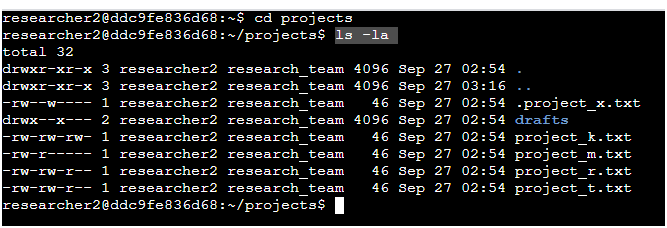
# File permissions in Linux

## Project description

In a laboratory activity, I was tasked to manage authorization on the specific resources in a system using Linux commands for configuration. I examined and managed the permissions on the files in the **/home/researcher2/projects** directory for the user files named **researcher2**. This step is essential for protecting sensitive information and maintaining the overall security of a system, which is one of the works of a security analyst.

## Check file and directory details

The first step I did was to explore the permissions on the directory assigned to me, which is named **projects**. I navigated to this directory, and typed some Linux commands for me to determine the contents and permissions on each file under the **projects** directory.



## Describe the permissions string

As we can see on the screenshot above, there are 10-character string in each of the entry. Those characters indicate the permissions status on each of the file. For a directory with full permissions for the owner, group and outsiders, the string combination will be **drwxrwxrwx.**

* The 1st character indicates the file type. The d indicates it’s a directory. When this character is a hyphen (-), it's a regular file.
* The 2nd-4th 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.
* The 5th-7th 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.
* 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. When one of these characters is a hyphen (-) instead, that indicates that this permission is not granted for other.

The second block of text in the expanded directory listing is the user who owns the file. The third block of text is the group owner of the file. In this case, the third block of text shows **research\_team**, which group is the owner of the files. Also, the files can be identified as hidden if the file name has a **(.).** In this case, the .project\_x.txt is one of the hidden files.

## Change file permissions

There were files that have incorrect permissions and needs changes. This action removes possibility of unauthorized access, and it strengthens the security on a system.



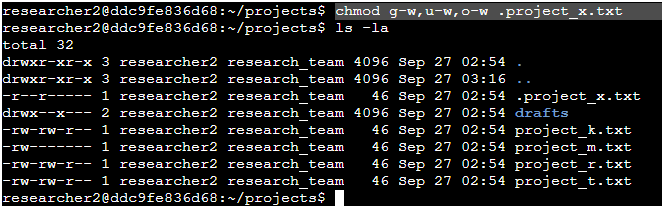
I found out that one of the files granted write permissions to others. I utilized the Linux command above to change the permission so that only the owner and the group member will have the write permission access to the file. The chmod command is for changing the permissions. There are three letters that can be utilized depending on the target type of owners: **u is for the owner of the file, g is for the group or team members, and o for others or outside the organization**.



For changing the permission settings on the **project\_m.txt file** that should be restricted and should not be readable or writable by the group or others, I utilized the command above to remove the current reading permission on the **project\_m.txt file**.

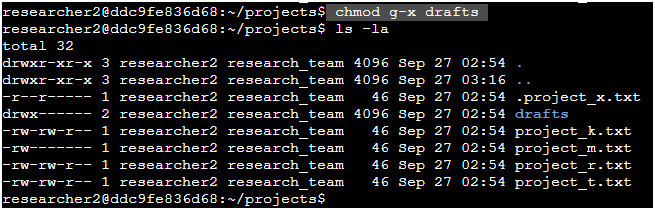
## Change file permissions on a hidden file

The **.project\_x.txt** is a hidden file that grants read and write permission to the main owner, but write only to the group. To fix the permission access, I utilized the command below so that both the user and the group can now read, but not write, to the file.



## Change directory permissions

I changed the permissions on **/home/researcher2/projects/drafts** directory. I ensured that only the **researcher2** user has the access on the drafts directory, especially on its contents. There was an execute permission for the group from the drafts directory. I utilized the Linux command below to remove that permission.



## Summary

By utilizing basic Linux Bash shell commands, I was able to examine file and directory permissions, change permissions on files, and change permissions on directories. I therefore conclude that mastering this foundational concepts is a must for any aspiring security analyst like me to manage authorization of the files inside any system. Basic commands such as **ls -la** for checking the permission status of a certain directory, **chmod** for changing the permissions even in multiple times for maintaining the security and accessibility of files and directories, and any other basic commands are fully utilized on the laboratory activity.