Current file permissions

This document displays the file structure of the `/home/researcher2/projects` directory and the permissions of the files and subdirectory it contains.

In the `/home/researcher2/projects` directory, there are five files with the following names and permissions:

- `project_k.txt`
  - User = read, write,
  - Group = read, write
  - Other = read, write
- `project_m.txt`
  - User = read, write
  - Group = read
  - Other = none
- `project_r.txt`
  - User= read, write
  - Group = read, write
  - Other = read
- `project_t.txt`
  - User = read, write
  - Group = read, write
  - Other = read
- `.project_x.txt`
  - User = read, write
  - Group = write
  - Other = none

There is also one subdirectory inside the `projects` directory named `drafts`. The permissions on `drafts` are:

- User = read, write, execute
- Group = execute
- Other = none

File permissions in Linux

Project description

This project focuses on configuring authorization using Linux commands. Participants will examine and manage file and directory permissions in the `/home/researcher2/projects` directory on a Linux system. The goal is to ensure that permissions are correctly set to restrict unauthorized access and maintain system security.

Check file and directory details

```
ls -lr /home/researcher2/projects
```

This command will recursively list all files and directories in `/home/researcher2/projects` displaying detailed information including permissions, ownership, size, and timestamp.

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

project_m.txt: -rw-r--r--

project_r.txt: -rw-rw-r--

project_t.txt: -rw-rw-r--

.project_x.txt: -rw-rw----

drafts directory: drwx--x---
```

These 10-character strings represent the permissions for each file and directory, respectively, where each triplet (rwx) denotes permissions for user, group, and others (read, write, execute)

Describe the permissions string

The 10-character string -rw-rw-r-- represents the permissions for the file project_k.txt in the Linux filesystem.

Explanation:

- The first character (-) indicates the type of the file. In this case, it's a regular file.
- Characters 2-4 (rw-) represent the permissions for the user (researcher2), where rw- means the user has read and write permissions, but not execute permissions.
- Characters 5-7 (rw-) denote the permissions for the group (research_team). Here, rw- indicates the group also has read and write permissions, but not execute permissions.
- Characters 8-10 (r--) specify the permissions for others (everyone else). r-- means others have only read permission, without write or execute permissions.
-

This permissions string (-rw-rw-r--) ensures that the file project_k.txt can be read and modified by the user researcher2 and members of the research_team group, while others can only read the file.

Change file permissions

project_m.txt currently allows write access (rw-) for others (everyone else), which violates the organization's policy of not allowing others to have write access to any files.
To modify the permissions of project_m.txt to remove write access for others, we use the chmod command in Linux.

```
chmod o-w /home/researcher2/projects/project_m.txt
```

The command chmod o-w /home/researcher2/projects/project_m.txt is used to modify the permissions of project_m.txt by removing write (-w) access for others (o). This ensures that only the user (rw-) and group (rw-) have read and write permissions, while others (r--) are restricted to only read access.

Change file permissions on a hidden file

To first find the hidden file, we can use a variation of the ls command.

```
ls -la
```

When combined (ls -la), this command will list all files and directories in the current directory (including hidden ones), showing detailed information for each.

Now assign the appropriate permissions to .project_x.txt where only the user and group have read access, and no write access is granted to anyone, we use the chmod command in Linux.

```
chmod ug=r,o= /home/researcher2/projects/.project_x.txt
```

The command chmod ug=r,o= /home/researcher2/projects/.project_x.txt is used to modify the permissions of .project_x.txt. Here's what each part of the command does:

- ug=r: Sets the permissions for the user (u) and group (g) to read (r) only.
- o=: Removes all permissions for others (o), ensuring they have no read, write, or execute permissions.

Change directory permissions

To modify the permissions of the drafts directory so that only the researcher2 user can access it, you can use the chmod command in Linux along with the appropriate options.

```
chmod g-rwx /home/researcher2/projects/drafts
chmod o-rwx /home/researcher2/projects/drafts
chmod u+rwx /home/researcher2/projects/drafts
```

These commands ensure that only the researcher2 user has full access to the drafts directory (rwx), while the group and others have no permissions (---). This setup aligns with the requirement to restrict access to only the owner (researcher2) of the directory.

Summary

In this project, we've used Linux commands like ls and chmod to effectively control file permissions. By reviewing and modifying permissions in /home/researcher2/projects, we've secured files such as .project_x.txt, ensuring that only approved users like researcher2 have

appropriate access. This approach strengthens data security by enforcing access controls that meet organizational standards.

Example of the terminal used: