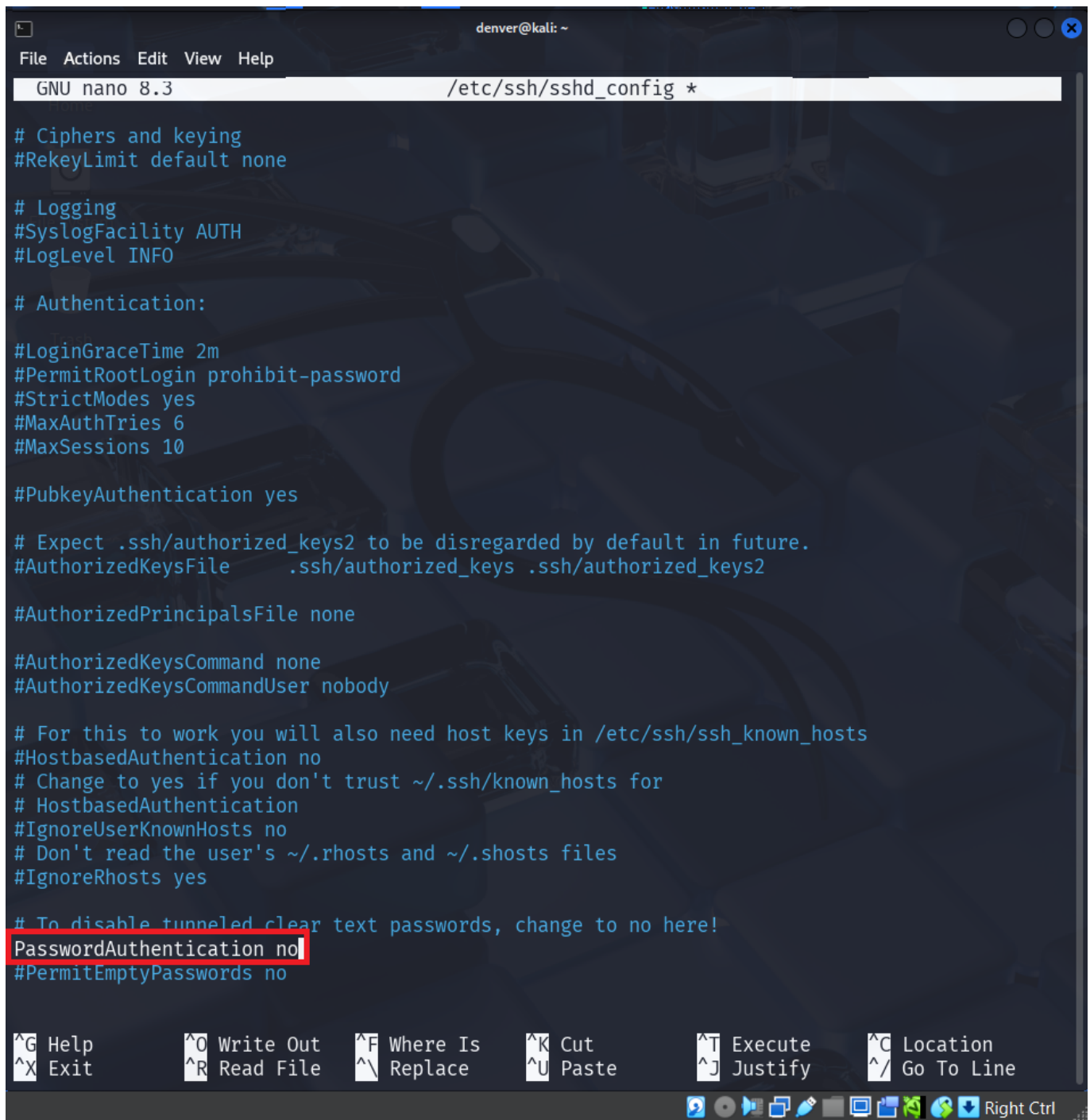
```
sudo nano /etc/ssh/sshd_config
```

Set:

```
PasswordAuthentication no
```

Then:

