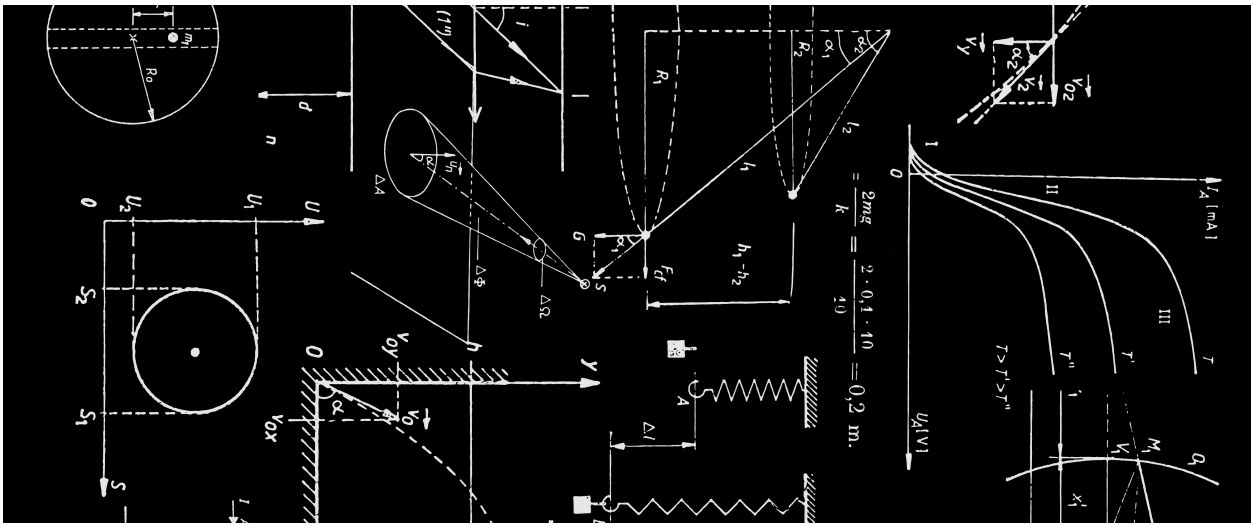

Linux Basics for Hackers [Notes]

▼ CHAPTER 1 - BASICS

▼ [1.1] Terminologies and Definitions

1. Binaries - refers to a file that can be executed. Generally stored (reside in) ***'/usr/bin'*** or ***'/usr/sbin'*** **.** It stores command utilities such as *ps*, *cat*, *cd* etc.
2. Case Sensitive - *Unlike Windows*, Linux is Case Sensitive. for eg. Front, front and fronT are all different from each other.
3. Directory - same as a *folder* in Windows.
4. Home - Each user has their own ***/home*** directory, where files created by user are saved by default.
5. Kali - A Linux distro, specifically designed for ***penetration testing***.
6. Root - Administrator / Superuser account, includes permissions such as reconfiguring the system, adding users/groups, changing passwords etc.
7. Script - series of commands that convert each line to source code in an ***interpretive environment***.

8. Shell - environment and interpreter for running commands in Linux. Most widely used shell is BASH ***[Bourne-again Shell]***, others include C Shell, Z Shell etc.
9. Terminal - The command line interface [CLI] - shortcut : CTRL + ALT + T.

▼ *REAL WORLD* TIPS

Linux vs. Windows – Know the Landscape

Why it matters: Thinking like an attacker or investigator requires you to understand both environments.

Windows hides many things by default. Linux exposes more control to the user.

Pro Tip: Practice privilege escalation in both — what works on Windows won't necessarily work on Linux, and vice versa.

▼ [1.2] Linux File System

1. [/] root - Superuser/root user's home directory (root is diff. from root user).
2. /etc - contains Linux configuration files- control when and how programs start.
3. /home - user's home directory.
4. /mnt - here other filesystems are attached or mounted to the filesystem.
5. /media - CDs and USB devices are usually attached or mounted to the filesystem.
6. /bin - application binaries [equivalent of executable files].
7. /lib - libraries [shared programs, similar to windows DLLs].

> 💡 It's ***important to note*** the one should not log in as root user when performing routine tasks, because anyone who hacks the system (yes, hackers sometimes get hacked) when you or me are logged in as root, the hacker would immediately gain root privileges and thus "own" our system. Log in as a regular user when starting regular applications, browsing the web, running tools like Wireshark, and so on.

▼ *REAL WORLD* TIPS

Linux File System Hierarchy

Why it matters: Many CTF flags, malware, or forensic evidence hide in `/tmp`, `/var/tmp`, `/dev/shm`.

Pro Tip: Always check these writable temp folders when analyzing a compromised system.

Insight: `/bin` is where your basic system binaries live — if this gets wiped or hijacked, the system can become unusable.

▼ [1.3] Basic Commands in Linux

1. pwd - ***present working directory***

   → returns your location within the directory structure.

