ARPIT SAGAR 4/04/2025

QUESTION 1.

Due to a security incident, you need to take immediate action to lock down certain user accounts and enforce stricter password policies.

Requirements: Lock User Accounts: Lock the accounts of users: Adam (adam), Eve (eve), and Jack (jack) to prevent them from logging in during the investigation.

Enforce Strong Password Policies: Set a minimum password length of 12 characters for all users.

Require all users to change their passwords immediately.

Account Auditing: Generate a list of all user accounts and their password status.

```
ubuntu@ip-172-31-32-198:~$ sudo useradd Adam
ubuntu@ip-172-31-32-198:~$ sudo useradd Eve
ubuntu@ip-172-31-32-198:~$ sudo Jack
sudo: Jack: command not found
ubuntu@ip-172-31-32-198:~$ sudo useradd Jack
ubuntu@ip-172-31-32-198:~$ sudo passwd -l Adam
passwd: password changed.
ubuntu@ip-172-31-32-198:~$ su Adam
Password:
su: Authentication failure
ubuntu@ip-172-31-32-198:~$ sudo passwd Adam
New password:
Retype new password:
passwd: password updated successfully
ubuntu@ip-172-31-32-198:~$ sudo passwd -l Adam
passwd: password changed.
ubuntu@ip-172-31-32-198:~$ ^C
ubuntu@ip-172-31-32-198:~$ su Adam
Password:
su: Authentication failure
ubuntu@ip-172-31-32-198:~$ su Adam
Password:
su: Authentication failure
ubuntu@ip-172-31-32-198:~$ sudo passwd -l Eve
passwd: password changed.
ubuntu@ip-172-31-32-198:~$ sudo passwd -l Jack
passwd: password changed.
ubuntu@ip-172-31-32-198:~$ su Jack
Password:
su: Authentication failure
ubuntu@ip-172-31-32-198:~$
```

created 3 users as Adam, Eve and Jack.
Locked their acc using “sudo passwd -l Adam”. (-l -> locked)
Verified using switching to locked acc, eg when switching to Adam, it shows Authentication Failure.

Now,

```
ubuntu@ip-172-31-32-198:~$ sudo nano /etc/pam.d/common-password
```

This opens a common-password file which contains all the password policies

```
# here are the per-package modules (the "Primary" block)
password        requisite                       pam_pwquality.so retry=3 minlen=12
password        [success=1 default=ignore]      pam_unix.so obscure use_authtok try_first_pass yescrypt
# here's the fallback if no module succeeds
password        requisite                       pam_deny.so
```

add minlen=12 meaning minimum length of password should be at least 12.

```
ubuntu@ip-172-31-32-198:~$ sudo useradd Adam
ubuntu@ip-172-31-32-198:~$ sudo useradd Eve
ubuntu@ip-172-31-32-198:~$ sudo useradd Jack
ubuntu@ip-172-31-32-198:~$ sud passwd Adma
Command 'sud' not found, but there are 15 similar ones.
ubuntu@ip-172-31-32-198:~$ sudo passwd Adam
New password:
BAD PASSWORD: The password is shorter than 12 characters
```

now when changing password for Adam in less than 12 chars, it's showing alert message as password is shorter than the required len(12).

```
ubuntu@ip-172-31-32-198:~$ sudo chage -d 0 Jack
ubuntu@ip-172-31-32-198:~$ su Jack
Password:
You are required to change your password immediately (administrator enforced).
Changing password for Jack.
Current password:
New password:
Retype new password:
$
```

Change password immediately on next login for user Jack:
using "sudo chage -d 0 Jack"
where "chage" is used for changing password expiry info and "-d 0" is used to set the last password change date to 0 (that is very long back (01/01/1970)) and forces to change it on next login.

```
ubuntu@ip-172-31-32-198:~$ sudo passwd -S Adam
Adam L 1970-01-01 0 99999 7 -1
ubuntu@ip-172-31-32-198:~$ sudo passwd -S Eve
Eve L 2025-04-01 0 99999 7 -1
ubuntu@ip-172-31-32-198:~$ sudo passwd -S Jack
Jack L 2025-04-01 0 99999 7 -1
ubuntu@ip-172-31-32-198:~$
```

Adam, Eve and Jack acc and password status (-S flag)
L means it is locked (we have done that earlier and date shows the last password change date)

QUESTION 2.

Scenario: You are the system administrator for a medium-sized company that uses a Linux-based server for its internal operations. Your company has recently undergone a reorganization, and there is a need to update the user groups to reflect the new structure.

The following changes are required:

1. Create New Groups: • A new department called "Research" has been formed. You need to create a new group named research. • Another new department called "Development" has also been established. Create a new group named development.

```
ubuntu@ip-172-31-32-198:~$ sudo groupadd research
ubuntu@ip-172-31-32-198:~$ sudo groupadd development
```

created two groups research and development using sudo (admininstrator privileges)

2. Modify Existing Groups: • The existing group engineering needs to be renamed to tech. • The existing group admin needs its group ID changed from 1001 to 2001.

```
ubuntu@ip-172-31-32-198:~$ sudo groupadd engineering
ubuntu@ip-172-31-32-198:~$ sudo groupmod -n tech engineering
```

created a group "engineering" and using "groupmod -n" renamed it to "tech".

```
ubuntu@ip-172-31-32-198:~$ getent group tech
tech:x:1115:
```

Verified using “getent”, showing name of the group, (x: means grp password is stored in /etc/shadow) and GID.

```
ubuntu@ip-172-31-32-198:~$ sudo groupmod -g 2001 tech
ubuntu@ip-172-31-32-198:~$ getent group tech
tech:x:2001:
```

Here I’ve changed the GID of group tech (for learning functionality) and verified it also.

