Carl Lochstampfor

CYSE 270: Linux System for Cybersecurity

Assignment: Lab 4 – Group and User Management

June 11, 2025

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:

1. Complete all tasks in **Task A** and **Task B** on your chosen Ubuntu/Kali VM.
2. Take screenshots for each step as evidence of successful Command execution.
3. Save all your screenshots and results in a single PDF or Word document.
4. Ensure that all **Commands are executed** correctly **AND**
    a. Include **detailed Explanations** for each step taken.

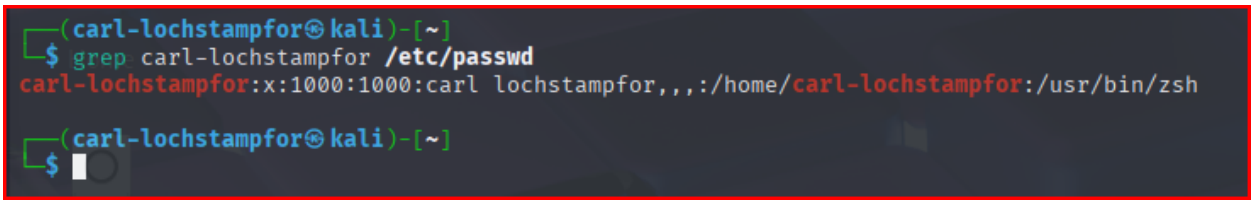
In this assignment, you should replace xxxxx (cloch001) with your MIDAS ID in all occurrences.

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**.

Command: **grep carl-lochstampfor /etc/passwd**

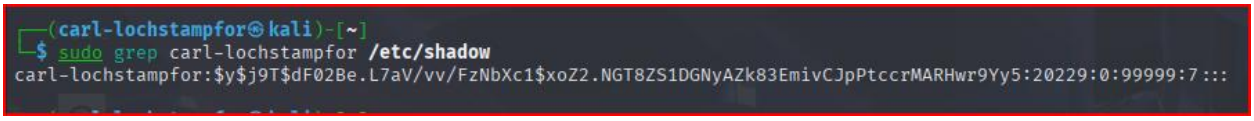
Explanation: **grep** searches for the string/pattern carl-lochstampfor, **carl-lochstampfor** is the target string/patter, and the **/etc/passwd** is the file that is being searched and contains user account information (i.e., carl-lochstampfor).



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

Command: **sudo grep carl-lochstampfor /etc/shadow**

Explanation: **Sudo** runs the Command in root privileges, **grep** searches for the string carl-lochstampfor, **/etc/shadow** is the file being searched and it contains shadowed group information like encrypted passwords, group admins and group members.



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

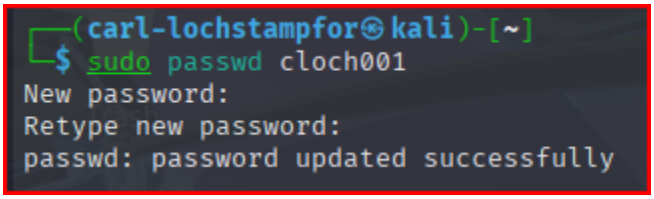
Command**: sudo useradd -m -d /home/cloch001 cloch001**

Explanation: Creating the home directory for the user cloch001. **Sudo** runs the Command in root privileges, **useradd** creates the new user (cloch001), **-m** creates a home directory for the user (because it doesn't currently exist), **-d /home/cloch001** explicitly sets the home directory path to /home/cloch001, and the **cloch001** is the new user.

4. Set a password for the new user.

Command: **sudo passwd cloch001**

Explanation: **Sudo** runs the Command in root privileges, **psswd** seeks to change the password for the following user, and **cloch001** is that user.

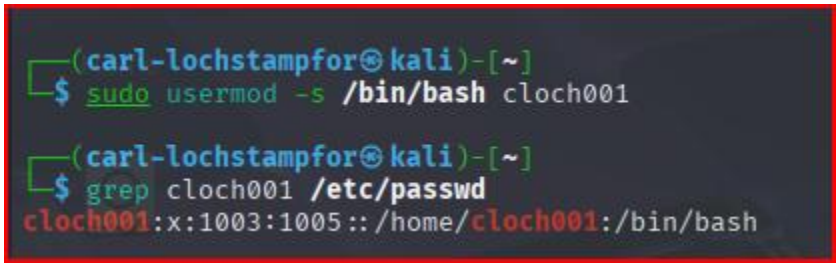


5. Set bash shell as the default login shell for the new user **xxxxx**, then verify the change.

Command: **sudo usermod -s /bin/bash cloch001, then grep cloch001 /etc/passwd** (verifies the change)

