LINUX USERS AND PERMISSIONS

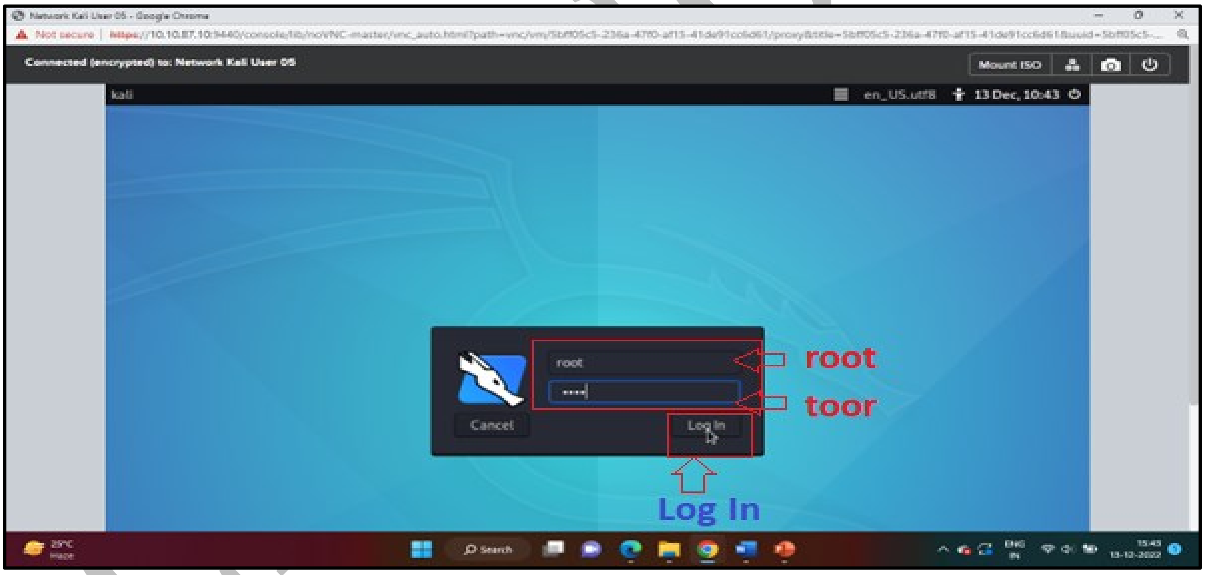
In Linux OS, A user is an entity, that can manipulate files and perform several other operations. Each user is assigned an ID that is unique for each user in the operating system.

After installation of the operating system, the ID 0 is assigned to the root user while the IDs 1 to 999 (both inclusive) are assigned to other system users and hence the id for the local user begins from 1000 onwards. Users are assigned permission to access, alter or execute the files.

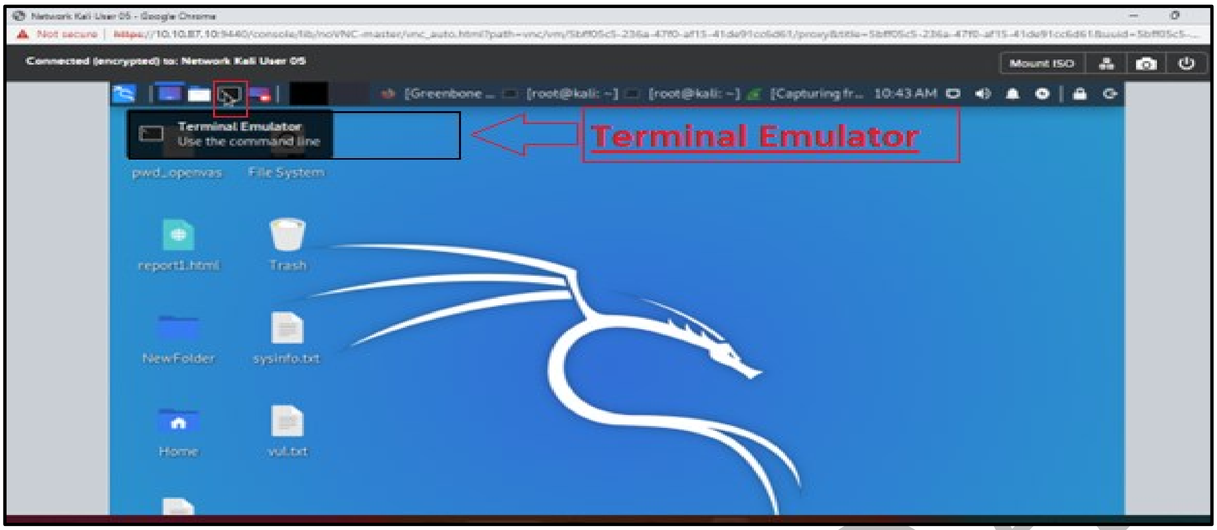
In this lab manual, you will learn about the Linux command used to manage users and permissions in a Linux operating system.

