File permissions in Linux

Project description

As a security analyst, I used Linux commands to inspect and modify file and directory permissions under `/home/researcher2/projects` to ensure authorization policies were properly enforced. This activity followed the principle of least privilege by removing unnecessary access, especially for the “other” and “group” user types, on sensitive files and directories.

Check file and directory details

Navigated to the `projects` directory:

```
cd /home/researcher2/projects
```

Listed the contents with detailed permissions:

```
ls -l
```

Output:

```
drwx--x--- 2 researcher2 research_team 4096 Oct 14 18:40 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 14 18:40 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 14 18:40 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 14 18:40 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 14 18:40 project_t.txt
```

Also checked for hidden files:

```
ls -la
```

Hidden file found:

```
-rw-rw-rw- 1 researcher2 research_team   46 Oct 14 18:40
.project_x.txt
```

Group that owns the files: `research_team`
Hidden file present: `.project_x.txt`

Describe the permissions string

Each file or directory has a permission string like:

```
-rw-rw-rw-
```

Breakdown:

- `-`: Regular file
- `rw-`: User (read, write)
- `rw-`: Group (read, write)
- `rw-`: Other (read, write) ← Too permissive

For secure configurations, files should not allow write access to "other", and sensitive files should restrict group access too.

Change file permissions

project_k.txt had:

```
-rw-rw-rw-  → other had write access
```

Removed write access for “other”:

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

Now:

```
-rw-rw-r--  project_k.txt
```

project_m.txt is restricted, so only the user should have access.

Existing:

```
-rw-r-----  → group has read access
```

Removed group access:

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

Result:

```
-rw-------  project_m.txt
```

Change file permissions on a hidden file

Hidden file `.project_x.txt` had:

```
-rw-rw-rw-  → user, group, and others had write access
```

As it's archived and should only be readable, ran:

```
chmod ug=r .project_x.txt
```

Now:

```
-r--r--r--  .project_x.txt
```

No write access remains. Only read access for user and group.

Change directory permissions

Directory `drafts` had:

`drwx--x--- → group had execute access`

Removed execute (`x`) from group:

`chmod g-x drafts`

Now:

`drwx------  drafts`

Only the `researcher2` user can access this directory.

Summary

In this activity, I reviewed and updated Linux file and directory permissions under `/home/researcher2/projects` to align with secure authorization practices:

- Verified permissions using `ls -l` and `ls -la`
- Removed unnecessary write/read access for group and others
- Restricted sensitive files like `project_m.txt` and `.project_x.txt`
- Ensured only the owner can access the `drafts` directory

This helps enforce least privilege, protecting sensitive research files from unauthorized modification or exposure.