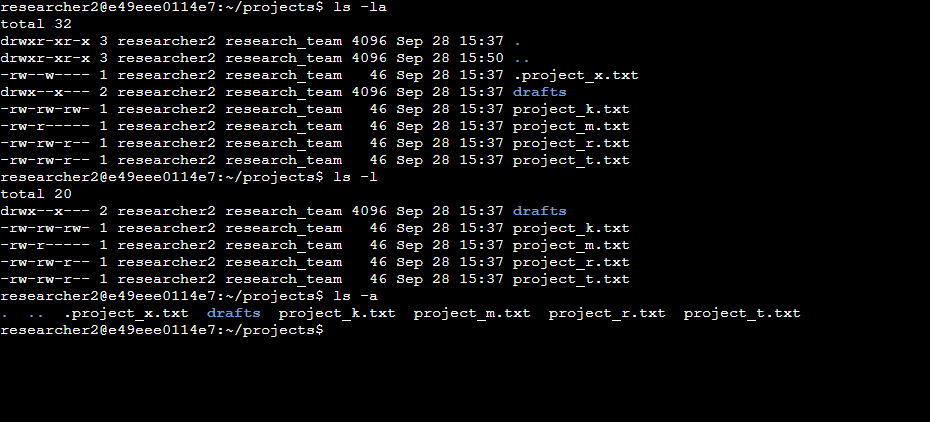
# File permissions in Linux

## Project description

The projects directory needs to be updated, the research team at my organization needs to update the file permissions for certain files and directories that do not reflect the level of authorization they should be given. Checking and updating the permissions to follow the minimum access policy helps keep the system safe. To complete this task, I performed the following tasks:

## 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.



The first line of code of the screenshot shows the command I entered ls -la to show the active files and the directories in the projects directory. ls -la to check files and directories details including the hidden files .project\_x.txt, ls -l returns permission for regular files and directories while ls -a returns permission for hidden files and directories. ls -la combines both commands to display all files and directories including the hidden ones as shown.

## Describe the permissions string

Permission strings are 10-character strings that can be deconstructed to determine who is authorized to access a file or directory and their specific permissions. The characters and what they represent are as follows:

* **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.

Choosing

-rw-rw-r-- 1 researcher2 research\_team 46 Sep 27 21:18 project\_r.txt as an example, -rw-rw-r-- is the permission string for the project\_r.txt .

The first character is a dash (-) which shows project\_r.txt is not a directory, the second, fifth and eighth characters are (r), shows user, group and other have read permissions. The third and sixth character is (w) which means user and group both have write permissions while the ninth character is a (-) which means the other does not have the write permission. The fourth, seventh and tenth characters all have a (-) meaning that none of them have the execute permission.

## Change file permissions

The company did not want any of the other to have write permissions. I referred to permissions displays and identified project\_k.txt other to have a write permission, this must be changed to comply with the company policy.

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

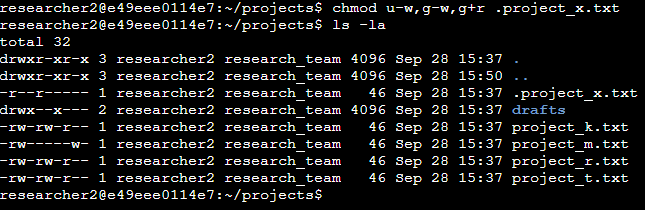
## 

The first line of the screenshot shows the command chmod o-w project\_k.txt which I used to change other write permission for project\_k.txt. The second line shows the command ls -la that displays the new updated details for the files and directories, shown in the other lines. 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 situation, I removed the write permission for other.

## Change file permissions on a hidden file

Project\_x.txt was recently archived by the research team and they do not want anyone to be able to edit this project. Write permission must be changed to achieve this task.

The following code demonstrates how I used Linux commands to change the hidden files permission:



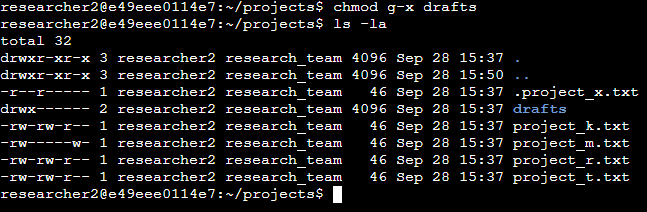
The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command ls -la. I know .project\_x.txt is a hidden file because it starts with a period (.). In this example, I removed write permissions from the user and group, and added read permissions to the group. 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.

All these changes with the command; chmod u-w,g-w,g+r .project\_x.txt

## Change directory permissions

The organization only wants the researcher2 user to have access to the drafts directory and its contents. This means no one else should have execute permission other than researcher2.

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



The first line is the command I entered chmod g-x drafts. This command removed the group execute permission leaving only the user with as the only one with the execute permission. . The researcher2 user already had execute permissions, so it did not need to be added.

## Summary

I changed multiple permissions to match the level of authorization my organization wanted for files and directories in the projects directory, these changes helped harden the organization's security. The first step was using ls -la to check the permissions for the directory. I then used the chmod command multiple times to change the permissions on files and directories in it as needed.