Explanation:

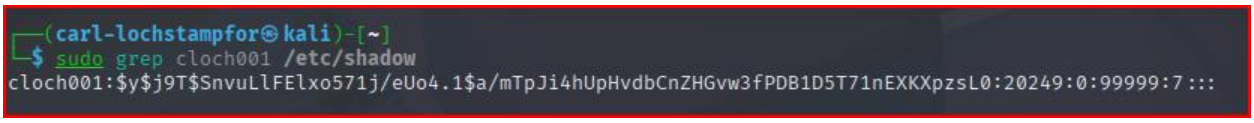
a. **Sudo** runs the Command in root privileges, **usermod** modifies the user account privileges, **-s /bin/bash** sets the login shell for the user to /bin/bash (path to the Bash shell), and **cloch001** is the target user account we are modifying.
b. **grep** searches for the string cloch001 **cloch001** is the target string, and the **/etc/passwd** contains user account information and is the file being searched for the exact string **cloch001**, showing the changes to default login shell to cloch001/bin/bash.



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

Command: **sudo grep cloch001 /etc/shadow**

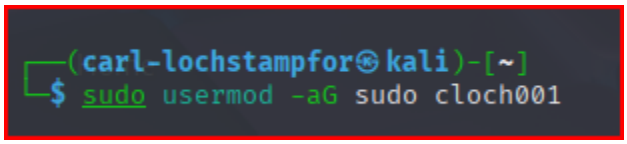
Explanation: **Sudo** runs the Command in root privileges, **grep** searches for the string cloch001, **/etc/shadow** is the file being searched and it contains shadowed group information like encrypted passwords, group admins and group members.



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

Command: **sudo usermod -aG sudo cloch001**

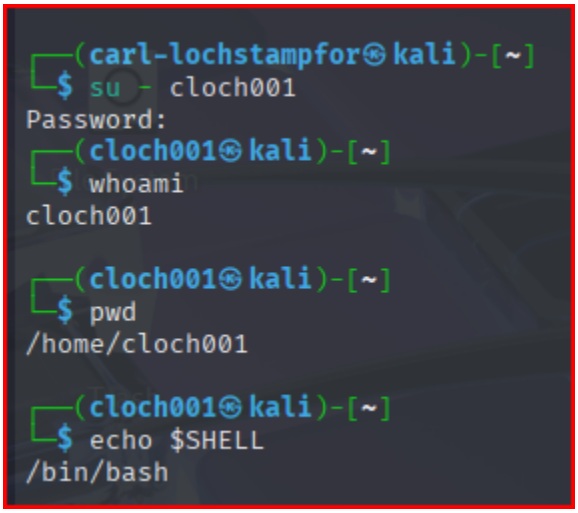
Explanation: **Sudo** runs the Command in root privileges, **usermod** modifies the target user's account, **-aG** appends the user to the target group (sudo) without removing existing group memberships, **sudo** is the target group, and **cloch001** is the username to modify.



8. Switch to the new user's account.

Command: **su – cloch001**

Explanation: Su – switches to the user, cloch001. Whoami verifies we switched to the new user's account.

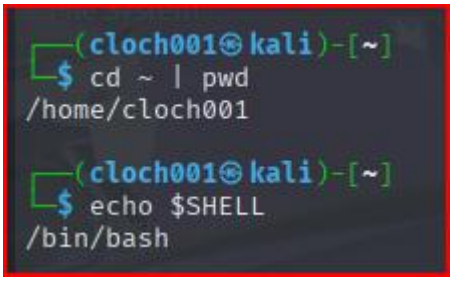


Task B – Group account management (12 * 5 = 60 points)

Use Linux Commands to execute the following tasks:

