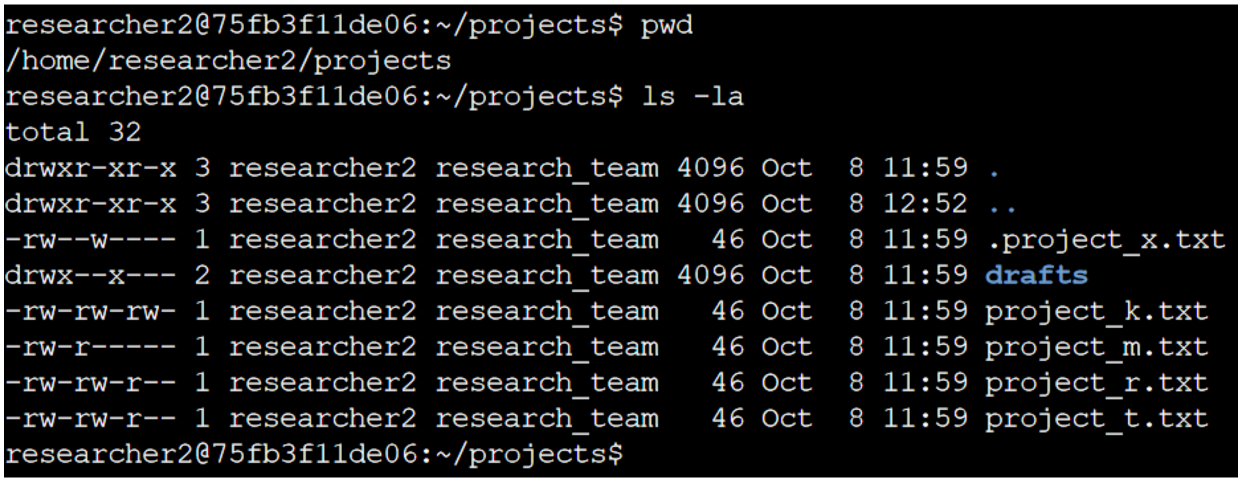
File permissions in Linux

Project Description

The research_team at my organization needs to update the file permissions for certain files and directories within the projects directory of user researcher2. The permissions do not currently reflect the level of authorization that shoul 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@75fb3f11de06:~/projects$ pwd
/home/researcher2/projects
researcher2@75fb3f11de06:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct  8 11:59 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct  8 12:52 ..
-rw--w---- 1 researcher2 research_team   46 Oct  8 11:59 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct  8 11:59 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct  8 11:59 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct  8 11:59 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct  8 11:59 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct  8 11:59 project_t.txt
researcher2@75fb3f11de06:~/projects$
```

The first line of the screenshot confirms that I verified the correct file path of the current directory using pwd. Following that is the command I executed, while the remaining lines display the output.

The code lists all the contents of the projects directory. I used the ls command with the -la option to provide a detailed listing, which includes hidden files. The output reveals:

- one directory named drafts
- a hidden file called .project_x.txt and
- 5 additional project files.

The 10-character string in the first column represents the permission settings for each file or directory.

Describe the permission 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 is either a d or a hypen (-) and indicates the **file type**. d if it's a directory, - if it's a regular file
- 2nd- 4th characters: These indicate the read (r), write (w) and execute (x) permissions for the **user.**
- 5th- 7th characters: These indicate the read (r), write (w) and execute (x) permissions for the **group.**
- 8th-10th characters: These indicate the read (r), write (w) and execute (x) permissions for the **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 hypen (-) instead, it indicates that this permission is not granted.

