File permissions in Linux

Project description

In this project, I explored how Linux file permissions work and how to manage them using terminal commands. I worked within a simulated research environment where proper access control was critical. Through a series of tasks, I examined existing permissions, interpreted what they meant, and applied the correct changes to ensure that files and directories were only accessible to the appropriate users. This hands-on exercise helped reinforce my understanding of Linux permissions and the importance of secure file management.

Check file and directory details

To see the current permissions for all files and folders in the projects directory, I used this command: ls -la /home/researcher2/projects

The -l flag gives a detailed (long) listing, and the -a flag includes hidden files (those that start with a dot). This allowed me to review each item's permissions, ownership, and other metadata to identify what changes were needed.

Describe the permissions string

One example from the directory is the file project_k.txt, which had the following permission string: -rw-rw-rw-
The permission string -rw-rw-rw- indicates a regular file where the owner and group can both read and write, and others also have read and write access, which poses a security risk since it allows anyone to modify the file—something that goes against the organization's policy.

Change file permissions

To fix the permission issue with project_k.txt and remove write access for "others," I ran this command: chmod o-w /home/researcher2/projects/project_k.txt

This command tells the system to take away (-) write access (w) from others (o). After running it, the updated permissions look like this: -rw-rw-r--

Change file permissions on a hidden file

The hidden file .project_x.txt should not be writable by anyone. However, it still needs to be readable by the user and group. To fix its permissions, I used this command:
chmod 440 /home/researcher2/projects/.project_x.txt
After this change, the file's permission string becomes: -r--r-----

Change directory permissions

The drafts directory is supposed to be private — only the owner (researcher2) should be able to see or edit it. To lock it down, I used this command:
chmod 700 /home/researcher2/projects/drafts
The result is: drwx------
This keeps the directory secure and prevents unauthorized users from accessing its contents.

Summary

This activity gave me a hands-on opportunity to manage file and directory permissions in a Linux system. I checked existing permissions, explained what they meant, and applied the appropriate changes using the chmod command. I also made sure that hidden files and directories were handled securely. These tasks helped me build a strong foundation in Linux access control and understand how to keep sensitive information protected.