File permissions in Linux

📁 Project Description

This project aims **set and manage access permissions** between the different roles of a work team, in order to **maintain the confidentiality, integrity and security of data** contained in various files and directories.

🔍 Verification of Permissions on Files and Directories

The command was used `ls -l` to list the current file permissions. The status of the permits is detailed below:

```
researcher2@2255d014d046:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 May 14 16:30 drafts
-rw-rw-rw- 1 researcher2 research_team   46 May 14 16:30 project_k.txt
-rw-r----- 1 researcher2 research_team   46 May 14 16:30 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 May 14 16:30 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 May 14 16:30 project_t.txt
researcher2@2255d014d046:~/projects$
```

Initial Permits

Archive	User	Group	Others
`project_k.txt`	read, write	read, write	read, write
`project_m.txt`	read, write	read	none
`project_r.txt`	read, write	read, write	read
`project_t.txt`	read, write	read, write	read
`.project_x.txt`	read, write	write	none

🔐 Permission Chain Explanation

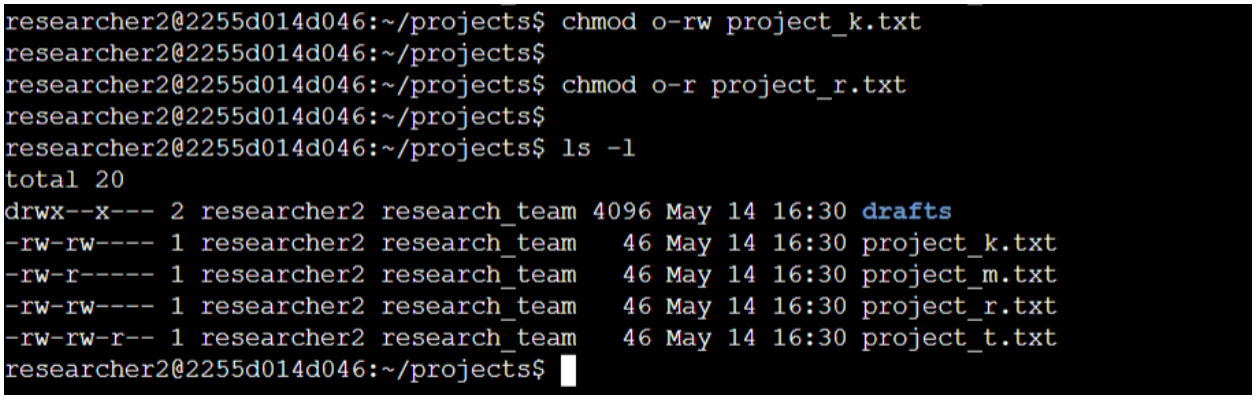
- `r` → Read permission: allows you to view the content of the file.
- `In` → Write permission: allows you to modify the content of the file.
- `x` → Execute permission: allows you to execute the file (in the case of scripts or binaries).
- `-` → Indicates absence of that permission.

⚙️ Changes Made to Permissions

1. Remove permissions from other users

To protect sensitive files and limit access to only authorized roles, the following commands were executed:

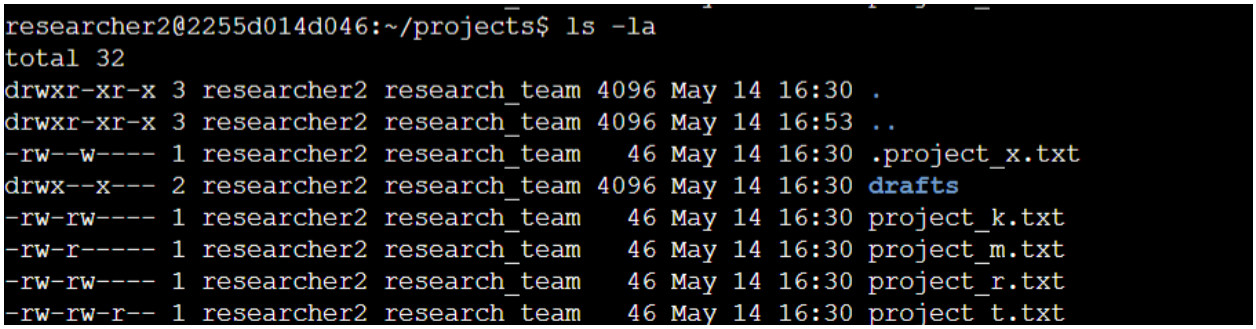
```
chmod o-rw project_k.txt
chmod o-r project_r.txt
```



- In `project_k.txt`, read and write permissions for "other" users were removed.
- In `project_r.txt`, only read permission for "others" was revoked.

