File permissions in Linux

Project description

In this project, the goal is to view the permissions given to the user, group, and others in the /home/researcher2/projects directory. After doing so, we will change file permissions to meet the proper criteria.

Scenario

You are a security professional at a large organization. You mainly work with their research team. Part of your job is to ensure users on this team are authorized with the appropriate permissions. This helps keep the system secure.

Your task is to examine existing permissions on the file system. You'll need to determine if the permissions match the authorization that should be given. If they do not match, you'll need to modify the permissions to authorize the appropriate users and remove any unauthorized access.

Note: This scenario involves investigating and updating the same file permissions as the ones in the **Manage authorization** lab. You can revisit the lab to get screenshots to include in your portfolio document. If you choose, it's also possible to complete this activity without revisiting the lab by typing your commands in the template.

Check file and directory details

●project_k.txt
○ User = read, write,
○ Group = read, write
○ Other = read, write

● project_m.txt
○ User = read, write
○ Group = read
○ Other = none

● project_r.txt

○ User= read, write
○ Group = read, write
○ Other = read

● project_t.txt
○ User = read, write
○ Group = read, write
○ Other = read

● .project_x.txt
○ User = read, write
○ Group = write
○ Other = none

There is also one subdirectory inside the projects directory named drafts. The permissions on drafts are:
● User = read, write, execute
● Group = execute
● Other = none

```
researcher2@49d95b889ddd:~$ pwd
/home/researcher2
researcher2@49d95b889ddd:~$ ls
projects
researcher2@49d95b889ddd:~$ cd projects/
researcher2@49d95b889ddd:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct  7
22:02 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct  7
22:15 ..
-rw--w---- 1 researcher2 research_team   46 Oct  7
22:02 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct  7
22:02 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct  7
22:02 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct  7
22:02 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct  7
22:02 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct  7
22:02 project_t.txt
researcher2@49d95b889ddd:~/projects$
```

Describe the permissions string

I'll be describing the directory named 'drafts'. The permissions string reads: "drwx--x---". The first character, "d", connotes that this is a directory. The next three characters, "rwx", show that the user has the permission to read, write, and execute. The three following characters, "--x", show that the group has no read or write permissions, but can execute. Finally, the last three characters, "---", shows that other users cannot read, write, or execute.

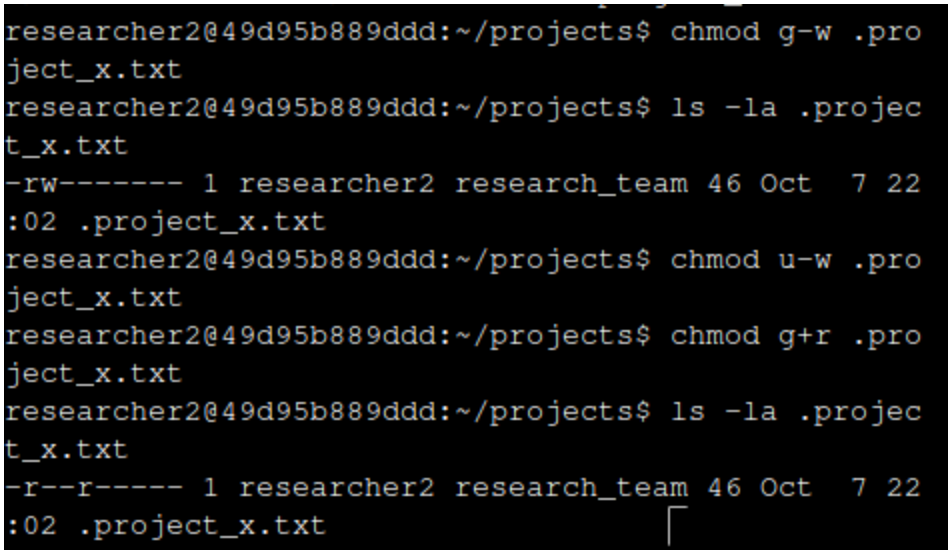
Change file permissions

The organization does not allow others to have write access to any files, yet, project_k.txt had write permissions given to others. I removed this permission by using `chmod o-w project_k.txt`.

```
researcher2@49d95b889ddd:~/projects$ chmod o-w proj
ect_k.txt
researcher2@49d95b889ddd:~/projects$ ls -l project_
k.txt
-rw-rw-r-- 1 researcher2 research_team 46 Oct  7 22
:02 project_k.txt
researcher2@49d95b889ddd:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct  7
22:02 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct  7
22:15 ..
-rw--w---- 1 researcher2 research_team   46 Oct  7
22:02 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct  7
22:02 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Oct  7
22:02 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct  7
22:02 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct  7
22:02 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct  7
22:02 project_t.txt
researcher2@49d95b889ddd:~/projects$
```

Change file permissions on a hidden file

The research team has archived **.project_x.txt**, which is why it's a hidden file. This file should not have write permissions for anyone, but the user and group should be able to read the file. Before the change, the user and group both had write permissions, and the group did not have read permissions. By using chmod u-w .project_x.txt, I removed write permissions from the user. Then, I used chmod g-w .project_x.txt and g+r .project_x.txt (though this can be one command), I removed the write permission from the group, and added the read permission.



Change directory permissions

According to the criteria, 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. Therefore, to fix this, as the group currently has execute permissions, I used `chmod g-x drafts/`.

```
researcher2@49d95b889ddd:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct  7
22:02 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct  7
22:15 ..
-r--r----- 1 researcher2 research_team   46 Oct  7
22:02 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct  7
22:02 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Oct  7
22:02 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct  7
22:02 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct  7
22:02 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct  7
22:02 project_t.txt
researcher2@49d95b889ddd:~/projects$ chmod g-x draf
ts/
researcher2@49d95b889ddd:~/projects$ ls -la drafts/
total 8
drwx------ 2 researcher2 research_team 4096 Oct  7
22:02 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct  7
22:02 ..
researcher2@49d95b889ddd:~/projects$
```

Summary

In this project, I worked with file permissions in Linux to manage access control to files and directories within a project directory. The goal was to view, modify, and interpret file and directory permissions to ensure they comply with the organization's guidelines. Commands such as `chmod` and `ls -la` were used to update and check permissions for users, groups, and others.

Check file and directory details:
Here, I used the command `ls -la`, as this lists all files, including hidden ones, along with detailed permissions in the current directory. It shows the 10-character permission string for each file or directory.

Change file permissions:

Here, I used `chmod o-w project_k.txt`in order to remove write permissions from project_k.txt for other users.

Change permissions for hidden files
Here, I used the longhand version of `chmod u-w,g-w,g+r .project_x.txt`. This command removed write permissions for the group and user, and added read permissions for the group on the hidden file .project_x.txt.

Change directory permissions
This directory should only belong to the user, so I used `chmod g-x drafts/` to remove the execute permissions for the group on the directory `drafts/`.

Interpreting the 10-Character Permission String
The 10-character string that appears in the `ls -la` output can be broken down as follows:

- First character: Indicates the type of file (d for directory, - for file).
- Next three characters (positions 2-4): Permissions for the user (owner). “rwx” gives the user read, write, and execute permissions.
- Next three characters (positions 5-7): Permissions for the group. “--x” gives the group execute permissions.
- Last three characters (positions 8-10): Permissions for others. “---” gives others no permissions.

In this project, I explored the process of viewing and modifying file and directory permissions in Linux using commands like `chmod` and `ls -la`. I also learned how to interpret permission strings and apply permission changes to both regular and hidden files. Ensuring correct permissions is critical for managing access control within a system, and the `chmod` command provides flexible options to update permissions for users, groups, and others effectively.