3. Add Users to Groups: • A new employee, Alice, is joining the Research department. Create a user account for Alice and add her to the research group. • Another new employee, Bob, is joining the Development department. Create a user account for Bob and add him to the development group. • Charlie, who is already a part of the engineering group, should now be part of the newly named tech group. • Dave, an existing member of the admin group, should remain in the group after the group ID change. Requirements: 1. Create

the new groups research and development. 2. Rename the engineering group to tech. 3. Change the group ID of admin to 2001.

```
ubuntu@ip-172-31-32-198:~$ sudo useradd Alice
```

```
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG research Alice
ubuntu@ip-172-31-32-198:~$ sudo useradd Bob
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG development Bob
```

Now, I’ve created two user “Alice” and “Bob”, and add both of them in “research” and “development” group respectively using “usermod -aG (append user + group_name)”

```
ubuntu@ip-172-31-32-198:~$ sudo usermod -g tech Charlie
```

```
ubuntu@ip-172-31-32-198:~$ groups Charlie
Charlie : tech
```

Changed the primary group of user “Charlie” to tech using “usermod -g (primary grp)”
and verified using “groups” cmd.

```
ubuntu@ip-172-31-32-198:~$ sudo useradd Dave
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG tech Dave
ubuntu@ip-172-31-32-198:~$ sudo groupmod -g 1221 tech
ubuntu@ip-172-31-32-198:~$ id Dave
uid=1022(Dave) gid=1022(Dave) groups=1022(Dave),1221(tech)
ubuntu@ip-172-31-32-198:~$ sudo groupmod -g 1233 tech
ubuntu@ip-172-31-32-198:~$ id Dave
uid=1022(Dave) gid=1022(Dave) groups=1022(Dave),1233(tech)
```

Dave still remains in the group even after the change of group id he belongs to (tech here for eg).
Created a user “Dave”, append him to the group tech using (usermod -aG) and checked the GID of group he belongs to (1221).
Now, changed the group id of tech using (groupmod -g 1233) and check again whether “Dave” still belongs to same group with updated group id of “tech”. And yes, he belongs to the updated GID (1233-tech).

Question 3.

You are a system administrator managing a shared directory /projects on a Linux server used by different teams in your organization. The directory contains subdirectories for different projects, and each project directory needs specific access permissions for different users and groups.

Requirements:

1. **Project Managers (group proj_managers) should have read, write, and execute permissions on all project directories.**
2. **Developers (group developers) should have read and execute permissions on all project directories, but they should not be able to delete or modify any files.**
3. **QA Engineers (group qa_engineers) should have read-only access to the project_alpha directory but no access to other project directories.**

Tasks:

Example Subdirectories in /projects:

/projects/project_alpha

/projects/project_beta

```
ubuntu@ip-172-31-32-198:~$ sudo groupadd developers
ubuntu@ip-172-31-32-198:~$ sudo groupadd qa_engineers
ubuntu@ip-172-31-32-198:~$ sudo groupadd proj_managers
```

Created 3 groups names “developers”, “qa_engineers” and “proj_managers”

```
ubuntu@ip-172-31-32-198:~$ sudo useradd man1
ubuntu@ip-172-31-32-198:~$ sudo useradd dev1
ubuntu@ip-172-31-32-198:~$ sudo useradd dev2
ubuntu@ip-172-31-32-198:~$ sudo useradd dev3
ubuntu@ip-172-31-32-198:~$ sudo useradd qa1
ubuntu@ip-172-31-32-198:~$ sudo useradd qa2
```

Created some sample users for all groups.

```
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG proj_managers man1
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG developers dev1
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG developers dev2
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG developers dev3
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG qa_engineers qa1
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG qa_engineers qa2
```

Appended all the created users to their respective groups.

```
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG developers Alice
ubuntu@ip-172-31-32-198:~$ id Alice
uid=1019(Alice) gid=1019(Alice) groups=1019(Alice),1113(research),1235(develo
pers)
```

```
ubuntu@ip-172-31-32-198:~$ groups Alice
Alice : Alice research developers
```

Created a user “Alice” and appended it to the group developers and checked whether it is grouped to developers using “id Alice” or “groups Alice”.

```
ubuntu@ip-172-31-32-198:~$ sudo mkdir projects
```

```
ubuntu@ip-172-31-32-198:~/projects$ sudo mkdir project_alpha
ubuntu@ip-172-31-32-198:~/projects$ sudo mkdir project_beta
```

Created “projects” directory, under which there are two sub-directories names “project_alpha” and “project_beta”

```
ubuntu@ip-172-31-32-198:~$ sudo chmod 770 projects
ubuntu@ip-172-31-32-198:~$ ls -ld projects
drwxrwx--- 4 root proj_managers 4096 Apr  3 06:39 projects
ubuntu@ip-172-31-32-198:~$ sudo chown -R root:proj_managers projects
```

Now I’ve set the permissions for “root” and “proj_managers” as both have access to read, write and execute but others(devlopers and qa_engineers) don’t so set “770” and verified the same using “ls -ld” and changed ownership to “root” as “owner” and “proj_mangers” as “group”

```
ubuntu@ip-172-31-32-198:~$ sudo setfacl -m g:developers:rx projects
ubuntu@ip-172-31-32-198:~$ sudo setfacl -m g:qa_engineers:r projects/project_
alpha
ubuntu@ip-172-31-32-198:~$ getfacl projects
# file: projects
# owner: root
# group: proj_managers
user::rwx
group::rwx
group:developers:r-x
mask::rwx
other::---
```

Set the required “read and execute permission” for developers and “read only” for qa_engineers using “setfacl -m(modify) g(group): group_name:<permissions(rwx)> file_name”.

```
ubuntu@ip-172-31-32-198:~$ sudo setfacl -m g:qa_engineers:--- projects/
```

```
ubuntu@ip-172-31-32-198:~$ ls -ld projects/*
ls: cannot access 'projects/*': Permission denied
```

Set setfacl for “qa_engineers:--- to” restrict them accessing the other project files.

4. **User alice (a senior developer) should have read, write, and execute permissions on the project_beta directory only.**

```
ubuntu@ip-172-31-32-198:~$ sudo setfacl -m u:Alice:rwx projects/project_beta
ubuntu@ip-172-31-32-198:~$ sudo getfacl projects/project_beta
# file: projects/project_beta
# owner: root
# group: proj_managers
user::rwx
user:Alice:rwx
group::r-x
mask::rwx
other::r-x
```

Gave “Alice” permissions to read, write and execute the project_beta files. And verified the same using “getfacl”

5. **Ensure that default ACLs are set so that any new files or subdirectories created within /projects inherit the correct permissions.**

```
ubuntu@ip-172-31-32-198:~$ sudo setfacl -d -m g:proj_managers:rwx projects
ubuntu@ip-172-31-32-198:~$ sudo setfacl -d -m g:developers:rx projects
ubuntu@ip-172-31-32-198:~$ sudo setfacl -d -m u:Alice:rwx projects/project_be
ta
```

To automatically apply these ACL rules to any new files did that using -d (default) flag

```
ubuntu@ip-172-31-32-198:~$ sudo getfacl projects/project_alpha
# file: projects/project_alpha
# owner: root
# group: proj_managers
user::rwx
group::r-x
group:qa_engineers:r--
mask::r-x
other::r-x

ubuntu@ip-172-31-32-198:~$ sudo getfacl projects/project_beta
# file: projects/project_beta
# owner: root
# group: proj_managers
user::rwx
user:Alice:rwx
group::r-x
mask::rwx
other::r-x
default:user::rwx
default:user:Alice:rwx
default:group::r-x
default:mask::rwx
default:other::r-x
```

Verified the default ACL rules and all working set as expected.

QUESTION 4.

Your company has a new project starting, and a temporary project team needs to be set up on the server. This involves creating user accounts, modifying permissions, and ensuring account security.
Requirements:

(a.) Create New User Accounts: Create user accounts for new team members: Alice (username alice), Bob (username bob), and Charlie (username charlie), and set initial passwords for each.

```
ubuntu@ip-172-31-32-198:~$ sudo useradd alice
ubuntu@ip-172-31-32-198:~$ sudo useradd bob
ubuntu@ip-172-31-32-198:~$ sudo useradd charlie
ubuntu@ip-172-31-32-198:~$ sudo passwd alice
New password:
Retype new password:
passwd: password updated successfully
ubuntu@ip-172-31-32-198:~$ sudo passwd bob
New password:
Retype new password:
passwd: password updated successfully
ubuntu@ip-172-31-32-198:~$ sudo passwd charlie
New password:
Retype new password:
passwd: password updated successfully
```

Created 3 users “alice”, “bob”, and “charlie” and set passwords for all of them.

(b.) Set the shell for all new users to /bin/bash.

```
ubuntu@ip-172-31-32-198:~$ getent passwd alice
alice:x:1030:1030::/home/alice:/bin/sh
```

First checked the current shell for “alice” and it is /bin/sh.
We need to change that. Used “usermod -s (new shell)” and set to /bin/bash

```
ubuntu@ip-172-31-32-198:~$ sudo usermod -s /bin/bash alice
ubuntu@ip-172-31-32-198:~$ getent passwd alice
alice:x:1030:1030::/home/alice:/bin/bash
ubuntu@ip-172-31-32-198:~$ sudo usermod -s /bin/bash bob
ubuntu@ip-172-31-32-198:~$ sudo usermod -s /bin/bash charlie
ubuntu@ip-172-31-32-198:~$ getent passwd bob
bob:x:1031:1031::/home/bob:/bin/bash
ubuntu@ip-172-31-32-198:~$ getent passwd charlie
charlie:x:1032:1032::/home/charlie:/bin/bash
```

Did the same for bob and charlie and checked the shell using “getent passwd alice/bob/charlie” and got the output “/bin/bash” as expected.

(c.) Modify User Permissions: Alice needs to be added to the developers group. Bob and Charlie need to be added to the testers group.

```
charlie:x:1032:1032::/home/charlie:/bin/bash
ubuntu@ip-172-31-32-198:~$ sudo groupadd developers
ubuntu@ip-172-31-32-198:~$ sudo groupadd testers
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG developers alice
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG testers bob
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG testers charlie
ubuntu@ip-172-31-32-198:~$ groups alice
alice : alice developers
ubuntu@ip-172-31-32-198:~$ groups bob
bob : bob testers
ubuntu@ip-172-31-32-198:~$ groups charlie
charlie : charlie testers
```

Created two groups “developers” and “testers”.

Added “alice” in “developers” group and “bob” and “charlie” to the testers group.
Checked whether all the users are assigned to correct groups using groups “alice/bob/charlie” and got expected results.

(d.) Password Policies: Set an expiry date for all user passwords to ensure they are changed in 30 days

```
ubuntu@ip-172-31-32-198:~$ sudo chage -M 30 alice
ubuntu@ip-172-31-32-198:~$ sudo chage -l alice
Last password change                                    : Apr 03, 2025
Password expires                                        : May 03, 2025
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 30
Number of days of warning before password expires       : 7
ubuntu@ip-172-31-32-198:~$ sudo chage -M 30 bob
ubuntu@ip-172-31-32-198:~$ sudo chage -l bob
Last password change                                    : Apr 03, 2025
Password expires                                        : May 03, 2025
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 30
Number of days of warning before password expires       : 7
ubuntu@ip-172-31-32-198:~$ sudo chage -M 30 charlie
ubuntu@ip-172-31-32-198:~$ sudo chage -l charlie
Last password change                                    : Apr 03, 2025
Password expires                                        : May 03, 2025
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 30
Number of days of warning before password expires       : 7
```

changed the password expiration date using “chage -M 30 alice” (password for alice will be expired after every 30 days) and checked using “sudo chage -l alice” which shows “max no. of days between password change: 30”. That’s what is needed.
Similarly, did the same for “bob” and “charlie” and verified the same.

Question 5.

You have recently been hired as a System Administrator at a mid-sized company. The company is restructuring its IT department, and you have been tasked with managing user accounts and groups on one of the company's Linux servers. Your tasks are as follows:

• Create a New User: A new employee, Alice Johnson, has joined the IT department as a Network Engineer. Create a user account for Alice with the username alicej.

```
ubuntu@ip-172-31-32-198:~$ sudo useradd alicej
```

Created a user “alicej”

• Ensure Alice’s home directory is located at /home/alicej, and set the default shell to /bin/bash.

```
ubuntu@ip-172-31-32-198:~$ getent passwd alicej
alicej:x:1033:1033::/home/alicej:/bin/sh
```

Checking the current home-dir for alicej, and it is as required /home/alicej

```
ubuntu@ip-172-31-32-198:~$ sudo usermod -d /home/alicej -m alicej
usermod: no changes
```

if need to change the home dir, we can use “usermod -d (home-dir) path -m (modify) alicej”
but as home dir already set as required, it shows no changes.

• Create a Group: The IT department has a special group for network engineers called neteng.

• Create this group on the system.

• Add the User to the Group:

• Add Alice to the neteng group.

• **Ensure that she is also part of the users group, which grants basic permissions.**

```
ubuntu@ip-172-31-32-198:~$ sudo groupadd neteng
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG neteng alicej
ubuntu@ip-172-31-32-198:~$ groups alicej
alicej : alicej neteng
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG users alicej
ubuntu@ip-172-31-32-198:~$ groups alicej
alicej : alicej users neteng
```

Created a group for network engineers called “neteng”.
Added “alicej” in that group using “usermod -aG” (append to group)
Similarly, added “alicej” in the “users” group.
Checked and verified to which groups user “alicej” belongs to. (users, neteng as expected).

