CYSE 270: Linux System for Cybersecurity
Assignment: Lab 4 – User and Group Accounts

Goal:

The goal of this lab is to familiarize students with the fundamental tasks of managing user and group accounts in Linux. By completing this lab, students will gain practical experience in creating, modifying, and deleting accounts, as well as managing group memberships and permissions, which are essential skills in system administration and cybersecurity.

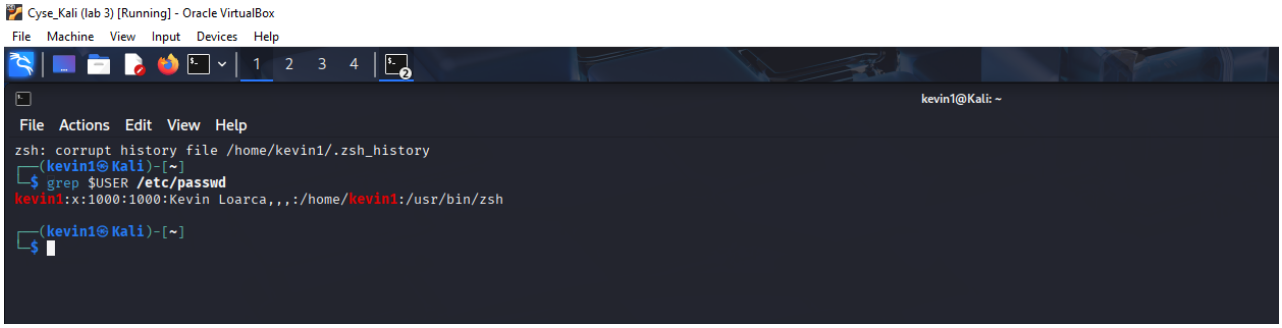
Submission Instructions:

- Complete all tasks in **Task A** and **Task B** on your chosen Ubuntu/Kali VM.
- Take screenshots for each step as evidence of successful command execution.
- Save all your screenshots and results in a single PDF or Word document.
- Ensure that all commands are executed correctly and include detailed explanations for each step taken.

In this assignment, you should replace xxxxx with your MIDAS ID in all occurrences.

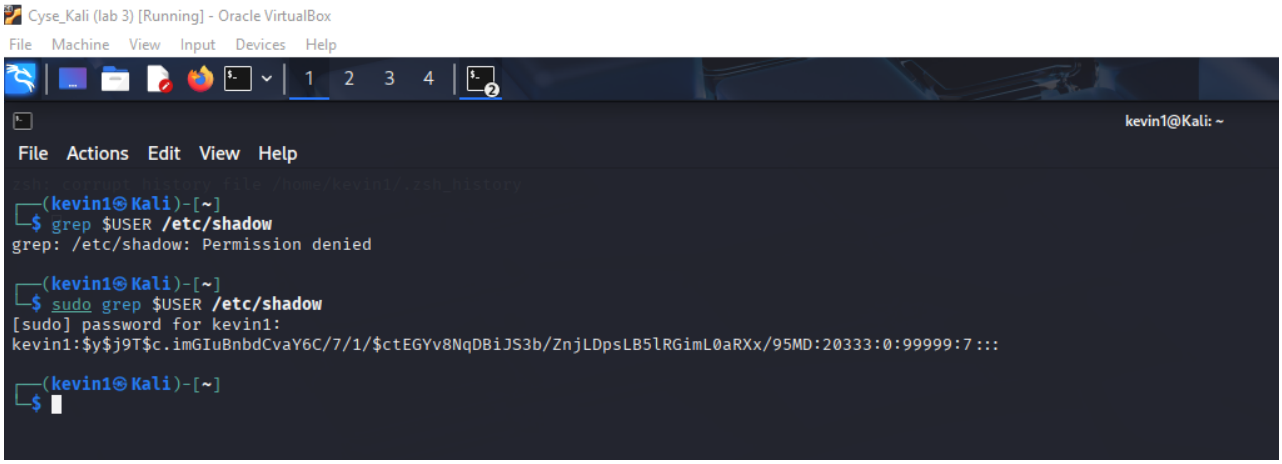
Task A – User Account management (8 * 5 = 40 points)

1. Open a terminal window in VM and execute the correct command to display user account information (including the login shell and home directory) for the current user using grep.