For executing the commands follow the below-given statements.

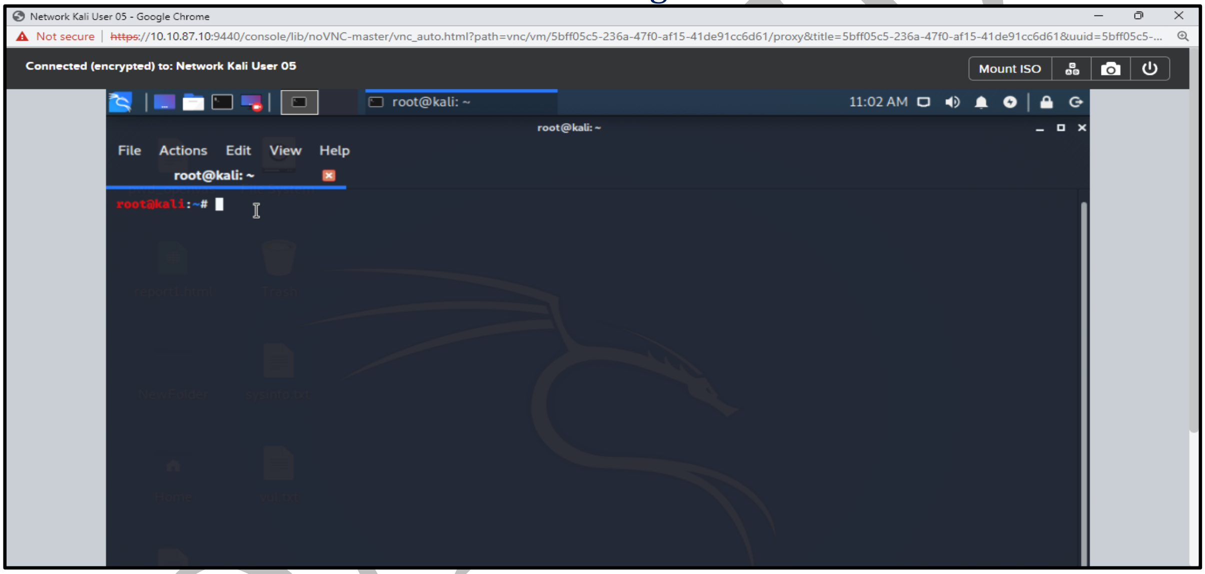
1. Connect to the kali Linux machine, created by you, using the RDP protocol.
2. When prompted for the username and password, enter root as username and toor as password. The root is the administrator user of the machine.


Network Kali User 05 - Google Chrome
Not secure
Connected (encrypted) to: Network Kali User 05
Mount ISO
kali
en_US.utf8
13 Dec, 10:43
root
Cancel
Log In
root
toor
Log In

3. click on the black box icon (Terminal Emulator) in the top left corner of the Kali Linux Desktop.



Running the terminal while using the root account, allows you to run various commands with administrator rights.


Network Kali User 05 - Google Chrome
Not secure
Connected (encrypted) to: Network Kali User 05
Mount ISO
root@kali: ~
11:02 AM
File
Actions
Edit
View
Help

```
root@kali:~#
```

Let's Understand user and permission commands one by one.

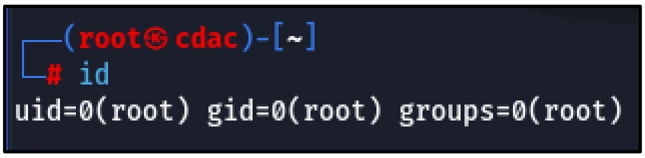
1. ID

Syntax : $id

In Linux Shell, id command is used to identify the current shell user.

