File permissions in Linux

Project description

The data science department in my organization needed several permission updates for files and folders stored inside the **projects** directory. The current permissions did not align with the level of access each user should have. It was necessary to review and adjust these permissions to better protect the system and the work stored inside it. To complete this task, I performed the following steps.

Check file and directory details

The following code demonstrates how I used Linux commands to view the current permissions applied to a directory in the file system.

```
ls -la/home/researcher2/projects
```

Describe the permissions string

The 10-character permissions string reveals who can access a file and what they are allowed to do.

- **Character 1:** Indicates file type — `d` for directory, `-` for regular file
- **Characters 2–4:** Permissions for the **user** (read, write, execute)
- **Characters 5–7:** Permissions for the **group**
- **Characters 8–10:** Permissions for **others** (everyone else)

For example, if a file shows `-rw-rw-r--`, the first `-` shows it is a regular file. The second and fifth characters being `r` indicate that both the user and group can read the file, and the third and sixth characters `w` show that the user and group can also write. The last three characters are `r--`, meaning others can only read and do not have permission to write or execute.

Change file permissions

```
chmod o-w /home/researcher2/projects/project_m.txt
```

To enforce access control, I removed write permission from others on the file project_m.txt. The chmod o-w command updated the permission string by removing the ability for other users to make changes to the file. This helps protect the file from unauthorized editing.

Change file permissions on a hidden file

```
chmod o-w /home/researcher2/projects/.project_x.txt
```

The hidden file .project_x.txt needed its access reduced because it contained archived project data. I used chmod o-w to remove write permission from others, preventing unwanted edits. Hidden files begin with a dot (.) and should be checked carefully since they may contain sensitive information.

Change directory permissions

```
chmod 700 /home/researcher2/projects/drafts
```

To restrict the drafts directory so only the owner could access it, I applied chmod 700. This permission setting grants full access (read, write, execute) only to the owner and blocks all access for group and others. This ensures the contents of the directory remain private to the user who owns it.

Summary

During this assignment, I examined and modified Linux file permissions to match the organization's security needs. I started by using ls -la to identify the current permissions on each item. Then I applied several chmod commands to update permissions for a regular file, a hidden file, and a directory. These changes strengthened access control and reduced the risk of unauthorized file modifications.