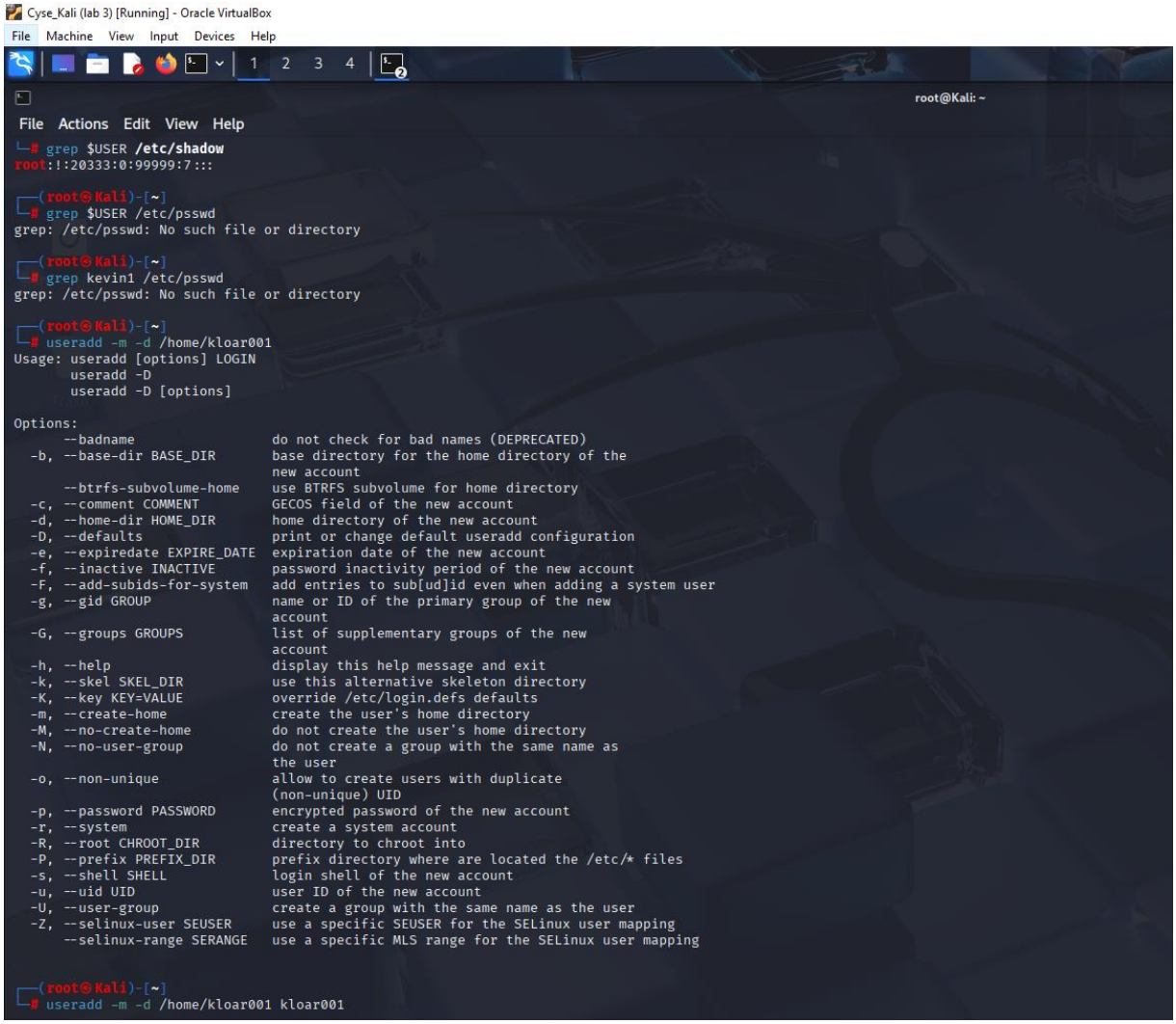
The "grep" command followed by “$USER /etc/passwd” will display the current user’s account information stored in the /etc/passwd.

