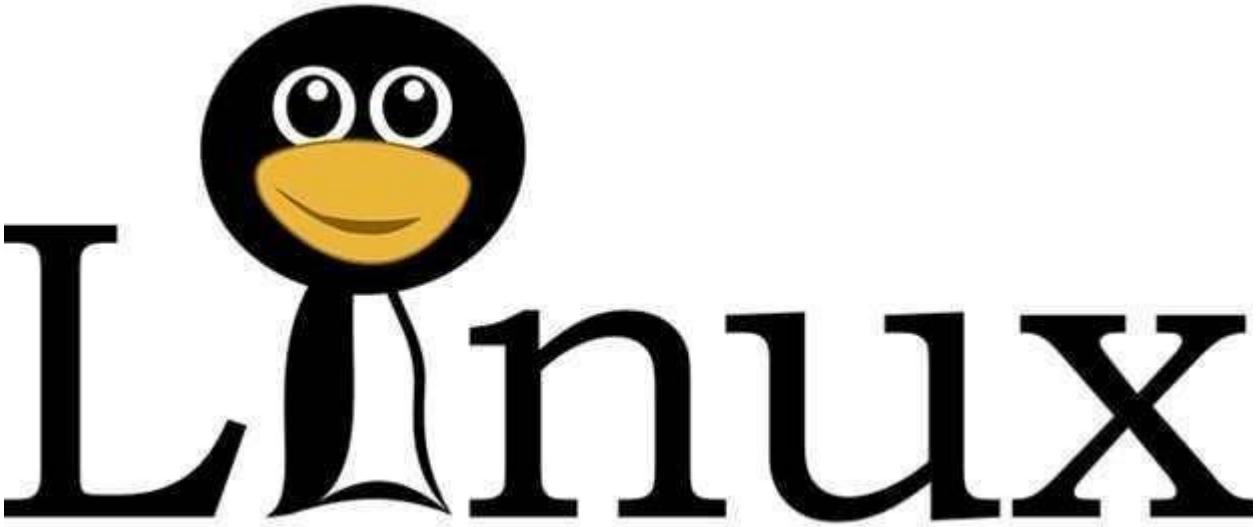

Assignment - 2

Points Covered in this Assignment-2

1. **User Administration** ○ Authentication, Authorization, and Auditing
2. **Commands to Learn** ○ useradd, passwd, userdel, usermod ○ groupadd, groupdel, groupmod ○ su and su -with examples

3. **User and Group Information** ○ User and group information files ○ Password information files
4. **Password Policies** ○ chage command and its options
5. **User Monitoring and Auditing** ○ Commands: w, last, lastb
6. **Sudo Power** ○ wheel group
7. **Default Configuration Files** ○ /etc/default/useradd ○ /etc/login.defs ○ /etc/security/limits.conf

---

1. **Create some users:**
   - Named **"alex"** with its home directory at **/home/user1** and give password **"pass1".**

     Command to create user:

     useradd -m -d /home/user1 alex

     passwd alex

```
[root@localhost ~]# useradd -m -d /home/user1 alex
[root@localhost ~]# passwd alex
Changing password for user alex.
\New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

   - Named **"brew"** with its home directory at **/mnt/user2** and give password **"pass2".**

     Command to create brew user with given home directory:

     useradd -m -d /mnt/user2 brew

     passwd brew

```
[root@localhost ~]# useradd -m -d /mnt/user2 brew
[root@localhost ~]# passwd brew
Changing password for user brew.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

- Named **“nora”** without its home directory

  Command to create user nora is **useradd nora**

```
[root@localhost ~]# useradd nora
```

- Named **“panny”** with custom UID **2112,** and assign password **“pass-4”**

  Command to create user “panny” with custom UID we use -u with the specified UID to a user Command: **sudo useradd -m -u 2112 panny**

```
[root@localhost ~]# sudo useradd -m -u 2112 panny
[root@localhost ~]# passwd panny
Changing password for user panny.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]# id panny
uid=2112(panny) gid=2112(panny) groups=2112(panny)
```

- Named **'texas'** without using the **useradd** or **adduser** commands.

