OS Security

Assignment 1:

Assignment report for creating two Linux user accounts and setting file permissions so that only one user can read and write to a file, while the other user has no access:

Objective

The objective of this assignment is to create two separate user accounts on a Linux system and configure a file with permissions allowing only one user to read and write to it, while explicitly denying any access to the other user.

Environment

Ubuntu 22.04 LTS running on a virtual or physical machine with root or sudo access.

Procedure

1. **Open a terminal with sudo privileges**
    - Access your Linux system and open a terminal.
    - Ensure you have administrative (sudo) rights to create users.
2. **Create the first user**
    - Run: **sudo adduser user1**
    - Follow the prompts to set a password and user details.
3. **Create the second user**
    - Run: **sudo adduser user2**
    - Similarly, set a password and complete the setup.
4. **Switch to user1**

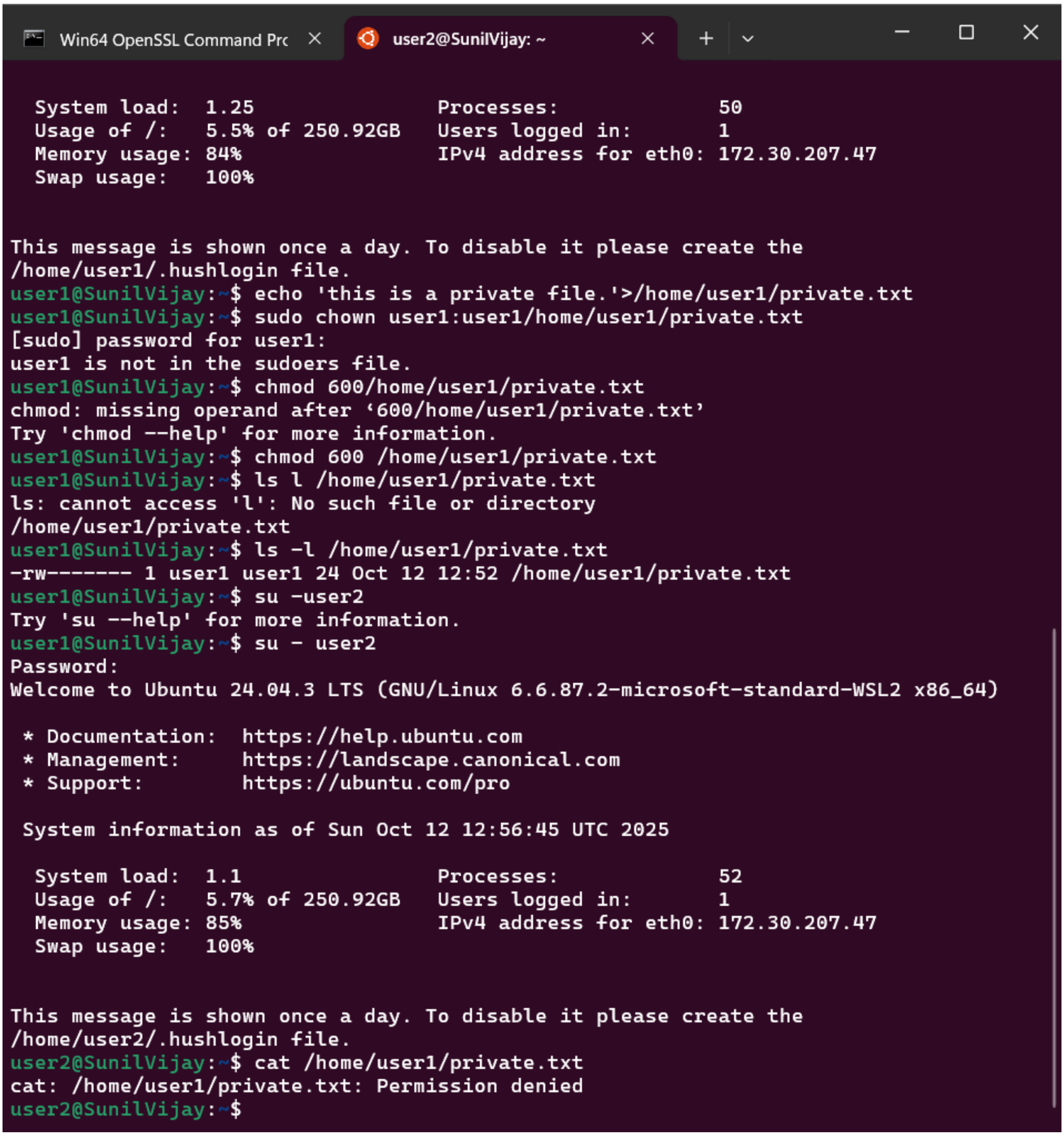
    Run: **su - user1**
    - Enter the password for user1 when prompted.
5. **Create a private file**
    - Run: echo 'This is a private file.' > /home/user1/private.txt
    - This creates a file in user1's home directory.
6. **Set file ownership**
    - Run: sudo chown user1:user1 /home/user1/private.txt
    - Ensures the file is owned by user1 and their primary group.
7. **Set restrictive permissions**
    - Run: chmod 600 /home/user1/private.txt
    - This allows only the owner (user1) to read and write the file.

