File permissions in Linux

Project description

The research team requires adjusted file permissions for specific files and directories within the "projects" directory, as the current settings do not align with the intended authorization levels. Correcting these permissions is crucial for maintaining system security. To achieve this, I undertook the following steps:

Check file and directory details

To check all file and directory detail in a specific directory, we should switch to the relevant directory with the cd command, followed by the directory path: 'cd /home/researcher2/projects'

```
researcher2@8e61093bf75a:~$ cd /home/researcher2/projects
researcher2@8e61093bf75a:~/projects$
```

To check the file and directory details, including hidden ones, I used the ls -la command

```
researcher2@e180b9b5bac5:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Apr 26 12:38 .
drwxr-xr-x 3 researcher2 research_team 4096 Apr 26 13:11 ..
-rw--w---- 1 researcher2 research_team   46 Apr 26 12:38 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Apr 26 12:38 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Apr 26 12:38 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Apr 26 12:38 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 26 12:38 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 26 12:38 project_t.txt
```

Hidden files can be seen starting with '.'
For example: '.project_x.txt'

Describe the permissions string

Let's choose the permission string of the first directory that appears in the example below:

```
drwxr-xr-x 3 researcher2 research_team 4096 Apr 26 12:38 .
```

Let me break it down for you:
d - means this is a directory
r - means the user have read permissions

w - means the user have write permissions
x - means the user have execute permissions
r - means the group have read permissions
- - means the group don't have write permissions
x - means the group have execute permissions
r - means the other have read permissions
- - means the other don't have write permissions
x - means the other have execute permissions

Change file permissions

Our organization doesn't allow others to have write access to any file.
In the example below, you can see a command that removes the write permissions from the file named project_k.txt. I used the following sentence to remove the write permission from 'other' on this file: 'chmod o-w project_k.txt'

```
researcher2@8e61093bf75a:~/projects$ chmod o-w project_k.txt
researcher2@8e61093bf75a:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Apr 26 12:40 .
drwxr-xr-x 3 researcher2 research_team 4096 Apr 26 13:33 ..
-rw--w---- 1 researcher2 research_team   46 Apr 26 12:40 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Apr 26 12:40 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Apr 26 12:40 project_k.txt
```

The chmod command, followed by the argument, enabled us to do this action.

Change file permissions on a hidden file

Our organization does not allow anyone to have write permission to classified documents such as lproject_x.txt. However, it does need this document to have a user and group to have read permissions. I change the .project_x.txt permissions according to the organization guidelines. This sentence will remove the write permissions from user and group and will add read permission to group: 'chmod u-w,g+r,g-w .project_x.txt'

```
researcher2@8e61093bf75a:~/projects$ chmod u-w,g+r,g-w .project_x.txt
researcher2@8e61093bf75a:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Apr 26 12:40 .
drwxr-xr-x 3 researcher2 research_team 4096 Apr 26 13:33 ..
-r--r----- 1 researcher2 research_team   46 Apr 26 12:40 .project_x.txt
```

Change directory permissions

Our organization allows only the user ‘researcher2’ to have any kind of access to the drafts directory.

```
researcher2@8e61093bf75a:~/projects$ chmod o-w project_k.txt
researcher2@8e61093bf75a:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Apr 26 12:40 .
drwxr-xr-x 3 researcher2 research_team 4096 Apr 26 13:33 ..
-rw--w---- 1 researcher2 research_team   46 Apr 26 12:40 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Apr 26 12:40 drafts
```

Currently the drafts directory has one access permission: The group owner type has execute permission. The following sentence will remove this excess permission:
‘chmod g-x drafts’

```
researcher2@8e61093bf75a:~/projects$ chmod g-x drafts
researcher2@8e61093bf75a:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Apr 26 12:40 .
drwxr-xr-x 3 researcher2 research_team 4096 Apr 26 13:33 ..
-r--r----- 1 researcher2 research_team   46 Apr 26 12:40 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Apr 26 12:40 drafts
```

Summary

I adjusted permissions for files and directories within the "projects" directory to align with my organization's authorization requirements. Initially, I utilized the "ls -la" command to assess existing permissions, which guided my subsequent actions. I then employed the "chmod" command iteratively to modify permissions as needed.