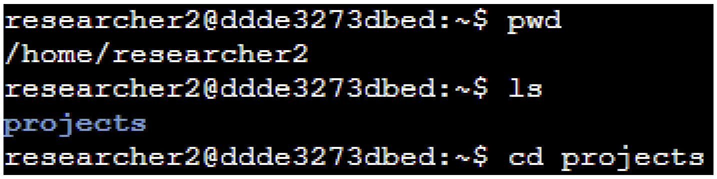
File permissions in Linux

Project description

In this activity, I am a security professional at a large organization. I am tasked with ensuring users on the team are authorized with the appropriate permissions. I need to examine permissions and determine if the permissions are matching the authorization given, and make changes to the permissions as necessary.

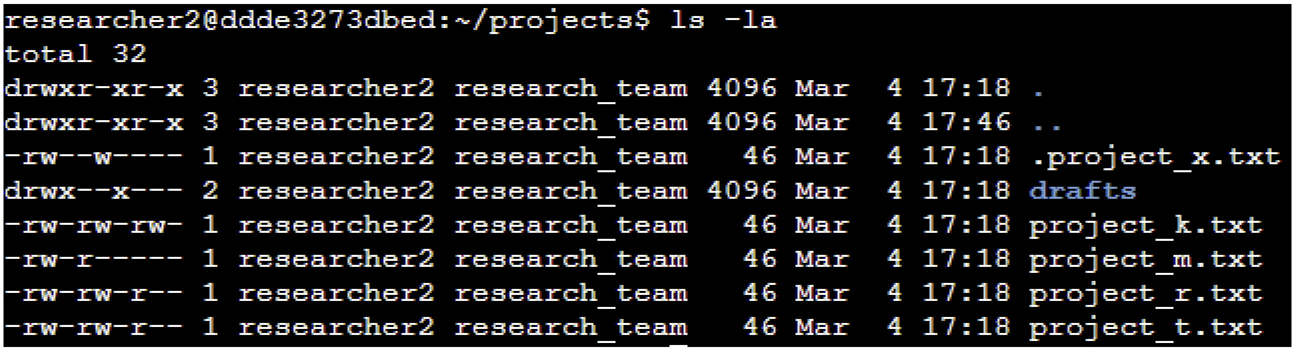
Check file and directory details

I am first going to navigate to the `project` directory using the `cd` command:



```
researcher2@ddde3273dbed:~$ pwd
/home/researcher2
researcher2@ddde3273dbed:~$ ls
projects
researcher2@ddde3273dbed:~$ cd projects
```

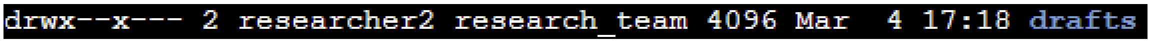
and display all permissions for hidden files and folders using the `ls` command:



```
researcher2@ddde3273dbed:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Mar  4 17:18 .
drwxr-xr-x 3 researcher2 research_team 4096 Mar  4 17:46 ..
-rw--w---- 1 researcher2 research_team   46 Mar  4 17:18 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Mar  4 17:18 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Mar  4 17:18 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Mar  4 17:18 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar  4 17:18 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar  4 17:18 project_t.txt
```

Describe the permissions string

In this command output line, there is a directory called `drafts`, the first 10 character string listed in the output is the directories listed permissions for each owner `user`, `group` and `other`.

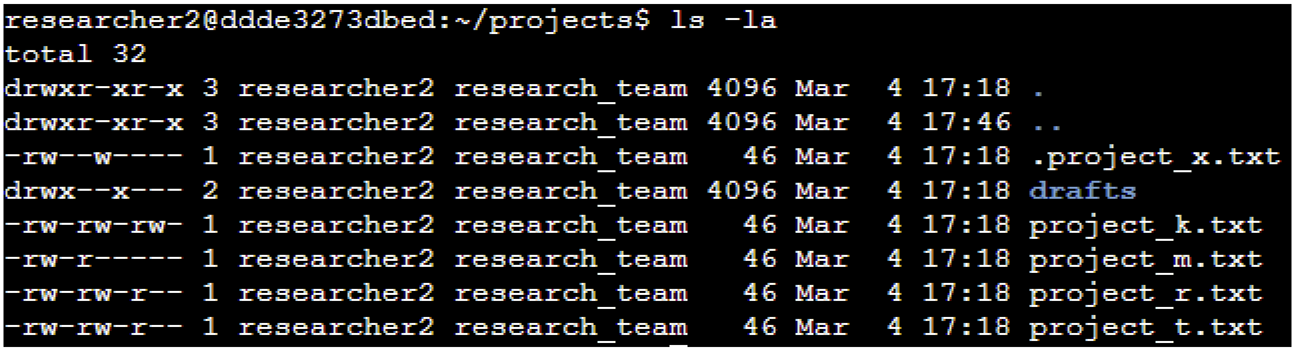


```
drwx--x--- 2 researcher2 research_team 4096 Mar  4 17:18 drafts
```

I can see that this is a directory indicated by the letter d.
The owner user `researcher2` has read, write and execute permissions indicated by the first 3 letters.
The owner group `research_team` has only execute permissions as indicated by the x.
The owner `other` has no permissions as indicated by the last 3 '- - -' dashes.

