File permissions in Linux

Project description

To ensure a secure system, the research team in my organization must revise the permissions for specific files and directories within the `projects` directory. Currently, these permissions do not align with the appropriate level of authorization. By inspecting and modifying these permissions, we guarantee system security. To accomplish this, I undertook the following steps:

Check file and directory details

The following code demonstrates how I used Linux commands to determine the existing permissions set for a specific directory in the file system.

```
researcher2@a0a0d2b66b9b:~$ ls
projects
researcher2@a0a0d2b66b9b:~$ cd projects/
researcher2@a0a0d2b66b9b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep 27 16:23 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep 27 16:59 ..
-rw--w---- 1 researcher2 research_team   46 Sep 27 16:23 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep 27 16:23 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Sep 27 16:23 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 27 16:23 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_t.txt
researcher2@a0a0d2b66b9b:~/projects$
```

We start off the lab in researcher2's directory. From there, I use the command `ls` to see the files inside of their directory, and it outputs the directory `projects`.

I then proceed to change the current working directory to the `projects` directory using the command `cd`. Once in the `projects` directory, I use the command `ls` again this time with the options `-la` which allows us to see a listed view of the output with things such as file permissions, file size, etc, as well as any hidden files.

The output indicates that there is one directory `drafts`, one hidden file titled `.project_x.txt` and then four other project files.

In the first column, the first 10 strings include the current file permissions for each of the files and one directory.

Describe the permissions string

The 10 strings each represent permissions of who can authorize certain files and what authorization they may have. The following below explain each of the 10 strings:

- **1st character**:
  - This character will be represented by either a `d` or a `-`. A `d` will represent a directory and a `-` will represent a file.
- **2nd-4th characters**:
  - These characters in this row represent the permissions as follows: read (`r`), write (`w`), and execute (`x`) for the permissions of the user. If there is a `-` instead, this will indicate the user does not have permissions for that field.
- **5th-7th characters**:
  - These characters in this row represent the permissions as follows: read (`r`), write (`w`), and execute (`x`) for the permissions of the group. If there is a `-` instead, this will indicate the group does not have permissions for that field.
- **8th-10th characters**:
  - These characters in this row represent the permissions as follows: read (`r`), write (`w`), and execute (`x`) for the permissions of other. What that means is this owner type will represent all other users/groups on the system. If there is a `-` instead, this will indicate others does not have permissions for that field.

Taking a look at the screenshot from the “check file and directory listing” segment, we can take a look at the permissions for `project_t.txt`. The permissions are as followed: `-rw-rw-r--`.

The first character is a `-`, indicating it is a file. Next, the 2nd-4th characters give the permissions for the owner of the file, in this case, the owner of the file can read (`r`), write (`w`), but not execute (`x`) as indicated by the `-`. Then, the 5th-7th characters represent the group file permissions. The group can read (`r`), write (`w`), but not execute (`x`) as indicated by the `-`. Lastly, the 8th to 10th characters represent others. Others only has read (`r`) permissions.

Change file permissions

The organization has stated that they do not want to allow other to have write access to any of the files.

```
drwxr-xr-x 3 researcher2 research_team 4096 Sep 27 16:23 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep 27 16:59 ..
-rw--w---- 1 researcher2 research_team   46 Sep 27 16:23 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep 27 16:23 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Sep 27 16:23 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 27 16:23 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_t.txt
researcher2@a0a0d2b66b9b:~/projects$
```

As shown above, the only file that the others group can write on is the file `project_k.txt`. In order to remove their write access, I will be using the command `chmod` which allows me to change file permissions of directories and files.

