

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.

CYSE 270: Linux System for Cybersecurity

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`.

```
Cyse_Kali (lab 3) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
kevin1@Kali: ~
File Actions Edit View Help
zsh: corrupt history file /home/kevin1/.zsh_history
(kevin1@Kali)-[~]
$ grep $USER /etc/passwd
kevin1:x:1000:1000:Kevin Loarca,,,:/home/kevin1:/usr/bin/zsh
(kevin1@Kali)-[~]
$
```

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.

```
Cyse_Kali (lab 3) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
kevin1@Kali: ~
File Actions Edit View Help
(kevin1@Kali)-[~]
$ grep $USER /etc/shadow
grep: /etc/shadow: Permission denied
(kevin1@Kali)-[~]
$ sudo grep $USER /etc/shadow
[sudo] password for kevin1:
kevin1:$y$j9T$c.imGIuBnbdCvaY6C/7/1/$ctEGYv8NqDBiJS3b/ZnjLDpsLB5lRGimL0aRXx/95MD:20333:0:99999:7 :::
(kevin1@Kali)-[~]
$
```

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.

```
Cyse_Kali (lab 3) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@Kali: ~
File Actions Edit View Help
└─$ grep $USER /etc/shadow
root:!:20333:0:99999:7:::

(root@Kali)~# grep $USER /etc/psswd
grep: /etc/psswd: No such file or directory

(root@Kali)~# grep kevin1 /etc/psswd
grep: /etc/psswd: No such file or directory

(root@Kali)~# useradd -m -d /home/kloar001
Usage: useradd [options] LOGIN
       useradd -D
       useradd -D [options]

Options:
  --badname                do not check for bad names (DEPRECATED)
  -b, --base-dir BASE_DIR  base directory for the home directory of the
                           new account
  --btrfs-subvolume-home  use BTRFS subvolume for home directory
  -c, --comment COMMENT    GECOS field of the new account
  -d, --home-dir HOME_DIR  home directory of the new account
  -D, --defaults           print or change default useradd configuration
  -e, --expiredate EXPIRE_DATE expiration date of the new account
  -f, --inactive INACTIVE  password inactivity period of the new account
  -F, --add-subids-for-system add entries to sub[uid]id even when adding a system user
  -g, --gid GROUP          name or ID of the primary group of the
                           new account
  -G, --groups GROUPS      list of supplementary groups of the
                           new account
  -h, --help              display this help message and exit
  -k, --skel SKEL_DIR     use this alternative skeleton directory
  -K, --key KEY=VALUE     override /etc/login.defs defaults
  -m, --create-home       create the user's home directory
  -M, --no-create-home    do not create the user's home directory
  -N, --no-user-group     do not create a group with the same name as
                           the user
  -o, --non-unique        allow to create users with duplicate
                           (non-unique) UID
  -p, --password PASSWORD encrypted password of the new account
  -r, --system            create a system account
  -R, --root CHROOT_DIR   directory to chroot into
  -P, --prefix PREFIX_DIR prefix directory where are located the /etc/* files
  -s, --shell SHELL       login shell of the new account
  -u, --uid UID           user ID of the new account
  -U, --user-group        create a group with the same name as the user
  -Z, --selinux-user SEUSER use a specific SEUSER for the SELinux user mapping
  --selinux-range SERANGE use a specific MLS range for the SELinux user mapping

(root@Kali)~# useradd -m -d /home/kloar001 kloar001
```

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.

```
Cyse_Kali (lab 3) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

root@Kali: ~
File Actions Edit View Help
-F, --add-subids-for-system add entries to sub[uid]id even when adding a system user
-g, --gid GROUP name or ID of the primary group of the new account
-G, --groups GROUPS list of supplementary groups of the new account
-h, --help display this help message and exit
-k, --skel SKEL_DIR use this alternative skeleton directory
-K, --key KEY=VALUE override /etc/login.defs defaults
-m, --create-home create the user's home directory
-M, --no-create-home do not create the user's home directory
-N, --no-user-group do not create a group with the same name as the user
-o, --non-unique allow to create users with duplicate (non-unique) UID
-p, --password PASSWORD encrypted password of the new account
-r, --system create a system account
-R, --root CHROOT_DIR directory to chroot into
-P, --prefix PREFIX_DIR prefix directory where are located the /etc/* files
-s, --shell SHELL login shell of the new account
-u, --uid UID user ID of the new account
-U, --user-group create a group with the same name as the user
-Z, --selinux-user SEUSER use a specific SEUSER for the SELinux user mapping
--selinux-range SERANGE use a specific MLS range for the SELinux user mapping

(root@Kali)-[~]
# useradd -m -d /home/kloar001 kloar001

(root@Kali)-[~]
# ls

(root@Kali)-[~]
# passwd kloar001
New password:
Retype new password:
passwd: password updated successfully

(root@Kali)-[~]
# useradd -s /bin/bash kloar001
useradd: user 'kloar001' already exists

(root@Kali)-[~]
# usermod -s /bin/bash kloar001

(root@Kali)-[~]
# getent passwd kloar001
kloar001:x:1001:1001::/home/kloar001:/bin/bash
```

