Project Title:

Managing File and Directory Permissions in Linux

Project Description:

The research team at my organisation needed to secure sensitive project files by configuring appropriate permissions. I used Linux terminal commands to identify incorrect permissions, remove unauthorised access, and ensure only the intended users had the appropriate read, write, or execute rights. This project demonstrated how correct file and directory permissions help protect shared systems from accidental or malicious changes.

Task 1: Check File and Directory Details

Task:

In this task, you must explore the permissions of the projects directory and the files it contains. The lab starts with /home/researcher2 as the current working directory. This is because you're changing permissions for files and directories belonging to the researcher2 user.

1. **Navigate to the projects directory.**
2. **List the contents and permissions of the projects directory.**

What I did:

- Verified I was in the correct working directory using pwd
- Listed current files and folders using ls
- Navigated to the projects directory with cd projects
- Used ls -l to check the contents and their permissions

```
researcher2@09abab2dd29a:~$ pwd
/home/researcher2
researcher2@09abab2dd29a:~$
```

```
researcher2@09abab2dd29a:~$ pwd
/home/researcher2
researcher2@09abab2dd29a:~$ ls
projects
researcher2@09abab2dd29a:~$ cd projects
researcher2@09abab2dd29a:~/projects$
```

Task 2: Describe the Permissions String

Explanation:

The 10-character string shown in ls -l output describes the file type and permissions for user, group, and others. Example: -rw-rw-r--

- The **first character** indicates the type:
  - d = directory
  - - = regular file
- Characters **2–4** = User permissions (r, w, x)
- Characters **5–7** = Group permissions (r, w, x)
- Characters **8–10** = Other permissions (r, w, x)

This string helps identify what level of access each owner type has on files and directories.The second block of text in the expanded directory listing is the user who owns the file. The third block of text is the group owner of the file.

```
researcher2@09abab2dd29a:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Apr 19 14:00 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Apr 19 14:00 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Apr 19 14:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 19 14:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 19 14:00 project_t.txt
researcher2@09abab2dd29a:~/projects$
```

Task 3: Change File Permissions

Task:

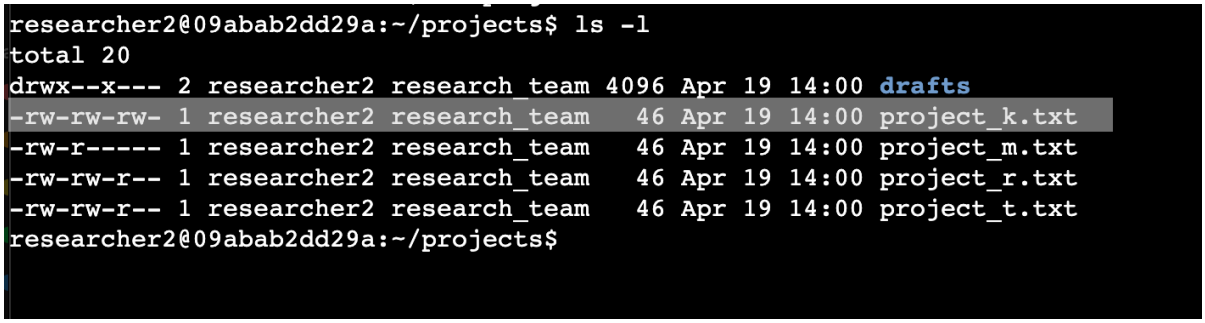
In this task, you must determine whether any files have incorrect permissions and then change the permissions as needed. This action will remove unauthorised access and strengthen security on the system.
None of the files should allow the other users to write to files.

1. **Check whether any files in the projects directory have write permissions for the owner type of other.**
2. **Change the permissions of the file identified in the previous step so that the owner type of other doesn't have write permissions.**

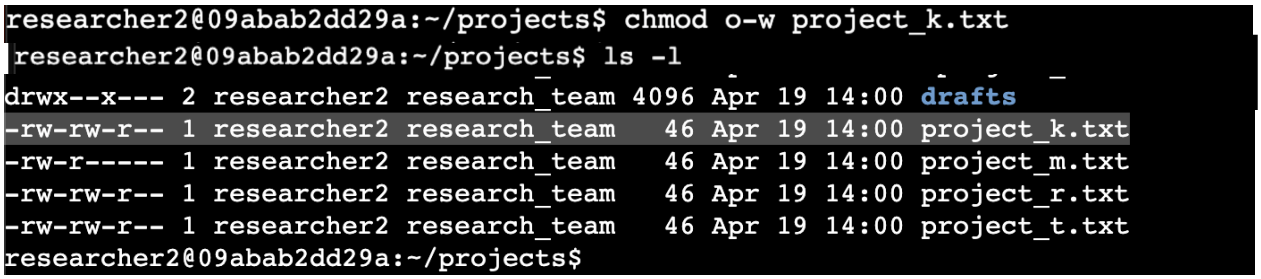
Note: *Permissions are granted for three different types of owners, namely user, group, and other.*

