Project Title: Use Linux Commands to Manage File Permissions

Project Description

In this activity, I acted as a **security professional** responsible for managing access control within a research team's shared project directory. The task involved examining file and directory permissions using Linux commands and adjusting them to meet the organization's security policy. Through this exercise, I ensured that users were authorized appropriately, unauthorized write access was removed, and sensitive files were restricted to the correct users and groups.

Check File and Directory Details

Command used:

```
ls -la
```

Explanation:
The `ls -la` command lists all files (including hidden files) in the directory, along with detailed information such as file type, permissions, owner, group, and timestamps. This helps identify which files have incorrect access rights.

Example output:

```
drwx--x--- 2 researcher2 research_team 4096 Oct 11 13:00 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 11 13:00
project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 11 13:00
project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 11 13:00
project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 11 13:00
project_t.txt
-rw--w---- 1 researcher2 research_team   46 Oct 11 13:00
.project_x.txt
```

Describe the Permissions String

Example permission string:
`-rw-rw-rw-`

Explanation:
The 10-character string represents file type and access rights.

- The first character `-` indicates a regular file (a `d` would mean a directory).
- The next three characters `rw-` are the **owner's** permissions (read and write).
- The following three `rw-` are for the **group** (read and write).
- The final three `rw-` are for **others** (read and write).
  This configuration allows anyone to edit the file, which violates the organization's security policy.

Change File Permissions

Command used:

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

Explanation:
The organization's rule states that "others" should not have write permissions.
This command removes write (`w`) permission from "others" on the file `project_k.txt`.

Before: `-rw-rw-rw-`
After: `-rw-rw-r--`

This ensures only the file owner and group members can modify the file, while others can only read it.

Change File Permissions on a Hidden File

Command used:

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

Explanation:
Hidden files (those starting with a `.`) are still accessible and must be secured properly.
This command ensures `.project_x.txt` cannot be written to by anyone, but both the user and group can read it.

Before: `-rw--w----`
After: `-r--r-----`

This change meets the requirement that archived or hidden files remain readable but protected from modification.

Change Directory Permissions

Command used:

```
chmod g-x drafts
```

Explanation:
The `drafts` directory initially allowed group members to enter it (`x` permission).
Since only the owner (`researcher2`) should have access, this command removes execute (`x`) permission from the group, preventing them from entering or listing its contents.

Before: `drwx--x---`
After: `drwx------`

Summary

In this project, I used Linux commands to review and modify file and directory permissions in the `projects` directory.
Using `ls -la`, I identified files with excessive write access. I then applied the `chmod` command to restrict permissions, ensuring compliance with organizational security standards. These updates improved system security by enforcing **least privilege**, protecting hidden files, and ensuring sensitive directories are accessible only to authorized users.

Key Commands Demonstrated

Purpose	Command	Example Result
View all file permissions (including hidden files)	`ls -la`	Displays full permission details
Remove write permission for others	`chmod o-w project_k.txt`	Sets others to read-only
Restrict hidden file access	`chmod u-w,g-w,g+r .project_x.txt`	User & group read-only
Limit directory access to owner	`chmod g-x drafts`	Only owner can enter directory

Skills Demonstrated

- File and directory permission management
- Applying least-privilege principles
- Using `chmod` for access control
- Interpreting Linux permission strings
- Handling hidden files securely

```
researcher2@435a894eb7de:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 11 13:00 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 11 13:00 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 11 13:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 11 13:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 11 13:00 project_t.txt
researcher2@435a894eb7de:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 11 13:00 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 11 13:40 ..
-rw--w---- 1 researcher2 research_team   46 Oct 11 13:00 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 11 13:00 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 11 13:00 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 11 13:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 11 13:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 11 13:00 project_t.txt
researcher2@435a894eb7de:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 11 13:00 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 11 13:00 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 11 13:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 11 13:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 11 13:00 project_t.txt
researcher2@435a894eb7de:~/projects$ chmod o-w project_k.txt
researcher2@435a894eb7de:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 11 13:00 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Oct 11 13:00 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 11 13:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 11 13:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 11 13:00 project_t.txt
researcher2@435a894eb7de:~/projects$ chmod g-r project_m.txt
researcher2@435a894eb7de:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 11 13:00 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 11 13:40 ..
-rw--w---- 1 researcher2 research_team   46 Oct 11 13:00 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 11 13:00 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Oct 11 13:00 project_k.txt
-rw------- 1 researcher2 research_team   46 Oct 11 13:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 11 13:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 11 13:00 project_t.txt
researcher2@435a894eb7de:~/projects$ chmod u-w, g-w, g+r .project_x.txt
chmod: invalid mode: 'u-w,'
Try 'chmod --help' for more information.
researcher2@435a894eb7de:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@435a894eb7de:~/projects$ chmod g-x drafts
researcher2@435a894eb7de:~/projects$
```