Basic Linux Commands

ls (list)	Use: Shows files and folders in the current directory. ls ls folder ls -a (list all the hidden file and directories)
cd (change directory)	Use: Move from one folder to another cd folder cd .. cd/folder1/folder2
cat (concatenate)	Use: View content of a file cat file.txt cat folder1/folder2/file.txt
pwd (print working directory)	Use: Shows current location/path pwd
find	Use: Search for files and directories in the filesystem find <path> <option> <value> find -name file.txt find folder1 -name *.txt find folder1 -type d (find directory in folder1) find folder1 -size +10M (find files more than 10 MB) find folder1 -size -10M (find files less than 10 MB)
wp	Use: Counts words inside file wc -l access.log (-l used for counting line inside a file)
grep	Use: Search specific text/word inside files grep "word" filename grep hello file.txt grep "81.143.211.90" access.log (it finds entries the IP visited) grep -R "PRETTY_NAME" /directory/

man (manual)	Use: It shows the official help/documentation of any Linux command inside the terminal. man ls man nmap

Linux Operators

& : Runs a command in the background so the terminal stays usable.

Example:
python script.py &

&& : Run multiple commands (if first succeeds)

Example:
mkdir test && cd test

> : Sends output of a command to a file and overwrites existing content.

Example:
echo hello > file.txt

>> : Adds output to end of a file without deleting old content.

Example:
echo hello >> file.txt

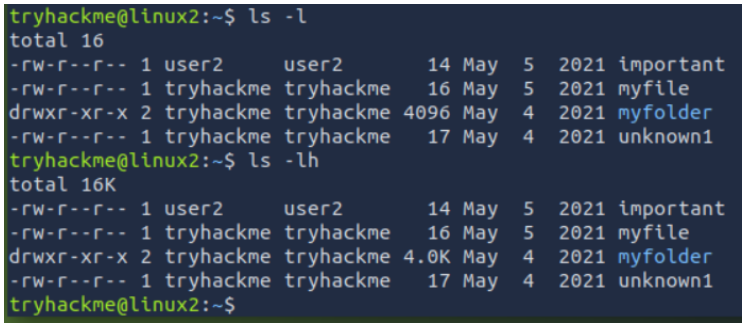
SSH

SSH (Secure Shell) is a protocol used to securely connect to another computer/server over a network. It allows you to control a remote system through the terminal — safely and encrypted.

If a server is in another country, you don't need to go there physically.
You can connect using SSH and control it from your laptop.

Example:
ssh user@192.168.1.10
ssh -p 2222 user@ip (connecting using port)

Commands for filesystem

touch	Use: Create a new empty file touch file.txt
ls -l (size in bytes) ls -lh (size in MB/KB/GB) (human readable)	Use: Shows detailed file info including permissions, owner, group ls -l ls -lh  -rw-r--r– → File permissions 1 → Number of links user → Owner user → Group 4096 → File size in bytes May 5 → Date important → File name
su user	Use: switch user in Linux su user su -l user (after using -l, our new session has dropped us into the home directory of "user" automatically.)

Permission format: rwxrwxrwx

This format is split into three groups:

Section	Applies To	Example
First 3	Owner	rwx
Next 3	Group	rwx
Last 3	Others	rwx

r → read
w → write
x → execute

Each permission has a numeric value:

Permission	Value
r (read)	4
w (write)	2
x (execute)	1

Common examples:

Symbolic	Numeric	Meaning
rwxr-xr-x	755	Owner can do everything, others can read and execute
rw-r--r--	644	Owner can read/write, others can only read
rwx------	700	Only the owner has access

Understanding numeric permissions is important because:

- Many Linux commands use numeric values (e.g. chmod 755 file)
- You can quickly identify security risks
- You can control who can access sensitive files

For example:

```
chmod 750 system_overview.txt
```

1. Owner: full access
2. Group: read + execute
3. Others: no access

Common linux directories

/etc (Editable Text Configuration)	Use: Stores system configuration files Contains: ● user accounts ● passwords (hashed) ● service configs ● network configs Example: ● /etc/passwd ● /etc/ssh/sshd_config
/var (Variable data)	Use: Files that change frequently Contains: ● logs ● cache ● mail ● web server data Examples: ● /var/log → system logs ● /var/www → website files ● /var/mail → user mails

/root	Use: Home directory of root user (admin) Only root users can access it by default. Example: /root This is like: /home/saif but for admin.
/tmp (Temporary files)	Use: Stores temporary data created by programs ● Anyone can access ● Auto-deleted after reboot (usually) Example: /tmp Used for: ● temporary downloads ● script output ● testing
/home	Use: Home folders of normal users Example: ● /home/saif ● /home/user2 Each user stores personal files here.
/bin	Use: Essential system commands (binary files) Examples: ● /bin/ls ● /bin/cp ● /bin/mv ● /bin/bash

/usr	Use: Installed software & user programs Examples: ● /usr/bin ● /usr/share ● /usr/lib
/dev	Use: Device files Represents hardware as files Examples: ● /dev/sda → hard disk ● /dev/null → black hole file ● /dev/tty → terminal
/opt	Use: Optional or third-party software Example: ● /opt/google ● /opt/lampp
/boot	Use: Bootloader & kernel files Needed to start Linux Example: ● /boot/vmlinuz

Terminal text editors

1. Nano: nano is a simple terminal text editor in Linux.

Example: nano notes.txt

If file exists → opens it
If not → creates new file

2. VIM: advanced terminal text editor.

Used for:

- Editing configs
- Writing scripts
- Exploit development
- Server work
- CTFs & pentesting

Almost every Linux server has vim installed.

Downloading files

wget = web get
Used to download files from the internet using a terminal.

Example:
wget URL
wget https://example.com/file.txt

wget -P /tmp -O saif.txt https://example.com/test.txt

wget reads options in order:

- -P /tmp → where to save
- -O saif.txt → rename file

So it will:

- download from URL
- save inside /tmp
- rename to saif.txt

SCP (Secure Copy Protocol)

SCP is used to securely transfer files between computers over SSH.

Basic Syntax
scp [file] user@ip:/destination

Send file (your PC → remote machine)
scp file.txt user@192.168.1.10:/home/user/

Download file (remote → your PC)
scp user@ip:/path/file.txt .

. means current folder.

Copy entire folder
scp -r folder user@ip:/home/user/

-r = recursive (for folders)

Rule (VERY IMPORTANT)

SCP destination path must exist on the machine where the command is running.

If command runs on THM → path must be THM path
If command runs on Kali → path must be Kali path

Serving Files From Your Host (WEB)

Hosting files from your own machine. So another machine (victim/THM/target) can download them via browser or wget.

Fastest method (Python web server
Inside folder where file exists: python3 -m http.server 8000

How target downloads file
From target machine: wget http://YOUR-IP:8000/file.txt

Processes (Basic idea)

A process is any running program on your system.
Each process has a unique PID (Process ID) assigned by Linux.

ps command (process Status)
Basic: ps
Shows processes of the current terminal session only.

All processes: ps aux

top command
Use: top
Purpose: Real-time process monitor.

- Shows live:
- CPU usage
- RAM usage
- Running processes
- System load
- Updates continuously every few seconds

Press q or ^c to exit.

kill command
Use: kill PID
Purpose: Stop/terminate a process using PID.

- Safely stops process
- Allows cleanup
- Recommended

Important kill signals
1. SIGTERM (default): kill PID

- Safely stops process
- Allows cleanup
- Recommended

2. SIGKILL (force kill): kill -9 PID

- Force stop immediately
- No cleanup
- Use when process stuck

3. SIGSTOP (pause): kill -STOP PID

- Pauses process
- Can resume later

Services & Boot Processes
Some programs run automatically when system starts:

- Web servers
- Databases
- SSH

Linux manages them using systemctl

systemctl Command
Used to control services.

Start service: sudo systemctl start apache2

Stop service: sudo systemctl stop apache2

Check status: sudo systemctl status apache2

Enable service at boot: sudo systemctl enable apache2

Disable service at boot: sudo systemctl disable apache2

Background & Foreground Processes

Processes run in two modes:

1. Foreground
2. Background

Foreground Process
A foreground process runs directly in the terminal and blocks the terminal until it finishes.

Example: `echo "Hello Saif"`

Output shows immediately and the terminal waits until command finishes.

Background Process
A background process runs without blocking the terminal.
You can continue using the terminal while it runs.

Run command in background:
Add & at end: `echo "Hi THM" &`

Output:

- Terminal returns process ID (PID)
- Command runs in background

Useful when:

- Copying large files
- Running scripts
- Scanning tools (nmap, etc)

Why Background Processes Are Useful

Very important in hacking/linux:

- Run scans while doing other work
- Run servers (python http server)
- Reverse shells
- Long brute-force tools
- Automation scripts

Example: `python3 -m http.server 8000 &`

The server runs in the background while you use the terminal.

&	→	run in background
Ctrl+Z	→	pause/send background
bg	→	resume in background
fg	→	bring to foreground
ps aux	→	view processes

Cron & Crontab

cron is a Linux background service (daemon) that runs scheduled tasks automatically.

