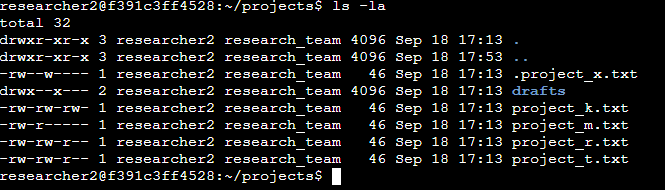
# File permissions in Linux

## Project description

I am a security professional responsible for ensuring the research team at my organization has the correct permissions on the file system. My task is to review the current permissions, verify that they align with the appropriate authorization, and modify them if necessary to grant access to authorized users and remove any unauthorized access to maintain system security.

## Check file and directory details

I first listed all the files and their permissions in my current directory, with the following command *ls -la*.



As seen in the screenshot there are 4 files, 1 directory **drafts** and 1 secret file **.project\_x.txt**. The first 10 characters in the output is what i'm looking for, this represents the permissions in each file/directory.

## Describe the permissions string

The **1st** character describes whether the listed item is a file or a directory if its a **d** then the item is a directory if its a hyphen (**-**)then the item is a file.

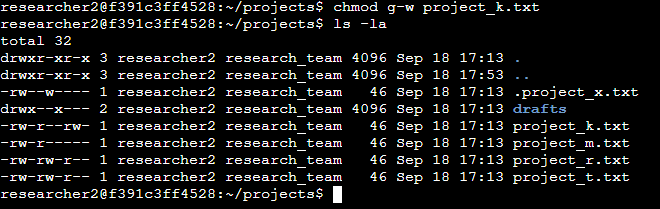
The **2nd** to **4th** characters represent the permissions the user has for this item. If they are "**rwx**" it means the user has **read**, **write**, and **execute** permissions for that item. If any of them is replaced by a hyphen (**-**) then that permission is not granted to the user. Ex. “**rw-**” this means the user has **read** and **write** permissions but not **execute** permissions.

The **5th** to **7th** characters are very similar to the previous one but they refer to group permissions instead.

Finally the **8th** to **10th** characters signify the same as the previous one but for the owner permissions.

## Change file permissions

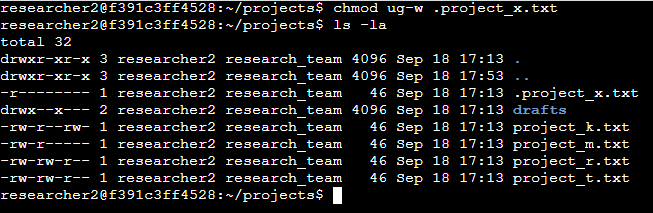
The organization determined that the group should not have writing permissions over **project\_k.txt** To comply I used the following command *chmod g-w project\_k.txt.*



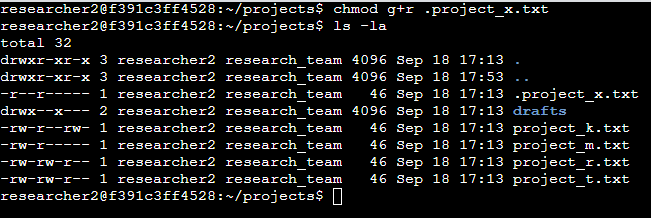
As you can see the **project\_k.txt** no longer has writing permissions for the group.

## Change file permissions on a hidden file

The organization decided to archive **.project\_x.txt** they have stated that they do not want anyone to have writing permissions but they still want the user and the group to have reading permissions. To do this I used two commands the first was *chmod ug-w .project\_x.txt*



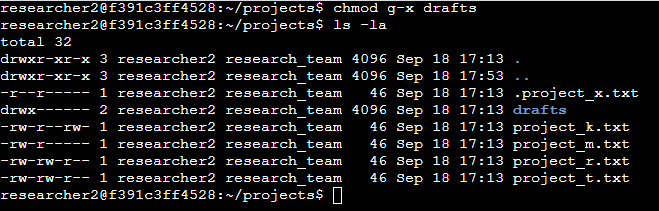
This removed the writing permissions for both the user and the group. I then used *chmod g+r*.



This command added reading permissions to the group.

## Change directory permissions

Finally my last task was to give only researcher2 (the user) execute permissions over the **drafts** directory. This just means that I need to remove the execute permissions of the group. To do this is simply used the following *chmod g-x drafts*.



As you can now see only the user has the execute permissions available.

## Summary

I listed all the files and permissions in my current directory using *ls -la*. The first 10 characters in the output represent permissions, with the 1st character indicating if the item is a file or directory, and the 2nd to 10th characters showing the user, group, and owner permissions.

I modified permissions for several items:

1. **project\_k.txt**: Removed group write permissions with  *chmod g-w project\_k.txt*.
2. **.project\_x.txt**: Archived the file by removing write permissions for both the user and group with *chmod ug-w .project\_x.txt*, then added group read permissions with *chmod g+r*.
3. **drafts directory**: Removed execute permissions from the group, leaving only the user with execute access, using *chmod g-x drafts*.