**File Permissions in Linux**

**Project Description**

In this project, I demonstrate my ability to manage file permissions using Linux commands. As a security professional working with a research team, my role involves ensuring that users have appropriate authorization for file access. Through this process, I verify existing permissions, interpret permission strings, and modify them using relevant Linux commands to enhance security.

**Check File and Directory Details**

To examine the current file and directory permissions, I used the following command:

This command lists all files, including hidden ones, along with their permissions, ownership details, and timestamps. Below is an example output:

**Describe the Permissions String**

A file permission string consists of 10 characters, structured as follows:

* *- :* Regular file (not a directory)
* *rw- :* Owner (read, write, no execute)
* *rw- :* Group (read, write, no execute)
* *r-- :* Others (read only, no write or execute)

This means that owner and group members can read and write the file, while others can only read it.

**Change File Permissions**

The organization does not allow others to have write access to any files. To remove write access from others, I used:

After running this command, the updated permissions became:

Now, others can only read the file but not modify it.

**Change File Permissions on a Hidden File**

The research team archived .project\_x.txt, and it should not have write permissions for anyone, but the user and group should be able to read it. To modify the permissions, I used:

The updated permissions:

Now, only the owner and group members can read the file, while others have no access.

**Change Directory Permissions**

The drafts directory should only be accessible to researcher2. To enforce this restriction, I executed:

This changed the permissions to:

Now, only researcher2 can access, modify, or execute files within the drafts directory.

**Summary**

In this project, I used ls -la to check file permissions and interpreted permission strings. I ensured secure authorization by modifying file permissions with chmod to remove unauthorized write access, manage hidden file permissions, and restrict directory access. This process enhances system security by enforcing proper access control in a research environment.