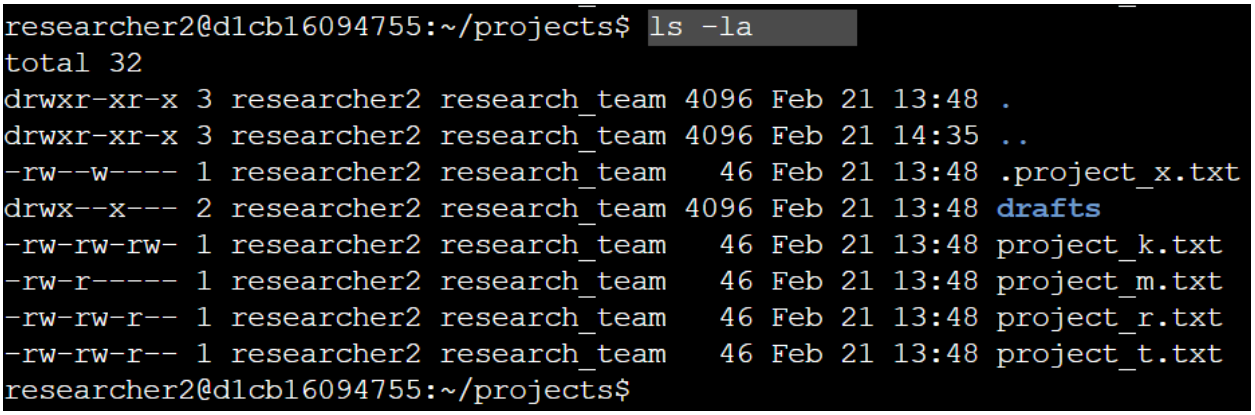
File System Security and Access Control in Linux

Project description

This project focuses on securing file system permissions within a research team's environment in a large organization. As a security professional, my role is to ensure that only authorized users have appropriate access while preventing unauthorized access. This helps maintain data confidentiality, integrity, and compliance with security policies.

Check file and directory details

This displays the file structure of the /home/researcher2/projects directory and the permissions of the files and subdirectory it contains.



```
researcher2@d1cb16094755:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb 21 13:48 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb 21 14:35 ..
-rw--w---- 1 researcher2 research_team   46 Feb 21 13:48 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Feb 21 13:48 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Feb 21 13:48 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Feb 21 13:48 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 21 13:48 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 21 13:48 project_t.txt
researcher2@d1cb16094755:~/projects$
```

The first line of the screenshot displays the command I entered and the other lines displays the output. The code lists all contents of the projects directory. I used the ls command with the -la option to display a detailed listing of the file contents that also returned hidden files. The output of my command indicates that there is one directory named drafts, one hidden file named .project_x.txt and five other project files. The 10-character string in the first column represents the permissions set on each file or directory.

Describe the permissions string

The 10-character string can be deconstructed to determine who is authorized to access the directory or file and their specific permissions. The character represent as follows:

1. **The 1st character** is either a d which indicates it is a directory or an hyphen(-) to indicate it's a regular file.

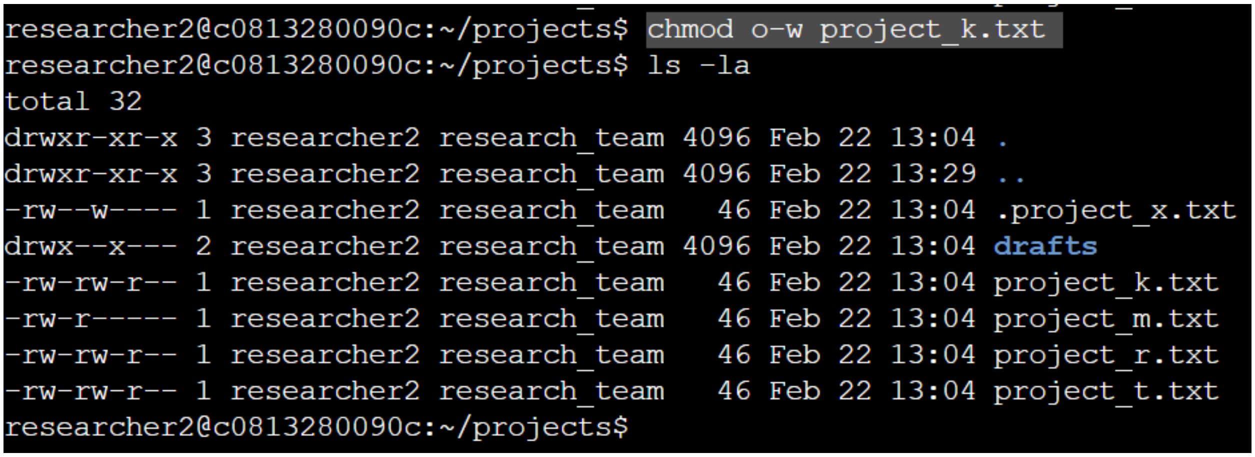
2. **The 2nd-4th characters:** tells the type of permission has been granted to the user type of owner, it reads a (r) if the read permission is enabled for a user, (w) if the write permission is enabled and (x) if the execute permission is enabled or an hyphen (-) is for each permission not granted.

