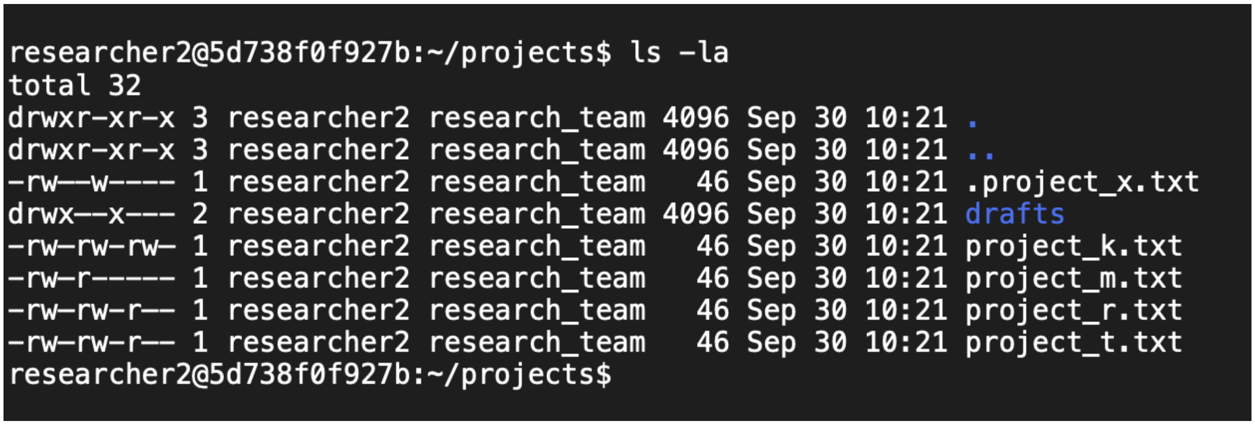
File permissions in Linux

Project description

The research team at my organization needs to revise the file permissions for specific files and directories within the projects folder. The current permissions do not match the appropriate authorization levels. Reviewing and adjusting these permissions will help maintain the security of their system.

Check file and directory details

Here is my code that demonstrates how I used Linux commands to know the 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 Sep 30 10:21 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep 30 10:21 ..
-rw--w---- 1 researcher2 research_team   46 Sep 30 10:21 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep 30 10:21 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Sep 30 10:21 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 30 10:21 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 30 10:21 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 30 10:21 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

The first line of the screenshot shows the command I entered, and the remaining lines show the output. The command lists all items within the projects directory. I used the ls command with the -la option to produce a detailed listing, including hidden files. The output indicates that the directory contains one folder named drafts, one hidden file called .project_x.txt, and five additional project files. The 10-character string in the first column represents the permission settings for each file or directory.

Describe the permissions string

The 10-character string can be broken down to identify who has access to the file and what specific permissions each group is granted. The characters and their corresponding meanings are listed below.

- **1st character**: This 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.

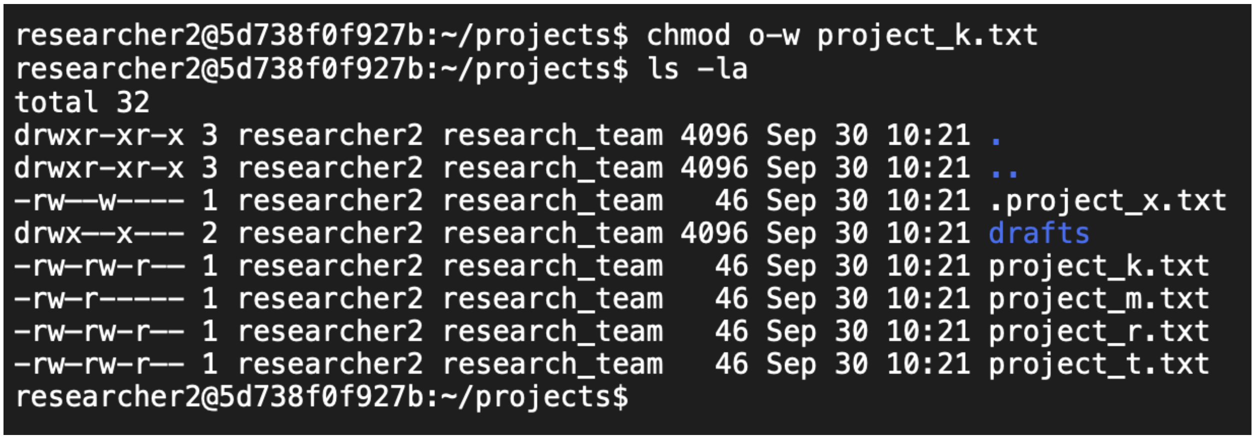
- **2nd-4th characters**: These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for the user. When one of these characters is a hyphen (`-`) instead, it indicates that this permission is not granted to the user.
- **5th-7th characters:** These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for the group. When one of these characters is a hyphen (`-`) instead, it indicates that this permission is not granted for the group.
- **8th-10th characters:** These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for other. This owner type consists of all other users on the system apart from the user and the group. When one of these characters is a hyphen (`-`) instead, that indicates that this permission is not granted for other.

In this case, the file permissions for project_t.txt are -rw-rw-r--. The leading hyphen (-) indicates that it is a regular file rather than a directory. The second, fifth, and eighth characters are r, meaning the user, group, and others all have read permissions. The third and sixth characters are w, showing that only the user and group have write permissions. No execute permissions are granted to any category for project_t.txt."

Change file permissions

The organization decided that 'other' should not have write access to any of its files. To meet this requirement, I reviewed the previously listed file permissions and identified that project_k.txt needed its write permission removed for the 'other' category.

Here is my code that shows how I used Linux commands to do this:



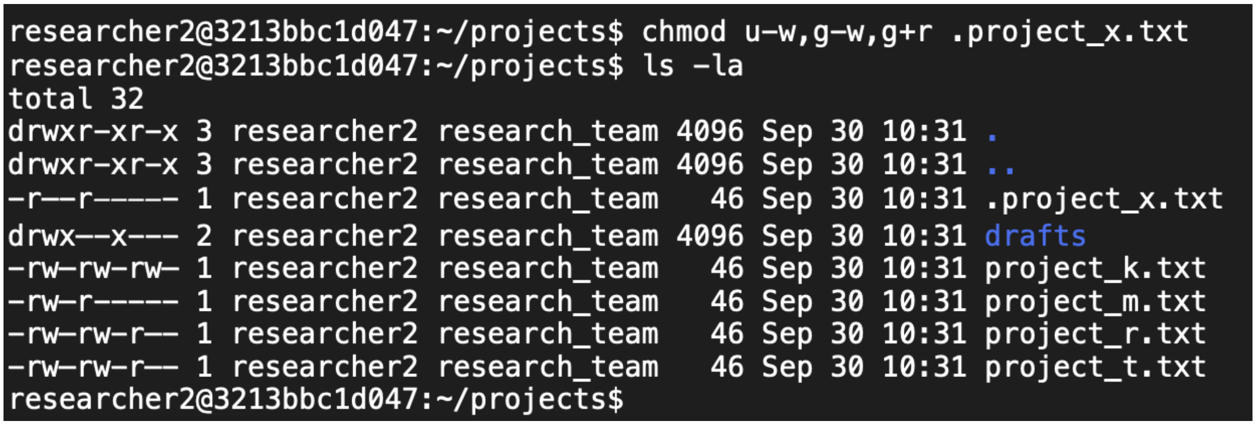
```
researcher2@5d738f0f927b:~/projects$ chmod o-w project_k.txt
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep 30 10:21 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep 30 10:21 ..
-rw--w---- 1 researcher2 research_team   46 Sep 30 10:21 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep 30 10:21 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Sep 30 10:21 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 30 10:21 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 30 10:21 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 30 10:21 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

The first two lines of my code show the commands that I entered, followed by the output of the second command. The chmod command is used to modify permissions on files and directories—the first argument specifies which permissions to change, and the second identifies the file or directory. In this case, I removed write permissions from the 'other' category for project_k.txt. After applying the change, I ran ls -la again to confirm the updated permissions.

Change file permissions on a hidden file

The research team at my organization recently archived project_x.txt. They want to ensure that no one has write access to this file, while still allowing the user and group to retain read access.

Here is my code showing how I used Linux commands to change the permissions:



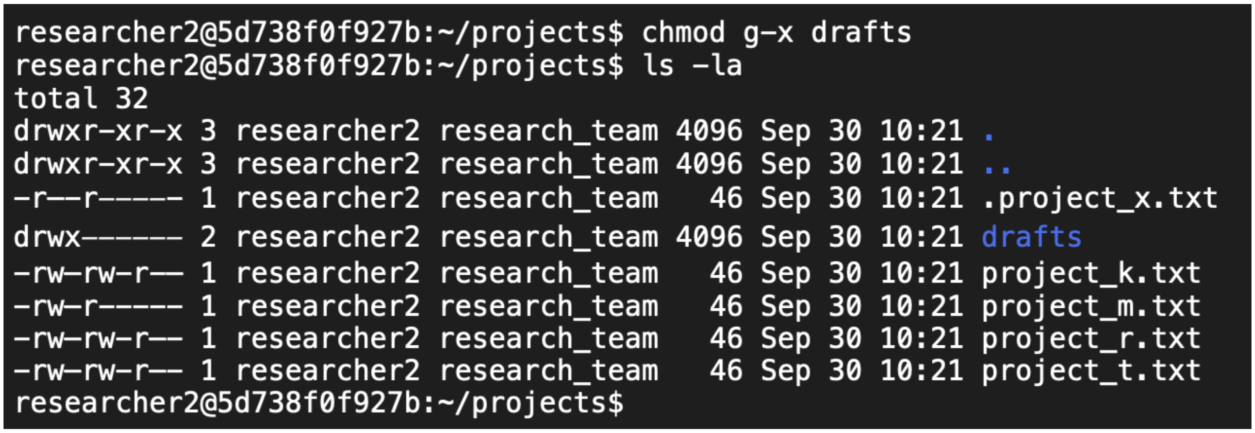
```
researcher2@3213bbc1d047:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@3213bbc1d047:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep 30 10:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep 30 10:31 ..
