File permissions in Linux

Project description

Our research team faced the critical task of updating file permissions within the projects directory. The existing permissions did not align with the required authorization levels, posing a security risk. My role was to ensure these permissions accurately reflected the necessary access controls.

Check file and directory details

Linux commands to determine existing permissions set for a specific directory in the file system:

```
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-rw--w---- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

Utilizing Linux commands, I assessed the current permissions for specific files and directories. This involved executing the ls -la command, which revealed a comprehensive list of all items in the projects directory, including hidden files. The output was crucial in understanding the permissions set on each item, displayed as a 10-character string.

Describe the permissions string

This string is a key to deciphering file access permissions. It comprises:

- The 1st character indicating the file type (a 'd' for directories, a '-' for files).
- Characters 2-4 representing the user's read (r), write (w), and execute (x) permissions.
- Characters 5-7 for group permissions.
- Characters 8-10 for other users.

For instance, '-rw-rw-r--' for a file signifies read and write permissions for the user and group, but only read for others, with no execute permissions for anyone.

Change file permissions

Addressing the need to restrict 'other' users from write access, I focused on modifying these permissions. I utilized the chmod command, carefully adjusting the access rights for specific files like 'project_k.txt', ensuring 'other' users could no longer modify them.

```
researcher2@5d738f0f927b:~/projects$ chmod o-w project_k.txt
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-rw--w---- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

Change file permissions on a hidden file

The team decided to restrict write access to the archived '.project_x.txt'. Again, I employed the chmod command, tailoring the permissions so that only read access was allowed for the user and group, while completely removing write privileges.

```
researcher2@3213bbc1d047:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@3213bbc1d047:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec 20 15:36 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec 20 15:36 ..
-r--r----- 1 researcher2 research_team   46 Dec 20 15:36 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec 20 15:36 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Dec 20 15:36 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec 20 15:36 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec 20 15:36 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec 20 15:36 project_t.txt
researcher2@3213bbc1d047:~/projects$
```

Change directory permissions

The requirement was exclusive access for the 'researcher2' user to the 'drafts' directory. My task involved removing execute permissions for all users except 'researcher2', ensuring controlled and secure access to this directory.

```
researcher2@5d738f0f927b:~/projects$ chmod g-x drafts
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-r--r----- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

Summary

This project entailed a meticulous process of evaluating and updating permissions using the ls -la and chmod commands. The objective was to realign the file and directory permissions with the organization's security policies, enhancing overall system security by carefully controlling access rights.