• **Modify User Account:**

• **After a security review, it's been decided that all Network Engineers must use /bin/zsh as their default shell.**

• **Modify Alice’s account to use /bin/zsh as the default shell.**

```
ubuntu@ip-172-31-32-198:~$ sudo usermod -s /bin/zsh alicej
usermod: Warning: missing or non-executable shell '/bin/zsh'
ubuntu@ip-172-31-32-198:~$ ^C
ubuntu@ip-172-31-32-198:~$ sudo apt update && sudo apt install -y zsh
```

To change the default shell using “usermod -s”, we got warning that zsh should be installed first.
So, I have installed it using “sudo apt update && sudo apt install -y zsh”

```
ubuntu@ip-172-31-32-198:~$ sudo usermod -s /bin/zsh alicej
```

Now, changed the default shell using “usermod -s (new shell) /bin/zsh alicej”.

```
ubuntu@ip-172-31-32-198:~$ getent passwd alicej
alicej:x:1033:1033::/home/alicej:/bin/zsh
```

Checked whether shell is updated using “getent passwd alicej” and it is updated.

• **Delete a User Account:**

• **Another employee, Bob Smith, has left the company. His username was bobsmith.**

• **Delete Bob’s user account along with his home directory and any associated files.**

```
ubuntu@ip-172-31-32-198:~$ sudo useradd bobsmith
ubuntu@ip-172-31-32-198:~$ getent passwd bobsmith
bobsmith:x:1034:1034::/home/bobsmith:/bin/sh
ubuntu@ip-172-31-32-198:~$ sudo userdel -r bobsmith
userdel: bobsmith mail spool (/var/mail/bobsmith) not found
userdel: bobsmith home directory (/home/bobsmith) not found
```

Created a user “bobsmith”.
Deleted his acc using “userdel -r (recurring) bobsmith” which del the “bobsmith” acc and home-dir and all the files associated with it.
Without “-r” home-dir and associated files will not be deleted and can only be deleted manually.

• **Additional Group Requirement:**

• **There is another group, admins, that needs to be created for users with administrative privileges.**
• **Create the admins group and add yourself (yourusername) to it.**

```
ubuntu@ip-172-31-32-198:~$ sudo groupadd admins
```
Created a group “admins”.

```
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG admins ubuntu
ubuntu@ip-172-31-32-198:~$ groups ubuntu
ubuntu : ubuntu adm cdrom sudo dip lxd admins
```
Appended the user “ubuntu” (because I’m using it as “ubuntu” name) using “usermod -aG (append-to-grp) admins ubuntu” and checked the status using “groups ubuntu” and got desired results.

Question: 6

File Permissions and User Management for a Development Server Scenario: You are managing a Linux development server used by multiple developers with different access needs.

• Password Reset for a Developer:

• A developer, john_r, has forgotten his password. Reset his password.

```
ubuntu@ip-172-31-32-198:~$ sudo useradd john_r
ubuntu@ip-172-31-32-198:~$ sudo passwd john_r
New password:
Retype new password:
passwd: password updated successfully
```
Created a user named “john_r” and set his password.

• Set a temporary password and require him to change it upon his next login.

```
ubuntu@ip-172-31-32-198:~$ sudo chage -d 0 john_r
```
To change password upon next login, I’ve used “chage -d 0 john_r” where “d” is last password change date and “0” means last password date is set to Unix Epoch Time which is very long ago.

```
ubuntu@ip-172-31-32-198:~$ su john_r
Password:
You are required to change your password immediately (administrator enforced)
.
Changing password for john_r.
Current password:
New password:
Retype new password:
```
Checked it using switching to user “john_r” and as soon as I switched to it, it asked for immediate password change.

• Switch User for Testing:

• John needs to test a configuration under the testuser account without logging out of his current session.

• Show John how to use the su command to switch to testuser and run a test, then return to his own account.

```
ubuntu@ip-172-31-32-198:~$ su john_r
Password:
$ su testuser
Password:
$ whoami
testuser
$ exit
$ whoami
john_r
$ exit
ubuntu@ip-172-31-32-198:~$ whoami
ubuntu
```

Created a user “testuser” which will run and test basic commands.
Now first, I switched to “john_r” using “su” and then from there again switched to “testuser”.
Under “testuser” I run a cmd “whoami” which showed result as “testuser” that means I’m running and testing cmds under “testuser” without coming out of “john_r” acc.
Then simple used “exit” to come out of “testuser” acc and checked it with “whoami” and it showed “john_r” and then “exit” again to come out of the “john_r” acc as well.

- **Grant sudo Privileges for Software Installation:**
- **John needs to install development tools but should not have full administrative rights.**
- **Add John to the sudo group with permissions limited to installing software packages using apt-get.**
- **Provide an example command for John to install the git package using sudo.**

```
ubuntu@ip-172-31-32-198:~$ su john_r
Password:
$ sudo apt-get install git
[sudo] password for john_r:
john_r is not in the sudoers file.
$ exit
```

Initially “john_r” is unable to install the git or we can say that he is restricted to install anything as sudoers file have not given him rights to do so.
Hence, first we have to give him installation rights through sudoers file.

```
ubuntu@ip-172-31-32-198:~$ sudo visudo
ubuntu@ip-172-31-32-198:~$ su john_r
Password:
$ sudo apt-get install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.2).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 28 not upgraded.
$
```

```
# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
john_r ALL=(ALL) NOPASSWD: /usr/bin/apt-get
```

(Above is the updated sudoers file and all line is appended to all “john_r” to use “apt-get”)
After that, when we switched to “john_r” and then try to install “git” we can able to do so as shown in the previous picture to above.

• **Set Directory Permissions for Shared Projects:**

• **John is working on a shared project in the /projects/shared/ directory.**

```
ubuntu@ip-172-31-32-198:~$ sudo mkdir -p projects/shared/
```

```
├── projects
│   ├── project_alpha
│   ├── project_beta
│   └── shared
```

Created a new directory “shared” having parent directory as “projects” (-p makes parent directory if not present or appends if already exists)

• **Use chmod to set the directory permissions so that John and his group (devteam) can read, write, and execute files, but no other users can access the directory.**

```
ubuntu@ip-172-31-32-198:~$ sudo chown :devteam projects/shared/
ubuntu@ip-172-31-32-198:~$ sudo chmod 770 projects/shared/
```

Here, I made “devteam” as the owner of “projects/shared” and allowed only John and devteam members to read, write, and execute files, and restricting access for others.