In a root shell, on executing id, it gives output of the current user's uid, gid, group

Uid 0 , indicated for root user.

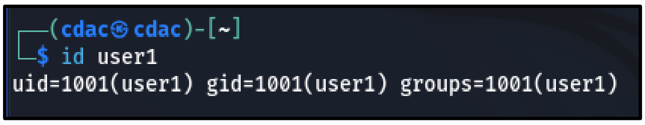


```
┌──(root㉿cdac)-[~]
└─# id
uid=0(root) gid=0(root) groups=0(root)
```

In a Normal User shell, to identify uid of a particular user.

execute with the username. It gives an output displaying the current user's uid, gid, and group.

Here, uid 1001 is for user1.



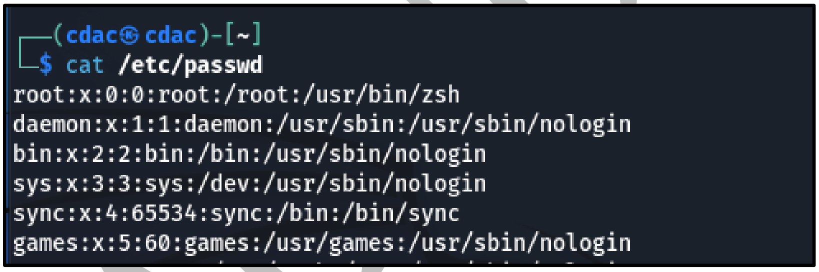
```
┌──(cdac㉿cdac)-[~]
└─$ id user1
uid=1001(user1) gid=1001(user1) groups=1001(user1)
```

the IDs 1 to 999 (both inclusive) are assigned to the system users; hence, the ids for local users begin from 1000 onwards. In a single directory, we can create up to 60,000 users.

2. Accessing a User Configuration file

In Linux, configurations (like shell used for a particular user) for all users are stored in /etc/passwd file

Syntax: $cat /etc/passwd



```
┌──(cdac㉿cdac)-[~]
└─$ cat /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
```

To access the user configuration file, execute "cat /etc/passwd". Here Output format is as follows

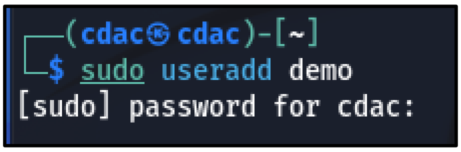
username: x : userid : user-group-id : : /home/username : /bin/bash

3. Add a User

useradd is a command in Linux that is used to add user accounts to the system.

USERADD

Syntax: $sudo useradd demo

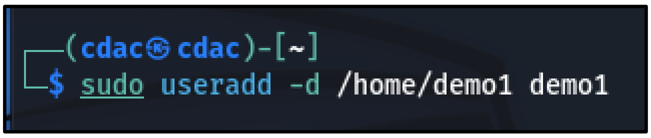


```
┌──(cdac㉿cdac)-[~]
└─$ sudo useradd demo
[sudo] password for cdac:
```

4. To give the home directory path for user demo1 added

You can use the useradd command to create a user with a home directory path

Syntax: $sudo useradd -d /home/demo1 demo1

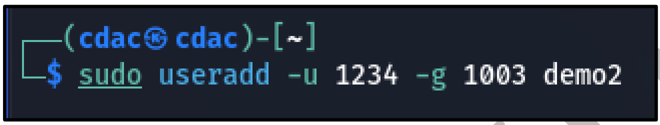


```
┌──(cdac㉿cdac)-[~]
└─$ sudo useradd -d /home/demo1 demo1
```

5. To create a user with specific user id and group id :

To create a user with specific user id and group id use useradd command with the syntax

Syntax: $sudo useradd –g 1234 –g 1003 demo2

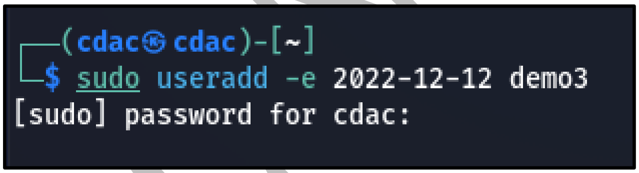


```
┌──(cdac㉿cdac)-[~]
└─$ sudo useradd -u 1234 -g 1003 demo2
```

