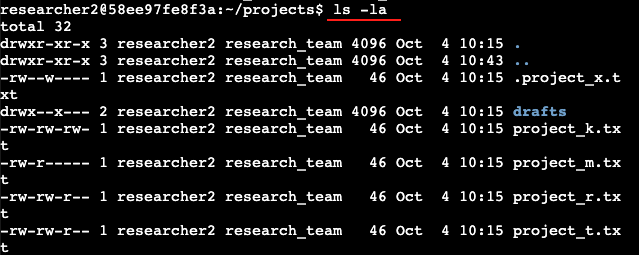
# File permissions in Linux

## Project description

The analyst task here was to examine existing permissions on the file system. The cybersecurity analyst needed to determine if the permissions match the authorization that should be given. If they do not match, the analyst has to modify the permissions to authorize the appropriate users and remove any unauthorized access.

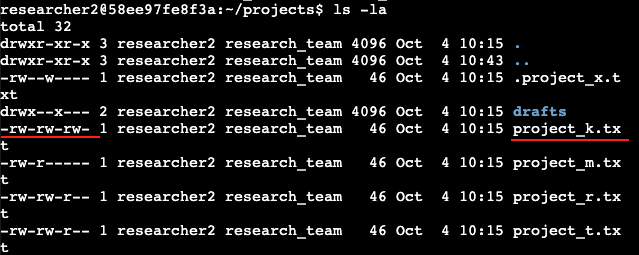
## Check file and directory details

First we started by checking the permissions set for files and subdirectories in the **projects** directory. Making sure we display all permissions, including hidden files using the **ls -la** command.



## Describe the permissions string

Next we identified the various permissions on the **project\_k.txt** file and we found out that the user, the group and others all had authorization to read and write on the file but none had the permission to execute it.



Permission **project\_k.txt**

○ User = read, write,

○ Group = read, write

○ Other = read, write

Describe the permissions 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 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.

For example, the file permissions for project\_t.txt are -rw-rw-r--. Since the first

character is a hyphen (-), this indicates that project\_t.txt is a file, not a directory. The

second, fifth, and eighth characters are all r, which indicates that user, group, and other all have

read permissions. The third and sixth characters are w, which indicates that only the user and group have write permissions. No one has executed permissions for project\_t.txt.

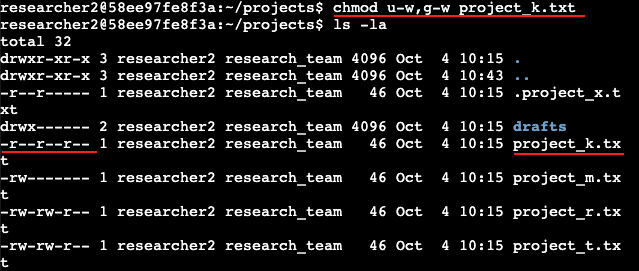
| **Character** | **Example** | **Meaning** |
| --- | --- | --- |
| 1st | **d**rwxrwxrwx | file type   * **d** for directory * **-** for a regular file |
| 2nd | d**r**wxrwxrwx | read permissions for the user   * **r** if the user has read permissions * **-** if the user lacks read permissions |
| 3rd | dr**w**xrwxrwx | write permissions for the user   * **w** if the user has write permissions * **-** if the user lacks write permissions |
| 4th | drw**x**rwxrwx | execute permissions for the user   * **x** if the user has execute permissions * **-** if the user lacks execute permissions |
| 5th | drwx**r**wxrwx | read permissions for the group   * **r** if the group has read permissions * **-** if the group lacks read permissions |
| 6th | drwxr**w**xrwx | write permissions for the group   * **w** if the group has write permissions * **-** if the group lacks write permissions |
| 7th | drwxrw**x**rwx | execute permissions for the group   * **x** if the group has execute permissions * **-** if the group lacks execute permissions |
| 8th | drwxrwx**r**wx | read permissions for other   * **r** if the other owner type has read permissions * **-** if the other owner type lacks read permissions |
| 9th | drwxrwxr**w**x | write permissions for other   * **w** if the other owner type has write permissions * **-** if the other owner type lacks write permissions |
| 10th | drwxrwxrw**x** | execute permissions for other   * **x** if the other owner type has execute permissions * **-** if the other owner type lacks execute permissions |

## Change file permissions

The organization does not allow **other** to have write access to **project\_k.txt** files. Based on the permissions established above, the analyst identified which file needs to have its permissions modified.