-r--r----- 1 researcher2 research_team   46 Sep 30 10:31 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep 30 10:31 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Sep 30 10:31 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 30 10:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 30 10:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 30 10:31 project_t.txt
researcher2@3213bbc1d047:~/projects$
```

The first two lines of the code show the commands I entered, and the remaining lines show the output of the second command. I know that .project_x.txt is a hidden file because its name begins with a period (.). In this example, I removed write permissions from both the user and the group, and then added read permissions for the group. I removed the user's write access using u-w, removed the group's write access with g-w, and granted the group read access using g+r.

Change directory permissions

My organization requires that only the researcher2 user has access to the drafts directory and everything inside it. This means that no users other than researcher2 should have execute permissions for that directory.

Here is my code demonstrating how I used Linux commands to change the permissions:



```
researcher2@5d738f0f927b:~/projects$ chmod g-x drafts
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep 30 10:21 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep 30 10:21 ..
-r--r----- 1 researcher2 research_team   46 Sep 30 10:21 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Sep 30 10:21 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Sep 30 10:21 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 30 10:21 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 30 10:21 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 30 10:21 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

The output shown lists the permissions for several files and directories. Line 1 represents the current directory/projects, and line 2 represents the parent directory (home). Line 3 shows a regular file named .project_x.txt. Line 4 displays the drafts directory, which now has restricted permissions. Here, only researcher2 has execute permissions. Since the group previously had execute access, I used the chmod command to remove it. The researcher2 user already had execute permissions, so no additional changes were required.

Summary

I updated several permissions to align with the authorization levels my organization required for the files and directories within the projects directory. I began by using ls -la to review the existing permissions, which guided the changes I needed to make. I then used the chmod command multiple times to adjust the permissions accordingly.