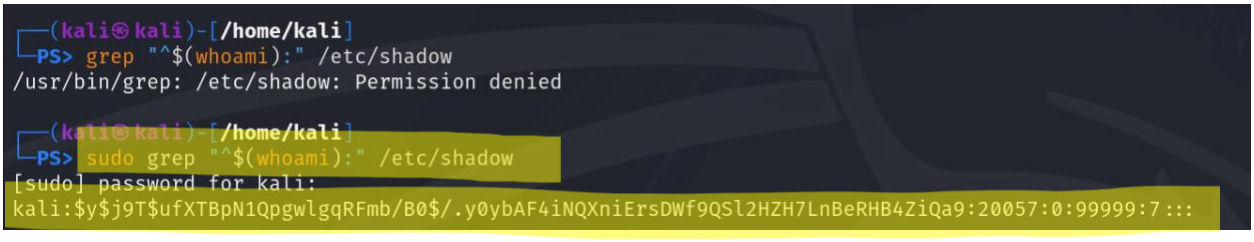
Lab 4: User and Group Accounts
Task A – User Account management

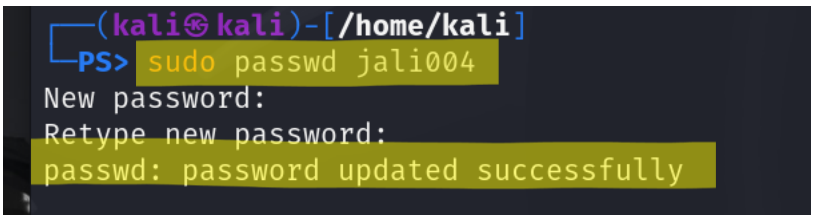
1. Open a terminal window in VM and execute the correct command to display user account information (including the login shell and home directory) for the current user using grep.
Explanation: I use **grep** command followed by **whoami** to display the required information.
Screenshot:

2. Execute the correct command to display user password information (including the encrypted password and password aging) for the current user using grep.
Explanation: I try the **grep** command and it does not work and then I use the **sudo** command followed by the **grep** command with the default password of kali to display the info required.
Screenshot:

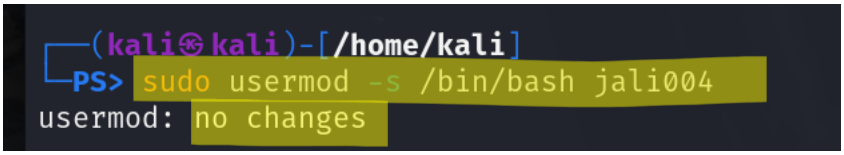


3. Create a new user named xxxxx and explicitly use options to create the home directory /home/xxxxx for this user. In this assignment, you should replace xxxxx with your MIDAS ID in all occurrences.
Explanation: I use **sudo useradd -m -d -s** to create a new user and explicitly use options to create the hd for this user. Notice **/home/** and my default shell for new user **/bin/bash**.
Screenshot:

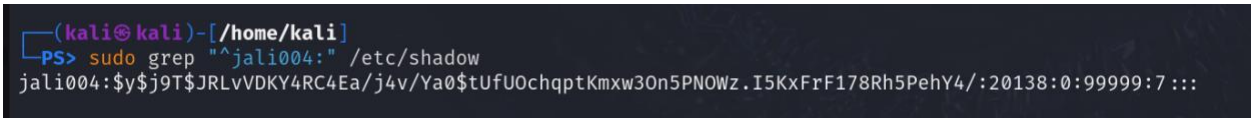
4. Set a password for the new user.
Explanation: Set password to “jess” with command **sudo passwd** followed by with user I am editing, **jali004**.
Screenshot:



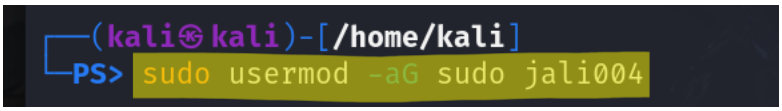
5. Set bash shell as the default login shell for the new user xxxxx, then verify the change.
Explanation: I had already set the default login shell for the new user jali004 in step 3 using **-s** command but I will verify in step 5. I use the **grep** command. I also added a screenshot of the required command if I did not do it in step 3.
Screenshot:



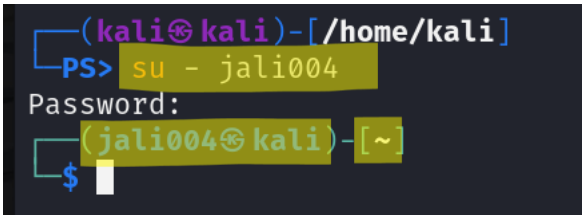
6. Execute the correct command to display user password information (including the encrypted password and password aging) for the new user xxxxx using grep.
Explanation: I use the **sudo grep** command to elevate my privileges within that command line again and display my new user's password within the shadow file. Which is where passwords are stored.
Screenshot:



7. Add the new user xxxxx to sudo group without overriding the existing group membership.
Explanation: Elevating my privilege within the command line with **sudo** and modifying user with **usermod** command followed by what type of modification **-ag** and then defining which user.
Screenshot:



8. Switch to the new user's account.
Explanation: **su** command to switch to my newly created user followed by the password I created. Successful as seen below.
Screenshot:

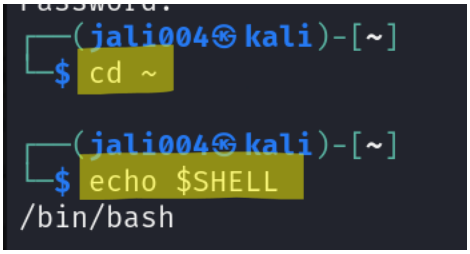


Task B – Group account management

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

Explanation: I return to my hd using command **cd ~** and then display the shell I am using with the **echo** command followed by **$SHELL** to display what I want to display.

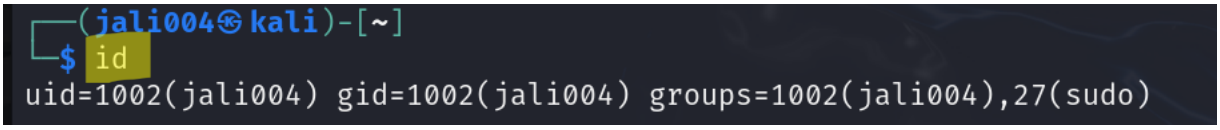
Screenshot:



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

Explanation: I use the command **id** to display required info.

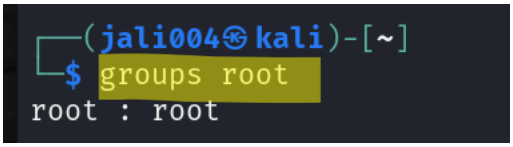
Screenshot:



3. Display the group membership of the root account.

Explanation: I use command **groups** to define that I want to display followed by what user, in this case I am looking at **root**.

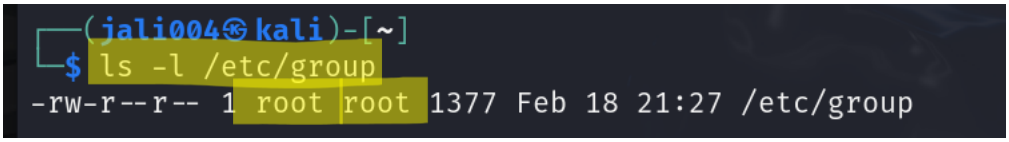
Screenshot:



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

Explanation: I use the **ls -l** command to get all of the info needed from a file. I define the file **/etc** followed by what info I am looking at, which in this case is group owner and user owner so I define with **/group**.

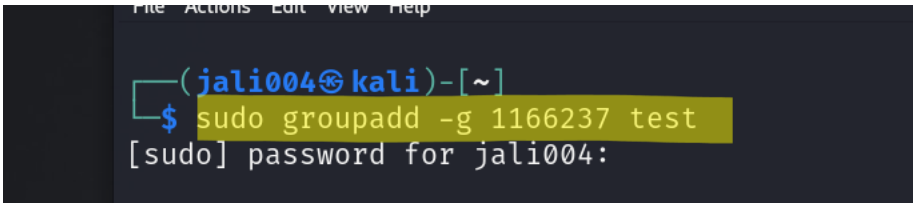
Screenshot:



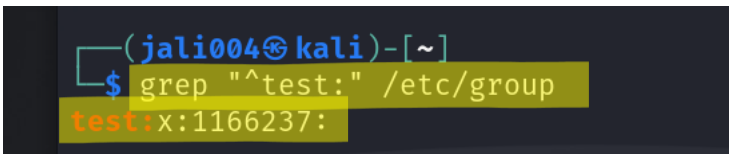
5. Create a new group named test and use your UIN as the GID.

