Assignment 3

Vishal Menariya

TR-1

Question 1: Basic Understanding of Users in Linux

- **How many types of users exist in a Linux system? What is the UID range of it?**
- **Write a Linux command to check which users have access to the shell for executing commands.**

Soln>

Types of user:

1. Root user: Has full administrative access to the system.
   UID – 0
2. System user: These are reserved for system processes and services.

   UID – 1 to 999

3. Regular user: These are normal users created by the administrator or system users.
   UID – 1000 and above

Linux Command to Check Users with Shell Access:

cat /etc/passwd | grep -E '/bin/bash|/bin/sh' | cut -d: -f1

Question 2: An organization “Copex Pvt Ltd” has set up some users and groups for a project. Perform the following tasks step-by-step:

User and Group Creation:

- **Create the following users and set a common password “pass” for all users: Nitesh, Mohan, Nitesh, Parul, Alex, Hitesh**
- **Create the following groups for this project: prod, test**

Collaborative Directory Setup:

- **As the root administrator, create a collaborative directory named “collaborative” under “/mnt”.**
- **Write a Linux command to change the owner & group-owner of the /mnt/collaborative directory to the “root & prod” group at a same time.**

Answer the following questions:

- **Write a Linux command to check the “default permissions, owner, and group owner” of the directory.**
- **Which users in this project fall under the "others" category for this directory?**

Soln>

User Creation:

useradd Nitesh

useradd Mohan

useradd Parul

useradd Alex

useradd Hitesh

echo "pass" | passwd --stdin Nitesh

echo "pass" | passwd --stdin Mohan

echo "pass" | passwd --stdin Parul

echo "pass" | passwd --stdin Alex

echo "pass" | passwd --stdin Hitesh

Group Creation:

groupadd prod

groupadd test

Collaborative Directory Setup:

Create a directory:

mkdir /mnt/collaborative

Change Owner & Group Owner Simultaneously:

chown root:prod /mnt/collaborative

Answer the following questions:

Linux command to check the “default permissions, owner, and group owner” of the directory:

ls -ld /mnt/collaborative

users project fall under the "others" category for this directory:

Nitesh Mohan Parul Alex Hitesh

```
[root@localhost ~]# useradd Nitesh
useradd Mohan
useradd Parul
useradd Alex
useradd Hitesh
echo "pass" | passwd --stdin Nitesh
echo "pass" | passwd --stdin Mohan
echo "pass" | passwd --stdin Parul
echo "pass" | passwd --stdin Alex
echo "pass" | passwd --stdin Hitesh
Changing password for user Nitesh.
passwd: all authentication tokens updated successfully.
Changing password for user Mohan.
passwd: all authentication tokens updated successfully.
Changing password for user Parul.
passwd: all authentication tokens updated successfully.
Changing password for user Alex.
passwd: all authentication tokens updated successfully.
Changing password for user Hitesh.
passwd: all authentication tokens updated successfully.
[root@localhost ~]# groupadd prod
groupadd test
[root@localhost ~]# mkdir /mnt/collaborative
[root@localhost ~]# chown root:prod /mnt/collaborative
[root@localhost ~]# ls -ld /mnt/collaborative
drwxr-xr-x. 2 root prod 6 Feb  3 21:11 /mnt/collaborative
[root@localhost ~]# S
```

Question 3: Advanced Permission Management.

Group Membership Assignment:

1. As the root administrator, add users Mohan and Nitesh to the prod group as secondary group members

SOln>

Command to add users to prod group as secondary members:

usermod -aG prod Mohan

usermod -aG prod Nitesh

Verifying user addition:

groups Mohan

groups Nitesh

Write the Linux commands to Apply the appropriate permissions as the root administrator and concepts to achieve this.