• **Change Ownership for File Maintenance:**

• **John needs to take ownership of some files within the /projects/shared/ directory that were created by another user.**

• **Use chown to change the ownership of these files to John, ensuring that he has full control over them.**

```
ubuntu@ip-172-31-32-198:~$ sudo touch projects/shared/file1.txt projects/shar
ed/file2.txt
ubuntu@ip-172-31-32-198:~$ sudo chown john_r:devteam projects/shared/file1.tx
t
ubuntu@ip-172-31-32-198:~$ sudo ls -ld projects/shared/
drwxrwx---+ 2 root devteam 4096 Apr  3 11:12 projects/shared/
ubuntu@ip-172-31-32-198:~$ sudo ls -ld projects/shared/file1.txt
-rw-rw----+ 1 john_r devteam 0 Apr  3 11:12 projects/shared/file1.txt
```

To change ownership of some files, we need to create some files.
Here, I’ve created file1.txt and file2.txt as sample files.
After that using “chown” gave full ownership to “john_r” for file1.txt (projects/shared)
and verified the same using “ls -ld file_path” and it showed the owner as “john_r”.

• **Revoke Temporary sudo Access:**

• **After the tools are installed, remove John’s sudo access to maintain security.**

• **Document the process to verify that his sudo privileges have been removed.**

```
# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d


Save modified buffer?
 Y Yes
 N No            ^C Cancel
```

To revoke temporary sudo access, opened the sudoers file “sudo visudo” and deleted the privilege given to “john_r” line and saved the file.

```
ubuntu@ip-172-31-32-198:~$ sudo deluser john_r sudo
```

To delete the “john_r” and also remove him from the “sudo” group as well.

```
ubuntu@ip-172-31-32-198:~$ sudo -l -U john_r
sudo: unknown user john_r
```

To list(-l) the sudo permissions for user(-U) “john_r” the above cmd is run.
It shows no “apt-get” privileges hence deletion is successful.

QUESTION 7.

You are managing a Linux server in a healthcare environment where data sensitivity is crucial. • Enforce Password Policies:

• The security policy requires all users to have passwords that expire every 60 days. Set this policy for the user dr_smith using the passwd command.

```
ubuntu@ip-172-31-32-198:~$ sudo useradd dr_smith
ubuntu@ip-172-31-32-198:~$ sudo passwd dr_smith
New password:
Retype new password:
passwd: password updated successfully
ubuntu@ip-172-31-32-198:~$ sudo passwd -x 60 dr_smith
passwd: password changed.
ubuntu@ip-172-31-32-198:~$ sudo chage -l dr_smith
Last password change                                    : Apr 03, 2025
Password expires                                        : Jun 02, 2025
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 60
Number of days of warning before password expires       : 7
```

First, created a user as “dr_smith” and set his password.
Now using “sudo passwd -x 60 dr_smith” (-x: maximum password age to 60 days.)
(we can also use: “sudo chage -M 60 dr_smith”)
Also verified the update using “sudo chage -l dr_chage”.

• Ensure that Dr. Smith is prompted to change the password the next time he logs in.

```
ubuntu@ip-172-31-32-198:~$ sudo chage -d 0 dr_smith
ubuntu@ip-172-31-32-198:~$ su dr_smith
Password:
You are required to change your password immediately (administrator enforced
.
Changing password for dr_smith.
Current password:
```

Using “sudo chage -d 0 dr_smith” changes the password change to next login time.
(-d 0: password expiration date changes to 0)

• Use of su for Secure Access:

• Dr. Smith needs to access another user’s account, nurse_jane, to review patient data. However, it is critical to ensure that this is done securely and logged.

• Guide Dr. Smith on how to use su to switch to Nurse Jane’s account and emphasize the importance of logging out afterward.

```
ubuntu@ip-172-31-32-198:~$ sudo useradd nurse_jane
ubuntu@ip-172-31-32-198:~$ sudo passwd nurse_jane
New password:
Retype new password:
passwd: password updated successfully
ubuntu@ip-172-31-32-198:~$ su dr_smith
Password:
$ su nurse_jane
Password:
$ whoami
nurse_jane
$ exit
$ whoami
dr_smith
$ exit
ubuntu@ip-172-31-32-198:~$
```

Created a user “nurse_jane” and set a password.
I switched to user “dr_smith” and then switched to user “nurse_jane”.

- **Granting Administrative Rights with sudo:**

- **The IT department needs to perform system maintenance, but you want to ensure that Dr. Smith can only perform specific administrative tasks, such as restarting a service.**

- **Add Dr. Smith to the sudo group with permissions limited to restarting the apache2 service.**

```
ubuntu@ip-172-31-32-198:~$ sudo visudo
```

opened sudoers file.

```
# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d

dr_smith ALL=(ALL) NOPASSWD: /bin/systemctl restart apache2
```

added the privilege required to given to “dr_smith”, saved it.

```
ubuntu@ip-172-31-32-198:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Before using the apache2 service, we need to install it.
So, installed it using “sudo apt install apache2”

- **Provide an example command Dr. Smith would use to restart the service with sudo.**

```
ubuntu@ip-172-31-32-198:~$ su dr_smith
Password:
$ sudo systemctl restart apache2
$
```

Now “dr.smith” is successfully able to restart apache2.

```
$ sudo -l -U dr_smith
Matching Defaults entries for dr_smith on ip-172-31-32-198:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
/bin\:/snap/bin,
    use_pty

User dr_smith may run the following commands on ip-172-31-32-198:
    (ALL) NOPASSWD: /bin/systemctl restart apache2
$
```

Checked assigned privileges using “sudo -l (list priviliges) -U(users) dr_smith” and got expected output.

• Setting Permissions on Sensitive Files:

• Dr. Smith has created a directory for storing patient data, located at /secure/patients/.

```
ubuntu@ip-172-31-32-198:~$ sudo mkdir -p /secure/patients
```

Created a patients dir whose parent dir is secure using “mkdir -p”, even if the parent dir doesn’t exist, a new parent dir is created.

• Use chmod to ensure that only Dr. Smith can access this directory and its files, with no read, write, or execute permissions for anyone else.

```
ubuntu@ip-172-31-32-198:~$ sudo chmod 700 secure/patients
ubuntu@ip-172-31-32-198:~$ ls -ld secure/patients
drwx------ 2 root root 4096 Apr  3 16:11 secure/patients
```

set chmod for “dr_smith” as 700 (because only he can read, write and execute it), and verified using “ls -ld” cmd.

• Change Ownership for Secure Collaboration:

• The patient data needs to be shared with Nurse Jane, but no one else should have access.

• Use chown to change the group ownership of the /secure/patients/ directory to nurses, allowing only members of the nurses group to access it.