Using a Linux **chmod** command to modify these permissions, the analyst removed the user and group writing permissions in **project\_k.txt** file with the command .

**chmod u-w,g-w project\_k.txt**



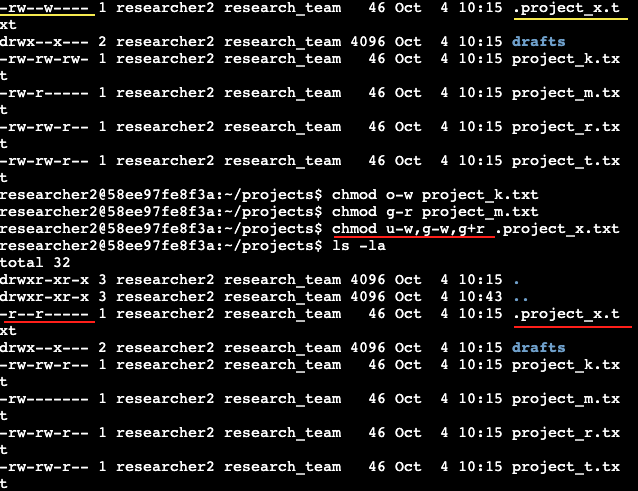
After this it can be observed that user, group and other ONLY have reading authorization to the **project\_k.txt** file as required by the organization.

## Change file permissions on a hidden file

The organization has archived **.project\_x.txt**, which is why it’s a hidden file. The organization also stipulates that the research team file should not have write permissions for anyone, but the user and group should be able to read the file.

Using a Linux **chmod** command to modify these permissions, the analyst removed the user and group writing permissions and added group reading permissions to the hidden file.

**chmod u-w,g-w,g+r .project\_x.txt**



It can now be observed that the research team file no longer has write permissions for anyone, but the user and group are able to read the file.

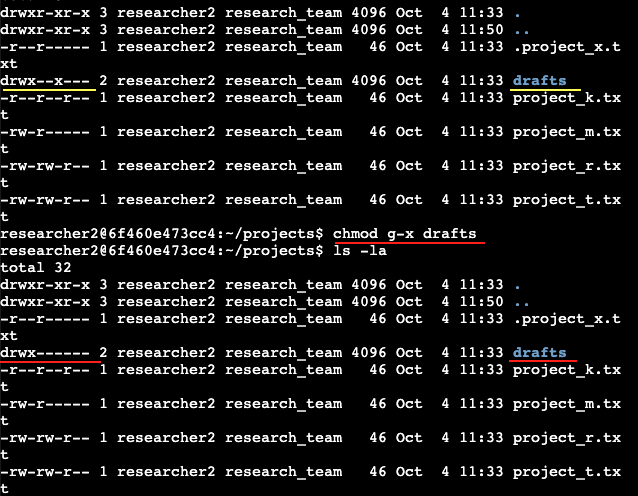
## Change directory permissions

The organization also stipulates that given that the files and directories in the projects directory belong to the **researcher2** user. Only **researcher2** should be allowed to access the **drafts** directory and its contents. (This means that only **researcher2** should have execute privileges.)

As such the analyst checked the permissions of the drafts directory and discovered that the group has permissions set to access the drafts directory and its contents.

So the analyst used Linux commands to modify the permissions accordingly.

**chmod g-x drafts**



It can be observed that now, only **researcher2**  is allowed to access the **drafts** directory have execute privileges

## Summary

These tasks demonstrate the practical application of cybersecurity measures related to file and directory permissions, ensuring that access to resources is aligned with organizational policies and requirements. Properly managing permissions is a critical aspect of cybersecurity, as it helps control who can access, modify, or execute files and directories, thereby enhancing security and reducing the risk of unauthorized access or data breaches.