# File permissions in Linux

## Project description

I have completed a lab activity where I used Linux commands to configure authorization. I explored file and directory permissions and changed the ownership of a file and a directory to control access. This was important to ensure security by restricting unauthorized users from accessing or modifying sensitive files. As a security analyst, I understand the importance of setting proper access permissions to protect information and maintain system security.

## Check file and directory details

I completed a task where I checked the details of files and directories, specifically exploring the permissions of the *projects* directory and its contents. I started in the */home/researcher2* directory since I was modifying permissions for files and directories belonging to the *researcher2* user. I navigated to the *projects* directory using `cd projects` and listed its contents and permissions with `ls -l`.

```
researcher2@b93d04377035:~$ ls
projects
researcher2@b93d04377035:~$ cd projects
researcher2@b93d04377035:~/projects$ ls
drafts  project_k.txt  project_m.txt  project_r.txt  project_t.txt
researcher2@b93d04377035:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Feb 19 11:12 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Feb 19 11:12 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Feb 19 11:12 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_t.txt
```

## Describe the permissions string

In Linux, file permissions are represented by a string of characters, as seen in `-rw-r-----` for the file *project_m.txt*. These permissions define what actions different categories of users can perform on the file. Here's a breakdown of what each part means:

1. **User (Owner) Permissions** (`rw-`):
    - The *researcher2* user (who owns the file) has **read (r)** and **write (w)** permissions.
    - This means *researcher2* can view and modify (edit, delete, or rename) the file.
    - However, the absence of **execute (x)** means the file cannot be executed as a program or script.
2. **Group Permissions** (`r--`):
    - The *research_team* group (which *researcher2* is a part of) has **read (r)** permission.
    - This means any user who belongs to the *research_team* group can only view the file but cannot modify or execute it.
3. **Other (Everyone Else) Permissions** (`---`):
    - Users who are neither the owner (*researcher2*) nor part of the *research_team* group have **no permissions** at all.
    - They cannot read, write, or execute the file.

### Summary:

- **User (Owner) = Read & Write**
- **Group = Read Only**
- **Other = No Access**

This setup ensures that only the file owner can modify the file, the group members can read it, and all other users have no access, enhancing security.

## Change file permissions

I completed a task where I checked and corrected file permissions to enhance system security. First, I listed the files in the *projects* directory using `ls -l` to determine if any files had incorrect permissions. I found that *project_k.txt* had write permissions for other users, which was a security risk. To remove unauthorized access, I used the command `chmod o-w project_k.txt`, ensuring that other users no longer had write permissions for the file.

```
researcher2@b93d04377035:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Feb 19 11:12 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Feb 19 11:12 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Feb 19 11:12 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_t.txt
researcher2@b93d04377035:~/projects$ ls -a
.  ..  .project_x.txt  drafts  project_k.txt  project_m.txt  project_r.txt  project_t.txt
researcher2@b93d04377035:~/projects$ chmod o-w project_k.txt
```

## Change file permissions on a hidden file

I completed a task where I checked and corrected the permissions of a hidden file to improve system security. First, I listed all files, including hidden ones, using `ls -la` to check the permissions of *.project_x.txt*. I found that both the user and group had incorrect write permissions, even though the file was archived and should not be modified.

To fix this, I attempted to run `chmod u-w,g-w,g+r .project_x.txt`, but I mistakenly included a space after the comma, causing the command to fail. To correct this, I executed the commands separately:

- `chmod u-w .project_x.txt` to remove write permissions for the user
- `chmod g-w .project_x.txt` to remove write permissions for the group

After these changes, the file was properly restricted, allowing only read access for the user and group.

```
researcher2@b93d04377035:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb 19 11:12 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb 19 12:13 ..
-rw--w---- 1 researcher2 research_team   46 Feb 19 11:12 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Feb 19 11:12 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_k.txt
-rw------- 1 researcher2 research_team   46 Feb 19 11:12 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_t.txt
```

```
researcher2@b93d04377035:~/projects$ chmod u-w, g-w .project_x.txt
chmod: invalid mode: 'u-w,'
Try 'chmod --help' for more information.
researcher2@b93d04377035:~/projects$ chmod u=r, g=r .project_x.txt
chmod: invalid mode: 'u=r,'
Try 'chmod --help' for more information.
researcher2@b93d04377035:~/projects$ chmod u -w, g -w .project_x.txt
chmod: invalid mode: '-w,,-w'
Try 'chmod --help' for more information.
researcher2@b93d04377035:~/projects$ chmod u-wx, g-wx .project_x.txt
chmod: invalid mode: 'u-wx,'
Try 'chmod --help' for more information.
researcher2@b93d04377035:~/projects$ chmod u-w
chmod: missing operand after 'u-w'
Try 'chmod --help' for more information.
researcher2@b93d04377035:~/projects$ chmod u-w .project.txt
chmod: cannot access '.project.txt': No such file or directory
researcher2@b93d04377035:~/projects$ chmod u-w .project_x.txt
researcher2@b93d04377035:~/projects$ chmod g-w .project_x.txt
researcher2@b93d04377035:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb 19 11:12 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb 19 12:13 ..
-r-------- 1 researcher2 research_team   46 Feb 19 11:12 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Feb 19 11:12 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_k.txt
-rw------- 1 researcher2 research_team   46 Feb 19 11:12 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_t.txt
researcher2@b93d04377035:~/projects$ chmod g+r .project_x.txt
researcher2@b93d04377035:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb 19 11:12 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb 19 12:13 ..
-r--r----- 1 researcher2 research_team   46 Feb 19 11:12 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Feb 19 11:12 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_k.txt
-rw------- 1 researcher2 research_team   46 Feb 19 11:12 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_t.txt
```

## Change directory permissions

Final task was to complete a task where I modified the permissions of a directory to restrict access. First, while in the *projects* directory, I checked the group permissions of the */home/researcher2/projects/drafts* directory. I needed to ensure that only the *researcher2* user could access the *drafts* directory and its contents, meaning only *researcher2* should have execute privileges.

To enforce this restriction, I used the command `chmod g-x drafts`, which removed execute permissions for the group. This ensured that no group members could access the directory, improving security by limiting access to the owner only.

```
researcher2@b93d04377035:~/projects$ chmod g-x drafts
researcher2@b93d04377035:~/projects$ ls -l
total 20
drwx------ 2 researcher2 research_team 4096 Feb 19 11:12 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_k.txt
-rw------- 1 researcher2 research_team   46 Feb 19 11:12 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 11:12 project_t.txt
researcher2@b93d04377035:~/projects$
```

## Summary

I completed several tasks to manage file and directory permissions in Linux, ensuring system security by restricting unauthorized access. I explored and modified file permissions, changing ownership where necessary. I removed write permissions for other users on *project_k.txt* and corrected incorrect permissions on the hidden file *.project_x.txt*. Additionally, I restricted access to the *drafts* directory, allowing only the *researcher2* user to access its contents. These actions helped strengthen security by enforcing proper authorization rules.