File permissions in Linux

Project description

The research team at my organization needs to update the file permissions for certain files and directories within the projects directory. The permissions do not currently reflect the level of authorization that should be given. Checking and updating these permissions will help keep their system secure. To complete this task, I performed the following tasks:

Check file and directory details

```
researcher2@71616e5f1e04:~$ cd projects
researcher2@71616e5f1e04:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug  1 03:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug  1 04:00 ..
-rw--w---- 1 researcher2 research_team   46 Aug  1 03:31 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug  1 03:31 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Aug  1 03:31 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug  1 03:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  1 03:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  1 03:31 project_t.txt
researcher2@71616e5f1e04:~/projects$
```

- The **first line** of the screenshot displays the command `cd` I entered to change directory to projects.
- The **second line** of the screenshot displays the command of the list of contents from the current directory that I have entered, and the other lines display the output. The code lists all contents of the projects directory. I used the `ls` command with the `-la` option to display a detailed listing of the file contents that also returned hidden files.

Describe the permissions string

Each character in (drwxrwxrwx) the 10-character string conveys different information about these permissions.

- The **1st** character indicates the file type. The d indicates it's a directory.
- The **2nd-4th** characters indicate the read (r), write (w), and execute (x) permissions for the **USER**.
- The **5th-7th** characters indicate the read (r), write (w), and execute (x) permissions for the **GROUP**.

- The **8th-10th** characters indicate the read (r), write (w), and execute (x) permissions for the owner type of **OTHER**. This owner type consists of all other users on the system apart from the user and the group.
- `(-)` if the other owner type lacks write permissions.

Change file permissions

The organization determined that others shouldn't have write access to any of their files. To comply with this, I referred to the file permissions that I previously returned.

```
researcher2@3560844c6dd5:~/projects$ chmod o-w project_k.txt
researcher2@3560844c6dd5:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug  1 04:17 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug  1 05:06 ..
-rw--w---- 1 researcher2 research_team   46 Aug  1 04:17 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug  1 04:17 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Aug  1 04:17 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug  1 04:17 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  1 04:17 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  1 04:17 project_t.txt
researcher2@3560844c6dd5:~/projects$
```

I determined **project_k.txt** must have the write access removed for other.
The others in **project_m.txt, project_r.txt, and project_t.txt** don't have permission to write. So it doesn't change.

Change file permissions on a hidden file

The research team has archived **.project_x.txt** which is why it's a hidden file. This file should not have write permissions for anyone, but the user and group should be able to read the file.

```
researcher2@3560844c6dd5:~/projects$ chmod u-w,g-w,o-w,u+r,g+r .project_x.txt
researcher2@3560844c6dd5:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug  1 04:17 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug  1 05:06 ..
-r--r----- 1 researcher2 research_team   46 Aug  1 04:17 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug  1 04:17 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Aug  1 04:17 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug  1 04:17 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  1 04:17 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  1 04:17 project_t.txt
researcher2@3560844c6dd5:~/projects$
```

I determined **.project_x.txt** must have the write access removed for all. But the user and group should be able to read the file.

Change directory permissions

My organization only wants the researcher2 user to have access to the drafts directory and its contents. This means that no one other than researcher2 should have execute permissions.

```
researcher2@3560844c6dd5:~/projects$ chmod g-x,o-x drafts
researcher2@3560844c6dd5:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug  1 04:17 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug  1 05:06 ..
-r--r----- 1 researcher2 research_team   46 Aug  1 04:17 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Aug  1 04:17 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Aug  1 04:17 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug  1 04:17 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  1 04:17 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  1 04:17 project_t.txt
researcher2@3560844c6dd5:~/projects$
```

Summary

I changed multiple permissions to match the level of authorization my organization wanted for files and directories in the projects directory.

1. The first step in this was using `ls -la` to check the permissions for the directory, this informed my decisions in the next steps.
2. I then used the `chmod` command multiple times to change the permissions on files and directories.