6. To create a user with expiry date :

To create a user with expiry date user useradd command with date and username

Syntax: $sudo useradd –e 2022-12-12 demo3



```
┌──(cdac㉿cdac)-[~]
└─$ sudo useradd -e 2022-12-12 demo3
[sudo] password for cdac:
```

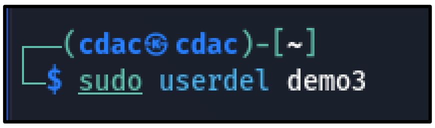
Delete a User

Users can be deleted in Linux using the userdel command. This command basically modifies the system account files, deleting all the entries which refer to the user being deleted. It is a low-level utility for removing users. To delete a user normally, use the userdel command with the name of the user to be deleted.

Note: Deleting Users created in Add Users Section.

7. To Delete User:

Syntax: $sudo userdel demo1

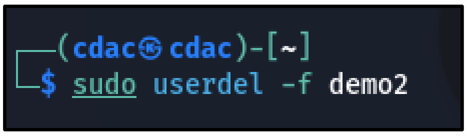


```
┌──(cdac㉿cdac)-[~]
└─$ sudo userdel demo3
```

8. Delete user forcibly

If you are not able to delete the user for any reason, use the –f option for forceful deletion.

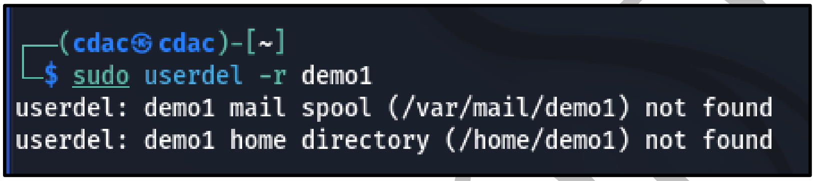
Syntax : $sudo userdel -f demo2



```
┌──(cdac㉿cdac)-[~]
└─$ sudo userdel -f demo2
```

Syntax: $sudo userdel -r demo1

Whenever we are deleting a user using this option then the files in the user's home directory will be removed along with the home directory itself and the user's mail spool.

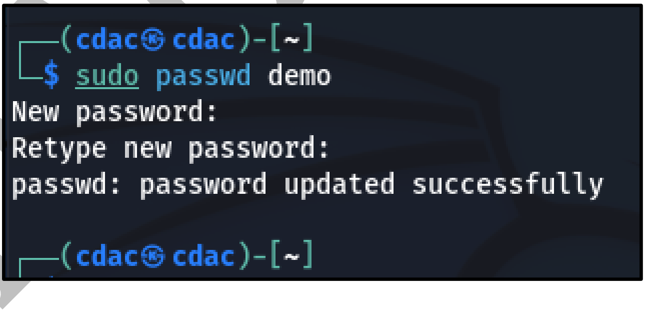


```
┌──(cdac㉿cdac)-[~]
└─$ sudo userdel -r demo1
userdel: demo1 mail spool (/var/mail/demo1) not found
userdel: demo1 home directory (/home/demo1) not found
```

9. To change the Password of a User:

The user Password can be reset using the password command. This command asks the user to enter the new password two times and then the password gets updated to the new password as shown in the command snapshot.

Syntax: $sudo passwd demo



```
┌──(cdac㉿cdac)-[~]
└─$ sudo passwd demo
New password:
Retype new password:
passwd: password updated successfully

┌──(cdac㉿cdac)-[~]
```

10. Managing Groups

Groups in Linux refer to the user groups. As you know uid 0 is for root, uid 1- 999 are for system users and normal users can take uid from 1000 to 60000.

