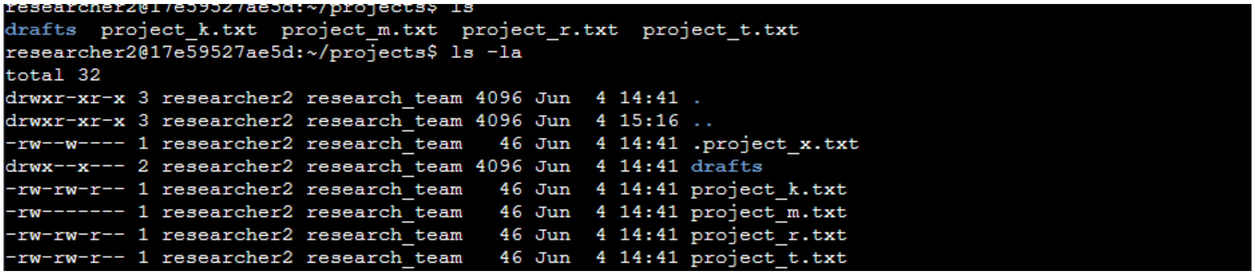
File permissions in Linux

Project description

The research team at my organization needs to update the file permissions for some files and directories within the project directory. The permission does not currently reflect the level of authorization that should be given, monitoring and updating these permissions is necessary in keeping their system secure. To complete this task, i performed the following

Check file and directory details

The following code shows how i used the linux command to determine the current permission for the files and subdirectories within the Project directory in the file system



```
researcher2@17e59527ae5d:~/projects$ ls
drafts  project_k.txt  project_m.txt  project_r.txt  project_t.txt
researcher2@17e59527ae5d:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun  4 14:41 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun  4 15:16 ..
-rw--w---- 1 researcher2 research_team   46 Jun  4 14:41 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun  4 14:41 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun  4 14:41 project_k.txt
-rw------- 1 researcher2 research_team   46 Jun  4 14:41 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun  4 14:41 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun  4 14:41 project_t.txt
```

The first line of the screenshot shows the command I entered, and the other lines display the output. This command lists all content of the project directory. I used the ls -la command which revealed contents of both hidden and unhidden files. The output of my command indicates that there's a directory named drafts, a hidden file named .project_x.txt and five other project files. The ten character strings on the first column describe the permissions set on each file or directory.

Describe the permissions string

The permission string is used to read and change authorizations to either prevent or give permission to the user, group or other, to read, write and execute a file or directory. The characters and what they represent are as follow

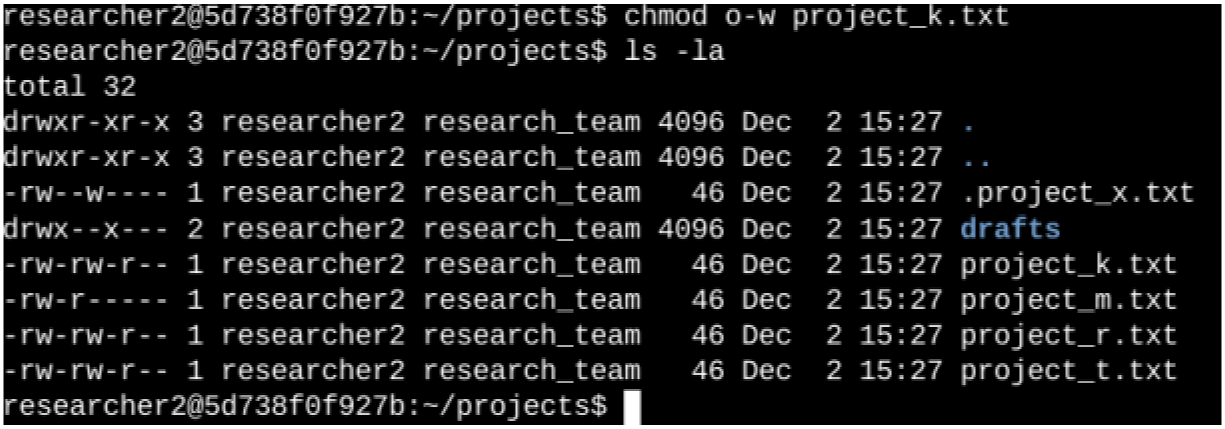
- 1st character : It’s either d or a Hyphen (-), whenever the first character is d, it means it’s a directory, and whenever the first character is a hyphen (-) it means it’s a regular file
- 2nd - 4th characters: These characters specify the read (r), write (w), and execute (x) permission for the user, when any of these characters has a hyphen (-) instead, it means that this permission isn’t granted to the user.
- 5th-7th characters: These characters specify the read (r), write (w), and execute (x) permission for the group, when any of these characters has a hyphen (-) instead, it means that this permission isn't granted to the group.
- 8th-10th characters: These characters specify the read(r), write (w), and execute (x) permission for other, this owner type consists of all other users on the system apart from user and group. when any of these characters has a hyphen (-) instead, it means that this permission isn't granted to other.

