File permissions in Linux

Project description

File permissions need to be updated for the research team since the current permissions do not have the appropriate level of authorization. Certain files and directories within the `projects` directory need to be checked and updated so that the system is secure. To that end, the following tasks were performed:

Check file and directory details

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

```
researcher2@5f0ee0b6867e:~$ cd projects
researcher2@5f0ee0b6867e:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Apr 29 13:00 .
drwxr-xr-x 3 researcher2 research_team 4096 Apr 29 13:33 ..
-rw--w---- 1 researcher2 research_team   46 Apr 29 13:00 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Apr 29 13:00 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Apr 29 13:00 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Apr 29 13:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 29 13:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 29 13:00 project_t.txt
researcher2@5f0ee0b6867e:~/projects$
```

The commands I entered is on the first and second lines and the other lines are the output of the second command. All contents of the `projects` directory are listed, including the hidden files because of the combination of the commands `ls-` and `-la`. The output indicates that there is one hidden file `.project_x.txt`, one directory `drafts`, and five more project files. The 10-character string in the first column represents the permissions set on each file or directory.

Describe the permissions string

The 10-character permissions string shows who is authorized to access a file and their specific permissions.
The first character is either a directory, indicated by (`d`), or a regular file, indicated by a hyphen (`-`), and shows the file type.

The second, third, and fourth characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for the user. If one of these characters is a hyphen (`-`), it means that the user does not have the respective permission.
The fifth, sixth, and seventh characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for the group. If one of these characters is a hyphen (`-`), it means that the group does not have the respective permission.
Lastly, the eighth, ninth, and tenth characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for other. If one of these characters is a hyphen (`-`), it means that other does not have the respective permission.
One could take the file permissions for `project_m.txt` as an example which are `-rw-rw-rw-`. Since the first character is a hyphen (`-`), this indicates that `project_m.txt` is a file, not a directory. The second and fifth characters are `r`, which indicates that user and grop have read permissions. The third character is `w`, which indicates that only the user has write permissions. No one has execute permissions for `project_m.txt`.

Change file permissions

Since other should not have write access to any of the organization's files, I determined that `project_k.txt` must have the its permissions changed.

In the following screenshot it can be seen how I achieved this.

```
researcher2@5f0ee0b6867e:~/projects$ chmod o-w project_k.txt
researcher2@5f0ee0b6867e:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Apr 29 13:00 .
drwxr-xr-x 3 researcher2 research_team 4096 Apr 29 13:33 ..
-rw--w---- 1 researcher2 research_team   46 Apr 29 13:00 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Apr 29 13:00 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Apr 29 13:00 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Apr 29 13:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 29 13:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 29 13:00 project_t.txt
researcher2@5f0ee0b6867e:~/projects$
```

Once again, the first two lines my input, and the other lines are the output of the second command. The `chmod` command allows me to change the permissions on files and directories. The first argument indicates what permissions are to be changed. In this case the first character (`o`) specifies that we are changing permissions for other, the second character (`-`) means that we are removing a certain permission from other, and the third character (`w`) refers to write access. After this, the command `ls -la` is used to review the updates I made.

Change file permissions on a hidden file

Because `project_x.txt` was archived recently, the research team does not want anyone to have write access to the file. However, they want the user and the group to have read access.

I used Linux to change permissions as demonstrated here:

```
researcher2@5f0ee0b6867e:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@5f0ee0b6867e:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Apr 29 13:00 .
drwxr-xr-x 3 researcher2 research_team 4096 Apr 29 13:33 ..
-r--r----- 1 researcher2 research_team   46 Apr 29 13:00 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Apr 29 13:00 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Apr 29 13:00 project_k.txt
-rw------- 1 researcher2 research_team   46 Apr 29 13:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 29 13:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 29 13:00 project_t.txt
researcher2@5f0ee0b6867e:~/projects$
```

As in the previous example, the first two lines are my input, and the other lines are the output of the second command. Because `.project_x.txt` starts with a period (`.`), I know that it is a hidden file. In the screenshot it can be seen how I removed write permissions for the user and the group using `u-w` and `g-w`, and then added read permissions for the group with `g+r`.

Change directory permissions

Since I was informed that only `researcher2` should have access to the `drafts` directory and its contents, I needed to make sure that only `researcher2` has execute permissions. This was, however, not the case since the group also has execute permissions to this directory.

I used Linux to remedy this as follows:

```
researcher2@5f0ee0b6867e:~/projects$ chmod g-x drafts
researcher2@5f0ee0b6867e:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Apr 29 13:00 .
drwxr-xr-x 3 researcher2 research_team 4096 Apr 29 13:33 ..
-r--r----- 1 researcher2 research_team   46 Apr 29 13:00 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Apr 29 13:00 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Apr 29 13:00 project_k.txt
-rw------- 1 researcher2 research_team   46 Apr 29 13:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 29 13:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 29 13:00 project_t.txt
researcher2@5f0ee0b6867e:~/projects$
```

The first two lines display my input and the rest display the output of the second command. Since the group should have execute permissions removed, I used the `chmod` command with the argument `g+x`. Nothing needed to be modified for the user `researcher2`.

Summary

Multiple permissions had to be changed to match the level of authorization the organization wanted for files and directories in the `projects` directory. Firstly, I used the `ls -la` command to check the permissions for the directory. I then used the `chmod` command to change the permissions on specific files and directories and then `ls -la` again to verify that my changes were successful.