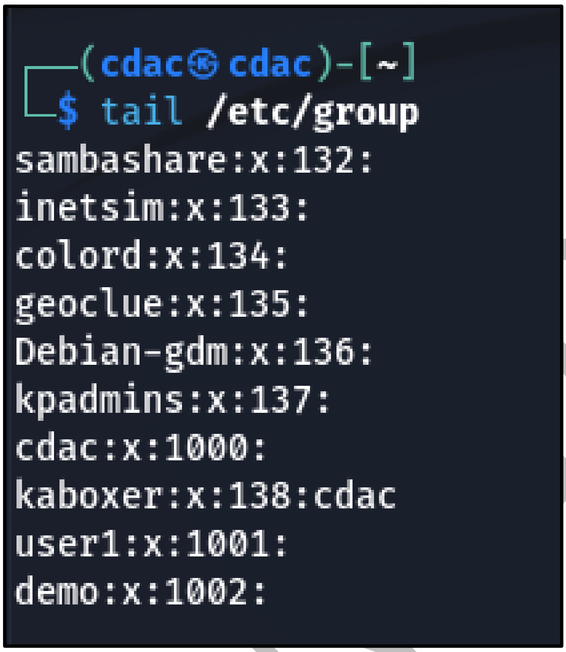
There might be some privileges that some users have, and some don't, and it becomes difficult to manage all the permissions at the individual user level. So, using groups, we can group together a number of users, and set privileges and permissions for the entire group.

/etc/group: It contains the account information of the Group.

/etc/gshadow: It contains the secure group account information.

This is a simple command to display the last few lines of /etc/group file about the account information of the group.

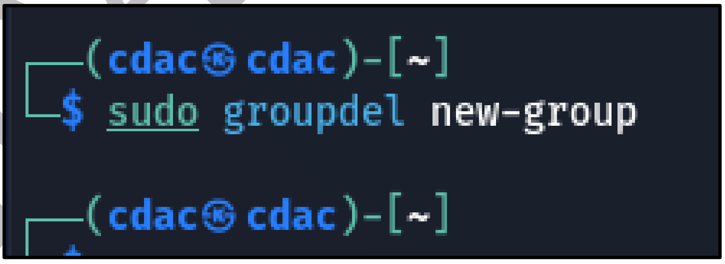
Syntax: $tail /etc/group



```
┌──(cdac㉿cdac)-[~]
└─$ tail /etc/group
sambashare:x:132:
inetsim:x:133:
colord:x:134:
geoclue:x:135:
Debian-gdm:x:136:
kpadmins:x:137:
cdac:x:1000:
kaboxer:x:138:cdac
user1:x:1001:
demo:x:1002:
```

11. Delete user group named as new-group

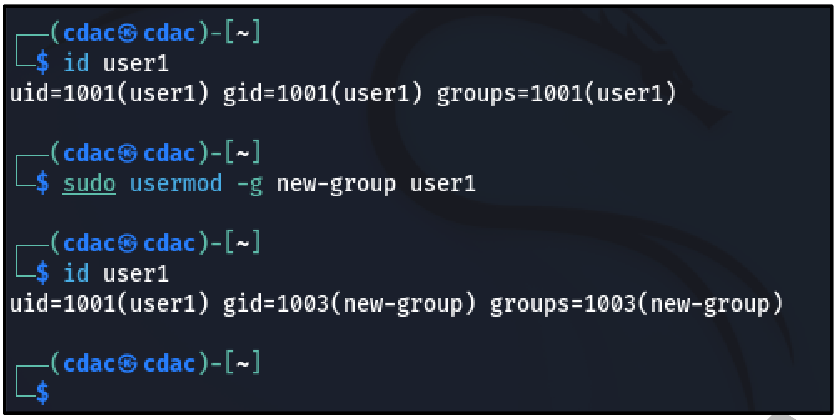
groupdel command is used to delete the user group.
The snapshots show the use of group add and group del commands.
Syntax: $sudo groupdel new-group



```
┌──(cdac㉿cdac)-[~]
└─$ sudo groupdel new-group

┌──(cdac㉿cdac)-[~]
```