8. **Verify the permissions**
    - Run: ls -l /home/user1/private.txt
    - You should see output like: *-rw------- 1 user1 user1 ...*
9. **Switch to user2**

    Open a new terminal or log out and run: su - user2

    - Enter user2's password.
10. **Test access from user2**
    - Try: cat /home/user1/private.txt
    - This should fail with a permission denied error, confirming user2 has no access.
11. **Return to user1 to confirm access**
    - Switch back to user1 and verify they can still read and write the file.

This setup ensures that only user1 has read and write access, while user2 (and all others) are denied access



```
  System load:  1.25               Processes:             50
  Usage of /:   5.5% of 250.92GB   Users logged in:       1
  Memory usage: 84%                IPv4 address for eth0: 172.30.207.47
  Swap usage:   100%


This message is shown once a day. To disable it please create the
/home/user1/.hushlogin file.
user1@SunilVijay:~$ echo 'this is a private file.'>/home/user1/private.txt
user1@SunilVijay:~$ sudo chown user1:user1/home/user1/private.txt
[sudo] password for user1:
user1 is not in the sudoers file.
user1@SunilVijay:~$ chmod 600/home/user1/private.txt
chmod: missing operand after ‘600/home/user1/private.txt’
Try 'chmod --help' for more information.
user1@SunilVijay:~$ chmod 600 /home/user1/private.txt
user1@SunilVijay:~$ ls l /home/user1/private.txt
ls: cannot access 'l': No such file or directory
/home/user1/private.txt
user1@SunilVijay:~$ ls -l /home/user1/private.txt
-rw------- 1 user1 user1 24 Oct 12 12:52 /home/user1/private.txt
user1@SunilVijay:~$ su -user2
Try 'su --help' for more information.
user1@SunilVijay:~$ su - user2
Password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.6.87.2-microsoft-standard-WSL2 x86_64)

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

 System information as of Sun Oct 12 12:56:45 UTC 2025

  System load:  1.1                Processes:             52
  Usage of /:   5.7% of 250.92GB   Users logged in:       1
  Memory usage: 85%                IPv4 address for eth0: 172.30.207.47
  Swap usage:   100%


This message is shown once a day. To disable it please create the
/home/user2/.hushlogin file.
user2@SunilVijay:~$ cat /home/user1/private.txt
cat: /home/user1/private.txt: Permission denied
user2@SunilVijay:~$
```

Results

- Two users successfully created (user1, user2).
- File exclusive_file.txt created with owner user1 and permission 600.
- user1 can read and write to the file.
- user2 is denied access, achieving the objective of exclusive access.

Conclusion

This assignment demonstrates Linux user and file permission management by enforcing strict access control on file resources. Using permission 600 effectively restricts the file to owner-only access, preventing unauthorized reading or modification by other users.

Commands Summary

Command	Purpose
sudo adduser user1	Create first user
sudo adduser user2	Create second user
echo "text" > file as user1	Create file owned by user1
chmod 600 file	Set owner read-write, no access others
ls -l file	Check permissions and ownership
cat file as user2	Confirm user2 access denial

Assaignment 2:

Assignment: Disable SSH Root Login on Linux and Verify

Objective

The objective of this assignment is to disable remote SSH login for the root user on a Linux system, restart the SSH service to apply changes, and verify that root login is effectively disabled by testing remote SSH access.

Environment

- Linux distribution (Ubuntu, Debian, CentOS, etc.) with sudo/root access
- SSH server installed and running

<u>Procedure</u>

Step 1: Edit SSH Configuration File

- Open the SSH daemon configuration file with a text editor:

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

- Locate the line containing *PermitRootLogin.*
- Change its value to no to disable root login remotely:

text

```
PermitRootLogin no
```

- Save and exit the editor.

Step 2: Restart SSH Service

- Apply changes by restarting the SSH service:

bash

```
sudo systemctl restart sshd
```

(On some distributions, use sudo service ssh restart or sudo service sshd restart.)

Step 3: Verify the Configuration

- *Try to login remotely as root user:*

bash

```
ssh root@<server-ip-address>
```

- The connection should be refused or denied due to disabled root login.

Results

- Root login over SSH was successfully disabled.
- Remote root login attempts were denied, enhancing system security by preventing direct root access.

Conclusion

Disabling SSH root login is a critical security measure that minimizes risk exposure by forcing administrators to log in as a non-root user and escalate privileges only when necessary. This approach improves accountability and reduces the attack surface.

Commands Summary

Command	**Purpose**
sudo nano /etc/ssh/sshd_config	Edit SSH daemon configuration
PermitRootLogin no	Disable remote root login
sudo systemctl restart sshd	Restart SSH service to apply changes

This report provides a complete overview of the process to disable SSH root login, restart the service, and verify the security configuration through testing.