2. Execute the correct command to display user password information (including the encrypted password and password aging) for the current user using grep.



After changing to root user (with sudo), it allowed me to use the “grep” followed by “$USER /etc/shadow”, which displays the encrypted password.

3. Create a new user named xxxxx and explicitly use options to create the home directory **/home/xxxxx** for this user.



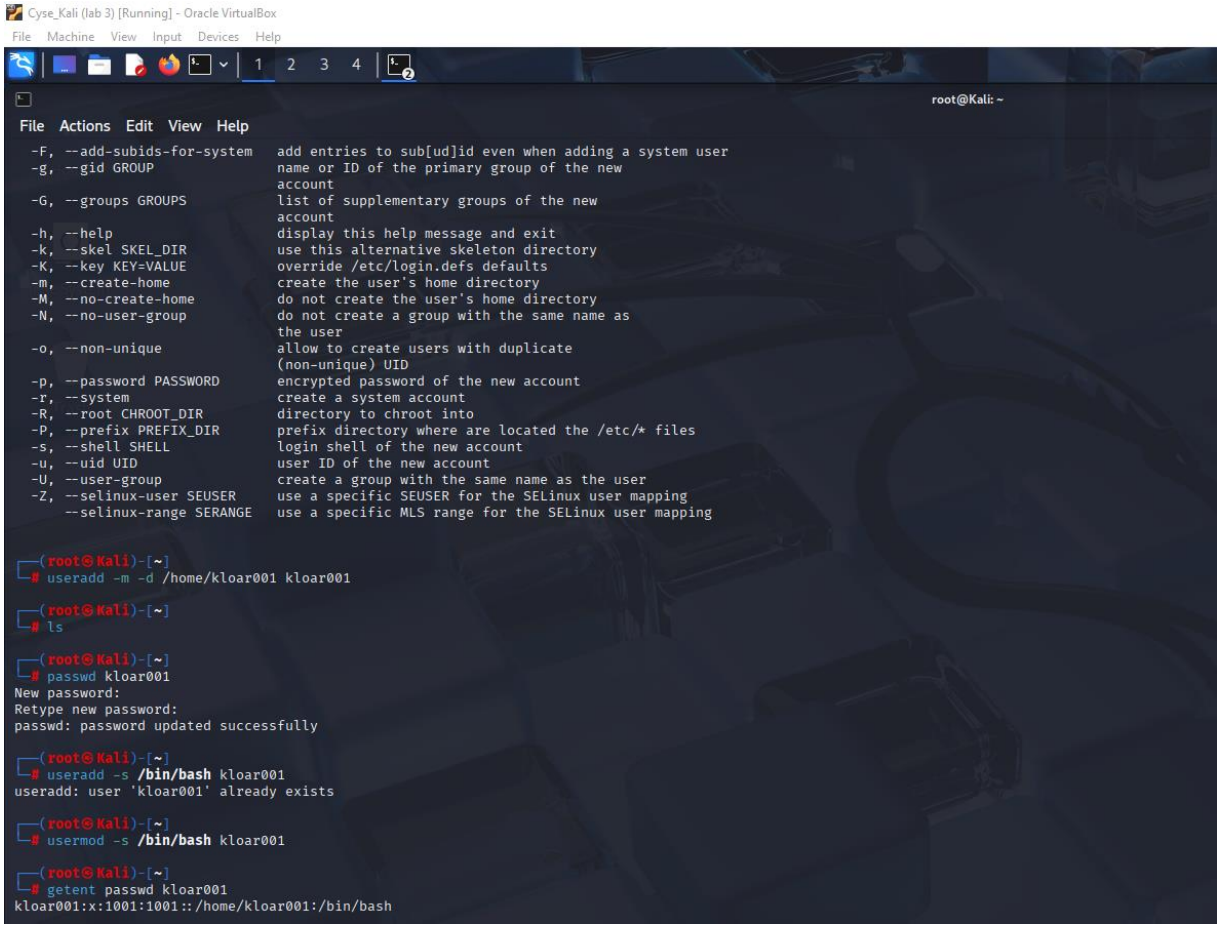
As root user, the use of “useradd -m -d /home/kloar001 kloar001” would create a new user with “m” and “d” creating the home directory of the new account.