12. To add an existing user to a group, use the usermod command:

Syntax: $sudo usermod -g newgroup user1

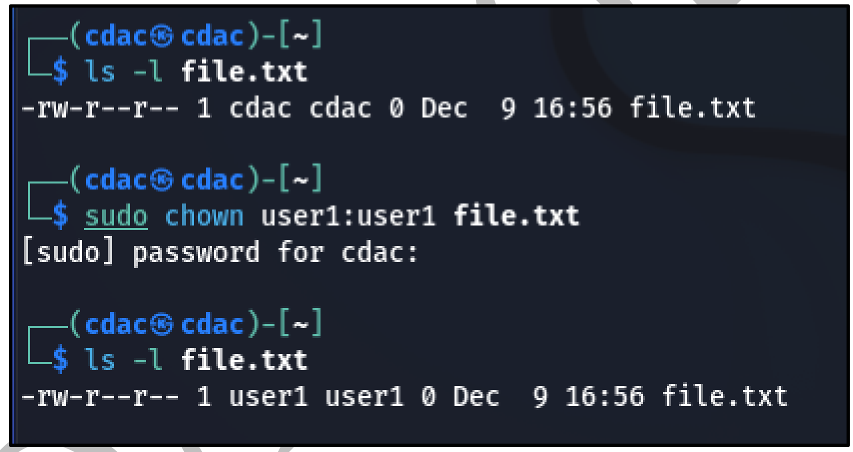


```
┌──(cdac㉿cdac)-[~]
└─$ id user1
uid=1001(user1) gid=1001(user1) groups=1001(user1)

┌──(cdac㉿cdac)-[~]
└─$ sudo usermod -g new-group user1

┌──(cdac㉿cdac)-[~]
└─$ id user1
uid=1001(user1) gid=1003(new-group) groups=1003(new-group)

┌──(cdac㉿cdac)-[~]
└─$
```

13. To change the owner of a file, use the chown command:

Each file in Linux has some permissions associated with reading, writing, and executing for the owners, groups, and other users of the system.

Syntax: $sudo chown user1:user1 file.txt



```
┌──(cdac㉿cdac)-[~]
└─$ ls -l file.txt
-rw-r--r-- 1 cdac cdac 0 Dec  9 16:56 file.txt

┌──(cdac㉿cdac)-[~]
└─$ sudo chown user1:user1 file.txt
[sudo] password for cdac:

┌──(cdac㉿cdac)-[~]
└─$ ls -l file.txt
-rw-r--r-- 1 user1 user1 0 Dec  9 16:56 file.txt
```

First view the Permission of a file.txt: **ls -l file.txt**

To change the owner of the file: chown owner_name filename: **chown user1:user1 file.txt**

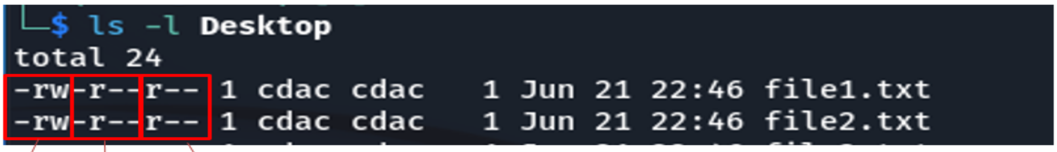
To verify the owner: **ls -l file.txt**

To change the permissions of a file, use the *chmod* command:

Understanding Security Permissions :Octal Notations

Octal	Binary	File Mode
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r—x
6	110	rw-
7	111	rwx

Let's understand the security permissions now. The files can have r, w and x permissions, associated with them where 'r' means "reading" the file's contents.
'w' means "writing" or modifying the file's contents and x means "executing" the file. This execute permission is given only if the file is a program. If any of the "r w x" characters are replaced by a '- ', in the output, it means that permission has been revoked.



```
└─$ ls -l Desktop
total 24
-rw-r--r-- 1 cdac cdac    1 Jun 21 22:46 file1.txt
-rw-r--r-- 1 cdac cdac    1 Jun 21 22:46 file2.txt
```

User	Group	Other
---	---	---
rwx	rwx	rwx

The first set of permissions in output represent the user permissions which are applicable only to the owner of the file or directory, they will not impact the actions of other users.