```
ubuntu@ip-172-31-32-198:~$ sudo groupadd nurses
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG nurses dr_smith
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG nurses nurse_jane
ubuntu@ip-172-31-32-198:~$ sudo chown :nurses secure/patients
ubuntu@ip-172-31-32-198:~$ ls -ld secure/patients
drwx------ 2 root nurses 4096 Apr  3 16:11 secure/patients
ubuntu@ip-172-31-32-198:~$ sudo chmod 770 secure/patients
ubuntu@ip-172-31-32-198:~$ ls -ld secure/patients
drwxrwx--- 2 root nurses 4096 Apr  3 16:11 secure/patients
ubuntu@ip-172-31-32-198:~$
```

Created a group “nurses” and add both “dr_smith” and “nurse_jane” to it using “usermod -aG (append to Group)”
Changed group ownership of the patient dir using “chown: nurses secure/patients”.
Checked it using “ls -ld” and group is changed to “nurses”.
Later using “chmod 770” gave read, write and execute access to “nurses” grp etc.

• Audit and Remove Unnecessary Privileges:

• After maintenance is complete, review and remove Dr. Smith's sudo privileges, ensuring no unnecessary access remains.

• Document how to check for any remaining sudo permissions and confirm their removal.

```
ubuntu@ip-172-31-32-198:~$ sudo visudo
ubuntu@ip-172-31-32-198:~$ sudo deluser dr_smith sudo
fatal: The user `dr_smith' is not a member of group `sudo'.
ubuntu@ip-172-31-32-198:~$ sudo -l -U dr_smith
User dr_smith is not allowed to run sudo on ip-172-31-32-198.
ubuntu@ip-172-31-32-198:~$
```

To remove special privileges, opened sudoers file using "sudo visudo" and deleted the extra permissions given to "dr_smith". (commenting it out can also work).
Now deleted the user "dr_smith" and all its sudo permissions.
Also verified the sudo removal for "dr_smith" using "sudo -l (list) -U (user) dr_smith".

Question. 8

Scenario: A company is working on two major projects, Project Alpha and Project Beta. Specific users need access to these projects, and security is critical.

• Create Project Groups: • Create two groups: alpha and beta for Project Alpha and Project Beta.

• Create User Accounts for Project Members:

• david_a and lisa_b are working on Project Alpha. Create their user accounts with usernames davida and lisab, respectively.

• nina_c and tom_d are working on Project Beta. Create their user accounts with usernames ninac and tomd.

• Assign each user to the appropriate project group (davida and lisab to alpha, ninac and tomd to beta).

```
ubuntu@ip-172-31-32-198:~$ sudo groupadd alpha
ubuntu@ip-172-31-32-198:~$ sudo groupadd beta
ubuntu@ip-172-31-32-198:~$ sudo useradd davida
ubuntu@ip-172-31-32-198:~$ sudo useradd lisab
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG alpha davida
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG alpha lisab
ubuntu@ip-172-31-32-198:~$ sudo useradd ninac
ubuntu@ip-172-31-32-198:~$ sudo useradd tomd
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG beta ninac
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG beta tomd
```

Created two groups "alpha" and "beta".
Created two "davida" and "lisab" and added them to "alpha" group and two more users "ninac" and "tomd" and added them to "beta" group.

```
ubuntu@ip-172-31-32-198:~$ getent group alpha
alpha:x:1245:davida,lisab
ubuntu@ip-172-31-32-198:~$ getent group beta
beta:x:1246:ninac,tomd
```

Verified the same using "getent group" cmd and all assigning is done correctly.

• **Cross-Project Access:**

• **David needs temporary access to Project Beta as well. Add him to the beta group.**

```
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG beta davida
ubuntu@ip-172-31-32-198:~$ groups davida
davida : davida alpha beta
```

Added “davida” to beta group as well.
and verified assigning using “groups” cmd. “davida” now belongs to both “alpha” and “beta” groups.

• **Security Update:**

• **Due to security policies, the default shell for all alpha project users must be changed to /bin/zsh.**

• **Apply this change to all users in the alpha group.**

```
ubuntu@ip-172-31-32-198:~$ sudo usermod -s /bin/zsh lisab
ubuntu@ip-172-31-32-198:~$ getent passwd lisab
lisab:x:1039:1039::/home/lisab:/bin/zsh
ubuntu@ip-172-31-32-198:~$ sudo usermod -s /bin/zsh davida
usermod: no changes
ubuntu@ip-172-31-32-198:~$ getent passwd davida
davida:x:1038:1038::/home/davida:/bin/zsh
ubuntu@ip-172-31-32-198:~$
```

Updated the shell for both users “davida” and “lisab” using “usermod -s (new-shell) /bin/zsh davida/lisab”.
Verified it using “getent passwd davida/lisab”.

• **Account Removal:**

• **Tom has completed his work on Project Beta and left the team. Remove his user account and all associated files.**

```
ubuntu@ip-172-31-32-198:~$ sudo userdel -r tomd
userdel: tomd mail spool (/var/mail/tomd) not found
userdel: tomd home directory (/home/tomd) not found
ubuntu@ip-172-31-32-198:~$ id tomd
id: ‘tomd’: no such user
ubuntu@ip-172-31-32-198:~$
```

Deleted acc for “tomd” using “userdel -r” (r- deletes home dir and its contents as well)
Checked whether “tomd” still exists using “id tomd”, and found no-user exists with such id.

• **Create a Shared Admin Group:**

• **Both projects need an admin group for managing project-specific permissions. Create an admin_alpha and admin_beta group.**

• **Add yourself to both groups for administrative purposes.**

```
ubuntu@ip-172-31-32-198:~$ sudo groupadd admin_alpha
ubuntu@ip-172-31-32-198:~$ sudo groupadd admin_beta
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG admin_alpha, admin_beta ubuntu
usermod: group '' does not exist
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG admin_alpha,admin_beta ubuntu
ubuntu@ip-172-31-32-198:~$ groups ubuntu
ubuntu : ubuntu adm cdrom sudo dip lxd admins admin_alpha admin_beta
ubuntu@ip-172-31-32-198:~$
```

Created 2 groups “admin_alpha” and “admin_beta”.
Then added “ubuntu” as user to both of them.
Verified the correct assigning using “groups ubuntu” and got desired results.

Question: 9

You are a System Administrator responsible for maintaining a secure environment on a shared Linux server used by various teams.

- **Set User Password:**
- **A new user, emma_w, has just joined the team. After creating her account, she needs to set a strong password.**

