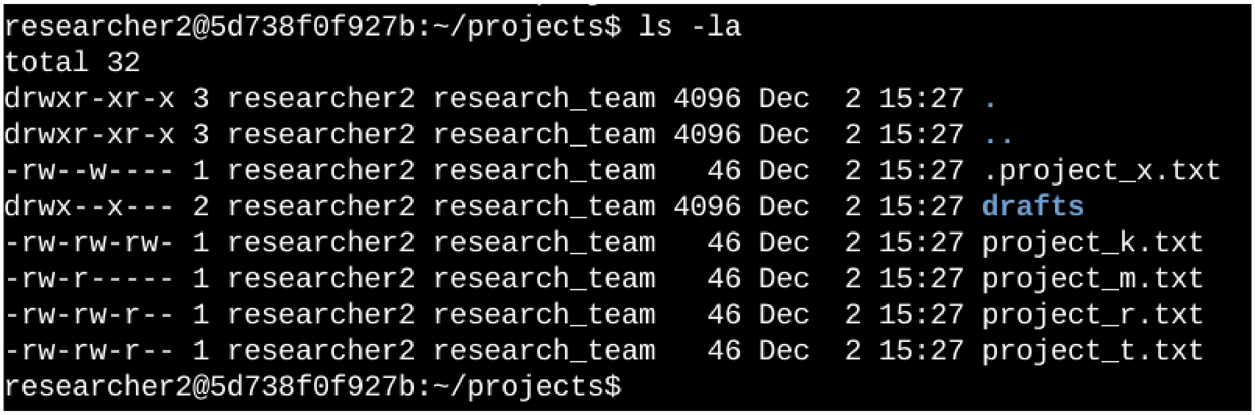
File permissions in Linux

Project description

In this scenario, my job is to manage and maintain file permissions within the `projects` directory for a large organization. This is to ensure alignment with the organization's security policies. Currently, the permissions of certain users do not reflect the level of authorization that should be given. The main goal is to review and update permissions to keep the file system secure. To complete this task, I performed the following tasks:

Check file and directory details

I entered the `ls -la` command to display a detailed listing of all contents of the `projects` directory, including hidden files and permissions.



```
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-rw--w---- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

Describe the permissions string

The 10-character string in the first column represents the permissions for each file and directory.

By analyzing this 10-character string, we can find out who is authorized to access the file and their specific permissions:

- **Character 1**: This character represents the file type. `d` for directory and `-` for file.
- **Characters 2 - 4**: These characters represent the **user**'s read `r`, write `w`, and execute (`x`) permissions. If any of these characters is a hyphen `-`, it means that particular permission is not granted to the user.

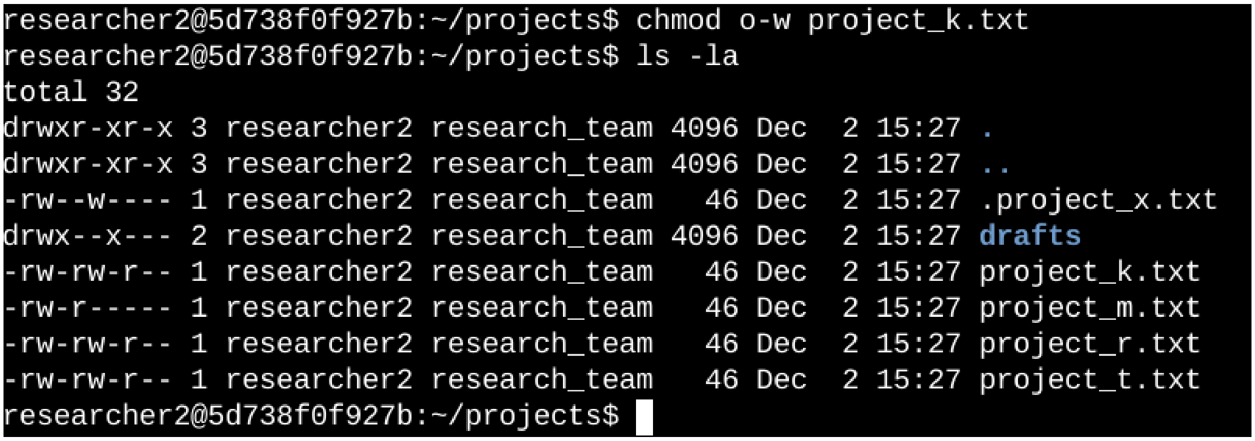
- **Characters 5 - 7:** These characters represent the **group**’s read (`r`), write (`w`), and execute (`x`) permissions. If any of these characters is a hyphen (`-`), it means that particular permission is not granted to the group.
- **Characters 8 - 10:** These characters represent **other**’s read (`r`), write (`w`), and execute (`x`) permissions. If any of these characters is a hyphen (`-`), it means that particular permission is not granted to other.

Example: The file permissions for `project_m.txt` are `-rw-r-----`. The first character is a hyphen `-`, so we can determine that `project_m.txt` is a file and not a directory. Analyzing Characters 2 - 4, we can determine that the user has read and write permissions, but not execute permission. Analyzing Characters 5 - 7, we can determine that the group only has read permission, but not write and execute permissions. Analyzing Characters 8 - 10, we can determine that other do not have any permissions to this file at all.

Change file permissions

I need to ensure that all write permissions are removed from other, since the organization does not allow other to have write access to any files. Referring to the detailed directory listing, other only has write permission to `project_k.txt`.

I entered the `chmod o-w project_k.txt` command to remove other’s write permission for the file.