  Adding user to passwd:

  echo "texas:x:1002:1002::/home/texas:/bin/bash" | sudo tee -a /etc/passwd

  Adding user to shadow:

  echo "texas:$(openssl passwd -6 pass123):0:0:99999:7:::" | sudo tee -a /etc/shadow

  Creating home directory

  sudo mkdir /home/texas

  sudo chown 1002:1002 /home/texas

  sudo chmod 700 /home/Texas

  Add Group:

  sudo groupadd -g 1002 texas

```
[root@localhost ~]# echo "texas:x:1002:1002::/home/texas:/bin/bash" | sudo tee -
a /etc/passwd
texas:x:1002:1002::/home/texas:/bin/bash
[root@localhost ~]# echo "texas:$(openssl passwd -6 pass123):0:0:99999:7:::" | s
udo tee -a /etc/shadow
texas:$6$dMwShKENG/5GckTa$nPJFf.Qx96HDJmtlIRMkuXWQ/o/Ouvig9FK5uCwJyjCsERMgDZ9IL7
LIypCvheBQ9EyQjQVCrs8a.gtP5z2xz1:0:0:99999:7:::
[root@localhost ~]# sudo mkdir /home/texas
sudo chown 1002:1002 /home/texas
sudo chmod 700 /home/texas
[root@localhost ~]# sudo groupadd -g 1002 texas
[root@localhost ~]# id texas
uid=1002(texas) gid=1002(texas) groups=1002(texas)
```

(Hint: Make changes in the 7 user configuration files)

2. Log in as user alex using the **su** and **su -** commands, and explain their differences.

Soln>

su = When you use su without a dash (-), it switches to the target user without loading the target user's environment.

i.e Switches to the user "alex" but keeps the current user's environment variables and working directory.

Command: su alex

```
[root@localhost ~]# su alex
[alex@localhost root]$
```

su - = When you use su -, it simulates a full login shell for the target user.

Switches to the user "alex" **and** loads their full login environment, as if "alex" had logged in directly.

This command starts a new login shell for "alex," updating environment variables (e.g., PATH, HOME) and changing the working directory to "alex's" home directory.

Command: su – alex

```
[alex@localhost root]$ su - alex
Password:
[alex@localhost ~]$
```

3. Set a password policy for all above users with the following requirements:
   - The maximum password age should be 30 days, and the minimum password age should be 10 days.
   - Set the password expiry date for all users to December 31, 2025.

Soln>

Password policy- A **password policy** is a set of rules designed to enhance the security of user accounts by enforcing good password practices.

Command for minimun password age 10 days and maximum password age 30 days:

chage -m 10 -M 30 <username> (alex)

here,

-m -> min pass age

-M -> max pass age

```
[root@localhost ~]# chage -m 10 -M 30 alex
```

```
[root@localhost ~]# chage -l alex
Last password change                                    : Jan 28, 2025
Password expires                                        : Feb 27, 2025
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 10
Maximum number of days between password change          : 30
Number of days of warning before password expires       : 7
```

4. **Modify the user "alex":**

- Add a comment: "I am alex"

  Command: **usermod -c "I am alex" alex**

- Change the UID to 2581

  Use the -u option to change the UID to 2581

  Command: **usermod -u 2581 alex**

- Change the shell to "nologin"
  use -s option
  Command: **usermod -s /sbin/nologin alex**

```
[root@localhost ~]# usermod -c "I am alex" alex
[root@localhost ~]# usermod -u 2581 alex
[root@localhost ~]# usermod -s /sbin/nologin alex
[root@localhost ~]# id alex
uid=2581(alex) gid=1006(alex) groups=1006(alex)
[root@localhost ~]# grep alex /etc/passwd
alex:x:2581:1006:I am alex:/home/user1:/sbin/nologin
[root@localhost ~]# grep alex /etc/passwd
alex:x:2581:1006:I am alex:/home/user1:/sbin/nologin
```

5. Create group with following configuration:
    - Named “**north**” with secondary group member “alex” & “texas”.
    - Named “**south**” with GID “2222”.