Used for:

- Backups
- Running scripts
- Updates
- Automation
- Starting programs at specific time

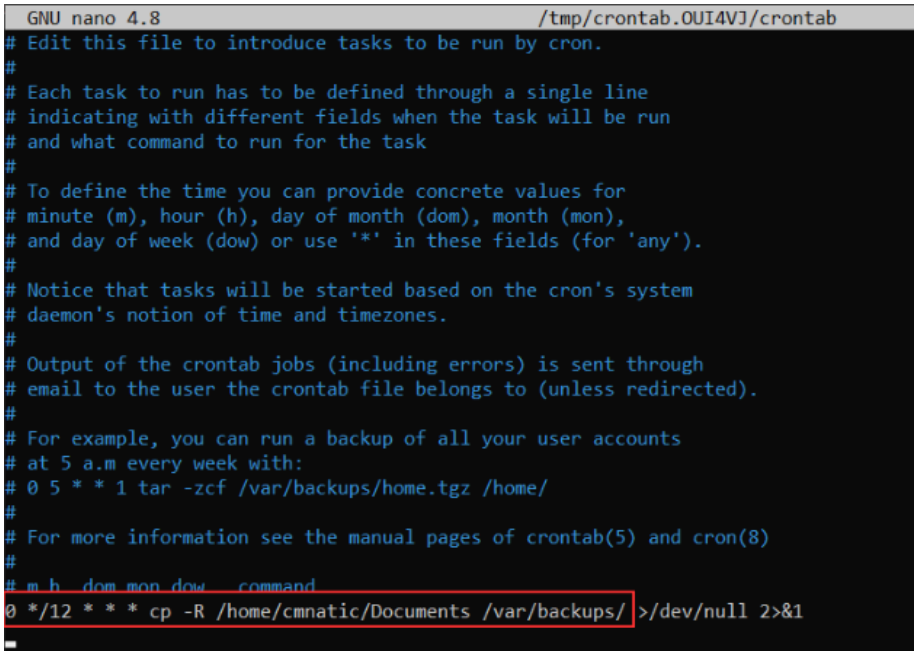
Cron runs in the background after system boot.

crontab (cron table)
A special file that stores scheduled tasks (cron jobs).

- Each line = one scheduled task.
- The system reads this file and executes commands automatically at specified time.

Open crontab editor: `crontab -e`

```
* * * * * command
| | | | |
| | | | └── Day of week (0-7) (Sun=0 or 7)
| | | └──── Month (1-12)
| | └────── Day of month (1-31)
| └──────── Hour (0-23)
└────────── Minute (0-59)
```



PACKAGES & SOFTWARE REPOSITORIES (APT)

Package

A package is a software/program in Linux

Example:

- firefox
- nmap
- nano
- apache

Linux installs software from repositories (repos).

Repository

Repository is an online storage where Linux downloads software.

Its like App Store / Play Store for Linux
Ubuntu uses: apt (package manager)

Where repos are stored?

Main config location: /etc/apt/

Important files:

- /etc/apt/sources.list
- /etc/apt/sources.list.d/

These contain download sources.

Basic APT commands (VERY IMPORTANT)

Update package list

- sudo apt update

Upgrade installed packages

- sudo apt upgrade

Install software

- sudo apt install nmap

Remove software

- sudo apt remove nmap

Search package

- apt search nmap

How to add a repository

EXAMPLE 1 — Install VLC (simple repo method)

Step 1: Add VLC repository
```
sudo add-apt-repository ppa:videolan/master-daily
```

Step 2: Update
```
sudo apt update
```

Step 3: Install VLC
```
sudo apt install vlc
```

Now VLC will install from the new repo.

EXAMPLE 2 — Manual repository method (IMPORTANT)

Now we do the manual way (important for cybersecurity & exam).
We will add Brave browser repo.

Step 1: Go to repo folder
```
cd /etc/apt/sources.list.d/
```

Step 2: Create repo file
```
sudo nano brave.list
```

Paste inside:
deb [arch=amd64] https://brave-browser-apt-release.s3.brave.com/ stable main

Step 3: Add GPG security key
```
sudo curl -fsSLo /usr/share/keyrings/brave-browser-archive-keyring.gpg
https://brave-browser-apt-release.s3.brave.com/brave-browser-archive-keyring.gpg
```

Step 4: Update system
```
sudo apt update
```

Step 5: Install
```
sudo apt install brave-browser
```

Linux Logs

Logs = everything happening in the system.

Stored in:
/var/log

1. Authentication logs (LOGIN HISTORY)
/var/log/auth.log

Shows:
- SSH login
- sudo usage
- hacking attempts

Check:
cat /var/log/auth.log

2. System log
/var/log/syslog

Shows:
- System events & errors.

3. Apache logs (Web server)
/var/log/apache2/access.log
/var/log/apache2/error.log

Access log → who visited
Error log → errors

4. Fail2ban logs (attack blocking)
/var/log/fail2ban.log

5. Firewall logs
/var/log/ufw.log