```
researcher2@5d738f0f927b:~/projects$ chmod o-w project_k.txt
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-rw--w---- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$ 
```

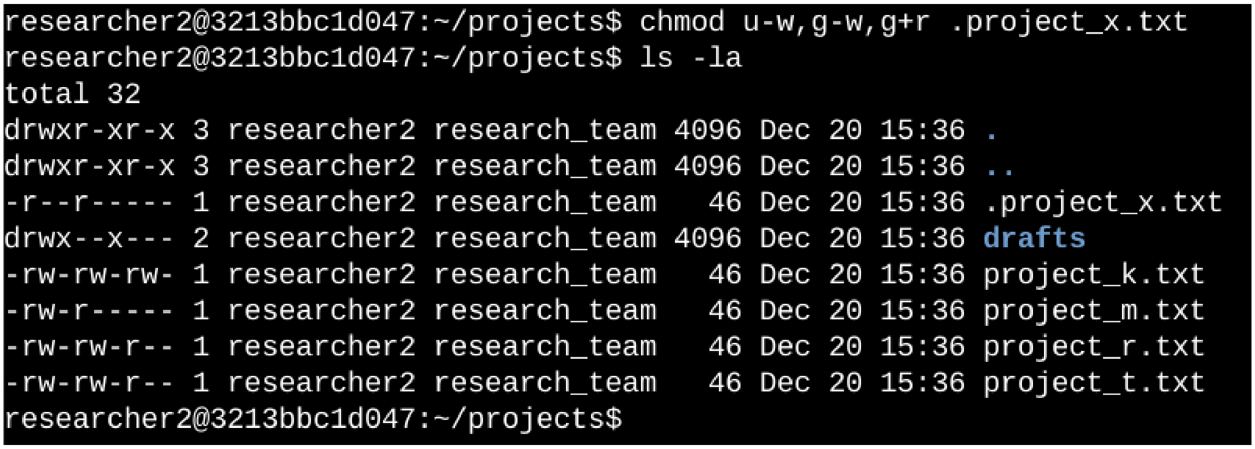
The `chmod` command changes the permissions on files and directories. The first argument `o-w` removes the write permission from other (o = other, -w = removes write permission), and the second argument specifies the file or directory that is being changed, which is `project_k.txt` in this case.

To confirm that the write permission has indeed been removed from other, I used `ls -la` to check the detailed listing again.

Change file permissions on a hidden file

Since `project_x.txt` is archived, the file should not have write permission for anyone and only user and group should have read permission.

I entered the `chmod u-w,g-w,g+r .project_x.txt` command to effect these changes.



```
researcher2@3213bbc1d047:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@3213bbc1d047:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec 20 15:36 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec 20 15:36 ..
-r--r----- 1 researcher2 research_team   46 Dec 20 15:36 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec 20 15:36 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Dec 20 15:36 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec 20 15:36 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec 20 15:36 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec 20 15:36 project_t.txt
researcher2@3213bbc1d047:~/projects$
```

The first argument consists of 3 portions. In the first portion `u-w`, I removed the write permission from user (u = user, -w = removes write permission). In the second portion `g-w`, I removed the write permission from group (g = group, -w = removes write permission). In the third portion `g+r`, I added the read permission for group (g = group, +r = adds read permission).

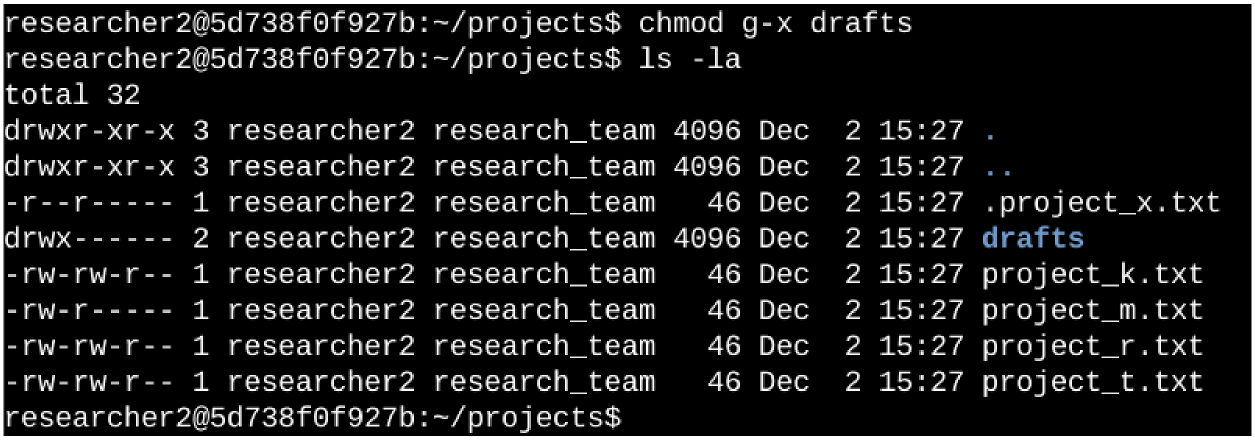
The second argument specifies the file or directory that is being changed, which is `.project_x.txt` in this case.

To confirm that the changes are done correctly, I used `ls -la` to check the detailed listing again.

Change directory permissions

Since only `researcher2` should have access to the `drafts` directory and its contents, I need to ensure that group and other do not have any permissions for the directory and its contents.

I entered the `chmod g-x drafts` command to remove execute permission from group.



```
researcher2@5d738f0f927b:~/projects$ chmod g-x drafts
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-r--r----- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

The first argument `g-x` removes the execute permission from group (g = group, -x = removes execute permission), and the second argument specifies the file or directory that is being changed, which is `drafts` in this case.

Summary

In this task, I analyzed and updated file and directory permissions on a Linux system according to the requirements of the organization. I used the `ls -la` command first to check the permissions for the directory. This enabled me to make decisions for the following steps to change permissions using the `chmod` command.