File permissions in Linux

Project description

This exercise was conducted as part of my Google Cybesecurity Professional course to simulate a companies request to manage user permissions via bash or CLI. The company requested that their permissions in the `projects` directory be updated to correspond with the level of authorization that the company deems appropriate. To conduct this change I took the following steps.

Check file and directory details

First, we had to check what the current permission settings were for the `projects` directory. To do this we utilized the following command `ls -la`. The screenshot shown below is what was displayed after the command was executed:

```
researcher2@e7a2bf62ba9f:~$ ls
projects
researcher2@e7a2bf62ba9f:~$ cd project
-bash: cd: project: No such file or directory
researcher2@e7a2bf62ba9f:~$ cd projects/
researcher2@e7a2bf62ba9f:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun  1 11:17 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun  1 12:44 ..
-rw--w---- 1 researcher2 research_team   46 Jun  1 11:17 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun  1 11:17 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jun  1 11:17 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun  1 11:17 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun  1 11:17 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun  1 11:17 project_t.txt
researcher2@e7a2bf62ba9f:~/projects$
```

I opted to utilize the `-la` option with the ls command to display the permissions for everything that was in the `projects` directory, including hidden ones such as `.project_x.txt`. This file is a hidden file because it contains a . at the beginning of the file name.

Describe the permissions string

On the left hand side, you can see a series of letters `(r)`, `(w)`, `(x)`, `(d)`, and hyphens `(-)`. These strings represent the permissions for each respective file and directory in the projects directory. Each character has a specific meaning which will be explained briefly for context.

The 1st character of the permissions string indicates whether the following is a directory or a file. If it is a file it will display with the letter `(d)`. If it is a normal file it will show as a hyphen `(-)`.

2-4 charactersdisplays the read `(r)`, write `(w)` and execute `(x)` permissions for the user. The absence of these letters is replaced with a hyphen `(-)` which indicates that this particular category cannot utilize that specific permission.

The 5-7 characters displays the read `(r)`, write `(w)` and execute `(x)` permissions for group. The absence of these letters is replaced with a hyphen `(-)` which indicates that this particular category cannot utilize that specific permission.

And finally, the 8-10 characters displays the read `(r)`, write `(w)` and execute `(x)` permissions for “others”. The absence of these letters is replaced with a hyphen `(-)` which indicates that this particular category cannot utilize that specific permission.

Change file permissions

To change the permissions for any of the directories you will need to either 1. Utilize bash as the root user (AKA super user) or utilize the super-user do command `(sudo)` to gain elevated authorization to alter permission settings for the files, or 2. Be the owner of the following files. In this example the user was the file owners which meant we were able to alter the permissions for the following files.

The company requested our team to remove the write`(w)` privilege from the “other” category for all of their files. To accomplish this we needed to change the permissions for `project_k.txt`.

I first utilized the chmod command with the argument o-w, indicating what specified which category `(o)`=other, and what I wanted to alter `(-w)`= remove write permission. Then my second argument was to specify which file I wanted this command to change in which i put `project_k.txt`.

After the command was executed II ran the `ls -la` command to ensure the changes were indeed made and as expected the write permission in the `project_k.txt` file for the other category has been changed successfully. The following screenshot revelas the changes and commands that I made

```
researcher2@e7a2bf62ba9f:~/projects$ chmod o-w project_k.txt
researcher2@e7a2bf62ba9f:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun  1 11:17 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun  1 12:44 ..
-rw--w---- 1 researcher2 research_team   46 Jun  1 11:17 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun  1 11:17 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun  1 11:17 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun  1 11:17 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun  1 11:17 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun  1 11:17 project_t.txt
researcher2@e7a2bf62ba9f:~/projects$
```

Change file permissions on a hidden file

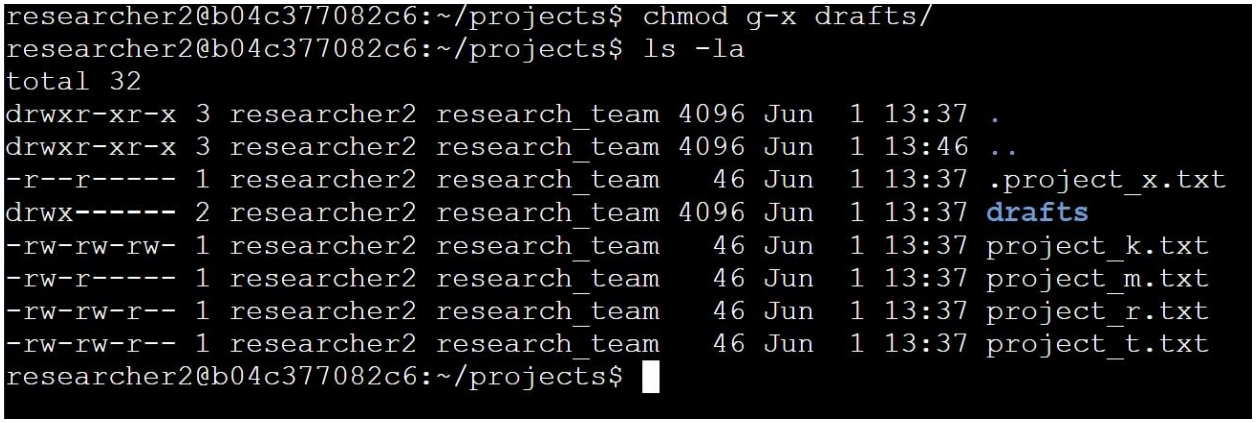
The company also request that our team to alter the permissions for the archived file `.project_x.txt` to have both the user and group categories to have read permissions only. In order to accomplish this we once again utilized the chmod command with the argument `u-w,g-w,g+r` for what category and changes we wanted to make, as well as `.project_x.txt` to specify which file we wanted this change to occur. After the command was executed, we double checked with the `ls -la` command to ensure the changes did take place. As shown in the screenshot below, the changes were successfully made.

```
researcher2@b04c377082c6:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@b04c377082c6:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun  1 13:37 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun  1 13:46 ..
-r--r----- 1 researcher2 research_team   46 Jun  1 13:37 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun  1 13:37 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jun  1 13:37 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun  1 13:37 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun  1 13:37 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun  1 13:37 project_t.txt
researcher2@b04c377082c6:~/projects$
```

Change directory permissions

Finally, the last change the organization wanted to make was to only have the current user researcher2 to have access to the `drafts` directory. Before the change the group category had execute permissions on the drafts directory. To change this we utilized the chmod command once more with the argument `g-x` and the corresponding directory we want the change to take place, `drafts/`. After executing the command, it is always the best practice to double check and confirm that the changes were made, so we once again use the `ls -la` command to check the permission settings for all the files and directories. As shown

in the screenshot below all changes were made successfully.



Summary

Utilizing bash (CLI) I was able to alter the user permissions for the files and directories in the `projects` directory, to one that was desired by the company. I started by first assessing the current user permissions with the `ls -la` command, and making the necessary changes with the `chmod` to change the permissions to our desired setting. After each change, we utilized the `ls -la` command to confirm that the changes did indeed take place.