1. Grant the prod group members permission to create and modify content in the /mnt/collaborative directory.
   - ⇨ **chmod 770 /mnt/collaborative**
     Here,
     7 – root (owner) – read(4), write(2), execute(1)
     7 – prod (group) - read(4), write(2), execute(1)
     0 – No permission
2. Restrict "others" from having no permissions in the /mnt/collaborative directory using the symbolic method.
   - ⇨ **chmod o-rwx /mnt/collaborative**
   - ⇨ **o- => remove from others**
     r-read, w-write, x-execute

3. Create some files and directories in /mnt/collaborative and ensure that any new content created in /mnt/collaborative automatically inherits the same group ownership as the parent directory.
   ⇨ **chmod g+s /mnt/collaborative**
   g+s -> adds groups special permission SGID
4. Additionally, ensure that no one can delete the files created by others, except the file's creator.
   ⇨ **chmod +t /mnt/collaborative**
   Now, only **the file owner (creator) or root** can delete their own files.

Verification Tasks:

Log in as the user “Mohan” and:

1. Verify that user “Mohan” can create content in the “/mnt/collaborative” directory or not.
   ⇨ **su – Mohan**
   touch /mnt/collaborative/testfile_Mohan.txt
   ls -l /mnt/collaborative

2. Now again what are the permissions for “Owner, Group & Other for ”/mnt/collaborative”, Describe the permission section of especially group & others.
   ⇨ **ls -ld /mnt/collaborative**

```
[root@localhost ~]# usermod -aG prod Mohan
usermod -aG prod Nitesh
[root@localhost ~]# groups Mohan
groups Nitesh
Mohan : Mohan prod
Nitesh : Nitesh prod
[root@localhost ~]# sudo chmod 770 /mnt/collaborative
[root@localhost ~]# chmod o-rwx /mnt/collaborative
[root@localhost ~]# chmod g+s /mnt/collaborative
[root@localhost ~]#  chmod +t /mnt/collaborative
[root@localhost ~]# su - Mohan
[Mohan@localhost ~]$ touch /mnt/collaborative/testfile_Mohan.txt
ls -l /mnt/collaborative
total 0
-rw-r--r--. 1 Mohan prod 0 Feb  3 21:30 testfile_Mohan.txt
[Mohan@localhost ~]$ ls -ld /mnt/collaborative
drwxrws--T. 2 root prod 32 Feb  3 21:30 /mnt/collaborative
[Mohan@localhost ~]$ chmod o+t /mnt/collaborative
chmod: changing permissions of '/mnt/collaborative': Operation not permitted
```

Question 4: Write a command to remove the SUID special permission from the file /usr/bin/passwd using the numerical method & explain the impact of this change

Soln>

Command Using the Numerical Method:

chmod 755 /usr/bin/passwd

Impact

The **SUID** bit allows **regular users** to run /usr/bin/passwd as root to change their own passwords.

After removing SUID (chmod 755):

- Regular users **CANNOT** change their passwords anymore.
- Only **root** can modify user passwords.

```
[root@localhost ~]# chmod 755 /usr/bin/passwd
[root@localhost ~]# ls -l /usr/bin/passwd
-rwxr-xr-x. 1 root root 32648 Aug 10  2021 /usr/bin/passwd
```

Question 5: Set the UMASK Value:

- Write the Linux command to check the current “umask” value for the user's shell.
- How would you change the “umask” setting so that all newly created users on the system have a default “umask” value of `0777`?

Soln>

Checking the Current umask Value for the User's Shell:

umask

change the “umask” setting so that all newly created users on the system have a default “umask” value of `0777`:

sudo nano /etc/profile

umask 0777

Modify /etc/bash.bashrc:

sudo nano /etc/bash.bashrc

umask 0777

Modify /etc/login.defs (For Newly Created Users):

sudo nano /etc/login.defs

UMASK 0777

Applying the Changes:

source /etc/profile

source ~/.bashrc

verify: **umask**

```
[root@localhost ~]# umask
0022
[root@localhost ~]# nano /etc/profile
[root@localhost ~]# nano /etc/login.defs
[root@localhost ~]# nano /etc/login.defs
[root@localhost ~]# source /etc/profile
source ~/.bashrc
source /etc/profile
source ~/.bashrc
[root@localhost ~]# umask
0777
```