2. whoami - ***returns logged in 'username'***

3. cd - ***changing directories***

   → [.] - current directory

   → [..] - parent directory

   → [.. ..] - moves up 2 levels in file system hierarchy

   → [.. .. ..] - moves up 3 levels in file system hierarchy and so on

4. ls - list command

   → lists the contents of the directory [files and directories within current directory]

→ 'ls -l' - long list format of files and directories

→ 'ls -a' - reveals hidden files in file system

5. ***[command/application/utility]*** + "- -help" - returns a manual of sorts with description, guidance and options on how to use the command

   → although many applications support all the three options i.e. - - help, -h and -?, there's no guarantee the application you're using will. So if one doesn't work try another. ***manual*** [man] page has more information and synopsis of the command or application.

   → eg. *nmap -h* or *grep - -help* or *man aircrack -ng.*

6. locate - easiest command to use, followed by a ***keyword*** denoting what is to be found, locates every occurrence of the ***keyword***.

   → The locate command is not perfect, however. Sometimes the results of locate can be overwhelming, giving you too much information. Also, locate uses a database that is usually only updated once a day, so if you just created a file a few minutes or a few hours ago, it might not appear in this list until the next day.

7. whereis - used to ***locate*** a binary file. It returns not only the location of the binary file but also its source and man page is available.

8. which - returns only the location of binaries in the ***PATH*** variable.

9. find - most powerful and flexible of the searching utilities. It is capable of beginning your search in any designated directory and looking for, filename, date of creation or modification, owner, group, permissions, and size.

   → Syntax: find *directory* ***options expression.***

10. grep - a filter to search for keywords. Often used when output is piped from another command. Referred to as *'piping'* .

11. ps [***processes***] - used to display information about processes running on the machine.

▼ *REAL WORLD* TIPS

Root vs. Normal Users

Why it matters: Attackers aim to become `root`. Every privilege escalation bug is built around this.

Pro Tip: Never run your browser or `wireshark` as root. Use:

```
sudo adduser charlie wireshark
```

Insight: Running `chmod 777` as root in `/etc` or `/usr` is like taping a grenade to your face.

▼ [1.4] Wildcards in Linux

? - represents a single character

[ ] - used to match characters that appear in the square brackets

* - matches any character(s) of any length, from none to an unlimited number of characters.

▼ *REAL WORLD* TIPS

1.4 Switching Users

Why it matters: During incident response, you'll often find attacker-created users in `/etc/passwd`.

Pro Tip: Use `grep -vE 'root|daemon|sys' /etc/passwd` to list only human users — great for forensics.

▼ [1.5] Modifying Files and Directories

▼ [1.5.1] Creating File(s)

1. using **cat** [concatenate] command

→ eg. ***cat filename***

> 💡 when you press ENTER, Linux will go into interactive mode and wait for content of the file to be entered. Press ***CTRL + D*** to quite the prompt.

2. using touch command

→ originally ***touch*** command was developed to change details, such as creation or modification date of a file. However, if the file is missing or doesn't exist, it automatically creates the file.

eg. ***touch newfile***

▼ *REAL WORLD* TIPS

`>` VS. `>>`

Why it matters: One overwrites, the other appends. Many admins accidentally destroy configs with `>`.

Pro Tip: When editing logs or appending payloads, always double-check with `ls -lh file` after your operation.

▼ [1.5.2]Creating Directory

mkdir [make directory] command is used to create a directory

→ eg. mkdir newdirectory

▼ [1.5.3]Copying a File

Use cp [copy] command to create a duplicate file in new location or in the same directory.

→ eg. cp *oldfile* **/root/newdirectory/newfile**

> 💡 Renaming file is optional and often done to create a distinction if not mentioned the original name is retained as default name for new duplicate file.

▼ *REAL WORLD* TIPS

Copying and Moving Files

Why it matters: You'll use `cp` and `mv` during malware containment, log collection, or forensics image prep.

```
cp -r /var/log ~/Evidence/log_backup/
mv suspicious.sh /tmp/hold/
```

Never move original evidence. Always **copy first, hash later**.

▼ [1.5.4]Renaming a File

No command intended solely for renaming a file in Linux, ***mv*** [move command can be used to achieve the same, **mv** command is multipurpose:

1. Move a file or directory to new location.

→ eg. mv file1 ~/Desktop/testdir/txtFiles

2. Or, rename an existing file.

→ eg. mv file1 newfile1.doc

▼ *REAL WORLD* TIPS

Deleting Files

Why it matters: Many attackers use `shred`, `rm -rf`, or overwrite logs.

Pro Tip: Use `alias rm='rm -i'` in your `.bashrc` to always prompt before deletion.

Incident Response Tip: Deleted files can still be found in memory or unallocated disk — tools like Autopsy and `foremost` help here.

▼ [1.5.5]Removing a File

rm [remove] command is used to delete (remove) a file.

→ eg. rm file1.doc

▼ [1.5.6]Removing a Director

rmdir [remove directory] command is used to delete (remove) a directory.

> 💡 rmdir will ***not*** remove a directory this is ***not empty***, to remove a directory with all the sub-contents of it including sub-directories and files, use;
>
> rmdir -r [-r option: recursively] - use thoughtfully can be dangerous.

→ eg. rmdir newdir

▼ BONUS TIP

Always use full paths in scripts and cron jobs.

It avoids PATH hijack attacks where `ls` or `cat` might be malicious.

Don't just trust GUI file managers. Use `file`, `md5sum`, and `strings` to verify content — attackers rename extensions all the time