Soln>

Create north group: groupadd north

Add alex and Texas to north a secondary members:

usermod -aG north alex

usermod -aG north texas

here -aG adds members to a group without removing them from previous group.

Create the "south" group with GID 2222:

groupadd -g 2222 south

```
[root@localhost ~]# groupadd north
[root@localhost ~]# usermod -aG north alex
[root@localhost ~]# usermod -aG north texas
[root@localhost ~]# groupadd -g 2222 south
[root@localhost ~]# grep north /etc/group
north:x:2113:alex,texas
[root@localhost ~]# grep south /etc/group
south:x:2222:
```

6. Grant user **Alex** administrative privileges through the wheel group so that Alex can add Panny to the admin group without requiring root access.

Soln>

Ensure the "wheel" Group Has Sudo Privileges

sudo visudo

uncomment **%wheel ALL=(ALL) ALL**

Save and exit (:wq)

Add Alex to the "wheel" Group

usermod -aG wheel alex

Add Panny to the "admin" group

sudo usermod -aG admin panny

Now check sudo previliges to alex

su – alex

sudo date

```
[root@localhost ~]# visudo
[root@localhost ~]# usermod -aG wheel alex
[root@localhost ~]# su - alex
[alex@localhost ~]$ sudo usermod -aG admin panny
[alex@localhost ~]$ sudo date
Wednesday 29 January 2025 06:52:49 PM IST
```

7. Change the group name from “**south**” to “**dakshin**”.

SOln>

groupmod -n dakshin south

Checking changes: grep dakshin /etc/group

```
[root@localhost ~]# groupmod -n dakshin south
[root@localhost ~]# grep dakshin /etc/group
dakshin:x:2222:
```

8. Create a system user named “**ping**” and check its UID.

Soln> Command: **useradd -r ping**

-r is used to create a system user

Check the UID of the User "ping": id ping

```
[root@localhost ~]# useradd -r ping
[root@localhost ~]# id ping
uid=978(ping) gid=977(ping) groups=977(ping)
```

9. Create a group named **goa** with GID 11000. Set this group as the supplementary group for “**brew**”

Soln> Command: **groupadd -g 11000 goa**

Add brew to goa as supplementary group: **usermod -aG goa brew**

Verify group of brew: groups brew

```
[root@localhost ~]# groupadd -g 11000 goa
[root@localhost ~]# usermod -aG goa brew
[root@localhost ~]# groups brew
brew : brew goa
```

10. Create a group named **“prod”**. Then, create two users, user2 and user1, and set both the user’s primary group to **prod**.

SOln> creating group prod: **groupadd prod**

Create the Users "user1" and "user2" with "prod" as their Primary Group

useradd -g prod user1

useradd -g prod user2

verify group of users: **id user1**

id user2

```
[root@localhost ~]# groupadd prod
[root@localhost ~]# useradd -g prod user1
useradd: warning: the home directory /home/user1 already exists.
useradd: Not copying any file from skel directory into it.
[root@localhost ~]# useradd -g prod user2
[root@localhost ~]# id user1
uid=2582(user1) gid=11001(prod) groups=11001(prod)
[root@localhost ~]# id user2
uid=2583(user2) gid=11001(prod) groups=11001(prod)
```