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

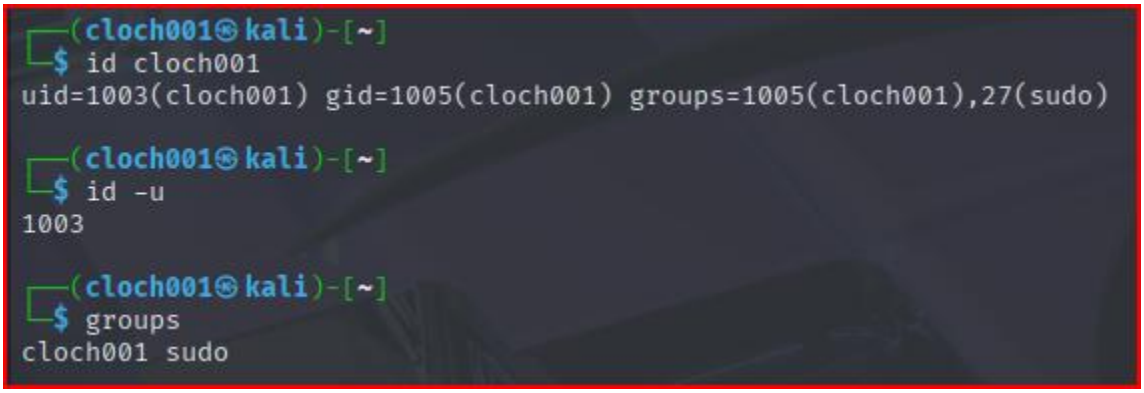
Command: **cd ~ pwd**, then **echo $SHELL**
Explanation: cd ~ returns to the home directory, pwd prints the working directory, and echo $SHELL displays the default shell for the user cloch001 (i.e., /bin/bash).



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

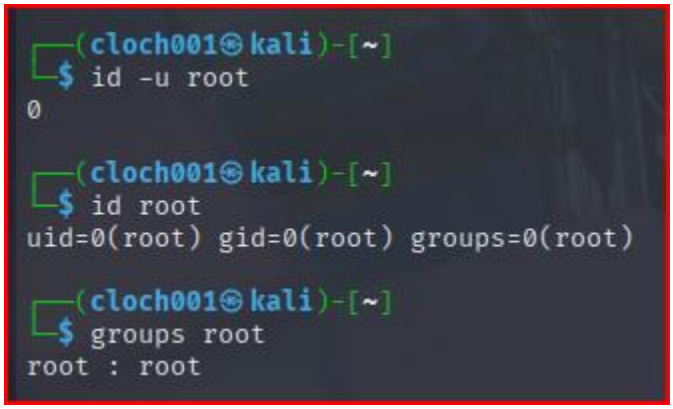
Command: **id cloch001**; can also use **id -u**, then **groups** to display only specific information.

Explanation: **id cloch001** displays the detailed information of the user, cloch001, on one line together; **id -u** displays the current user id number, and **groups** lists all of the groups cloch001 is a member of (i.e., sudo).



3. Display the group membership of the root account.

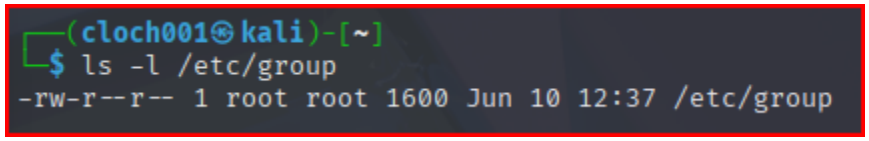
Command: **id -u root; id root; root : root**
Explanation: **id root** displays detailed information about the root account, including the user id number, group id number, and the number of additional groups the root account is a member of (i.e., zero). The root account is only a member of the root group.



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

Command: **ls -l /etc/group**

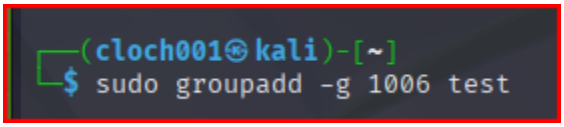
Explanation: The **ls -l** Command lists the files and directories in long format, providing detailed information about **/etc/group**. The User Owner is root with permissions (rw-r—r—) that owns the /etc/group file; the second field shows the root is the Group Owner of the file (1600 June 10 12:37 /etc/group).



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

Command: **sudo groupadd -g 1006 test**

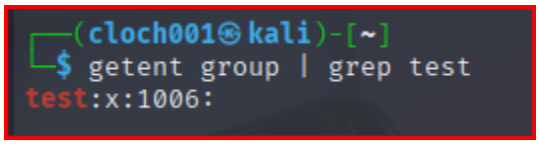
Explanation: **Sudo** runs the Command in root privileges, **groupadd** creates a group, **-g** assigns the GID to the group (i.e., 1006), and **test** is the group name.
** **Note** ** My UIN was assigned to a group already, so I manually assigned **1006** to the **test** group.