For instance the file permission for drafts are drwx--x--- , since the first character is d, this implies that drafts is a directory and not a file. The 2nd, 3rd and 4th characters are r,w,x which implies that user permission to read (r), write (w) and execute (x) are granted. The 5th and 6th characters are hyphens (-) which implies that group permission to read(r) and write (w) isn’t granted. The 8th. 9th, 10th characters are also hyphens (-) which implies that other permission to read (r) write (w) and execute (x) isn’t granted. The 7th character isn;t a hyphen (-), instead it’s x which implies that group permission to execute (x) is granted.

Change file permissions

The organization decided that other shouldn't have write access to project_k.txt. To make sure the permissions align with the organization request, i decided that project_k.txt must deny other access to write (x)

The following code shows how i achieved this with linux command



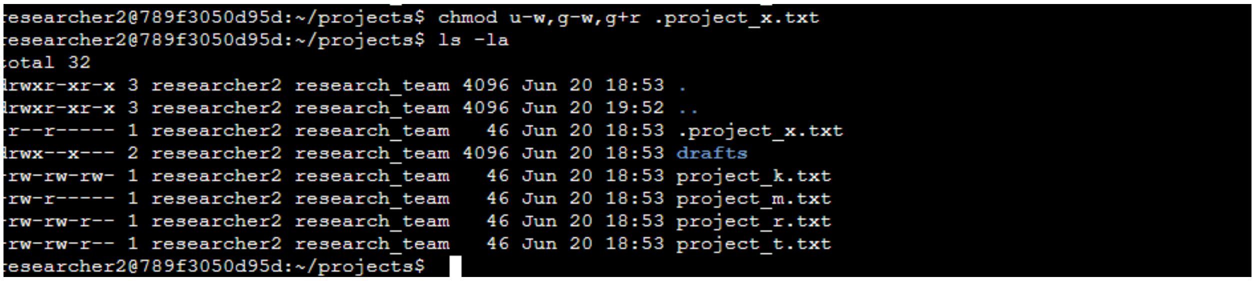
```
researcher2@5d738f0f927b:~/projects$ chmod o-w project_k.txt
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-rw--w---- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/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 specifies the file or directory. In this example, i removed the write permission from other for project_k.txt, after this i used ls -la to review the update 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 access to this project, but the user and group should have read access.

The following code demonstrates how I used Linux commands to change the permissions:



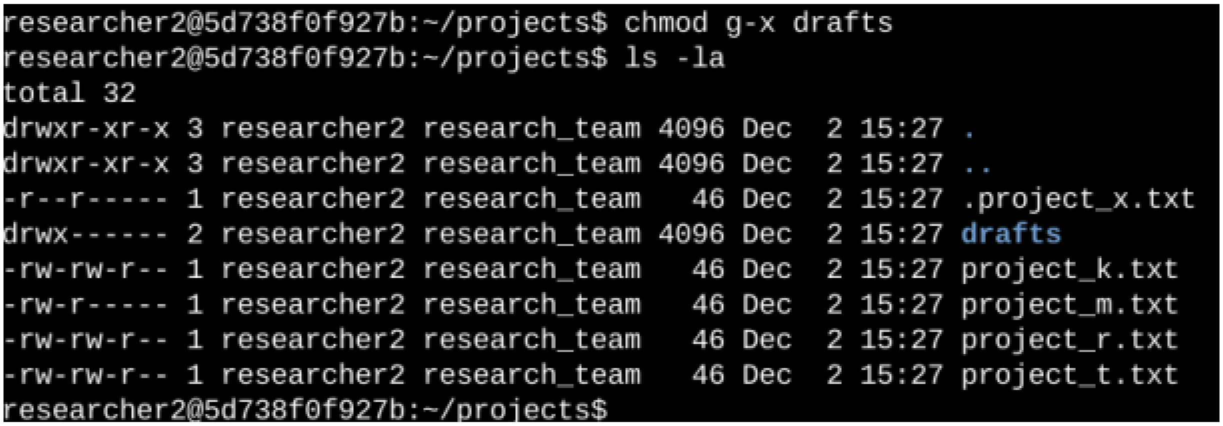
```
esearcher2@789f3050d95d:~/projects$ chmod u-w,g-w,g+r .project_x.txt
esearcher2@789f3050d95d:~/projects$ ls -la
otal 32
rwxr-xr-x 3 researcher2 research_team 4096 Jun 20 18:53 .
rwxr-xr-x 3 researcher2 research_team 4096 Jun 20 19:52 ..
r--r----- 1 researcher2 research_team   46 Jun 20 18:53 .project_x.txt
rwx--x--- 2 researcher2 research_team 4096 Jun 20 18:53 drafts
rw-rw-rw- 1 researcher2 research_team   46 Jun 20 18:53 project_k.txt
rw-r----- 1 researcher2 research_team   46 Jun 20 18:53 project_m.txt
rw-rw-r-- 1 researcher2 research_team   46 Jun 20 18:53 project_r.txt
rw-rw-r-- 1 researcher2 research_team   46 Jun 20 18:53 project_t.txt
esearcher2@789f3050d95d:~/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 researcher2 should have execute permissions.

The following code demonstrates how I used Linux commands to change the permissions:



```
researcher2@5d738f0f927b:~/projects$ chmod g-x drafts
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-r--r----- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/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. I then used the chmod command multiple times to change the permissions on files and directories.