File permissions in Linux

Project description

The team at my organization's research department must revise the file permissions of specific files and directories within the projects directory. The existing permissions are not aligned with the intended level of authorization. Adjusting these permissions is essential for maintaining system security. In order to accomplish this, I carried out these commands

Check file and directory details

```
researcher2@465e95bb2ad9:~/projects$ ls
drafts  project_k.txt  project_m.txt  project_r.txt  project_t.txt
researcher2@465e95bb2ad9:~/projects$ ls -a
.   .project_x.txt  project_k.txt  project_r.txt
..  drafts          project_m.txt  project_t.txt
researcher2@465e95bb2ad9:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 17 22:14 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 17 22:14 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 17 22:14 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 17 22:14 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 17 22:14 project_t.txt
researcher2@465e95bb2ad9:~/projects$
```

After the command cd projects, I listed the files. To analyze the permissions I used ls with the option -1. This shows all files within the directory and each file's permissions.

Describe the permissions string

The 10-character string can be deconstructed to determine who is authorized to access the file and their specific permissions. The characters and what they represent are as follows:

The 1st character is either a d or hyphen and indicates the file type. If it's a d, it's a directory. If it's a hyphen, it's a regular file.

The 2nd-4th characters indicate the read, write, and execute permissions for the user.

The 5th-7th character indicates the read, write, and execute permissions for the group.

The 8th-10th characters indicate the read, write, and execute permissions for others.

When one of these characters is a hyphen, it indicates that this permission is not granted for the user, group, or other.

Change file permissions

```
researcher2@dcf9b50c7633:~$ cd projects/
researcher2@dcf9b50c7633:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 17 22:05 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 17 22:05 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 17 22:05 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 17 22:05 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 17 22:05 project_t.txt
researcher2@dcf9b50c7633:~/projects$ chmod o-w project_k.txt
researcher2@dcf9b50c7633:~/projects$ ls-l
-bash: ls-l: command not found
researcher2@dcf9b50c7633:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 17 22:05 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Oct 17 22:05 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 17 22:05 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 17 22:05 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 17 22:05 project_t.txt
researcher2@dcf9b50c7633:~/projects$
```

The organization does not want others to have write permissions. THis means the user and group permissions do not need to be adjusted. Only the file project_k.txt allows others to write. To change the permission, I run the following command chmod o-w project_k.txt. This eliminates writing permissions for others on the file.

Change file permissions on a hidden file

```
researcher2@dcf9b50c7633:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@dcf9b50c7633:~/projects$ ls -l .project_x.txt
-r--r----- 1 researcher2 research_team 46 Oct 17 22:05 .project_x.txt
researcher2@dcf9b50c7633:~/projects$
```

The organization has a hidden file that the user and group should only have read permissions. To do this I ran the command chmod u-w,g-w,g+r .project_x.txt. I removed write permissions from the user and group, and added read permissions to the group. Now the file only has read permissions for the user and the group.

Change directory permissions

```
researcher2@dcf9b50c7633:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 17 22:05 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Oct 17 22:05 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 17 22:05 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 17 22:05 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 17 22:05 project_t.txt
researcher2@dcf9b50c7633:~/projects$ chmod g-x drafts
researcher2@dcf9b50c7633:~/projects$ ls -l
total 20
drwx------ 2 researcher2 research_team 4096 Oct 17 22:05 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Oct 17 22:05 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 17 22:05 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 17 22:05 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 17 22:05 project_t.txt
researcher2@dcf9b50c7633:~/projects$
```

The organization only wants the user to have permissions for the drafts directory. I determined that the group had execute permissions. To change the permissions, I use the command chmod g-x drafts. The researcher2 user already had execute permissions, so they did not need to be added.

Summary

I adjusted various permissions to align with the desired authorization level set by my organization for files and directories within the projects directory. The initial action involved utilizing the 'ls -l' command to examine the directory's permissions, which guided my subsequent decisions. Subsequently, I employed the 'chmod' command repeatedly to modify permissions on both files and directories.