```
ubuntu@ip-172-31-32-198:~$ sudo useradd emma_w
ubuntu@ip-172-31-32-198:~$ sudo passwd emma_w
New password:
Retype new password:
passwd: password updated successfully
```

- Created a new user “emma_w” and set her password.
- **Guide her to set her password using the passwd command. Ensure the password meets the company’s security policies.**

```
ubuntu@ip-172-31-32-198:~$ sudo nano /etc/pam.d/common-password
```

Opened the common-password configuration file using above cmd

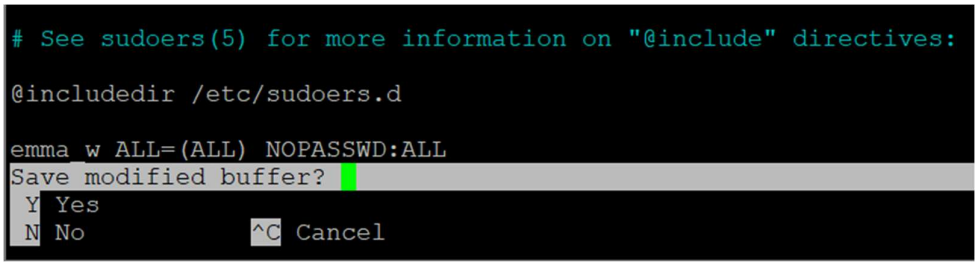
```
# here are the per-package modules (the "Primary" block)
<=3 minlen=12 dcredit=-1 ucredit-1 lcredit=-1 ocredit=-1
```

set policies such as “minlen=12” (minimum password length should be at least 12), “dcredit=-1” (password should contain as least 1 numeric digit), “lcredit=-1” (password should contain as least 1 lowercase alphabet), “ucredit=-1” (password should contain as least 1 uppercase alphabet), “ocredit=-1” (password should contain as least 1 special character).

```
ubuntu@ip-172-31-32-198:~$ sudo passwd emma_w
New password:
BAD PASSWORD: The password contains less than 1 digits
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password contains less than 1 non-alphanumeric characters
Retype new password:
```

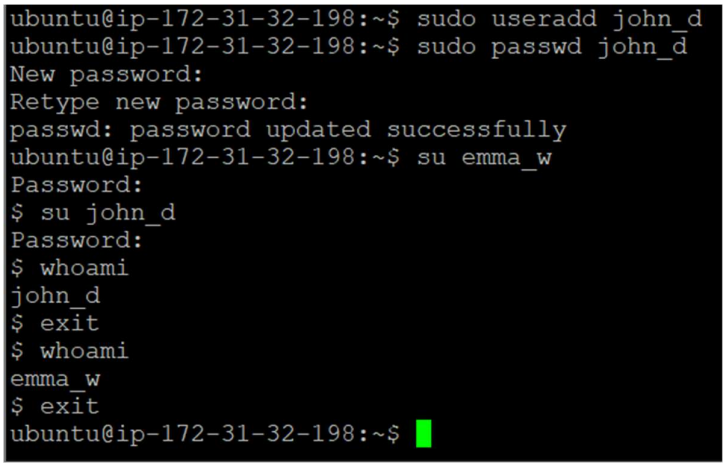
When updating password, it shows if password is strong enough to sustain standard security policies.

- **Temporary Root Access:**
- **For a critical system update, Emma needs temporary root access to perform administrative tasks.**
- **As a security measure, instead of sharing the root password, provide her with sudo privileges.**
- **Document the steps she would take to gain root access using the sudo command and how to perform a secure task, such as updating the system.**



Opened the sudoers file using “sudo visudo”.
Provided “emma_w” with temporary root access using “emma_w ALL=(ALL) NOPASSWD:ALL”, which will require no password when “emma_w” run any cmds from sudo group.
Verified by trying to update “apt” after switching to “emma_w” acc and successfully did so.

- **Switch User Role:**
- **After finishing her work, Emma needs to switch to another user's account, john_d, to verify some configurations.**
- **Explain how Emma can use the su command to switch to John’s account, and specify the importance of logging out after the task.**



First created a user “john_d” and set his password.
Then switched to “emma_w” user using “su” (switch) cmd.
From there switched to “john_d” user using “su” cmd.
Checked whether I’m at correct acc using “whoami”.
After checking successfully returned to original user using “exit” cmd.

- **Modify File Permissions:**
- **Emma notices that a script she needs to execute does not have the proper permissions. The script is located at /home/emma_w/scripts/update.sh.**
- **Change the permissions of the script to make it executable only by Emma using the chmod command.**

```
ubuntu@ip-172-31-32-198:~$ sudo mkdir -p /home/emma_w/scripts
```

Created a “scripts” dir at the required location using “mkdir”

```
ubuntu@ip-172-31-32-198:~$ sudo chown emma_w:emma_w /home/emma_w/scripts
ubuntu@ip-172-31-32-198:~$ sudo touch /home/emma_w/scripts/update.sh
ubuntu@ip-172-31-32-198:~$ sudo chown emma_w:emma_w /home/emma_w/scripts/upda
te.sh
ubuntu@ip-172-31-32-198:~$ sudo chmod 700 /home/emma_w/scripts/update.sh
ubuntu@ip-172-31-32-198:~$ ls -ld  /home/emma_w/scripts/update.sh
-rwx------ 1 emma_w emma_w 0 Apr  4 05:38 /home/emma_w/scripts/update.sh
```

Made “emma_w” owner of the “scripts” dir.
Created a script file named “update.sh”.
Now gave her permission for “update.sh” as well.
And at last set the permissions as (chmod 700) so that only emma_w can access it (update.sh).
Verified using “ls -ld” and It shows that only emma_w has access (rwx------).

- **Change File Ownership:**
- **The script Emma worked on is now ready to be shared with the entire team. To ensure proper access, the ownership of the script should be transferred to the team group.**
- **Use the chown command to change the group ownership of the script to team, while keeping Emma as the file owner.**

```
ubuntu@ip-172-31-32-198:/$ sudo groupadd team
ubuntu@ip-172-31-32-198:/$ sudo usermod -aG team emma_w
ubuntu@ip-172-31-32-198:/$ groups emma_w
emma_w : emma_w sudo team
```

Created a group “team” and added “emma_w” in that.
Check whether “emma_w” belongs to that group.
Verified successfully using “groups emma_w”.

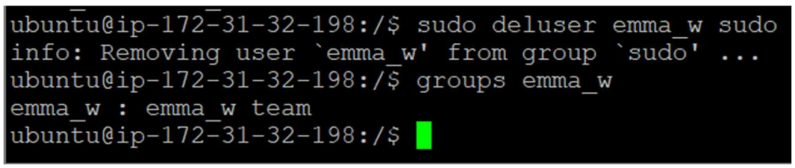
```
ubuntu@ip-172-31-32-198:/$ sudo chown emma_w:team /home/emma_w/scripts/update
.sh
```