What I did:

- Used ls -l to identify that project_k.txt had write permissions for others
- Ran chmod o-w project_k.txt to remove write access from the "other" owner type
- Verified the update using ls -l



- As seen in the image above the project_k.txt file has read and write permissions for all the owner types, according to the instructions none of the files should allow the other owner type to have write permissions so I need to change permissions for the project_k.txt file.
- I used the command chmod(change mode) o-w(owner type other - remove the write permission) project_k.txt(file) in the (CLI). I then checked to make sure that the change has occurred using the command ls -l in the (CLI) which would show all files.



Task 4: Change Hidden File Permissions

Task:

In this task, you must determine if a hidden file has incorrect permissions and then change the permissions as needed. This action will further remove unauthorised access and strengthen security on the system. The file .project_x.txt is a hidden file that has been archived and should not be written to by anyone. (The user and group should still be able to read this file.)

What I did:

- Ran ls -la to show all files, including hidden ones
- Observed that .project_x.txt had write permissions for both user and group
- Ran chmod u=r,g=r .project_x.txt to set both to read-only
- Verified change using ls -la

```
researcher2@09abab2dd29a:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Apr 19 14:00 .
drwxr-xr-x 3 researcher2 research_team 4096 Apr 19 14:33 ..
-rw--w---- 1 researcher2 research_team   46 Apr 19 14:00 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Apr 19 14:00 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Apr 19 14:00 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Apr 19 14:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 19 14:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 19 14:00 project_t.txt
researcher2@09abab2dd29a:~/projects$
researcher2@09abab2dd29a:~/projects$ chmod u=r,g=r .project_x.txt
researcher2@09abab2dd29a:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Apr 19 14:00 .
drwxr-xr-x 3 researcher2 research_team 4096 Apr 19 14:33 ..
-r--r----- 1 researcher2 research_team   46 Apr 19 14:00 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Apr 19 14:00 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Apr 19 14:00 project_k.txt
-rw------- 1 researcher2 research_team   46 Apr 19 14:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 19 14:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 19 14:00 project_t.txt
researcher2@09abab2dd29a:~/projects$
```

Task 5: Change Directory Permissions

Task:

In this task, you must change the permissions of a directory. First, you'll check the group permissions of the /home/researcher2/projects/drafts directory and then modify the permissions as required. (You should be in the projects directory while managing the permissions of its subdirectory drafts.)

Only the researcher2 user should be allowed to access the drafts directory and its contents. (This means that only researcher2 should have execute privileges.)

1. **Check the permissions of the drafts directory and determine if they are correct.**
2. **Remove the execute permission for the group from the drafts directory.**

What I did:

- Ran ls -l to inspect the drafts directory's current permissions
- Noticed the group also had execute permission
- Ran chmod g-x drafts to remove execute access from the group
- Verified change using ls -l

```
researcher2@09abab2dd29a:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Apr 19 14:00 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Apr 19 14:00 project_k.txt
-rw------- 1 researcher2 research_team   46 Apr 19 14:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 19 14:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 19 14:00 project_t.txt
researcher2@09abab2dd29a:~/projects$
```

```
researcher2@09abab2dd29a:~/projects$ chmod g-x drafts
researcher2@09abab2dd29a:~/projects$ ls -l
total 20
drwx------ 2 researcher2 research_team 4096 Apr 19 14:00 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Apr 19 14:00 project_k.txt
-rw------- 1 researcher2 research_team   46 Apr 19 14:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 19 14:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 19 14:00 project_t.txt
researcher2@09abab2dd29a:~/projects$
```

Summary:

In this project, I managed file and directory permissions in a Linux system to improve security. I used commands like ls -l, ls -la, and chmod to inspect and modify access controls. I removed unauthorised write permissions from regular and hidden files and ensured that only the correct users had access to sensitive directories. This hands-on experience helped reinforce my understanding of Linux file permission structures and their importance in multi-user environments.