**ls -> list down all the files in your directory**

**---------------------------------------------------------------------------------------------**

**nano <file name> -> open a particular file.**

**vim <file name> -> open a particular file.**

**gedit <file name> -> open a particular file in graphical user interface.**

**---------------------------------------------------------------------------------------------**

**Ctrl + O(letter) -> save the changes we had done to file.**

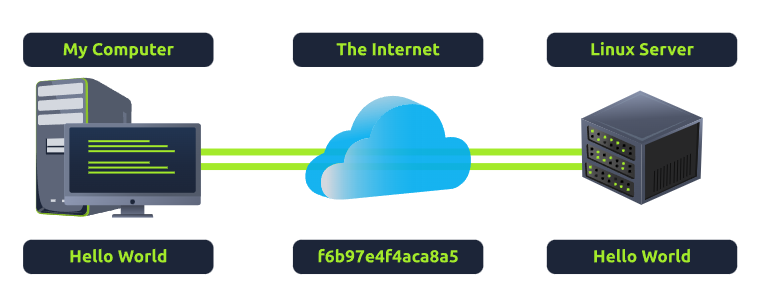
**Ctrl + X(letter) -> exit from a particular file.**

**cat <file name> -> print the content of the file without opening the file.**

**---------------------------------------------------------------------------------------------**

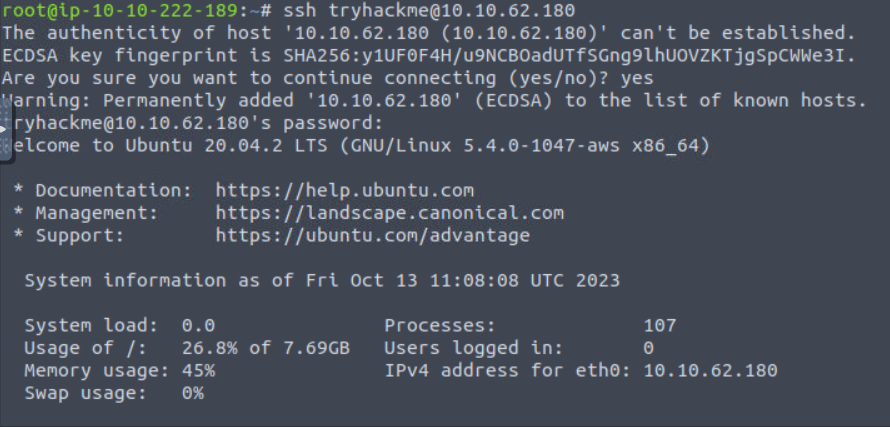
**Secure Shell or SSH**

* the common means of connecting to and interacting with the command line of a remote Linux machine.
* SSH simply is a protocol between devices in an encrypted form.



* SSH allows us to remotely execute commands on another device remotely.

**The way to access another device using SSH**

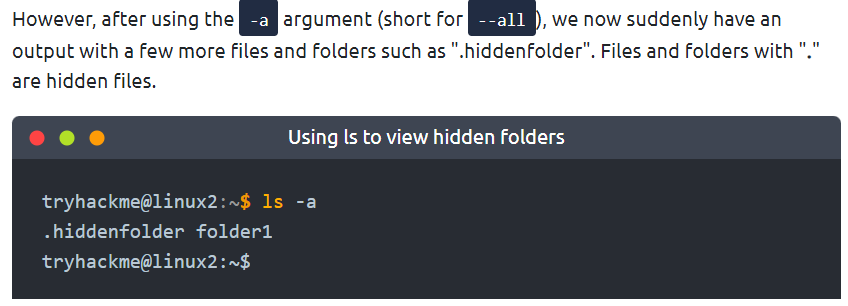


**ssh <username>@<target\_IP\_address>**

**Importance of flags and switches,**

 ls lists the contents of the working directory. However, hidden files are not shown. We can use flags and switches to extend the behavior of commands.