```
ubuntu@ip-172-31-32-198:/$ sudo ls -ld /home/emma_w/scripts/update.sh
-rwx------ 1 emma_w team 0 Apr  4 05:38 /home/emma_w/scripts/update.sh
```

now change the group ownership for “update.sh” to “team” and verified successfully using “ls -ld …../scripts/update.sh”.

- **Remove Temporary Privileges:**
- **After the system update is complete, revoke Emma’s sudo privileges to maintain security. Document the process to ensure the removal is verified.**

Delete or comment out the “privilege” added in the “soders” file to discard all the privileges given to “emma_w”



Removed “emma_w” from sudo group and checked if she still exists in group “sudo”.
Successfully removal “emma_w” is done.

Question: 10

Scenario: You are managing a Linux server that hosts files for various projects. Each project has specific access requirements.

• Set Password Expiry:

• The company's security policy requires users to change their passwords every 90 days. Set this policy for the user mike_b using the passwd command.

```
ubuntu@ip-172-31-32-198:~$ sudo useradd mike_b
ubuntu@ip-172-31-32-198:~$ sudo passwd mike_b
New password:
Retype new password:
passwd: password updated successfully
ubuntu@ip-172-31-32-198:~$ sudo chage -M 90 mike_b
ubuntu@ip-172-31-32-198:~$ sudo chage -l mike_b
Last password change                                    : Apr 04, 2025
Password expires                                        : Jul 03, 2025
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 90
Number of days of warning before password expires       : 7
ubuntu@ip-172-31-32-198:~$
```

Created a user "mike_b" and set his password.

Changed his password expiry change date to 90 days using "passwd chage -M 90" and we can also use "sudo passwd -x 90 mike_b".
Verified the update using "passwd chage -l mike_b".

• Project File Permissions:

• Mike_b is working on a confidential project. The project files are stored in /projects/alpha/.

```
ubuntu@ip-172-31-32-198:~$ sudo mkdir -p projects/alpha
```

Created a parent dir as "project" under which there is a sub dir "alpha"

• Set permissions on this directory so that only Mike can read, write, and execute files within it. Use chmod to restrict access for all other users.

```
ubuntu@ip-172-31-32-198:~$ sudo chown mike_b:mike_b projects/alpha
ubuntu@ip-172-31-32-198:~$ sudo chmod 700 projects/alpha
ubuntu@ip-172-31-32-198:~$ ls -ld projects/alpha
drwx------ 2 mike_b mike_b 4096 Apr  4 06:21 projects/alpha
ubuntu@ip-172-31-32-198:~$
```

Set "mike_b" as the owner for "projects/alpha".
and gave only him access to read, write and execute the "project/alpha" (used chmod 700).
And verified using "ls -ld" (rwx------) that only "mike_b" has access to the dir.

• **Switch User Context:**

• **Mike needs to temporarily assume the identity of another user, sara_c, to check some configurations. Explain how he can switch to Sara's account using the su command.**

```
ubuntu@ip-172-31-32-198:~$ sudo useradd sara_c
ubuntu@ip-172-31-32-198:~$ sudo passwd sara_c
New password:
Retype new password:
passwd: password updated successfully
ubuntu@ip-172-31-32-198:~$
```

For that first created the user "sara_c" and set her password

```
ubuntu@ip-172-31-32-198:~$ su mike_b
Password:
$ su sara_c
Password:
$ whoami
sara_c
$ exit
$ whoami
mike_b
$ exit
ubuntu@ip-172-31-32-198:~$
```

In order to switch to "sara_c" we switched to "mike_b" and then using "su" cmd switched to "sara_c" and checked it using "whoami" and then logout using "exit" cmd.

• **Grant Limited Administrative Access:**

• **Mike needs to install some software but should not have full root access. Add him to the sudo group with limited privileges to install software packages only.**

• **Provide an example of how Mike would install a package using sudo.**

```
# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d

mike_b ALL=(ALL) NOPASSWD: /usr/bin/apt-get
```

Using "sudo visudo", opened sudoers file and append the permission line allowing "mike_b" can only install software packages using sudo.

```
ubuntu@ip-172-31-32-198:~$ su mike_b
Password:
$ sudo apt-get install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.2).
0 upgraded, 0 newly installed, 0 to remove and 26 not upgraded.
$
```

Switched to "mike_b" user and tried installing git.
Successfully able to install git.

• **Ownership Transfer for Collaboration:**

• The project is now in a collaborative phase, and the files need to be accessible by the devteam group.

• Use the chown command to change the ownership of the files in /projects/alpha/ to the devteam group while retaining Mike as the file owner.

```
ubuntu@ip-172-31-32-198:~$ sudo groupadd devteam
ubuntu@ip-172-31-32-198:~$ sudo usermod -aG devteam mike_b
```

Made a group “devteam” and appended “mike_b” to it.

```
ubuntu@ip-172-31-32-198:~$ sudo chown mike_b:devteam projects/alpha
ubuntu@ip-172-31-32-198:~$ sudo ls -ld projects/alpha
drwx------ 2 mike_b devteam 4096 Apr  4 06:21 projects/alpha
ubuntu@ip-172-31-32-198:~$ sudo chmod 770 projects/alpha
ubuntu@ip-172-31-32-198:~$ ls -ld projects/alpha
drwxrwx--- 2 mike_b devteam 4096 Apr  4 06:21 projects/alpha
ubuntu@ip-172-31-32-198:~$
```

Changed the group ownership of “project/alpha” to “devteam” and verified using “ls -ld projects/alpha” cmd.
Then gave group members read, write and excute permissions using chmod (770) to “projects/alpha”

• Revoke User Access:

• Mike is transferring to a different project. Remove his access to the /projects/alpha/ directory and ensure he can no longer use sudo on the system. Document the steps to verify these changes.

```
ubuntu@ip-172-31-32-198:~$ sudo deluser mike_b devteam
info: Removing user `mike_b' from group `devteam' ...
```

Removed “mike_b” from “devteam” using “deluser” cmd.
Now also deleted the extra privilege line added in the sudoers file (using “sudo visudo” to go to sudoers file) for “mike_b” to install software packages.

```
ubuntu@ip-172-31-32-198:~$ groups mike_b
mike_b : mike_b
ubuntu@ip-172-31-32-198:~$
```

And verified that “mike_b” is not present in sudo group any longer.