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

- 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$whghB7DR3IjIjNeBv56sFk0$in3iHtuHMI0AAMT1P5pM5kVyd5bBUA1F2KFP.e9cQ98:20347:0:99999:7:::
```

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

- 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.

- Switch to the new user’s account.

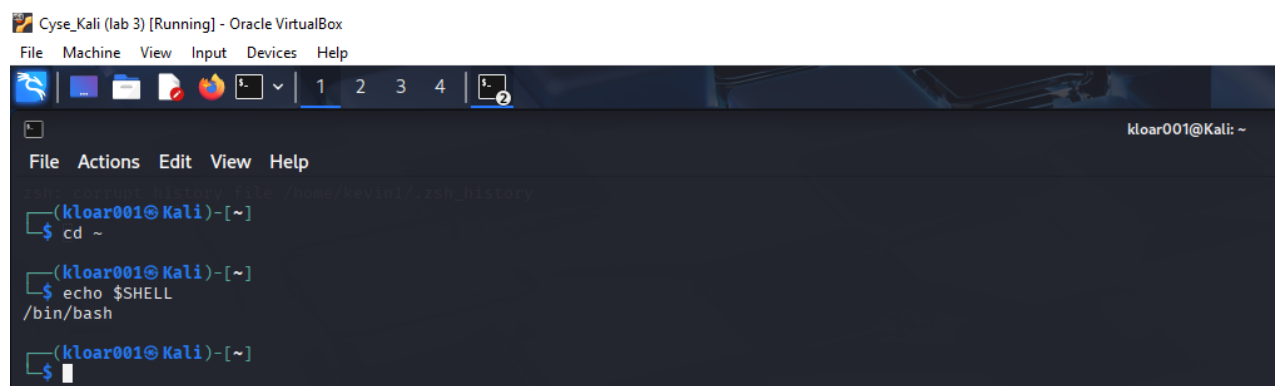
```
(kevin1@Kali)-[~]
$ su - kloar001
Password:
(kloar001@Kali)-[~]
$
```

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:

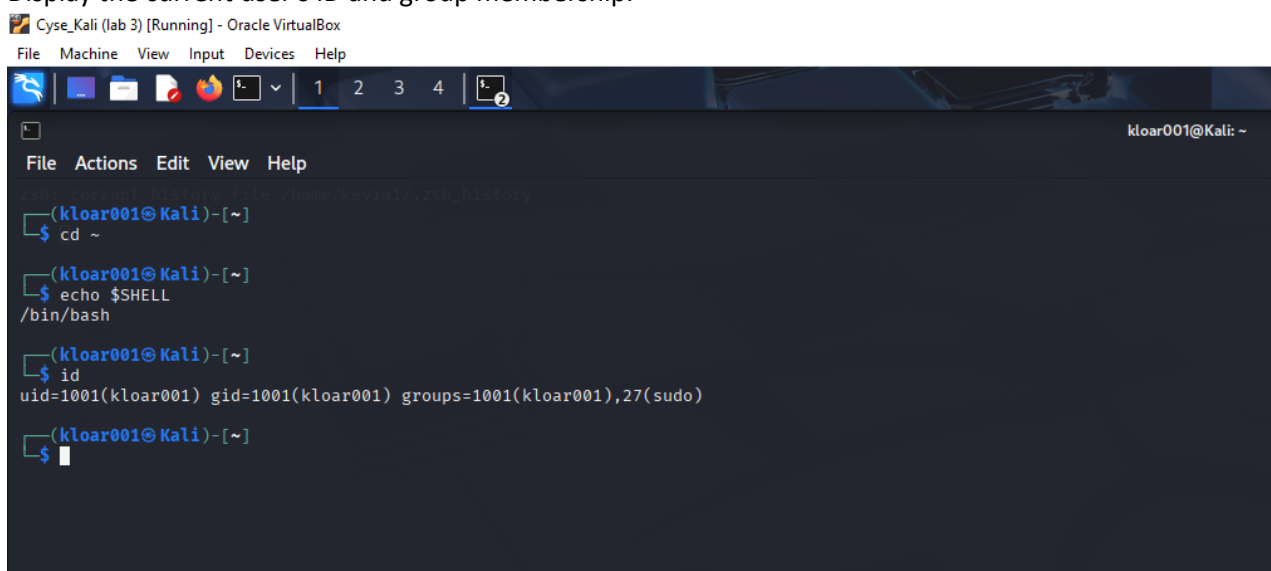
- Return to your home directory and determine the shell you are using.



