File Permissions in Linux

Project Description

In this project, I reviewed and modified file and directory permissions in a Linux environment to ensure only authorized users had proper access. This task simulated real-world system hardening by identifying overly permissive settings and locking down access for users outside the research team. The goal was to align with organizational security standards and restrict write access where not explicitly authorized.

Check File and Directory Details

To check the current file and directory permissions, I used the `ls -la` command inside the /home/researcher2/projects directory:

```
ls -la
```

This command displays a detailed list of files and directories, including hidden files (those starting with a dot .), along with their permissions, ownership, and timestamps.

Example output:

```
-rw-rw-rw- 1 researcher2 research_team 46 Apr 5 18:50 project_k.txt
```

Describe the Permissions String

Example: -rw-rw-rw-

This string consists of 10 characters and breaks down as follows:

- The first character (-) indicates a regular file (as opposed to a directory d).
- The next three (rw-) represent the user (owner) permissions: read and write.
- The next three (rw-) represent the group permissions: read and write.
- The final three (rw-) represent the other (everyone else) permissions: read and write.

In this case, project_k.txt was readable and writable by everyone, which violates the organization's policy. This needed to be fixed.

Change File Permissions

File Permissions in Linux

According to the policy, 'others' should not have write access. Two files needed updates:

1. project_k.txt

   To remove write permission from 'others':

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

2. project_m.txt

   To remove group execute and read permissions:

   ```
   chmod g-rx project_m.txt
   ```

Both commands are shown in the screenshots. The permission strings were successfully updated after running ls -l.

Change File Permissions on a Hidden File

The hidden file .project_x.txt had incorrect permissions. It was writable by the group and unreadable by others, which needed to be adjusted to read-only for user and group, and no permissions for others.

To apply this:

```
chmod u=r,g=r,o= .project_x.txt
```

Resulting permissions:

```
-r--r-- --- 1 researcher2 research_team 46 Apr 5 18:50 .project_x.txt
```

This ensures only the user and group can read it, and nobody can write or execute it.

Change Directory Permissions

The drafts directory should only be accessible by the user researcher2. Group and others should not have access. Initially, group had execute (x) permission.

To remove group execute access:

File Permissions in Linux

```
chmod g-x drafts
```

Updated directory permissions:

```
drwx--x--- 2 researcher2 research_team 4096 Apr 5 18:50 drafts
```

This ensures only the owner has full access, effectively locking out everyone else from reading or modifying contents.

Summary

In this project, I evaluated and modified file and directory permissions for a Linux system in a research-focused organization. I used ls -la to view permissions and applied chmod to enforce policies restricting write and execute access for groups and others. This exercise reinforced the importance of precise access control and provided hands-on experience with interpreting and securing Linux file systems, a critical skill in cybersecurity and system administration roles.