4. Set a password for the new user.

```
┌──(root㉿Kali)-[~]
└─# passwd kloar001
New password:
Retype new password:
passwd: password updated successfully

┌──(root㉿Kali)-[~]
└─#
```

While in root user, the use of “passwd” followed by my user account name allows the password to be created for that account.

5. Set bash shell as the default login shell for the new user xxxxx, then verify the change.



While in the root user, the “usermod -s /bin/bash kloar001” would edit the user account information to create shell to bash for the kloar001 account. To confirm that it was created with bash shell, use “getent passwd” with the user account name

6. Execute the correct command to display user password information (including the encrypted password and password aging) for the new user xxxxx using grep.

```
┌──(root㉿Kali)-[~]
└─# grep kloar001 /etc/shadow
kloar001:$y$j9T$whghB7DR3IJiNeBv56sFk0$in3iHtuHMI0AAMTiPSpM5kVyd5bBUA1f2KFP.e9cQ98:20347:0:99999:7:::
```

In the root user, the "grep" command would grab the encrypted password for "kloar001"in the "/etc/passwd".

7. Add the new user xxxxx to sudo group without overriding the existing group membership.

```
┌──(root㉿Kali)-[~]
└─# usermod -aG sudo kloar001

┌──(root㉿Kali)-[~]
└─#
```

With the "usermod" command and followed by "-aG sudo kloar001", it would add the new user without overriding existing group membership.

8. Switch to the new user's account.

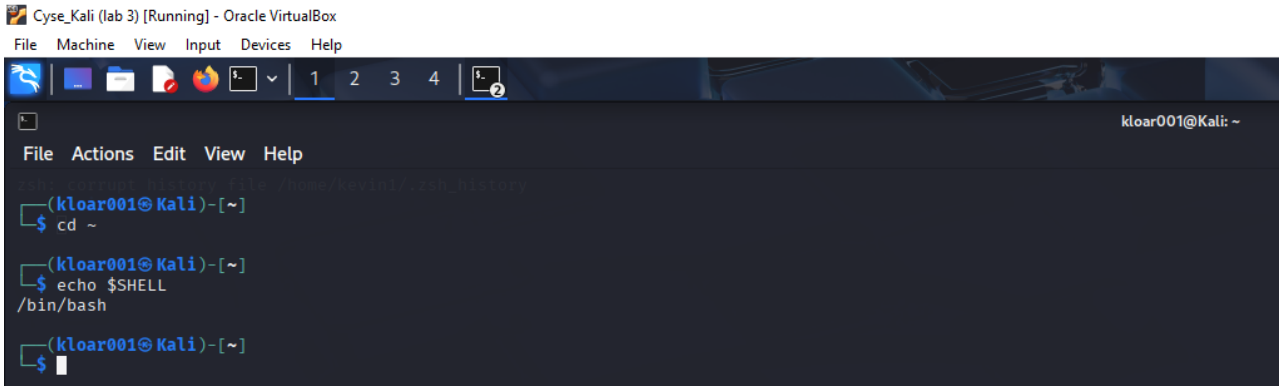
```
┌──(kevin1㉿Kali)-[~]
└─$ su - kloar001
Password:
┌──(kloar001㉿Kali)-[~]
└─$
```

The "su –" followed by the user account name would change to the new account.

Task B – Group account management (12 * 5 = 60 points)

