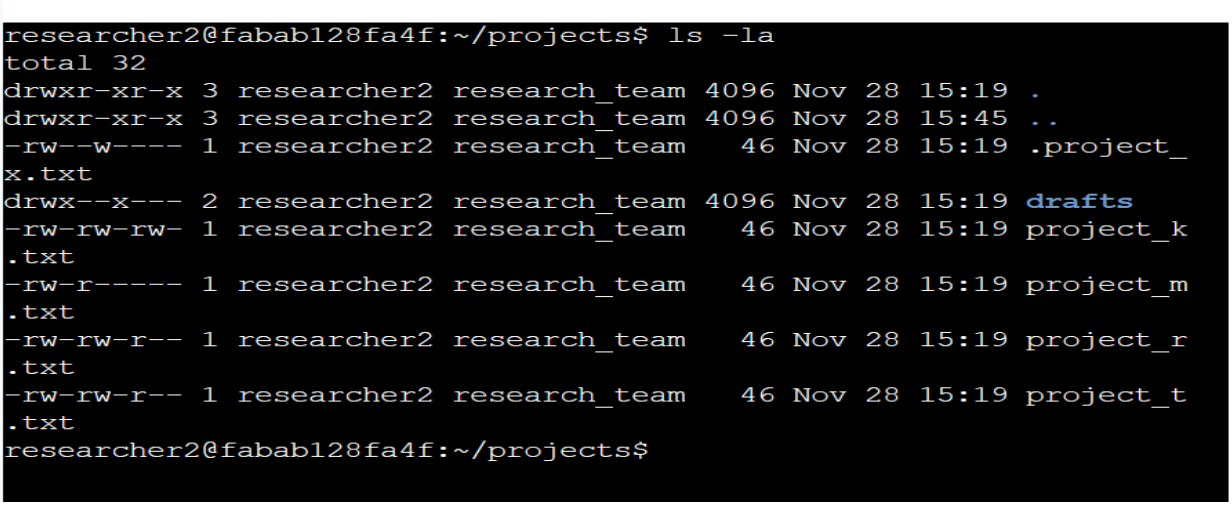
File permissions in Linux project

Description

I am in charge of security at a large organisation. Part of this role is to ensure users have the correct permissions to ensure the security of data within the organisations network. I have been tasked to manage the permissions of the research team members who have access to the projects directory. This is a vital step in ensuring that data is kept secure within the system and that all measures are up to date. Below are the steps which i followed to carry out this task

Check file and directory details

The following screenshot shows how I used LInux commands in the Bash shell to check the current permissions within the Projects directory. I utilised the ls -la code to show a list of all files contained within the directory, including any which have been hidden. I will be using the 10 character permission string at the beginning of each output to check current permissions



Permissions string

The 10-character permissions string located at the beginning of each output is used to determine which users permissions each user group has. The string is constructed as follows:

- The 1st character indicates the file type. A d indicates that this is a directory while a hyphen (-) indicates that it is a regular file
- The 2nd-4th characters indicate the read, write and execute permissions for the user. This is determined by the characters (r) for read permissions, (w) for write permissions and (x) for the execute permissions. A hyphen is used to indicate that the user has no permissions granted for this file/directory.
- The 5th-7th characters indicate the read, write and execute permissions for the group. A hyphen is used to indicate that the group has no permissions granted for this file/directory.
- The 8th-10th characters indicate the read, write and execute permissions for the owner type of "other". This consists of all other users on the system apart from the user and the group which we discussed previously. A hyphen is used to indicate that the other group has no permissions granted for this file/directory.

An example of this is included below.

```
drwxr-xr-x 3 researcher2 research_team 4096
```

This indicates that, as the first character is a d, this is a directory and not a file. The remaining characters show that the user has read (r) write (w) and execute (e) permissions for this directory. The group has read (r) and execute (e) permissions, but do not have write permissions which is indicated by the hyphen (-) in place of the write (w) character.

