File permissions in Linux

Project description

This project involves ensuring proper access control for a research team within a large organization. As the security professional, the primary goal is to review the current file system permissions, assess whether they align with the authorized access levels, and make necessary adjustments. The end goal is to maintain a secure system by ensuring that only authorized users have the appropriate permissions while removing any accessibility from any unauthorized parties.

Check file and directory details

```
researcher2@df616911e63e:~$ pwd
/home/researcher2
researcher2@df616911e63e:~$ ls
projects
researcher2@df616911e63e:~$ cd /home/researcher2/projects
researcher2@df616911e63e:~/projects$ la -l
-bash: la: command not found
researcher2@df616911e63e:~/projects$ ls
drafts  project_k.txt  project_m.txt  project_r.txt  project_t.txt
researcher2@df616911e63e:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Sep  7 01:19 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Sep  7 01:19 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep  7 01:19 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep  7 01:19 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep  7 01:19 project_t.txt
researcher2@df616911e63e:~/projects$
```

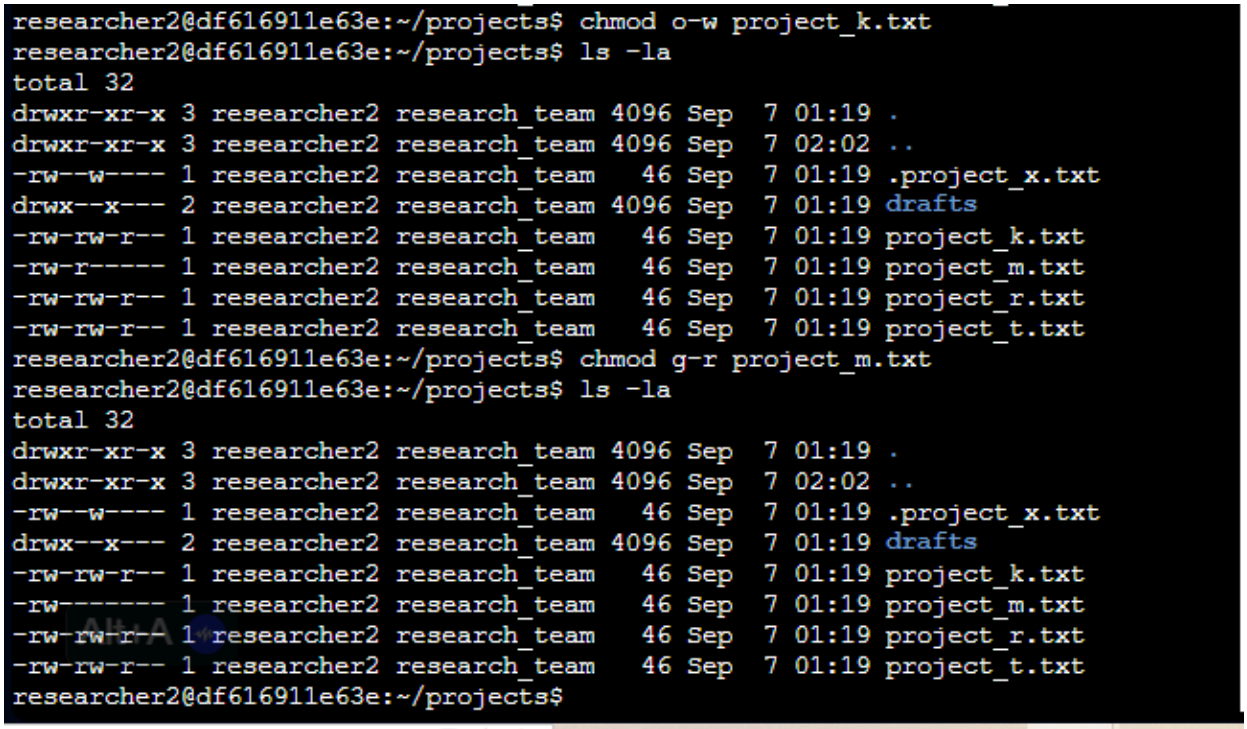
The first command line entered was `pwd`, which allowed me to find out which directory I was currently in. Once the current directory was displayed, the command line `ls` was used to identify the sub-directories and files that could be accessed from the researcher2 home directory. The command line `cd` was then used to travel to the directory of `projects` where the list of all project files could be found and accessed. The command line `ls` gave a list of files available within the `projects` directory and the command line `ls -l` gave a comprehensive look as to the permissions of each file and which parties had which permissions.