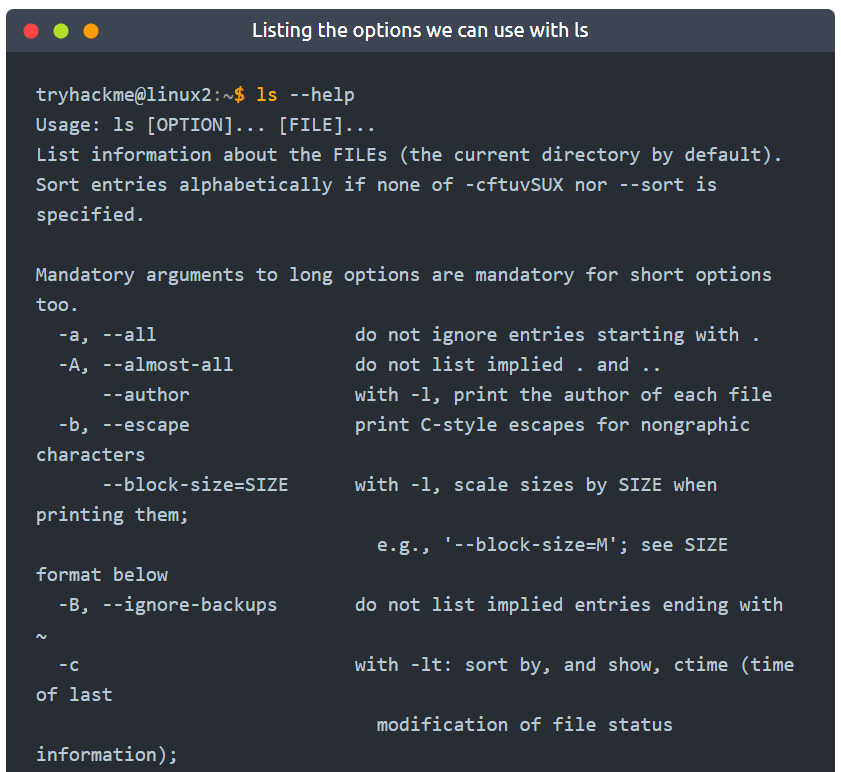
Example of **-a**,



**To see all the flags and switches behavior of particular command**

**<Command> --help**

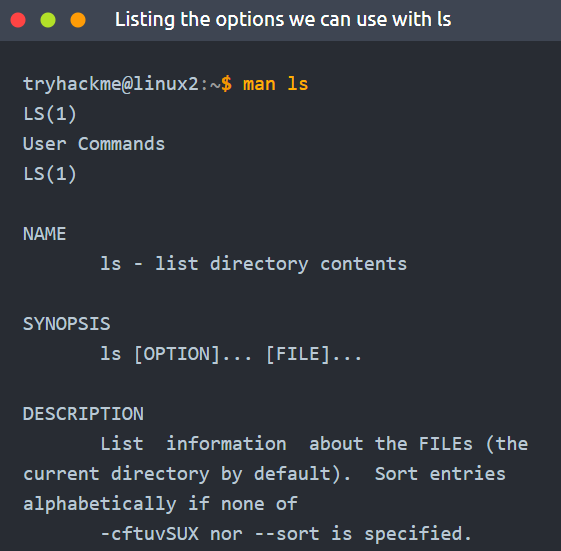
If you want to know what the capabilities have particular command, we can simply type “--help” after the command.



**Way to access manual page.**

Manual page contains about flags and describe the functionality of each of the flags.

Type “man ls” command



Ex:

* What flag use to display the output in a “human-readable” way?

-h

**File system related commands**

**A white grid with black text

Description automatically generated**

* create files and folders

touch command

touch <file name>

this creates blank file according to the name of the file.

mkdir command

mkdir <directory name>

this creates blank directory.

* Remove files and folders

Remove a file

rm <file name>

remove a directory

rm -R <directory name>

* cp command

this takes two arguments.