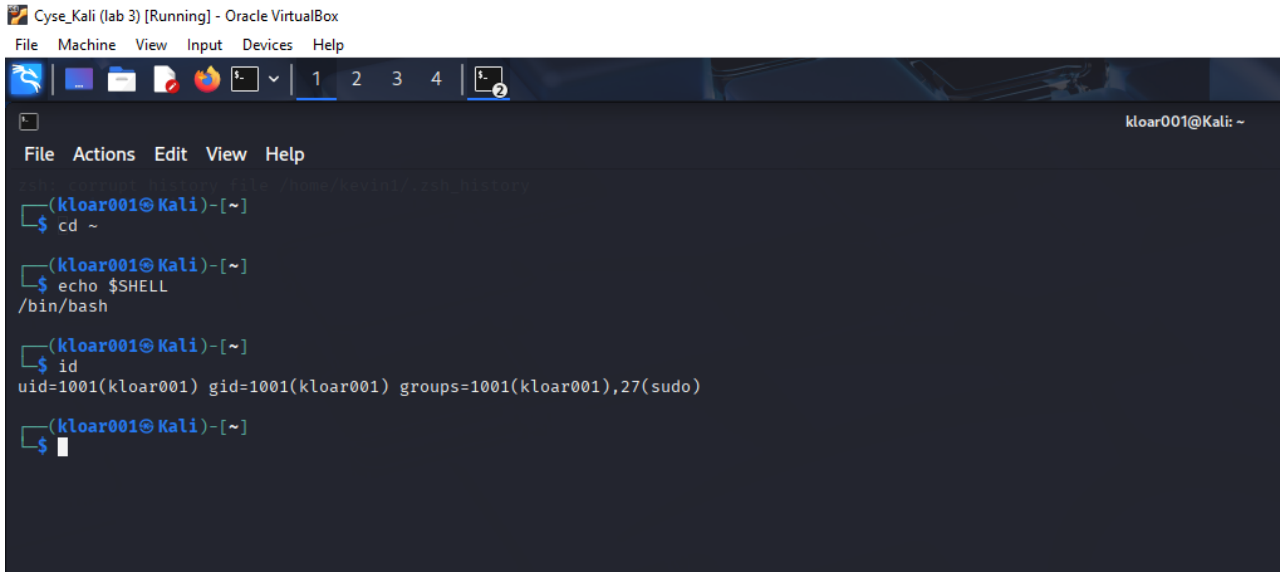
Use Linux commands to execute the following tasks:

1. Return to your home directory and determine the shell you are using.



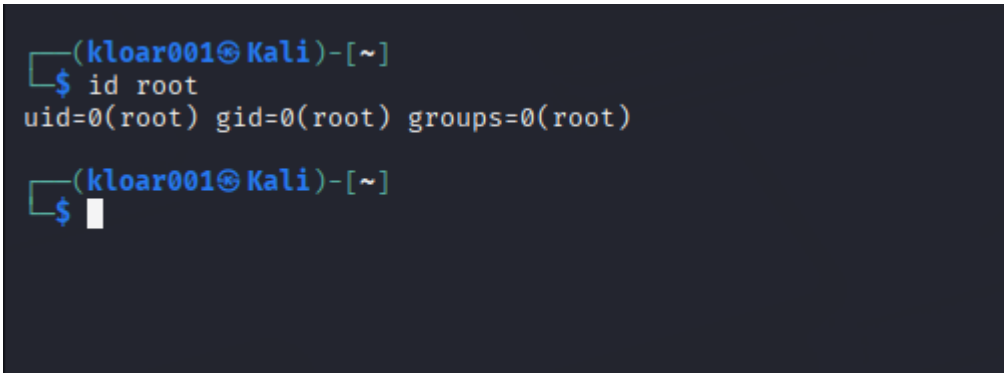
The “cd ~” is to return to home directory. Then the “echo $SHELL” command displays the type of shell.

2. Display the current user’s ID and group membership.



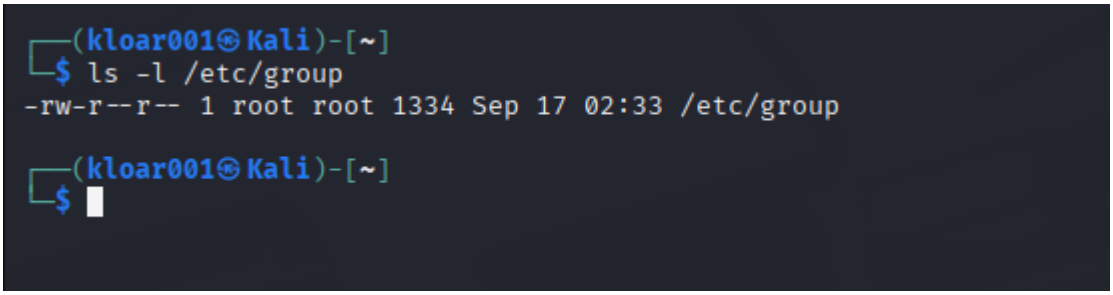
The “id” command displays UID and group membership.

