Linux Commands Activity

Name: Hridhima Karmakar

Project Description:
In this activity, I explored how to manage file and directory permissions in a Linux environment using basic command-line tools. I used the `ls -la` command to inspect permissions, identified and managed hidden files, and applied `chmod` to adjust access rights for users, groups, and others. This exercise enhanced my understanding of authorization and access control—key components of system security.

Task 1: Checking File and Directory Permissions

Command used:

```
ls -la
```

Explanation:
This command lists all files, including hidden ones (. prefix), in long format. It displays the file type, permissions, ownership, size, and modification date.

Interpreting the permission string:
Each line begins with a 10-character string

- 1st character: file type (`-` for file, `d` for directory)
- Next 3: user permissions
- Next 3: group permissions
- Last 3: others' permissions

Task 2: Identifying and Correcting Permissions

A. Removing write permissions for others on `project_k.txt`:

Command:

```
chmod o-w project_k.txt
```

Explanation:
Removes write access for "others" (the public) to strengthen file security.

B. Removing group read permissions from `project_m.txt`:

Command:

```
chmod g-r project_m.txt
```

Explanation:
This ensures only the file owner can read the file, restricting group access.

C. Managing a hidden file (`.project_x.txt`):

Command:

```
chmod u-w,g-w,g+r .project_x.txt
```

Explanation:
This corrects over-permissive access by removing write rights for both user and group, while ensuring the group retains read access.

Task 3: Directory Permissions

Checking directory permissions on `drafts/`:

```
ls -ld drafts
```

Removing execute permission for group:

```
chmod g-x drafts
```

Explanation:
This removes the ability for group members to enter the directory, securing its contents.

Summary

In this activity, I used Linux commands to audit and adjust file and directory permissions. I learned to:

- Interpret the Linux permission string format
- Detect and manage hidden files
- Use `chmod` to customize access control
- Secure directories by managing execute permissions

This hands-on task reinforced the principles of least privilege and access control, essential to maintaining secure Linux-based systems.