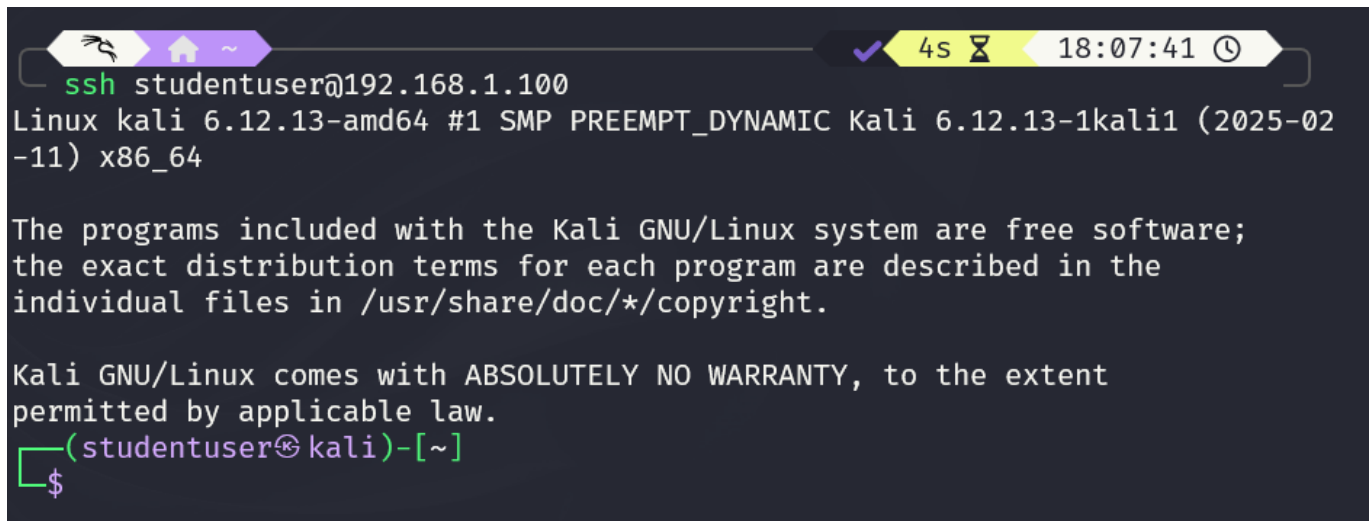
```
sudo systemctl restart sshd
```

Screenshot:

```
denver@kali: ~  
File Actions Edit View Help  
GNU nano 8.3 /etc/ssh/sshd_config *  
  
# Ciphers and keying  
#RekeyLimit default none  
  
# Logging  
#SyslogFacility AUTH  
#LogLevel INFO  
  
# Authentication:  
  
#LoginGraceTime 2m  
#PermitRootLogin prohibit-password  
#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  
#IgnoreUserKnownHosts no  
# Don't read the user's ~/.rhosts and ~/.shosts files  
#IgnoreRhosts yes  
  
# To disable tunneled clear text passwords, change to no here!  
PasswordAuthentication no  
#PermitEmptyPasswords no  
  
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  
Right Ctrl
```

3.5 Result:

Logging in with `ssh studentuser@192.168.1.100` will succeed without asking for a password if the correct private key is available. If trying to login without a key or as a different user, access will be denied.



A terminal window showing a successful SSH login. The top status bar includes a terminal icon, a home icon, a tilde icon, a checkmark, a 4s timer, and a clock showing 18:07:41. The command `ssh studentuser@192.168.1.100` is entered. The output shows the system information: `Linux kali 6.12.13-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.13-1kali1 (2025-02-11) x86_64`. Below this, a message states: "The programs included with the Kali GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright." Another message follows: "Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law." The prompt changes to `(studentuser@kali)-[~]` and a dollar sign `$` is entered.

```
ssh studentuser@192.168.1.100
Linux kali 6.12.13-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.13-1kali1 (2025-02-11) x86_64

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

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
(studentuser@kali)-[~]
$
```

SSH login for *studentuser* using key-based authentication.



A terminal window showing failed SSH login attempts. The top status bar includes a terminal icon, a home icon, a tilde icon, a checkmark, and a clock showing 18:08:43. The command `ssh denver@192.168.1.100` is entered. The output is `denver@192.168.1.100: Permission denied (publickey).` Below this, the same command is entered again. The bottom status bar shows a red 255 X error indicator and a clock showing 18:10:48.

```
ssh denver@192.168.1.100
denver@192.168.1.100: Permission denied (publickey).

ssh denver@192.168.1.100
denver@192.168.1.100: Permission denied (publickey).
```

Login attempt using wrong user or without key failing.