11. Change the password policy for the USER3 and USER4 accounts to expire on 2026-01-15.

Soln> USER3 and USER4 doesn't exists so we have to create them first:

useradd USER3

useradd USER4

Use the chage Command to Set Password Expiry

sudo chage -E 2026-01-15 USER3

sudo chage -E 2026-01-15 USER4

verify password expiry date:

sudo chage -l USER3

sudo chage -l USER4

```
[root@localhost ~]# useradd USER3
[root@localhost ~]# useradd USER4
[root@localhost ~]# sudo chage -E 2026-01-15 USER3
[root@localhost ~]# sudo chage -E 2026-01-15 USER4
```

```
[root@localhost ~]# sudo chage -l USER3
Last password change                                    : Jan 29, 2025
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : Jan 15, 2026
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
[root@localhost ~]# sudo chage -l USER4
Last password change                                    : Jan 29, 2025
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : Jan 15, 2026
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

12. Configure administrative rights for all members of the **Goa** group to execute any command as any user.

Soln>

sudo visudo

%goa ALL=(ALL) ALL

Save and edit

Verify: sudo usermod -aG goa <username>

Test: sudo ls /root

```
[root@localhost ~]# visudo
[root@localhost ~]# su - brew
[brew@localhost ~]$ sudo usermod -aG goa alex

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for brew:
[brew@localhost ~]$ sudo ls /root
anaconda-ks.cfg  file10.txt   file14.txt  file19.txt  file4.txt  file9.txt  Templates
demo             file11.txt   file15.txt  file1.txt   file5.txt  --help     test
Desktop          {file1..20}  file16.txt  file20.txt  file6.txt  Music      Videos
Documents        file12.txt   file17.txt  file2.txt   file7.txt  Pictures   vishal
Downloads        file13.txt   file18.txt  file3.txt   file8.txt  Public
[brew@localhost ~]$ id alex
uid=2581(alex) gid=1006(alex) groups=1006(alex),10(wheel),2113(north),11000(goa)
```

13. How would you check all failed login attempts on the system from the last 10 days? Write the command and display the output.

Soln>

journalctl --since "10 days ago" -u sshd | grep "Failed password"

journalctl – fetches system logs

--since “10 days ago” – Filters logs from the last **10 days**.

-u sshd - filter logs for the SSH device

grep "Failed password" - Extracts lines containing **failed login attempts**.

```
[root@localhost ~]# journalctl --since "10 days ago" -u sshd | grep "Failed password"
Jan 29 19:22:11 localhost.localdomain sshd[4511]: Failed password for invalid user wronguser f
rom 127.0.0.1 port 38878 ssh2
Jan 29 19:22:17 localhost.localdomain sshd[4511]: Failed password for invalid user wronguser f
rom 127.0.0.1 port 38878 ssh2
Jan 29 19:22:20 localhost.localdomain sshd[4511]: Failed password for invalid user wronguser f
rom 127.0.0.1 port 38878 ssh2
```

14. How would you determine how many users are currently logged into the system? Write the command to achieve this.

Soln>

Command: who | wc -l

Who – Displays logged in users

wc -l – Counts the no. of lines, giving the total no. of logged-in users.

```
[root@localhost ~]# who | wc -l
2
```

15. Add the user "**sara**" to the "wheel" group and create a collaborative directory /collaborative/infodir.

Soln>

First add user sara: **useradd sara**

Add "sara" to the wheel group: **sudo usermod -aG wheel sara**

Change user to sara: **su - sara**

Create a collaborative directory:

sudo mkdir -p /collaborative/infodir

sudo chmod 2775 /collaborative/infodir

sudo chown :wheel /collaborative/infodir

chmod 2775 → Enables group collaboration.

chown :wheel → Sets wheel as the group owner.