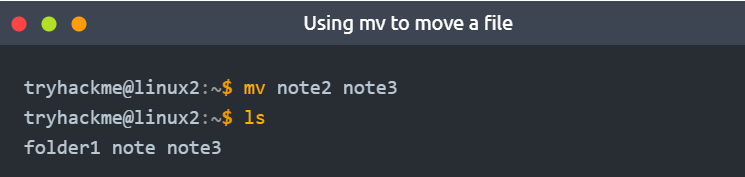
1. The name of the existing file.
2. The name we wish to assign.

A screenshot of a computer

Description automatically generated

“note” is already existing file. But “note2” is newly created file with this command.

* mv command



you use mv to move a file to a new folder. But that folder must already create before run this command.

you can also use mv to rename a file or folder.

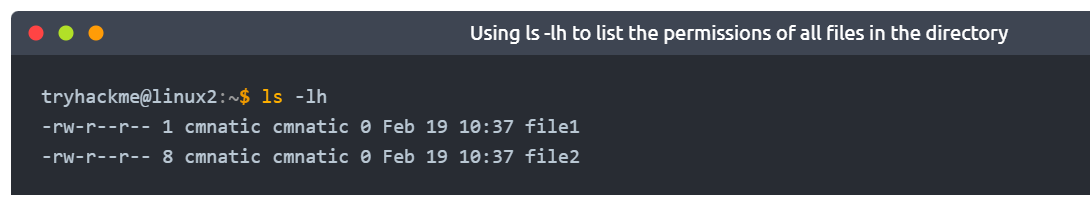
* To view the Data format of a file.

file [file name]

A screenshot of a computer code

Description automatically generated

Permissions



"-" indicator highlighting that it is a file

then "rw", This means that only the owner of the file can read and write to this"cmnatic.pem" file but cannot execute it.

If we can see “r”,” w” and “x” then all the privileges are granted. Read, write and execute.

One awesome feature of Linux is that you can control who can do what with a file very precisely. Even if one person owns a file, you can let a group of people have their own specific permissions for that file, without changing the file's owner.

**Switching Between Users**

On a Linux system, you can switch between users using the **su** command.

To switch users:

* The username we wish to switch to and
* The user's password

Ex:

currently logged in as the user "user1," and you want to switch to "user2." Type,

su - user2

The difference between users & groups

In Linux, users and groups are different. A regular user and a system user have differences, but Linux allows for precise permission control. Even if a file is owned by a user, it can be shared with a group of users, each having different permissions. For example, a web server system user needs to read and write files for a web app, but web hosting companies want customers to upload files without compromising security.

## Common directories

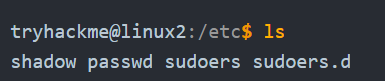
**/etc**

The root directory is super important, and the "etc" folder is where your system keeps important files. which is like a storage place for special files that your computer needs to work.

For example, there's a file in there called "sudoers": which decides who can do special things on the computer.

There are also files called "passwd" and "shadow": that hold your passwords in a secret way called sha512, which is unique to Linux.

Go to inside of the “etc” folder and type “ls”,

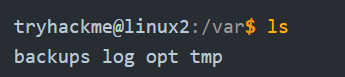


**/var**

"/var" directory, with "var" being short for “variable data”.

 This folder stores data that is frequently accessed or written by services or applications running on the system.

Ex: log files from running services and applications are written here (/var/log)

****

/**root**

"root" user, whose home directory is "/root" and not "/home/root."

The /root folder is like the home for the "root" system user. It's just where the "root" user's stuff is.

**A black background with white and orange letters

Description automatically generated**

**/tmp**

The "/tmp" directory in Linux is like a temporary storage space. It's used for data that you only need briefly, and it gets cleared when you restart your computer.

For penetration testing, it's handy because anyone can put files in there. So, when we're testing a system, we can store our tools and scripts there.