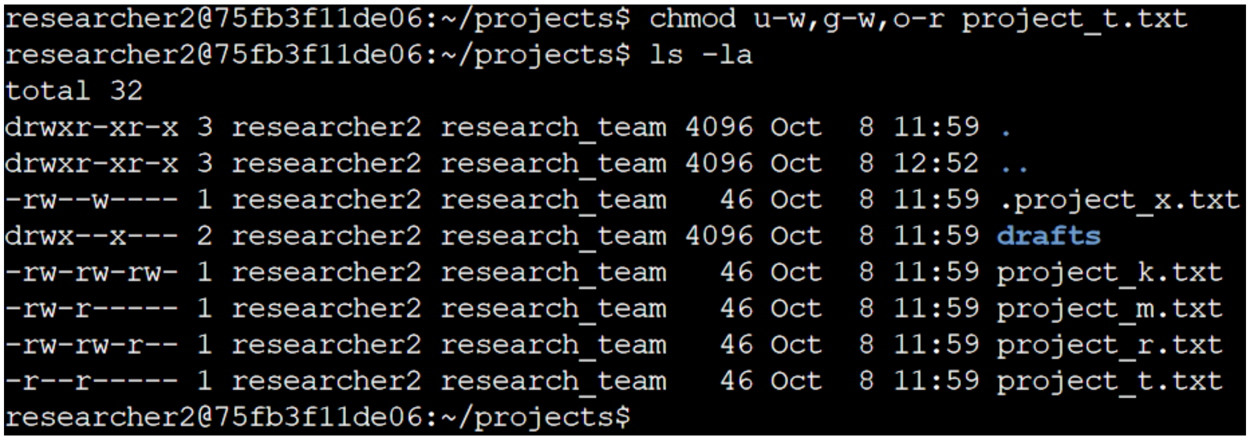
For example, the file permissions for project_t.txt are -rw-rw-r--

- 1st character: The hypen (-) indicates it's a regular file, not a directory
- 2nd- 4th characters: rw- indicate the user have read and write access
- 5th- 7th characters: rw- indicate the group have read and write access
- 8th- 10th characters: r-- indicate the other have read access only

  No one has execute permissions for project_t.txt

Change file permissions

The research team at my organization recently archived project_t.txt. They do not want anyone to have write access to this project, but the user and group should have read access. The following code demonstrates how I used Linux commands to change the permissions from -rw-rw-r-- to -r--r----- using chmod u-w,g-w,o-r projects_t.txt



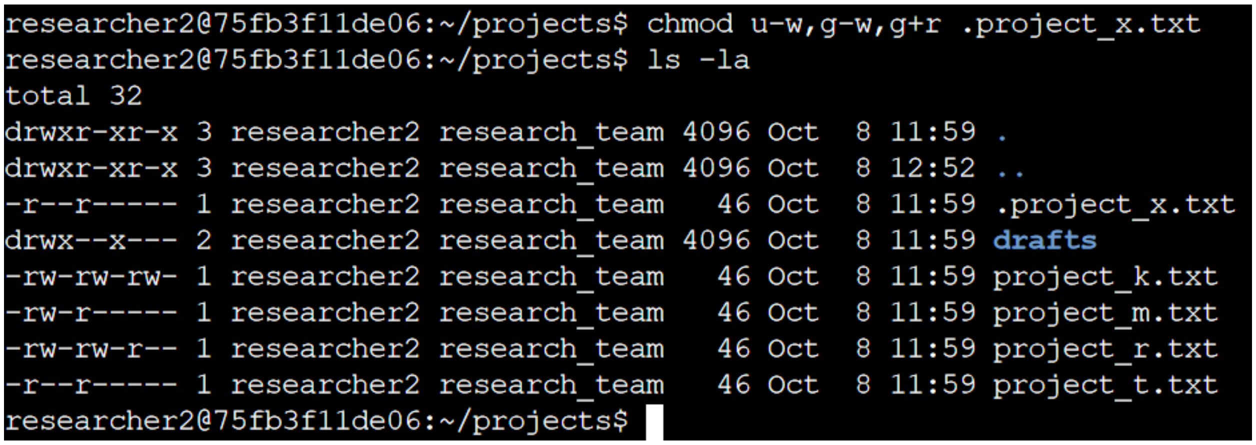
```
researcher2@75fb3f11de06:~/projects$ chmod u-w,g-w,o-r project_t.txt
researcher2@75fb3f11de06:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct  8 11:59 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct  8 12:52 ..
-rw--w---- 1 researcher2 research_team   46 Oct  8 11:59 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct  8 11:59 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct  8 11:59 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct  8 11:59 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct  8 11:59 project_r.txt
-r--r----- 1 researcher2 research_team   46 Oct  8 11:59 project_t.txt
researcher2@75fb3f11de06:~/projects$
```

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. The first argument indicates what permissions should be changed, and the second argument specifies the file or directory.