```
[root@localhost ~]# usermod -aG wheel sara
[root@localhost ~]# su - sara
[sara@localhost ~]$ sudo mkdir -p /collaborative/infodir

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for sara:
[sara@localhost ~]$ sudo chmod 2775 /collaborative/infodir
[sara@localhost ~]$ sudo chown :wheel /collaborative/infodir
```

16. **Configure login/logout messages**:

- When you log in with a new user, display a message: "Hello, you are logged in as USER" (where USER is replaced with the logged-in username).

Soln> **echo 'echo "Hello, you are logged in as $(whoami)"' | sudo tee -a /etc/profile**

- When you log out, display: "You are logged out now".

Soln> **echo 'echo "You are logged out now"' | sudo tee -a /etc/bash.bash_logout**

```
[root@localhost ~]# echo 'echo "Hello, you are logged in as $(whoami)"' | sudo tee -a /etc/pro
file
echo "Hello, you are logged in as $(whoami)"
[root@localhost ~]# echo 'echo "You are logged out now"' | sudo tee -a /etc/bash.bash_logout
echo "You are logged out now"
[root@localhost ~]# su - alex
Hello, you are logged in as alex
Hello, you are logged in as alex
[alex@localhost ~]$ exit
logout
You are logged out now
You are logged out now
```

17. **Configure system parameters for newly created users**:

- Warning period for password expiry: 5 days
- Minimum user UID: 2000 ○ Maximum user UID: 70000

Soln>

Edit the /etc/login.defs file:

sudo nano /etc/login.defs

Set password expiry warning:

PASS_WARN_AGE 5

Set minimum and maximum UID:

UID_MIN 2000

UID_MAX 70000

```
root@localhost:~ — nano /etc/login.defs
  GNU nano 5.6.1                    /etc/login.defs
# Password aging controls:
#
#       PASS_MAX_DAYS   Maximum number of days a password may be used.
#       PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#       PASS_MIN_LEN    Minimum acceptable password length.
#       PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_WARN_AGE   5

# Currently PASS_MIN_LEN is not supported

# Currently SU_WHEEL_ONLY is not supported

# Currently CRACKLIB_DICTPATH is not supported

#
# Min/max values for automatic uid selection in useradd(8)
#
UID_MIN                  2000
UID_MAX                 70000
```

18. **Create a directory /data** and configure the system so that all newly created users get /data as their home directory by default.

Soln>

Set default home directory to /data for new users

Edit /etc/default/useradd: **nano /etc/default/useradd**

Change: **HOME=/data**

Create the /data directory:

mkdir -p /data

chmod 755 /data

```
[root@localhost ~]# nano /etc/default/useradd
[root@localhost ~]# mkdir -p /data
[root@localhost ~]# chmod 755 /data
```

19. Name a file where we can set a file size limit upto 200 MB for a single file.

Soln>

Edit /etc/security/limits.conf: **nano /etc/security/limits.conf**

Add: *** hard fsize 204800**

```
#<domain>      <type>  <item>         <value>
#
* hard fsize 204800
```

20. **Check the last three users** who logged into your system.

Soln>

Command: **last -n 3**

```
[root@localhost ~]# last -n 3
root     tty2         tty2             Wed Jan 29 18:20    gone - no logout
root     seat0        login screen     Wed Jan 29 18:20    gone - no logout
reboot   system boot  5.14.0-362.8.1.e Wed Jan 29 18:20   still running

wtmp begins Wed Jan 22 22:20:58 2025
```

21. As a system administrator, how would you configure the system to ensure that:

- Automatically create an instructions.txt file in the home directory of every new user upon account creation.
- Ensure that the mail directory for every newly created user is set to /home/spool/mail/ by default?"

Soln>

Automatically create instructions.txt in new users' home directories:

echo "Welcome to the system. Please follow instructions carefully." |tee /etc/skel/instructions.txt

Set default mail directory to /home/spool/mail/ Edit /etc/login.defs:

sudo nano /etc/login.defs

Change: **MAIL_DIR /home/spool/mail/**

```
[root@localhost ~]#     echo "Welcome to the system. Please follow instructions carefully." |t
ee /etc/skel/instructions.txt
Welcome to the system. Please follow instructions carefully.
[root@localhost ~]# nano /etc/login.defs
```

22. Delete some users ○ Named **'alex' and 'brew'** with its all data contents including mail data.

Soln>

Command:

sudo userdel -r alex

sudo userdel -r brew