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

Command: **getent group | grep test**

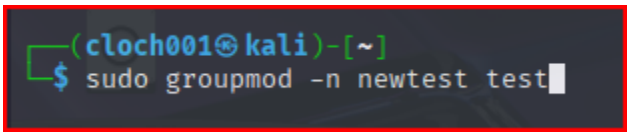
Explanation: **getent group** retrieves a list of groups from the system's group database (under cloch001), **grep test** filters the output to only display group test.



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

Command: **sudo groupmod -n newtest test**

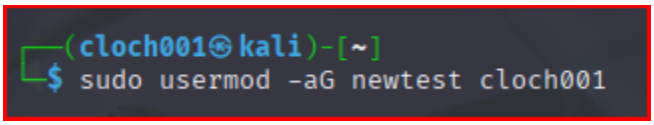
Explanation: **Sudo** runs the Command in root privileges, **groupmod -n** changes the group name from **test** to **newtest**.



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

Command: **sudo usermod -aG newtest cloch001**

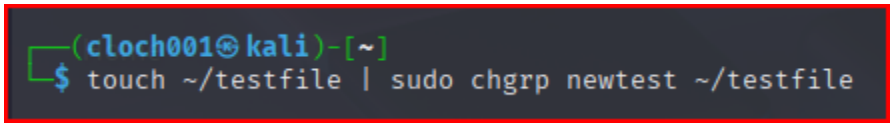
Explanation: **Sudo** runs the Command in root privileges, **usermod** modifies the target user's account, **-aG** appends the user to the target group (sudo) without removing existing group memberships, **newtest** is the target group to add the user to, and **cloch001** is the username to modify.



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

Command: **touch ~/testfile | sudo chgrp newtest ~/testfile**

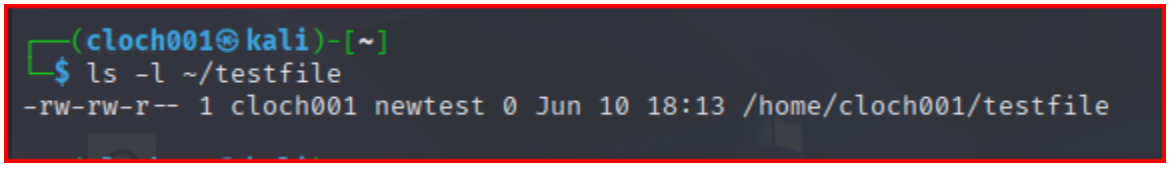
Explanation: **touch** creates an empty new file called **testfile**; **Sudo** runs the Command in root privileges, **chgrp** changes the group owner of **testfile** to **newtest**. ~/testfile is the specific path to testfile in the home directory.



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

Command: **ls -l ~/testfile**

Explanation: The **ls -l** lists the file in long format, displaying its details like the user owner and group owner, and ~/testfile is the specific path to testfile in the home directory. The file is owned by cloch001, and the group owner is newtest.



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

Command: **sudo groupdel newtest | ls -l ~/testfile**

Explanation:

1. **Sudo** runs the Command in root privileges, **groupdel** deletes the group newtest.
2. The **ls -l** lists the file in long format, displaying its details like the user owner and group owner, and **~/testfile** is the specific path to **testfile** in the home directory.
3. **What did you find?** The group owner, newtest, no longer exists. The user owner remains cloch001, and the GID remains the same and still belongs to newtest (cloch001 is 1003/1005). Linux now shows only the numeric GID (1006) instead of the deleted group name (newtest).

```
┌──(cloch001㉿kali)-[~]
└─$ sudo groupdel newtest

┌──(cloch001㉿kali)-[~]
└─$ ls -l ~/testfile
-rw-rw-r-- 1 cloch001 1006 0 Jun 10 18:13 /home/cloch001/testfile
```

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

Command: **sudo userdel -r cloch001**

Explanation: **Sudo** runs the Command in root privileges, **userdel** deletes a user account, **-r** removes the user's home directory, and the targeted **username** deleted is cloch001.

```
┌──(cloch001㉿kali)-[~]
└─$ sudo userdel -r cloch001
[sudo] password for cloch001:
userdel: user cloch001 is currently used by process 235523

┌──(cloch001㉿kali)-[~]
└─$ exit
logout

┌──(carl-lochstampfor㉿kali)-[~]
└─$ sudo userdel -r cloch001
[sudo] password for carl-lochstampfor:
userdel: cloch001 mail spool (/var/mail/cloch001) not found

┌──(carl-lochstampfor㉿kali)-[~]
└─$ su - cloch001
su: user cloch001 does not exist or the user entry does not contain all the required fields
```