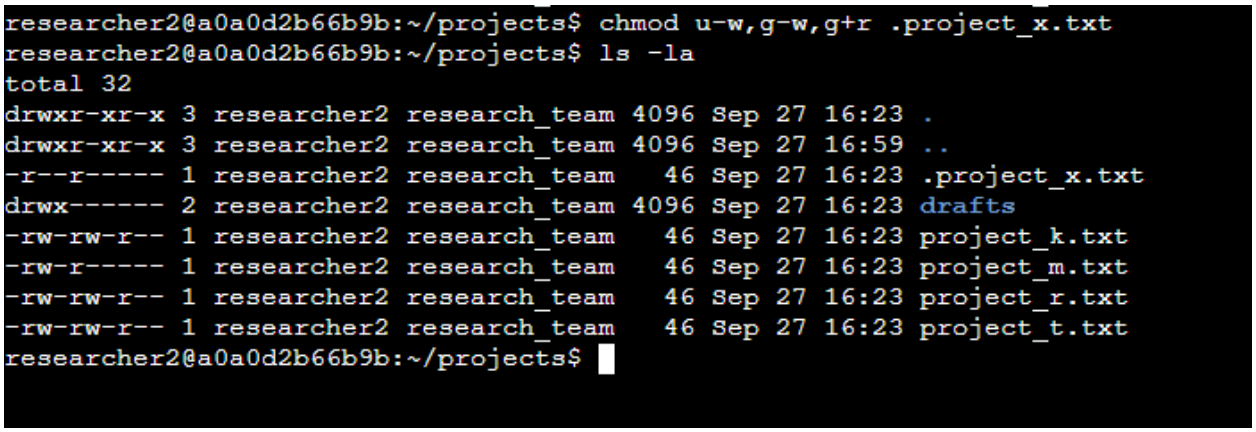
Inputting the full command and arguments I will type `chmod o-w project_k.txt`, where `chmod` is the command to change directory/file permissions. Next, `o` indicates I am changing the permission for other and `-w` indicating I am removing their write permission. Lastly `project_k.txt` being the file in which I am removing their write access on.

After removing write access for other, I use the command `ls -l` once again to show all the files in `projects`, as well as ful details, which include the updated user permissions. Now, the others group no longer has write access for `project_k.txt`, which can be seen in the image below.

```
drwxr-xr-x 3 researcher2 research_team 4096 Sep 27 16:23 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep 27 16:59 ..
-rw--w---- 1 researcher2 research_team   46 Sep 27 16:23 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep 27 16:23 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Sep 27 16:23 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 27 16:23 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_t.txt
researcher2@a0a0d2b66b9b:~/projects$ chmod o-w project_k.txt
researcher2@a0a0d2b66b9b:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Sep 27 16:23 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 27 16:23 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_t.txt
researcher2@a0a0d2b66b9b:~/projects$
```

Change file permissions on a hidden file

Recently a hidden file was archived titled `.project_x.txt`. My organization wants the file to have no write permissions, but the user and group may read it. We know that `.project_x.txt` is a hidden file because it starts with a "`.`".



Here I used the `chmod` command to remove the write access of the file using `u-w`. I then used `g-w` to remove the write access for the group. Next, I used g+r to give read access to the group, which it did not have before. Lastly I used `ls -la` to confirm the new permissions for the file.

Change directory permissions

My organization would like `researcher2` and only `researcher2` to be allowed to access the `drafts` directory and its contents. In order to do this, I will need to edit the execute (`x`) permissions. In the command line, I used `chmod` to edit drafts by using `g-x`. This removed access for group. Now, using `ls -l` you can see that only user (`researcher2`) has access to the drafts folder.

```
researcher2@a0a0d2b66b9b:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Sep 27 16:23 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 27 16:23 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_t.txt
researcher2@a0a0d2b66b9b:~/projects$ chmod g-x drafts
researcher2@a0a0d2b66b9b:~/projects$ ls -l
total 20
drwx------ 2 researcher2 research_team 4096 Sep 27 16:23 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 27 16:23 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 27 16:23 project_t.txt
researcher2@a0a0d2b66b9b:~/projects$
```

Summary

I used various Linux commands to check, edit, and confirm different permission levels for directories and files in `projects`. Using `chmod` I was able to edit the permission levels. Lastly, using `ls -la` I was able to see a list of the updated permissions for every file.