Managing File Permissions in Linux

Project description

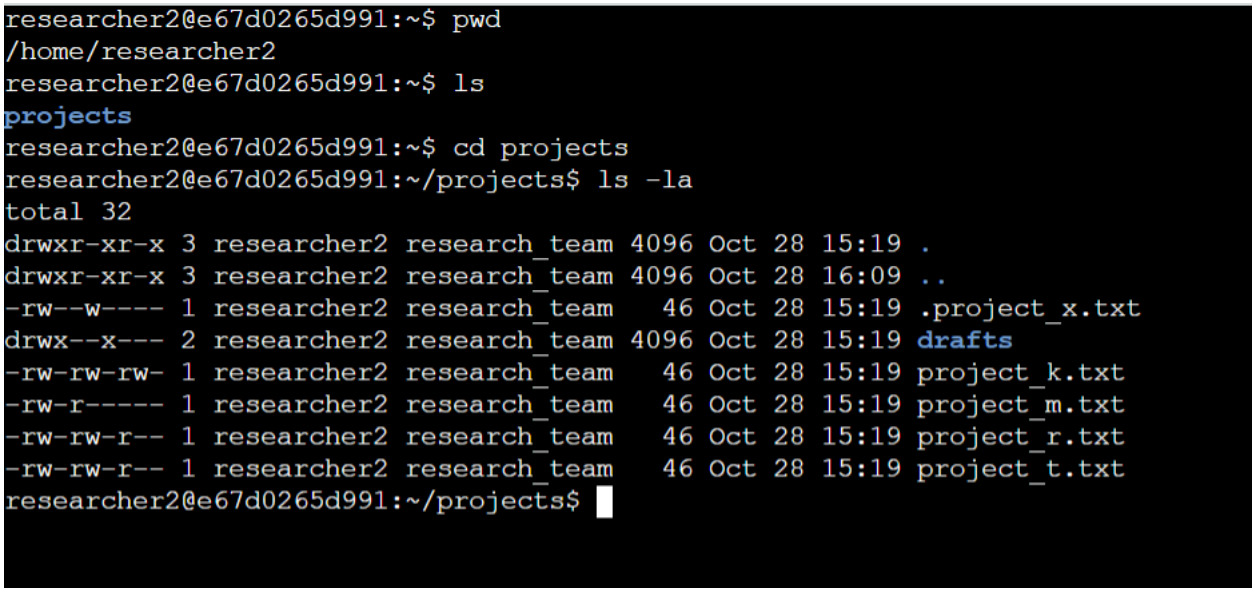
The research team tasked me with examining the existing permission sets for files and subdirectories within the projects directory. I was then to determine whether these permissions aligned with the intended authorization. If discrepancies were found, I would modify the permissions to authorize appropriate users and revoke unauthorized access.

Check file and directory details

I navigated to the projects directory. Then I input the command `ls` to display all the available directories. As a result, `projects` was the only directory listed.

By using `ls -la`, I displayed permissions to files and directories, including hidden files.

Hidden file naming conventions start with a period (`.`), followed by the file name. In this case, `.project_x.txt` was the hidden file.



Describe the permissions string

A 10-character string begins each entry and indicates how the permissions on the file are set. For instance, a directory with full permissions for all owner types would be:

drwxrwxrwx

- 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. A hyphen (-) instead 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. A hyphen (-) instead indicates that this permission is not granted for others.

Change file permissions

1. I wrote the command `chmod o-w project_k.txt` to remove write permissions from the file.
2. I wrote the command `chmod g-r project_m.txt` to remove read permissions from the file.

```
researcher2@e67d0265d991:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 28 15:19 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 28 16:09 ..
-rw--w---- 1 researcher2 research_team   46 Oct 28 15:19 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 28 15:19 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Oct 28 15:19 project_k.txt
-rw------- 1 researcher2 research_team   46 Oct 28 15:19 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 28 15:19 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 28 15:19 project_t.txt
researcher2@e67d0265d991:~/projects$
```

Change file permissions on a hidden file

Regarding the file `.project_x.txt`, I wanted to remove write permissions for both users and the group while maintaining read permissions for the group. The following code achieved this in a single line of code:

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

```
researcher2@e67d0265d991:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@e67d0265d991:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 28 15:19 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 28 16:09 ..
-r--r----- 1 researcher2 research_team   46 Oct 28 15:19 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 28 15:19 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Oct 28 15:19 project_k.txt
-rw------- 1 researcher2 research_team   46 Oct 28 15:19 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 28 15:19 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 28 15:19 project_t.txt
researcher2@e67d0265d991:~/projects$
```

Change directory permissions

To allow only `researcher2` access to the `drafts` directory, I used the command `chmod g-x drafts`. This command removed the group execute permission, ensuring that only users with specific individual permissions could enter the directory.

```
drwxr-xr-x 3 researcher2 research_team 4096 Oct 28 15:19 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 28 16:09 ..
-r--r----- 1 researcher2 research_team   46 Oct 28 15:19 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Oct 28 15:19 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Oct 28 15:19 project_k.txt
-rw------- 1 researcher2 research_team   46 Oct 28 15:19 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 28 15:19 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 28 15:19 project_t.txt
researcher2@e67d0265d991:~/projects$
```

Summary

As a security analyst, I found that setting appropriate access permissions was critical to protecting sensitive information and maintaining the overall security of a system. This scenario demonstrates my ability to use basic Linux Bash shell commands to:

- Examine file and directory permissions.
- Change permissions on files.
- Change permissions on directories.