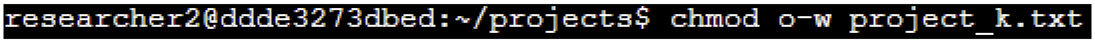
Change file permissions

The organization does not allow the owner `other` to have write permissions on any files or directories.



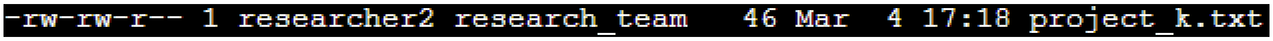
```
researcher2@ddde3273dbed:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Mar  4 17:18 .
drwxr-xr-x 3 researcher2 research_team 4096 Mar  4 17:46 ..
-rw--w---- 1 researcher2 research_team   46 Mar  4 17:18 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Mar  4 17:18 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Mar  4 17:18 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Mar  4 17:18 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar  4 17:18 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar  4 17:18 project_t.txt
```

Using the `ls -la` command from before, I can see that `project_k.txt` currently allows other write permissions, which we will remove with the `chmod` command.



```
researcher2@ddde3273dbed:~/projects$ chmod o-w project_k.txt
```

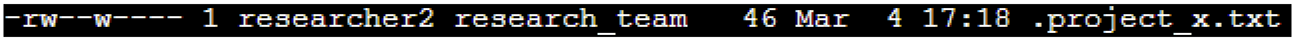
Using the `ls -la` command again, we can see that the write permissions for other have been removed,
leaving only read permissions.



```
-rw-rw-r-- 1 researcher2 research_team   46 Mar  4 17:18 project_k.txt
```

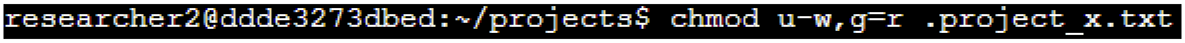
Change file permissions on a hidden file

The research team has archived `.project_x.txt.` This file should not have write permissions for anyone, but the `user` and `group` should still have read permissions.



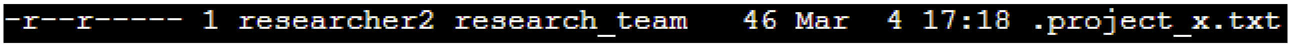
```
-rw--w---- 1 researcher2 research_team   46 Mar  4 17:18 .project_x.txt
```

Using the `ls -la` command from before, we can see that `user` currently has both read and write permissions, and `group` has only read permissions, with `other` having no permissions. We will use the `chmod` command to remove write permissions from both `user`, and `group`, and give read permissions to `group`.



```
researcher2@ddde3273dbed:~/projects$ chmod u-w,g=r .project_x.txt
```

And upon using the `ls -la` command to confirm the changes have been applied correctly:

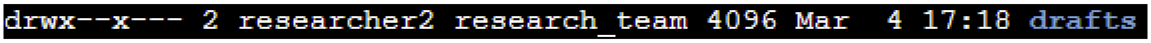


```
-r--r----- 1 researcher2 research_team   46 Mar  4 17:18 .project_x.txt
```

We can see that `user` and `group` have only read permissions.

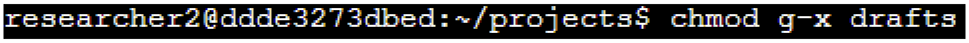
Change directory permissions

The files and directories in the projects directory belong to user `researcher2`. User `researcher2` should be the only one able to access drafts.



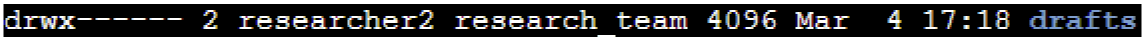
```
drwx--x--- 2 researcher2 research_team 4096 Mar  4 17:18 drafts
```

Using the `ls -la` command from before, we can see that the permissions for the drafts directory indicate that `user` has read, write and execute permissions, however `group` has execute permissions as well. We will again use `chmod` to change permissions.



```
researcher2@ddde3273dbed:~/projects$ chmod g-x drafts
```

And upon reviewing if the changes are correct:



```
drwx------ 2 researcher2 research_team 4096 Mar  4 17:18 drafts
```

We can see that `user` has read, write and execute permissions, and `group` has no permissions.

Summary

Using permission commands in Linux, I was able to modify and view permissions for each owner and make changes as necessary. This ensures only authorized users are able to access sensitive files and directories, and prevents any unauthorized access.