Question 6: Set the default permissions for the user Parul on newly created files and directories as follows:

- **Set the default permissions for all newly created files to r--r--r--.**
- **Set the default permissions for all newly created directories to r-xr-xr-x..**

Soln>

Calculating umask:

Default file permission: 666

Desired file permission: 444(rrr)

umask = 666-444 -> 222

Default directory permissions: 777

Desired file permission: 555(rxrxrx)

umask = 777-555 -> 222

set Parul user umask – 0222

sudo nano /home/Parul/.bashrc

add this line at end: **umask 0222**

apply changes: **source /home/Parul/.bashrc**

verify setting:

touch testfile

ls -l testfile

Create a test directory:

mkdir testdir

ls -ld testdir

```
[root@localhost ~]# nano /home/Parul/.bashrc
[root@localhost ~]# source /home/Parul/.bashrc
[root@localhost ~]# touch testfile
ls -l testfile
-r--r--r--. 1 root root 0 Feb  3 22:17 testfile
[root@localhost ~]# mkdir testdir
ls -ld testdir
dr-xr-xr-x. 2 root root 6 Feb  3 22:19 testdir
```

Question 7: As a system administrator, configure the system to ensure that only the user Nitesh and the root user can modify the /etc/chrony.conf file, while all other users should have read-only access to it. Write the commands.

Soln>

Change the File Ownership:

chown Nitesh:root /etc/chrony.conf

Set the Correct Permissions:

chmod 644 /etc/chrony.conf

644(rw-r--r--)

Verify:

ls -l /etc/chrony.conf

```
[root@localhost ~]# chown Nitesh:root /etc/chrony.conf
[root@localhost ~]#  chmod 644 /etc/chrony.conf
[root@localhost ~]# ls -l /etc/chrony.conf
-rw-r--r--. 1 Nitesh root 1369 Aug 29  2022 /etc/chrony.conf
```

Question 8: User Alex needs to be granted administrative privileges equivalent to the root user to manage the system, while ensuring that all other users retain their restricted access based on their roles. Describe how you would implement this configuration. Write the commands.

Soln>

Add Alex to the wheel group:

usermod -aG wheel Alex

Verify groups of alex:

groups Alex

visudo

uncomment the line: **%wheel ALL=(ALL) ALL**

su - Alex

sudo ls /root

```
[root@localhost ~]# usermod -aG wheel Alex
[root@localhost ~]# groups Alex
Alex : Alex wheel
[root@localhost ~]# visudo
visudo: /etc/sudoers.tmp unchanged
[root@localhost ~]# su - Alex
sudo ls /root
[Alex@localhost ~]$
```

Question 9: User Hitesh, a senior team member, requires full access to the system for daily operations. However, to prevent accidental shutdowns or reboots, configure the system so that Hitesh can execute all commands except poweroff and reboot. Write the commands.

Soln>

Edit sudoers file:

visudo

Define a Custom Command Alias for Restricted Commands:

 # Command Alias for restricted commands

Cmnd_Alias RESTRICTED = /sbin/poweroff, /sbin/reboot

Deny poweroff and reboot for Hitesh:

 # Allow Hitesh to execute all commands except poweroff and reboot

Hitesh ALL=(ALL) ALL, !RESTRICTED

Check;

su – Hitesh

sudo poweroff #not work

```
[root@localhost ~]# visudo
[root@localhost ~]# su - Hitesh
[Hitesh@localhost ~]$ sudo poweroff
[sudo] password for Hitesh:
Sorry, user Hitesh is not allowed to execute '/sbin/poweroff' as root on localho
st.localdomain.
```

Question 10: To safeguard all-important and critical system directories, ensure they cannot be deleted or removed by the root user. Write the commands you would use to implement this protection. *Hint: (/ is a top-level file system directory)

Soln>

chattr +i /usr/bin /usr/sbin /root /boot /etc /var

```
[root@localhost ~]# chattr +i /usr/bin /usr/sbin /root /boot /etc /var
```