In this example, I removed write access from the user and the group for the project_t.txt file, leaving only read access. I removed write permissions from the user with u-w. Then, I removed write permissions from the group with g-w and removed read permissions to the other with o-r. The comma separates the different changes applied to each category of users. After this, I used ls -la to verify the updates I made.

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 this project, but the user and group should have read access. The following code demonstrates how I used Linux commands to change the permissions from -rw--w---- to -r--r----- using chmod u-w,g-w,g+r .projects_x.txt

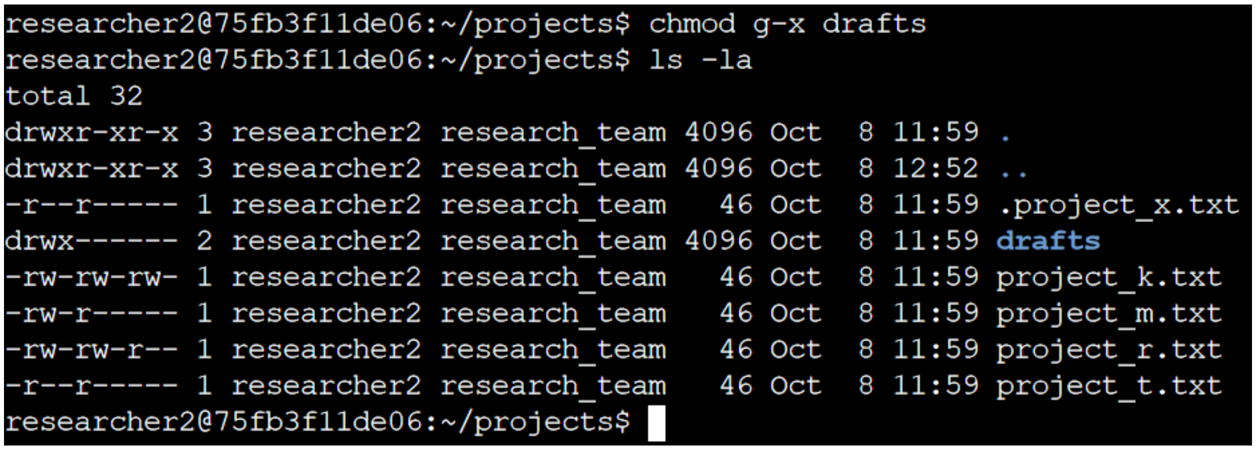


```
researcher2@75fb3f11de06:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@75fb3f11de06:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct  8 11:59 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct  8 12:52 ..
-r--r----- 1 researcher2 research_team   46 Oct  8 11:59 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct  8 11:59 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct  8 11:59 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct  8 11:59 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct  8 11:59 project_r.txt
-r--r----- 1 researcher2 research_team   46 Oct  8 11:59 project_t.txt
researcher2@75fb3f11de06:~/projects$
```

In this example, I know .project_x.txt is a hidden file because it starts with a period (.). I removed write permissions from the user with u-w. Then, I removed write permissions from the group with g-w, and added read permissions to the group with g+r. After this, I used ls -la to verify the updates I made.

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. The following code demonstrates how I used Linux commands to change the permissions from drwx--x--- to drwx------ using chmod g-x drafts



```
researcher2@75fb3f11de06:~/projects$ chmod g-x drafts
researcher2@75fb3f11de06:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct  8 11:59 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct  8 12:52 ..
-r--r----- 1 researcher2 research_team   46 Oct  8 11:59 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Oct  8 11:59 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct  8 11:59 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct  8 11:59 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct  8 11:59 project_r.txt
-r--r----- 1 researcher2 research_team   46 Oct  8 11:59 project_t.txt
researcher2@75fb3f11de06:~/projects$ 
```

In this example, the 1st character d indicates drafts is a directory. I removed execute permissions from the group with g-x . This allows only the user to have read, write and execute access. After this, I used ls -la to verify the updates I made.

Summary

I changed multiple permissions to match the level of authorization for the research_team at my organization within the projects directory of user researcher2. The first step in this was using ls -la to review the permissions for the directory. This informed my decisions in the following steps. I then used the chmod command multiple times to change the read, write and execute permissions on requested files and directories to each category of users.