Explanation: I elevate my privs again for the command line with **sudo** and then use the command **groupadd -g**, define the GID, which is my UIN minus the 0s because it didn't like

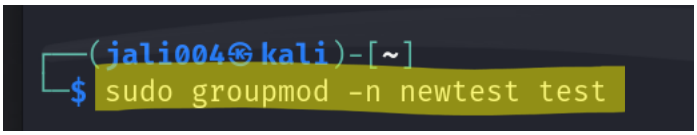
the 0s last time and then finally the title of the group, which in this case is **test**. I also typed in my password for this user, “jess”.
Screenshot:



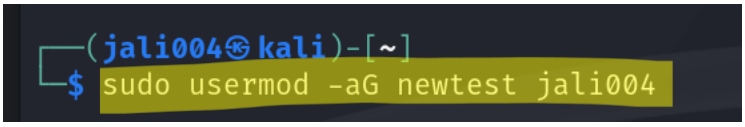
6. Display the group account information for the test group using grep.
Explanation: I use **grep** command followed by a caret to search for strings/lines containing test withing the group file. Which is why I followed the command with **/etc/group**. I needed to check if the command in step 5 was successful, and it was.
Screenshot:



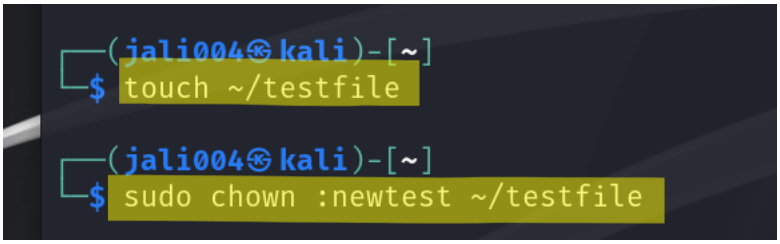
7. Change the group name of the test group to newtest.
Explanation: Elevating my privs with **sudo** command followed by **groupmod** because I am modifying a file to a new name **-n**, followed by the new name and defining what file I am changing.
Screenshot:



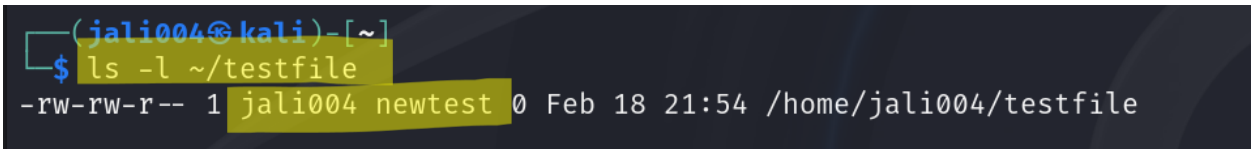
8. Add the current account (xxxxx) as a secondary member of the newtest group without overriding this user’s current group membership.
Explanation: Elevating my privs with **sudo** command followed by **groupmod** because I am modifying where the user will be followed by **-aG** for secondary member and the location **newtest**.
Screenshot:

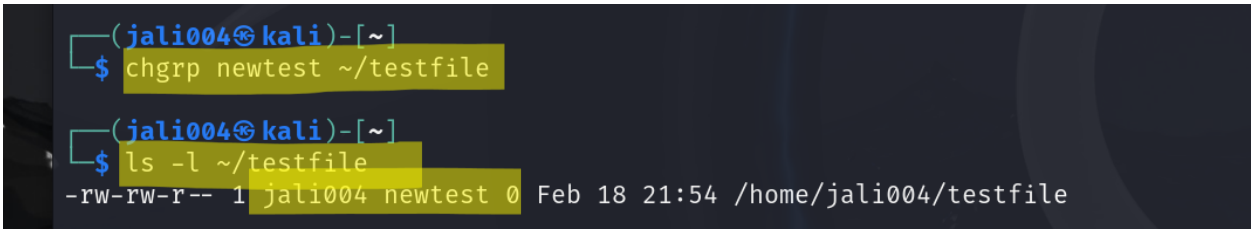


9. Create a new file testfile in the account’s home directory, then change the group owner to newtest.
Explanation: Creating a new file with **touch** command in home directory **~**, followed in the next command line, elevating privs with **sudo** followed by **chown :newtest ~/testfile** to then change the group owner to newtest.
Screenshot:

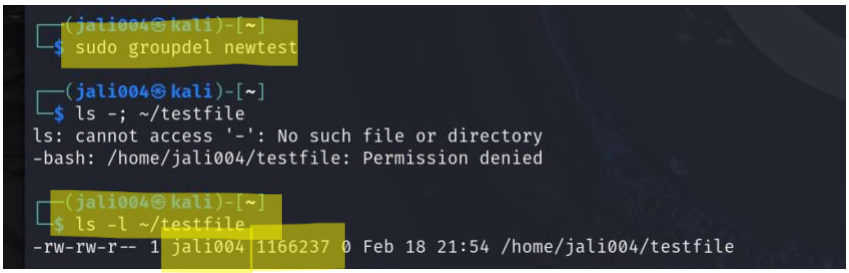


10. Display the user owner and group owner information of the file testfile.
Explanation: To check the owner and group of testfile I use the **ls -l** command followed by the location of the file **~/testfile**. I also used another command later to double check because things were going to smooth, **chgrp**.
Screenshot:



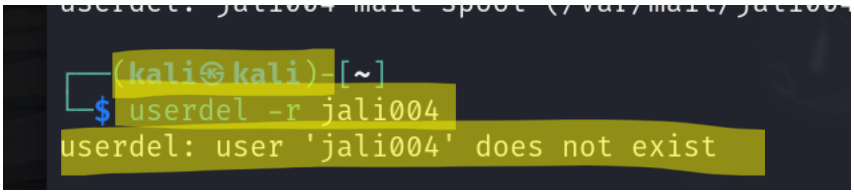


11. Delete the newtest group, then repeat the previous step. What do you find?
Explanation: Elevate my privs for the command, delete the newtest group with the **groupdel** command. I mess up and then I long list the testfile. You can also see the previous user, my UIN, 1166237 from a different day. The newtest group is gone.
Screenshot:



12. Delete the user xxxxx along with the home directory using a single command.
Explanation: I switch back to user kali with command **su -** , followed by the user I want, which in this case is kali. I then from this shell raise my privs again for this command line because to delete we need root privs, **sudo**, followed by **userdel** to delete, ending with a flag **-r** because I also want to delete the home directory of the user. Then I ran into an error, switched back to jali004 using **su –** command and killed the process with **pkill -KILL -u jali004**. Double checked deletion with **userdel -r jali004** .
Screenshot:


┌──(jali004㉿kali)-[~]
└─$ su - kali
Password:
┌──(kali㉿kali)-[~]
└─$ sudo userdel -r jali004
[sudo] password for kali:
userdel: user jali004 is currently used by process 96821
┌──(kali㉿kali)-[~]
└─$ userdel -r jali004
userdel: Permission denied.
userdel: cannot lock /etc/passwd; try again later.
┌──(kali㉿kali)-[~]
└─$ sudo userdel -r jali004
userdel: user jali004 is currently used by process 96821
┌──(kali㉿kali)-[~]
└─$ pkill
pkill: no matching criteria specified
Try `pkill --help' for more information.
┌──(kali㉿kali)-[~]
└─$ pkill -KILL -u jali004
pkill: killing pid 96821 failed: Operation not permitted
┌──(kali㉿kali)-[~]
└─$ su -jali004
su: invalid option -- 'j'
Try 'su --help' for more information.
┌──(kali㉿kali)-[~]
└─$ su - jali004
Password:
┌──(jali004㉿kali)-[~]
└─$ pkill -KILL -u jali004
Killed
Killed
Process terminated. Input/output error
at System.Environment.FailFast(System.String, System.Exception)
at Microsoft.PowerShell.UnmanagedPSEntry.Start(System.String[], Int32)
at Microsoft.PowerShell.ManagedPSEntry.Main(System.String[])
System.IO.IOException: Input/output error
at System.ConsolePal.TryGetCursorPosition(Int32& left, Int32& top, Boo
at System.ConsolePal.GetCursorPosition()


┌──(kali㉿kali)-[~]
└─$ userdel -r jali004
userdel: user 'jali004' does not exist