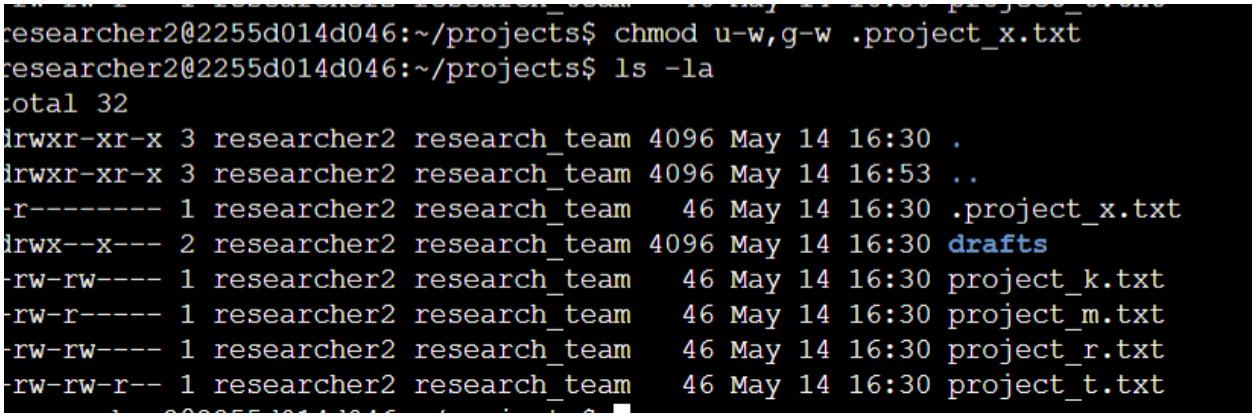
2. Hidden file management

The command was used `ls - her` to also list hidden files. It was identified that the file `.project_x.txt` it had write permissions for the user and group, which was unwanted due to its sensitive nature.



To correct it, we executed:

```
chmod u-w,g-w .project_x.txt
```

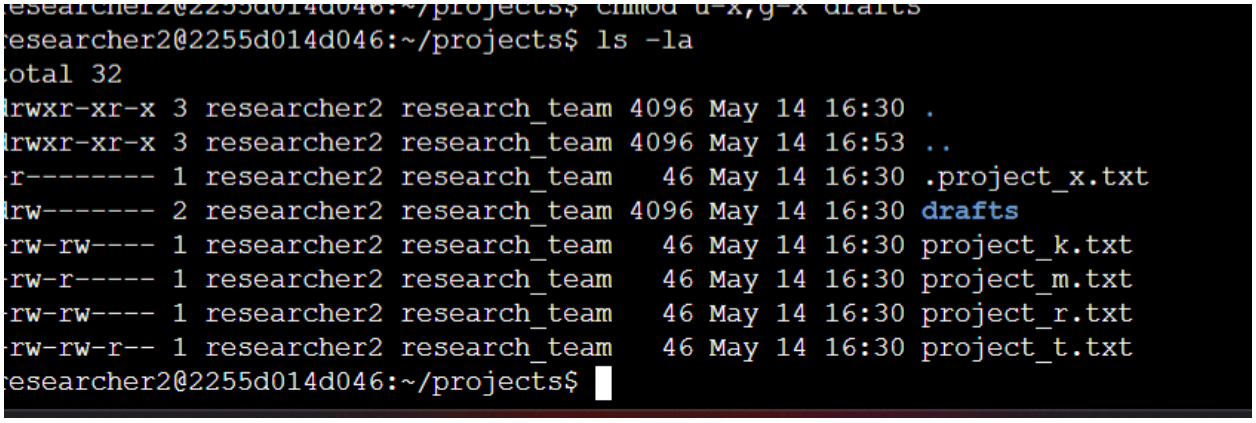


With this, the write permission for both the owner and the group was removed.

3. Changing permissions on a directory

To restrict execution of a directory named `drafts`, it was executed:

```
chmod u-x,g-x drafts
```



This prevents both the user and group members from being able to access the contents of the directory by browsing (`cd`), thus reinforcing security in collaborative work environments.

📝 Summary

In this exercise, basic principles of **file security on Unix/Linux systems**, focused on managing permissions using the command `chmod`. Privacy and role-based access control were strengthened, which is essential in professional environments where collaborative work involves sensitive data.