```
Cyse_Kali (lab 3) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
kloar001@Kali: ~
File Actions Edit View Help
(kloar001@Kali)-[~]
$ cd ~
(kloar001@Kali)-[~]
$ echo $SHELL
/bin/bash
(kloar001@Kali)-[~]
$
```

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.



```
Cyse_Kali (lab 3) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
kloar001@Kali: ~
File Actions Edit View Help
(kloar001@Kali)-[~]
$ cd ~
(kloar001@Kali)-[~]
$ echo $SHELL
/bin/bash
(kloar001@Kali)-[~]
$ id
uid=1001(kloar001) gid=1001(kloar001) groups=1001(kloar001),27(sudo)
(kloar001@Kali)-[~]
$
```

The “id” command displays UID and group membership.

3. Display the group membership of the root account.

```
(kloar001@Kali)-[~]
└─$ id root
uid=0(root) gid=0(root) groups=0(root)

(kloar001@Kali)-[~]
└─$
```

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.

```
(kloar001@Kali)-[~]
└─$ ls -l /etc/group
-rw-r--r-- 1 root root 1334 Sep 17 02:33 /etc/group

(kloar001@Kali)-[~]
└─$
```

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.

```
(kloar001@Kali)-[~]
└─$ sudo groupadd -g 01315442 test
[sudo] password for kloar001:

(kloar001@Kali)-[~]
└─$
```

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**.

```
(kloar001@Kali)-[~]
└─$ touch testfile

(kloar001@Kali)-[~]
└─$ sudo chgrp newtest ~/testfile

(kloar001@Kali)-[~]
└─$ █
```

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**.

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

(kloar001@Kali)-[~]
└─$ █
```

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

11. Delete the **newtest** 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

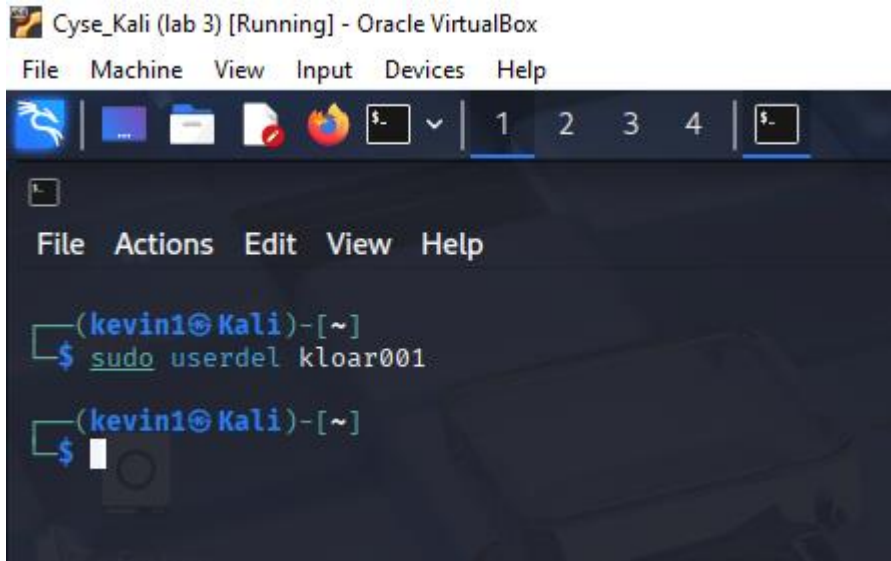
(kloar001@Kali)-[~]
└─$ sudo groupdel newtest

(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.



```
Cyse_Kali (lab 3) [Running] - Oracle VirtualBox
File Machine View Input Devices Help

(kevin1@Kali)-[~]
└─$ sudo userdel kloar001

(kevin1@Kali)-[~]
└─$ █
```

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.