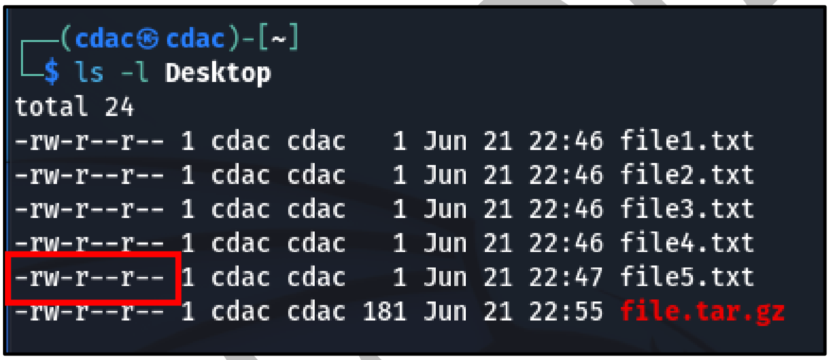
The second set represents the group permissions which apply only to the group that has been assigned to the file or directory, they will not affect the actions of other users.

The third set represents other permissions which apply to all other users on the system, this is the permission group that you want to watch the most as ethical hacker.
Syntax Commands to change file permissions:

Chmod ugo+rwx [file_name]
Chmod 777 [file_name]

14. Changing permission of File file5.txt

Before changing permissions



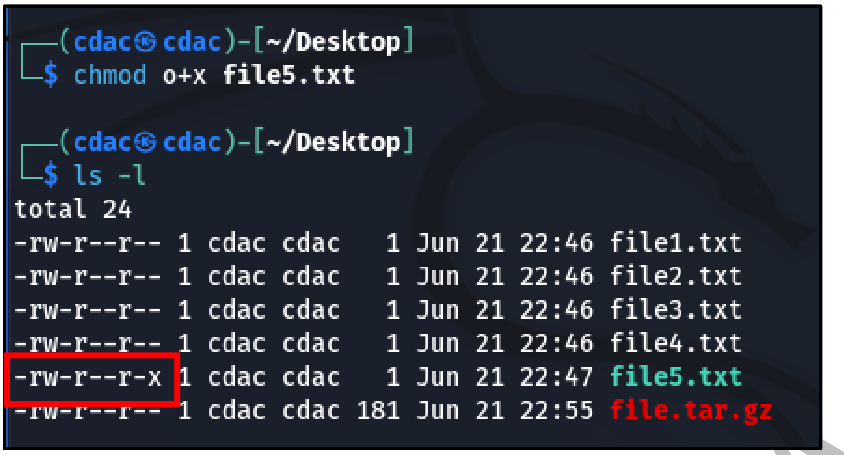
```
┌──(cdac㉿cdac)-[~]
└─$ ls -l Desktop
total 24
-rw-r--r-- 1 cdac cdac   1 Jun 21 22:46 file1.txt
-rw-r--r-- 1 cdac cdac   1 Jun 21 22:46 file2.txt
-rw-r--r-- 1 cdac cdac   1 Jun 21 22:46 file3.txt
-rw-r--r-- 1 cdac cdac   1 Jun 21 22:46 file4.txt
-rw-r--r-- 1 cdac cdac   1 Jun 21 22:47 file5.txt
-rw-r--r-- 1 cdac cdac 181 Jun 21 22:55 file.tar.gz
```

You can see that the user's permissions for some files is "rw-" as the first three characters. This means that the owner of the file ("cdac here") can "read" it and "write" it (modify its contents) but he cannot execute it because it is not a program; it is a text file.
The second set "r--" of characters means that the members of the group "cdac" can only read the files.

The final three characters show the permissions allowed to anyone who has a UserID on this Linux system. This shows that permission ("r– "). Allows anyone in Linux machine to read, but they cannot modify the contents of the files or execute it.

Syntax: $chmod o+x file5.txt

This Command will give execute permission to the third user



```
┌──(cdac㉿cdac)-[~/Desktop]
└─$ chmod o+x file5.txt

┌──(cdac㉿cdac)-[~/Desktop]
└─$ ls -l
total 24
-rw-r--r-- 1 cdac cdac   1 Jun 21 22:46 file1.txt
-rw-r--r-- 1 cdac cdac   1 Jun 21 22:46 file2.txt
-rw-r--r-- 1 cdac cdac   1 Jun 21 22:46 file3.txt
-rw-r--r-- 1 cdac cdac   1 Jun 21 22:46 file4.txt
-rw-r--r-x 1 cdac cdac   1 Jun 21 22:47 file5.txt
-rw-r--r-- 1 cdac cdac 181 Jun 21 22:55 file.tar.gz
```