Change file permissions

The organization determined that the "other" group should not have write permissions to any of the files. When checking the returned list of permissions, I can see that the other grouping have wite permissions for Project_k.txt
In order to change these permissions, I will utilise the "chmod" command.

```
researcher2@fabab128fa4f:~/projects$ chmod o-w project_k.txt
researcher2@fabab128fa4f:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Nov 28 15:19 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 28 15:45 ..
-rw--w---- 1 researcher2 research_team   46 Nov 28 15:19 .project_
x.txt
drwx--x--- 2 researcher2 research_team 4096 Nov 28 15:19 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 15:19 project_k
.txt
-rw-r----- 1 researcher2 research_team   46 Nov 28 15:19 project_m
.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 15:19 project_r
.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 15:19 project_t
.txt
researcher2@fabab128fa4f:~/projects$
```

The chmod command is used to change permissions on a file or directory. The first argument indicates what permissions should be changed, and the second argument specifies the file or directory. In this example, I removed write permissions from “other” for the project_k.txt file. In order to check if this had been carried out correctly, I repeated the ls -la command in order to list all contents and their permissions again. We can see from the screenshot above that the “other” grouping no longer has write permissions for the Project_k.txt file.

Change file permissions on a hidden file

The research team at my organization recently archived project_x.txt. We can tell this is now a hidden file as it has a . before the file name (.project_x.txt) No users should have write access to this file at all, however, the user and group should have permissions to read the file.
The following code demonstrates how I used Linux commands to change the permissions to match the criteria above:

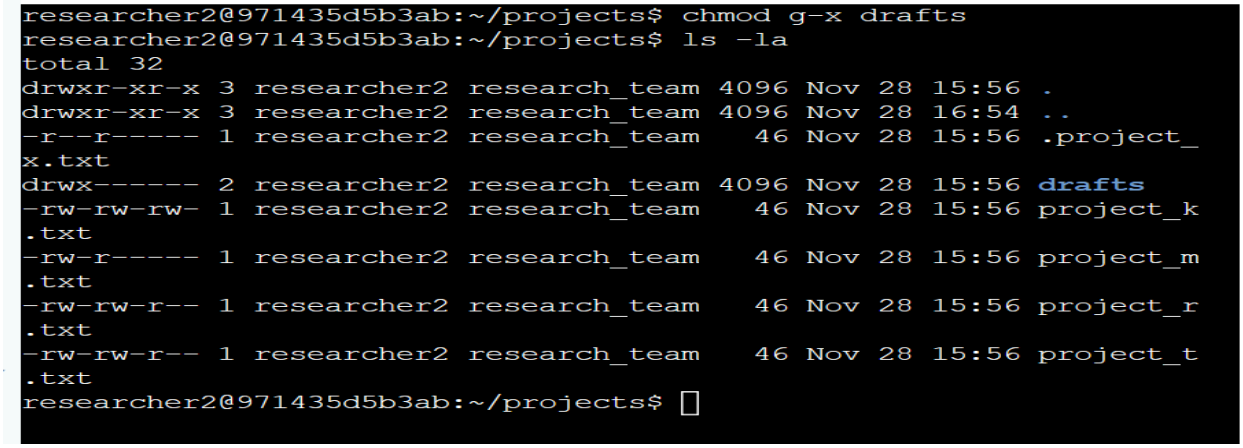
```
researcher2@971435d5b3ab:~/projects$ chmod u-w,g-w,g+r .project_x.
txt
researcher2@971435d5b3ab:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Nov 28 15:56 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 28 16:54 ..
-r--r----- 1 researcher2 research_team   46 Nov 28 15:56 .project_
x.txt
drwx--x--- 2 researcher2 research_team 4096 Nov 28 15:56 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Nov 28 15:56 project_k
.txt
-rw-r----- 1 researcher2 research_team   46 Nov 28 15:56 project_m
.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 15:56 project_r
.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 15:56 project_t
.txt
researcher2@971435d5b3ab:~/projects$
```

I used the chmod command to modify the permissions for this file. As shown in previous screenshots, I needed to remove write permission from both the user and group, whilst adding read permissions to group. I completed this by using the following command:
chmod u-w,g-w,g+r and confirming the file to carry this out to by finishing with .project_x.txt

Change directory permissions

My organization has stated that only “researcher2” should have access to the drafts directory and its contents. This means that only “researcher2” should have execute permissions for this directory and the user and group should have no permissions at all.

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



As the group previously had execute permissions for the drafts directory, I was required to remove these permissions. I used the chmod command to do so. The command which i utilised was chmod g-x drafts. This removes the execute permissions from the group, leaving only the user with execute permissions as required.

Summary

I have carried out permissions checks on the projects section of my organisations network. I used ls -la commands to check both the contents and permissions of the projects directory. I was then able to cross reference this with the organisations instructions to ensure that everything matched and was correct. I then used the chmod command to rectify any issues and change the permissions to the correct rules as per the organisations request.
Upon completion, reusing the ls- la command allows me to confirm all permissions are now correct.