Describe the permissions string

- 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.

Change file permissions



The organization decided that other users, meaning users who aren't in the group or who arent me, did not need permission to write on the file. The permission to write on a file allows users to alter the file, the company decided to remove the permission from other users to ensure no accidental or purposeful deletion or alteration of the file labeled `project_k.txt`. It was also decided that the file `project_m.txt` is a restricted file and should not be readable or writable by the group or other; only the user should have these permissions on this file so the `chmod` command was used to both remove the write permission of other(o) using `chmod o-w project_k.txt` and to remove the read permission from group on `project_m.txt` using the command `chmod g-r project_m.txt`. After each change the command `ls -la` was used to make sure the updates were properly made.

Change file permissions on a hidden file

```
researcher2@df616911e63e:~/projects$ chmod u-w,g-w .project_x.txt
researcher2@df616911e63e:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep  7 01:19 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep  7 02:02 ..
-r-------- 1 researcher2 research_team   46 Sep  7 01:19 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep  7 01:19 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Sep  7 01:19 project_k.txt
-rw------- 1 researcher2 research_team   46 Sep  7 01:19 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep  7 01:19 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep  7 01:19 project_t.txt
researcher2@df616911e63e:~/projects$ chmod g+r .project_x.txt
researcher2@df616911e63e:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep  7 01:19 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep  7 02:02 ..
-r--r----- 1 researcher2 research_team   46 Sep  7 01:19 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep  7 01:19 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Sep  7 01:19 project_k.txt
-rw------- 1 researcher2 research_team   46 Sep  7 01:19 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep  7 01:19 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep  7 01:19 project_t.txt
```

You can tell that `.project_x.txt` is a hidden file because it starts with a period (.), you can find hidden files using the `ls -a` command, and you can see the user permissions of hidden files using the command `ls -la` which combines both the `ls -a` and the `ls -l` commands. After locating the hidden file and viewing its user permissions, I first removed write permissions from both the user and the group using the `chmod u-w,g-w .project_x.txt` command. The `u-w,g-w` strain indicates that I want to remove the write permission from both the `researcher2` user and the group users. Then, in a separate command, I granted read permissions to the group using `chmod g+r .project_x.txt`, in this case, the `g+r` indicates I want to give the group users permission to read the file.

Change directory permissions

```
researcher2@df616911e63e:~/projects$ chmod g-x /home/researcher2/projects/draft
s
researcher2@df616911e63e:~/projects$ ls -l
total 20
drwx------ 2 researcher2 research_team 4096 Sep  7 01:19 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Sep  7 01:19 project_k.txt
-rw------- 1 researcher2 research_team   46 Sep  7 01:19 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep  7 01:19 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep  7 01:19 project_t.txt
researcher2@df616911e63e:~/projects$
```

The organization decided that only the researcher2 user should have access to the drafts directory and its contents. This means that no one other than researcher2 should have execute

Permission. The command line `chmod` in Linux is used to change the permissions of a user, so I used the command `chmod g-x /home/researcher2/projects/drafts` to remove the execute permission of the group (g), the (-x) functions the same as in math by subtracting the permission. The strain `/home/researcher2/projects/drafts` is the absolute file path to the directory I need to make the change.

Summary

I was instructed by my organization to modify multiple permissions to align with the required level of authorization for files and directories in the `projects` directory to comply with the principle of least privilege. The first step involved using `ls -la` to review the current permissions, which guided my decisions in the subsequent steps. I then executed the `chmod` command multiple times to adjust the permissions on the necessary files and directories.