3. Display the group membership of the root account.



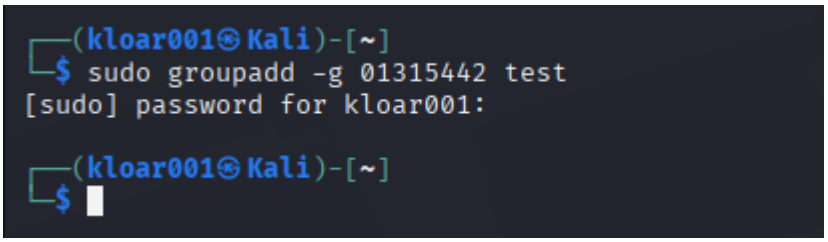
The command “id” and then adding root would display the UID and group membership of root

4. Run the correct command to determine the user owner and group owner of the /etc/group file.



The command “ls -l /etc/group” would display the information in the group, which shows to be root.

5. Create a new group named **test** and use your UIN as the GID.



As the root user, the command “groupadd -g” and my UIN would create the new group named test with the UIN as the GID as my UIN.

6. Display the group account information for the test group using grep.

```
┌──(kloar001㉿Kali)-[~]
└─$ grep test /etc/group
test:x:1315442:

┌──(kloar001㉿Kali)-[~]
└─$
```

The “grep” command grabs the group information from test.

7. Change the group name of the test group to **newtest**.

```
Options:
  -a, --append                  append the users mentioned by -U option to the group
                                without removing existing user members
  -g, --gid GID                 change the group ID to GID
  -h, --help                    display this help message and exit
  -n, --new-name NEW_GROUP      change the name to NEW_GROUP
  -o, --non-unique              allow to use a duplicate (non-unique) GID
  -p, --password PASSWORD       change the password to this (encrypted)
                                PASSWORD
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
  -U, --users USERS             list of user members of this group


┌──(kloar001㉿Kali)-[~]
└─$ sudo groupmod -n newtest test

┌──(kloar001㉿Kali)-[~]
└─$
```

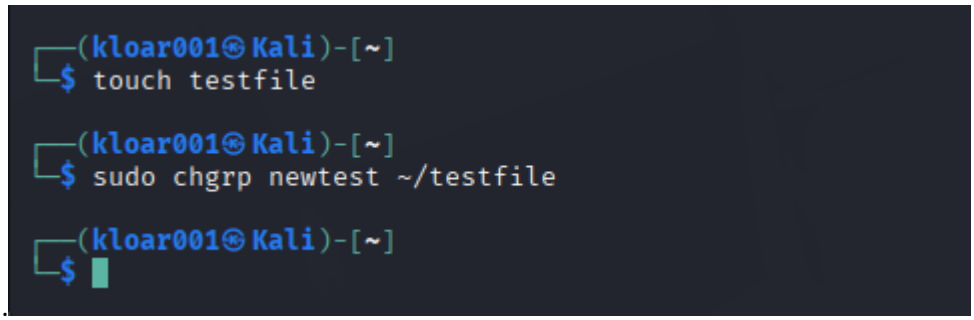
As the root user, the “groupmod -n newtest test” would change the group name (-n does this) test to newtest.

8. Add the current account (xxxxx) as a secondary member of the **newtest** group without overriding this user’s current group membership.

```
└─$ sudo groupmod -n newtest test

┌──(kloar001㉿Kali)-[~]
└─$ sudo usermod -aG newtest kloar001

┌──(kloar001㉿Kali)-[~]
└─$
```