3. **The 5th-7th characters:** tells the type of permission has been granted to the group type of owner, it reads a (r) if the read permission is granted for a user, (w) if the write permission is granted and (x) if the execute permission is granted or an hyphen (-) is for each permission not granted.

4. **The 8th-10th characters:** tells the type of permission has been granted to the others type of owner, it reads a (r) if the read permission is granted for a user, (w) if the write permission is granted and (x) if the execute permission is granted or an hyphen (-) is for each permission not granted.

Change file permissions

The organization determined that other shouldn't have write access to any of their files. To comply with this, I referred to the file permissions to confirm which file the other had permission to write. It showed that other have write permission on the project_k.txt files, I used the Linux command below to remove this permission.

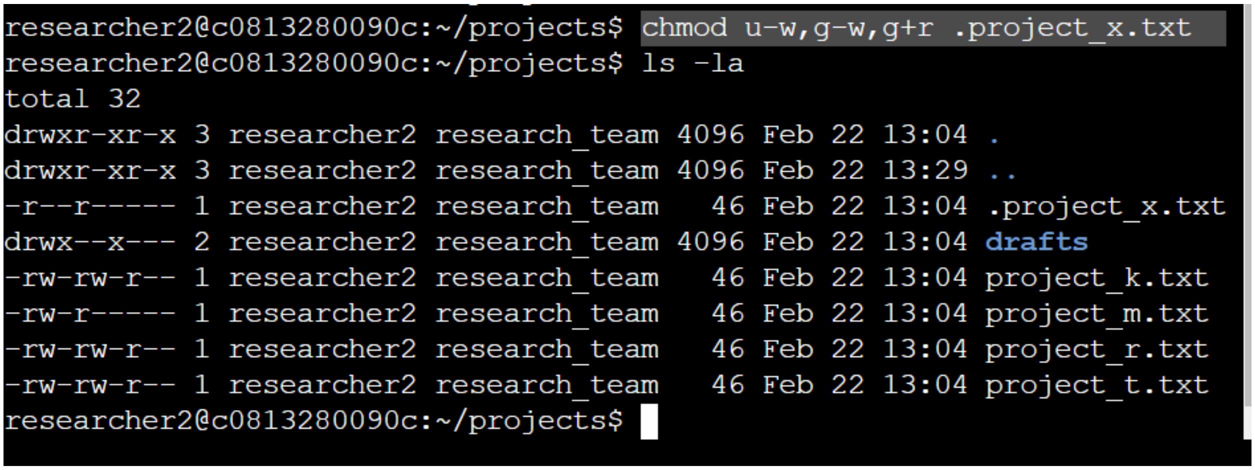


```
researcher2@c0813280090c:~/projects$ chmod o-w project_k.txt
researcher2@c0813280090c:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb 22 13:04 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb 22 13:29 ..
-rw--w---- 1 researcher2 research_team   46 Feb 22 13:04 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Feb 22 13:04 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Feb 22 13:04 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Feb 22 13:04 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 22 13:04 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 22 13:04 project_t.txt
researcher2@c0813280090c:~/projects$
```

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. The chmod command changes the permissions on files and directories. The first argument indicates what permissions should be changed, and the second argument specifes the file or directory. In this example, I removed write permissions from other for the project_k.txt file. After this, I used ls -la to review the updates I made.

