File permissions in Linux by Chidubem Onochie-Onyetenu

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 helps keep their system secure. To complete this task, I performed the following steps:

Check file and directory details

The following code demonstrates how I used Linux commands to determine the existing permissions set for a specific directory in the file system.

```
researcher2@076eb0275988:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 17 10:51 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 17 11:39 ..
-rw--w---- 1 researcher2 research_team   46 Aug 17 10:51 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug 17 10:51 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Aug 17 10:51 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 17 10:51 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 17 10:51 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 17 10:51 project_t.txt
researcher2@076eb0275988:~/projects$
```

The first line of the screenshot displays the command I entered, and the other lines display the output. The command lists all contents of the projects directory. I used the ls command with the -la option to display a detailed listing that also includes hidden files. The output shows one directory named drafts, one hidden file named .project_x.txt, and four other project files. The 10-character string in the first column shows the permissions set on each file or directory.

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:

- **1st character**: This character is either d or a hyphen (-) and indicates the file type. A d means the item is a directory, and a hyphen means it is a regular file.
- **2nd-4th characters**: These characters show the read (r), write (w), and execute (x) permissions for the user. A hyphen (-) in place of one of these characters means that permission is not granted to the user.

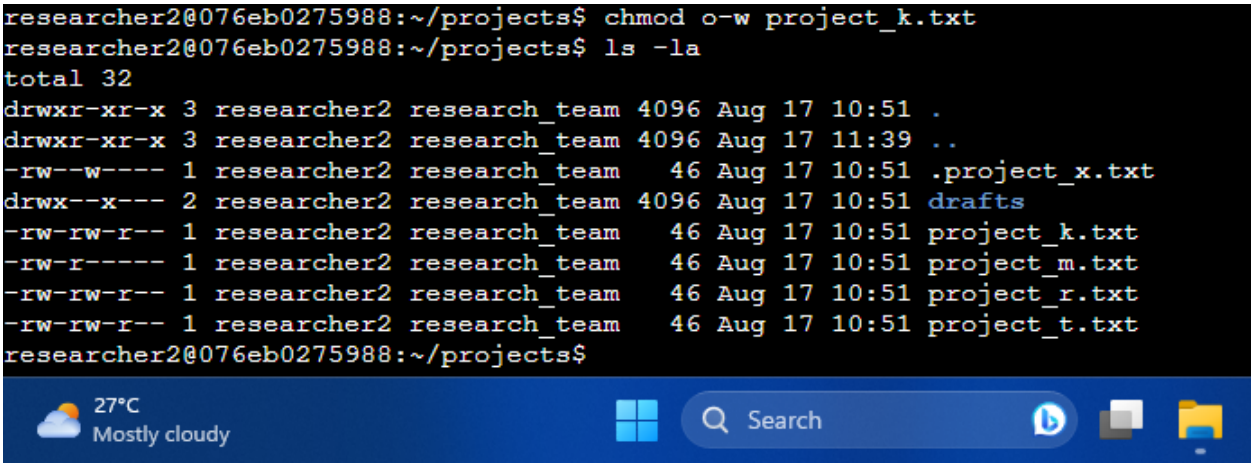
- **5th-7th characters**: These characters show the read (r), write (w), and execute (x) permissions for the group. A hyphen (-) in place of one of these characters means that permission is not granted to the group.
- **8th-10th characters**: These characters show the read (r), write (w), and execute (x) permissions for other. Other consists of all users on the system apart from the user and the group. A hyphen (-) in place of one of these characters means that permission is not granted to other.

For example, the permissions for project_t.txt are -rw-rw-r--. Since the first character is a hyphen (-), project_t.txt is a file, not a directory. The second, fifth, and eighth characters are all r, which means the user, group, and other all have read permission. The third and sixth characters are w, which means only the user and group have write permission. No one has execute permission for project_t.txt.

Change file permissions

The organization determined that other should not have write access to any of its files. To comply, I referred to the file permissions I had previously returned and determined that project_k.txt must have write access removed for other.

The following code demonstrates how I used Linux commands to do this:



The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. The chmod command changes the permissions on files and directories. Its first argument specifies which permissions to change, and its second argument specifies the file or directory. Here, I used o-w to remove write permission from other for project_k.txt. I then used ls -la to review the update.

Change file permissions on a hidden file

The research team at my organization recently archived project_x.txt. They do not want anyone to have write access to it, but the user and group should have read access.

The following code demonstrates how I used Linux commands to change the permissions:

```
researcher2@076eb0275988:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@076eb0275988:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 17 10:51 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 17 11:39 ..
-r--r----- 1 researcher2 research_team   46 Aug 17 10:51 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug 17 10:51 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Aug 17 10:51 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 17 10:51 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 17 10:51 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 17 10:51 project_t.txt
researcher2@076eb0275988:~/projects$
```



The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. I know .project_x.txt is a hidden file because its name starts with a period (.). In this example, I removed write permission from the user with u-w, removed write permission from the group with g-w, and added read permission to the group with g+r. The permissions changed from -rw--w---- to -r--r-----.

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 permission.

The following code demonstrates how I used Linux commands to change the permissions:

```
researcher2@076eb0275988:~/projects$ chmod g-x drafts
researcher2@076eb0275988:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 17 10:51 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 17 11:39 ..
-r--r----- 1 researcher2 research_team   46 Aug 17 10:51 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Aug 17 10:51 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Aug 17 10:51 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 17 10:51 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 17 10:51 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 17 10:51 project_t.txt
researcher2@076eb0275988:~/projects$
```



The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. I had previously determined that the group had execute permission on drafts, so I used chmod with g-x to remove it. The researcher2 user already had execute permission, so it did not need to be added. The permissions changed from drwx--x--- to drwx------.

Summary

I changed multiple permissions to match the level of authorization my organization wanted for files and directories in the projects directory. First, I used ls -la to check the existing permissions, which informed my decisions in the following steps. I then used the chmod command to change the permissions on project_k.txt, .project_x.txt, and the drafts directory.