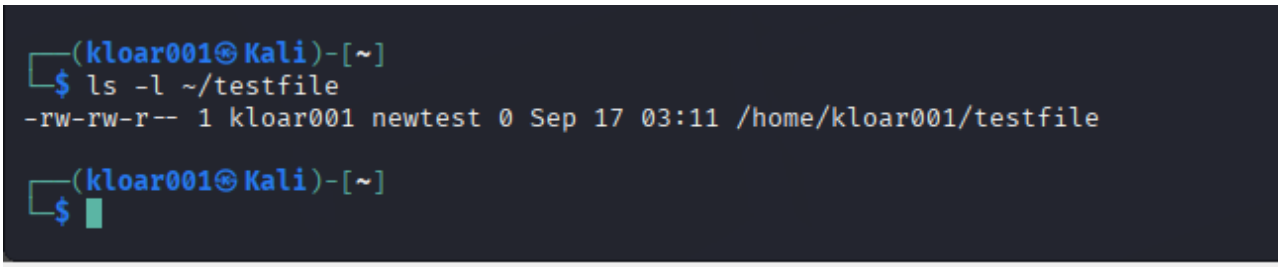
As the root user, the command “usermod -aG newtest kloar001” would put the kloar001 as the secondary member of the newtest group.

9. Create a new file testfile in the account’s home directory, then change the group owner to **newtest**.



The command “touch” would create a file. Then the “sudo chgrp newtest ~/testfile” would change the group owner to newtest.

10. Display the user owner and group owner information of the file **testfile**.



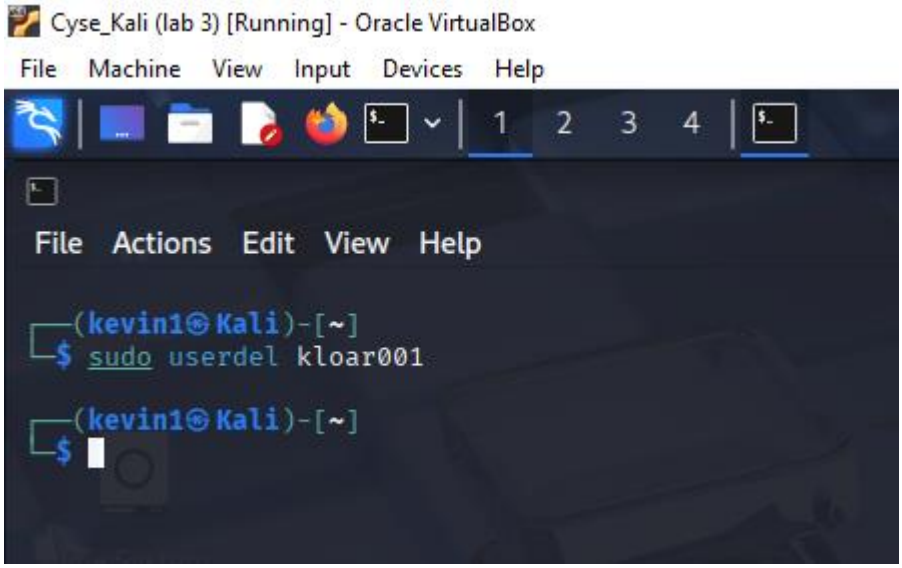
“ls -l” would display the contents of the file.

11. Delete the **newtes**t group, then repeat the previous step. What do you find?

```
┌──(kloar001㉿Kali)-[~]
└─$ ls -l ~/testfile
-rw-rw-r-- 1 kloar001 newtest 0 Sep 17 03:11 /home/kloar001/testfile

┌──(kloar001㉿Kali)-[~]
└─$ sudo groupdel newtest

┌──(kloar001㉿Kali)-[~]
└─$ ls -l ~/testfile
-rw-rw-r-- 1 kloar001 1315442 0 Sep 17 03:11 /home/kloar001/testfile

┌──(kloar001㉿Kali)-[~]
└─$
```

As the root user, the “groupdel newtest” would delete the group. When repeating the previous step, newtest is no longer there.

12. Delete the user xxxxx along with the home directory using a single command.



As the root user, the “userdel” ends up deleting the kloar001 user.

Reflection

In this lab, I practiced several different commands that relate to group and user management. I utilize commands such as usermod to change the properties of a user in Linux. I had also used the grep command to search for specific words in different files. There were some challenges in this lab where I would mistakenly set the users in the wrong member order.