Change file permissions on a hidden file

The research team at my organization recently archived project_x.txt. They do not want anyone to have write acces to this project, but the user and the group should have read access. The command below displays how I changed these permission

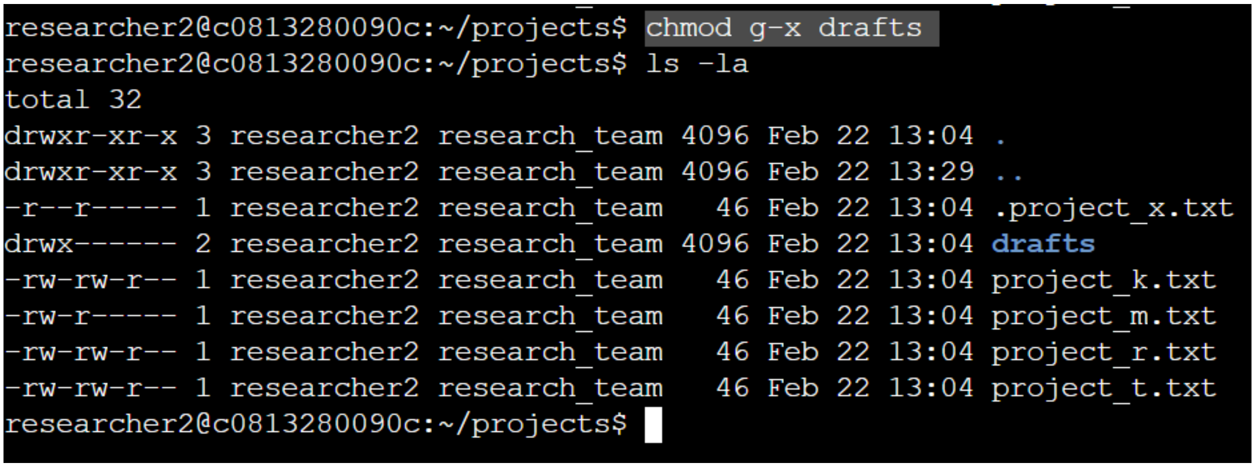


```
researcher2@c0813280090c:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@c0813280090c:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb 22 13:04 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb 22 13:29 ..
-r--r----- 1 researcher2 research_team   46 Feb 22 13:04 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Feb 22 13:04 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Feb 22 13:04 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Feb 22 13:04 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 22 13:04 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 22 13:04 project_t.txt
researcher2@c0813280090c:~/projects$
```

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. I know .project_x.txt is a hidden file because it starts with a period (.). In this example, I removed write permissions from the user and group, and added read permissions to the group. I removed write permissions from the user with u-w.
Then, I removed write permissions from the group with g-w, and added read permissions to the group with g+r.

Change directory permissions

My organization only wants the researcher2 user to have access to the drafts directory and its contents. This means that no one other than reseracher2 should have execute access. This following command demonstrates how I changed the permissions.



```
researcher2@c0813280090c:~/projects$ chmod g-x drafts
researcher2@c0813280090c:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb 22 13:04 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb 22 13:29 ..
-r--r----- 1 researcher2 research_team   46 Feb 22 13:04 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Feb 22 13:04 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Feb 22 13:04 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Feb 22 13:04 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 22 13:04 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 22 13:04 project_t.txt
researcher2@c0813280090c:~/projects$
```

The output here displays the permission listing for several files and directories. Line 1 indicates the current directory (projects), and line 2 indicates the parent directory (home). Line 3 indicates a regular file titled .project_x.txt. Line 4 is the directory (drafts) with restricted permissions. Here you can see that only researcher2 has execute permissions. It was previously determined that the group had execute permissions, so I used the chmod command to remove them. The researcher2 user already had execute permissions, so they did not need to be added.

Summary

I changed multiple permissions to match the level of authorization my organization wanted for files and directories in the projects directory. The first step in this was using ls -la to check the permissions for the directory. This informed my decisions in the following steps. Then, I used the chmod command multiple times to change the permissions on files and directories.