File permissions in Linux

Project description

This project demonstrates how Linux can be used to check and modify directory and file access permissions for users, groups or others.

Check file and directory details

After navigating to the correct directory to check the command [ls -la] is used to display permissions to access files. This includes hidden files as well [a].

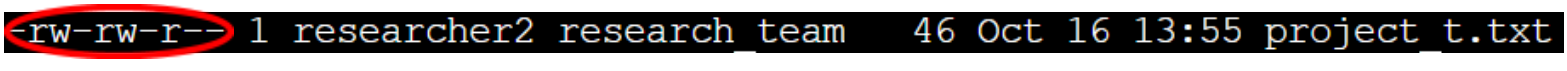
```
researcher2@faf016320716:~$ cd /home/researcher2/projects
researcher2@faf016320716:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 16 13:55 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 16 14:49 ..
-rw--w---- 1 researcher2 research_team   46 Oct 16 13:55 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 16 13:55 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 16 13:55 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 16 13:55 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 16 13:55 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 16 13:55 project_t.txt
researcher2@faf016320716:~/projects$
```

Here we can see that there are 5 files including one hidden files with the following permissions:

- .project_x.txt - users: read + write, group: write and others have no access.
- project_k.txt - users: read + write, group: read _ write, others: read + w.
- project_m.txt - users: read + write, group: read and others have no access.
- project_r.txt - users: read + write, group: read and write, others: read.
- project_t.txt - users: read + write, group: read + write, others: read.

Describe the permissions string

Take for example the final file listed project_t.txt:
The permissions string is circled in red. The ten character string represents either [r](read), [w](write), [x](execute). Given this is a file and not a directory [-] begins the string rather than [d]. Thereafter, each 3 character increment represents the user, group and other, in that order. So for example users have read and write permissions (characters 2-4), group also has read and write permissions (characters 5-7) and others can only read the file (characters 8-10). Nobody has permission to execute the file, hence why the final character in each increment is [-] rather than [x].



Change file permissions

It was noted that others had permission to write (file project_k.txt). In this instance, others should not have authorisation to write to files therefore permissions were modified so others could only read the file by implementing the [command chmod o-w] followed by the file name.

```
-rw-rw-rw- 1 researcher2 research_team   46 Oct 16 13:55 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 16 13:55 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 16 13:55 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 16 13:55 project_t.txt
researcher2@faf016320716:~/projects$ chmod o-w project_k.txt
```

Change file permissions on a hidden file

When looking at permissions it became apparent that the hidden file (.project_x.txt) has permissions for writing on by both user and group. These need to be restored to read-only. Here using the command [chmod u-r,g=r] followed by the file name removed writing permissions for the user (who had read and write access) and the group (who had write access) and modified both to read only.

```
researcher2@e750948c551c:~/projects$ chmod u-w,g=r .project_x.txt
```

Afterwards I checked to make sure the file permissions were modified correctly and it is certain that the file is now read only for users and group.

```
researcher2@e750948c551c:~/projects$ ls -l .project_x.txt
-r--r----- 1 researcher2 research_team 46 Oct 16 15:07 .project_x.txt
```

Change directory permissions

It became evident when checking permissions that the group had permission to execute[x] on the drafts folder. In this instance, only the user “researcher2”, should have this permission. I implemented the command [chmod g-x] followed by drafts to modify permissions to allow only the user to access the drafts folder.

```
researcher2@e750948c551c:~/projects$ chmod g-x drafts
```

Summary

This project was designed to utilise some of the commands I’ve learnt to modify permissions in Linux Bash for the user, group and others in order to make sure only the people who need certain permissions have them. With regards to cybersecurity, this works with the principle of least privilege in mind and is a necessity when it comes to protecting data.