Linux File Permissions Management Portfolio

Name: Abubakari-Sadique Hamidu

Program: Mentor Me Collective - Grow With Google

LinkedIn: https://www.linkedin.com/in/sadique1

1. Project Description

As a security professional supporting a research team, my responsibility includes managing user permissions on the file system. This ensures that only authorized users have access to critical data. This portfolio demonstrates how I reviewed and updated file and directory permissions in the `/home/researcher2/projects` directory using Linux commands.

2. File Structure and Current Permissions

Files and their initial permissions:

- **project_k.txt**: User=rw, Group=rw, Other=rw
- **project_m.txt**: User=rw, Group=r, Other=none
- **project_r.txt**: User=rw, Group=rw, Other=r
- **project_t.txt**: User=rw, Group=rw, Other=r
- **.project_x.txt**: User=rw, Group=w, Other=none (hidden file)

Directory:

- **drafts/**: User=rwx, Group=x, Other=none

3. Checking File Permissions

To check file and directory permissions, I used the command:

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

This command lists all files, including hidden ones, along with their permissions and ownership. Example output:

```
 -rw-rw-rw- 1 researcher2 researcher2  234 Apr 18 10:00 project_k.txt
-rw-r----- 1 researcher2 researcher2  543 Apr 18 10:02 project_m.txt
-rw-rw-r-- 1 researcher2 researcher2  198 Apr 18 10:04 project_r.txt
-rw-rw-r-- 1 researcher2 researcher2  876 Apr 18 10:06 project_t.txt
--rw--w--- 1 researcher2 researcher2  765 Apr 18 10:08 .project_x.txt
drwx--x--- 2 researcher2 researcher2 4096 Apr 18 10:10 drafts
```

4. Modifying File Permissions

To ensure appropriate access, I modified permissions using the `chmod` command. Examples:

```
 chmod 664 project_k.txt   # Removes write for others
chmod 640 project_m.txt   # Removes read from group
chmod 664 project_r.txt   # Keeps rw for user/group, read for others
chmod 600 .project_x.txt  # Restricts hidden file access
```

These changes ensure only authorized access is granted.

5. Interpreting Permission Strings

The 10-character permission string (e.g., `-rw-r--r--`) breaks down as:

- 1st char: File type (`-` for regular file, `d` for directory)
- Next 3: User permissions (r, w, x)
- Next 3: Group permissions
- Last 3: Other permissions

6. Hidden Files and Directories

Files starting with a dot (.) like `.project_x.txt` are hidden. They are not shown by default with `ls` but are revealed using `ls -a` or `ls -la`. These are often used to store configuration or sensitive data.

7. Summary

This activity reinforced my understanding of Linux file permissions. I learned to interpret permission strings, identify unauthorized access, and apply secure permission settings using `chmod`. Mastering these skills helps maintain secure and organized systems, a critical responsibility in cybersecurity roles.