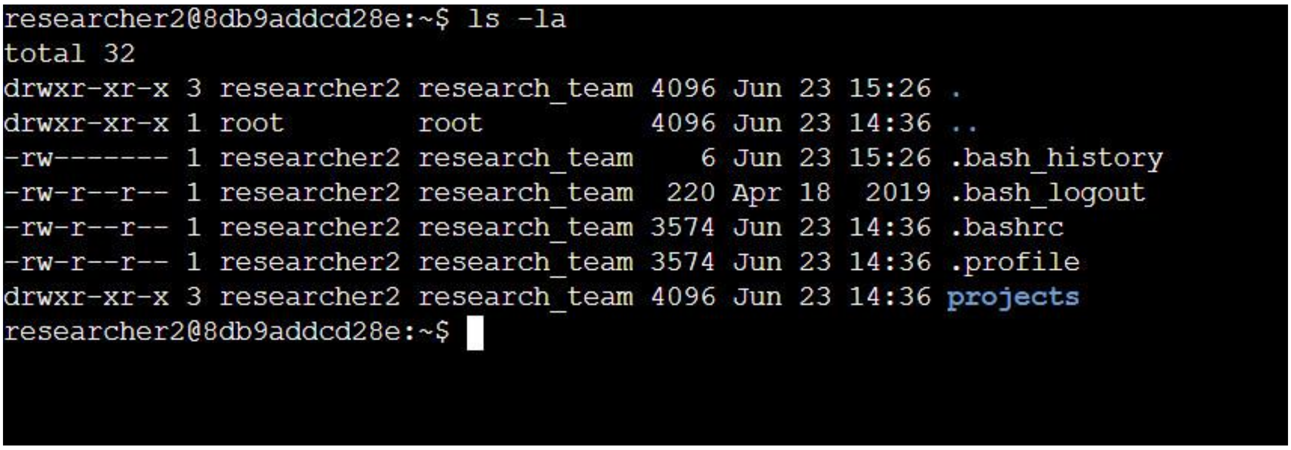
File permissions in Linux

Project description

In this portfolio, I am considered a security professional at a large organization. I mainly work with the research team. As part of my job, I should provide users of the research team with correct authorization and appropriate permissions to the files and directories in `projects` directory they use in their work. This helps create a secure system for storing the results of their research. To accomplish this task, I took the following actions:

Check file and directory details

To review the current situation and evaluate the existing permissions assigned to each item in the `projects`, I first checked the directory. In the screenshot below, you can see the Linux commands I used to display all files and directories, including hidden ones.
The table provides information about the user, group, and others, as well as the access level for each item in the list.



```
researcher2@8db9addcd28e:~$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 23 15:26 .
drwxr-xr-x 1 root        root          4096 Jun 23 14:36 ..
-rw------- 1 researcher2 research_team    6 Jun 23 15:26 .bash_history
-rw-r--r-- 1 researcher2 research_team  220 Apr 18  2019 .bash_logout
-rw-r--r-- 1 researcher2 research_team 3574 Jun 23 14:36 .bashrc
-rw-r--r-- 1 researcher2 research_team 3574 Jun 23 14:36 .profile
drwxr-xr-x 3 researcher2 research_team 4096 Jun 23 14:36 projects
researcher2@8db9addcd28e:~$ 
```

Describe the permissions string

As you may notice in the screenshot above, the 10-character string at the beginning of each entry contains specific symbols. These characters indicate to a specialist what set of permissions each file or directory has. When a file has this string fully filled out as `-rwxrwxrwx`, it means there are no access restrictions for the user, group, or others. This is the most dangerous combination of permissions in terms of security.

Let's consider the meaning of each character in the 10-character string and how it affects authorization:
The first character indicates whether the item is a file or a directory. If it's a `d`, it represents a directory; a hyphen `-` indicates a regular file.
There are three types of letters that appear in the fully filled string `rwxrwxrwx`, each repeated three times. Each group corresponds to a specific category: `u` – user, `g` – group, and `o` – others.
The first three characters refer to the user category, the second three to the group, and the last three to others.
Each of the `rwx` characters has its own specific meaning:

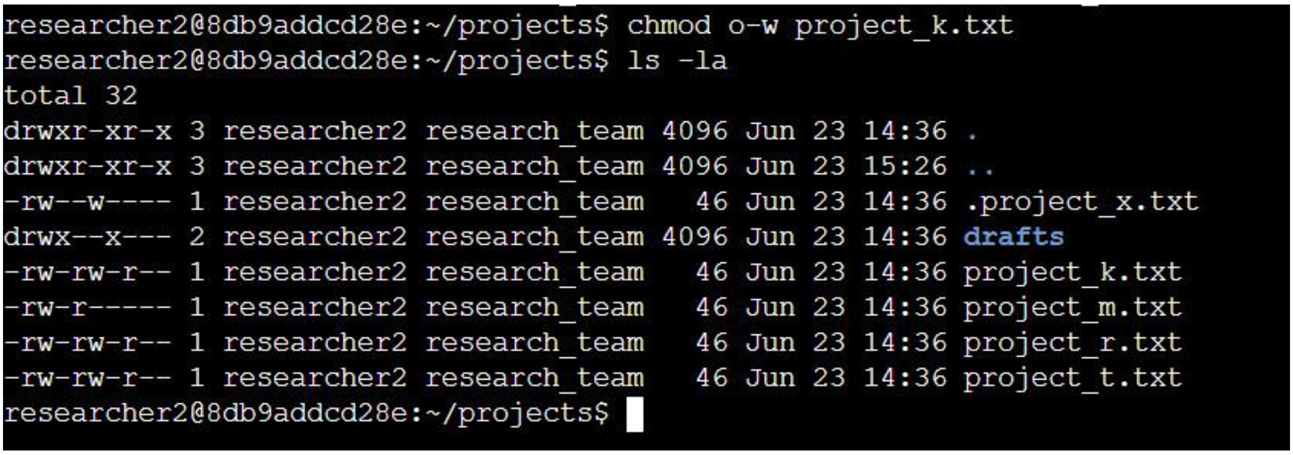
- `r` stands for read permission
- `w` stands for write permission
- `x` stands for execute permission

In cases where one or more (or even all) of the `rwx` characters are replaced by a hyphen `-`, it indicates that permission for that category is not granted, and the user, group, or others have no access to the file or directory.

Additionally, to this 10-character marks in the screenshot you can also see two blocks of information. Next after 10-character string goes the user (e.g. `researcher2`) who owns the file and then at the third block of text is the group owner of the file (e.g. `research_team`).

Change file permissions

It is also important to close the writing permissions for others to the files of the research team. To finish this task, I have made next steps:
I decided to close writing access for `others` at the `project_k.txt` file with the help of the Linux command `chmod`.

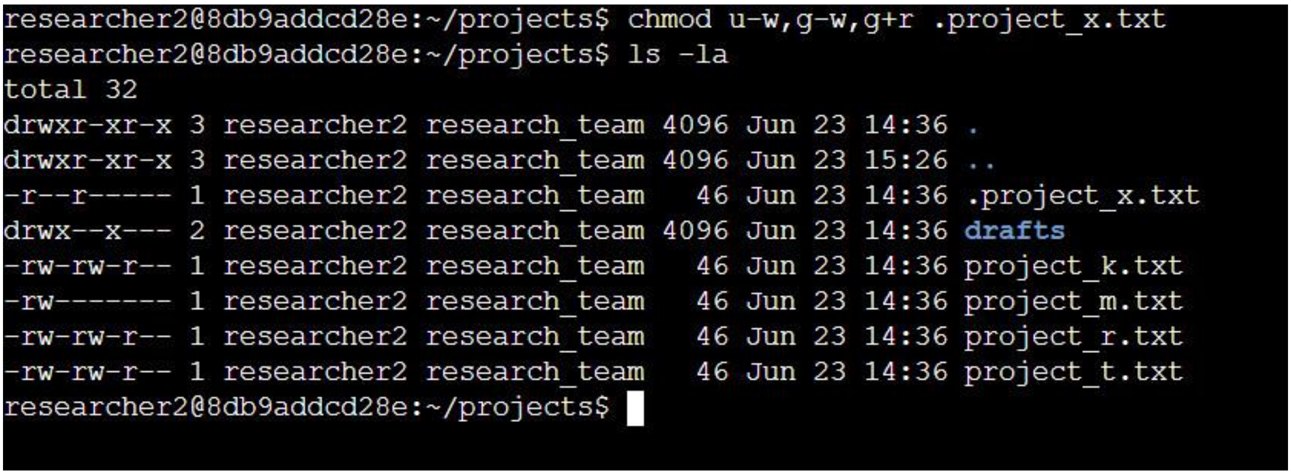


```
researcher2@8db9addcd28e:~/projects$ chmod o-w project_k.txt
researcher2@8db9addcd28e:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 23 14:36 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 23 15:26 ..
-rw--w---- 1 researcher2 research_team   46 Jun 23 14:36 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 23 14:36 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun 23 14:36 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun 23 14:36 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 23 14:36 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 23 14:36 project_t.txt
researcher2@8db9addcd28e:~/projects$
```

In the screenshot above, I ran the command `chmod o-w project_k.txt` in the terminal. This command removed the write permission from the others category for the file `project_k.txt`.
The first argument specified the category where permissions would be changed: `u`, `g` or `o`. This argument has options `+` (to add), `=` (to assign) or `-` (to remove) to indicate what to do with permissions. Also, it has the option to choose the type of permissions `r` (to read), `w` (to write) or `x` (to execute). The second argument contains the file name to which new permissions should be applied.
It is important to keep in mind, that options in argument should be written without spaces between them. Especially when making multiple permission changes in one command.
In this case, I have removed the access to the file `project_k.txt` for the category of `others`. To check the results, I entered the `ls -la` command and viewed the updated list with the new permission indicators.

Change file permissions on a hidden file

There is a hidden file in the `projects` directory, that was hidden by the research team to avoid unnecessary changes to its content. They recently archived the `project_x.txt` file, but it is important to preserve read permissions for the user and the group.
To manage this task, I entered the following command: `chmod u-w,g-w,g+r project_x.txt`

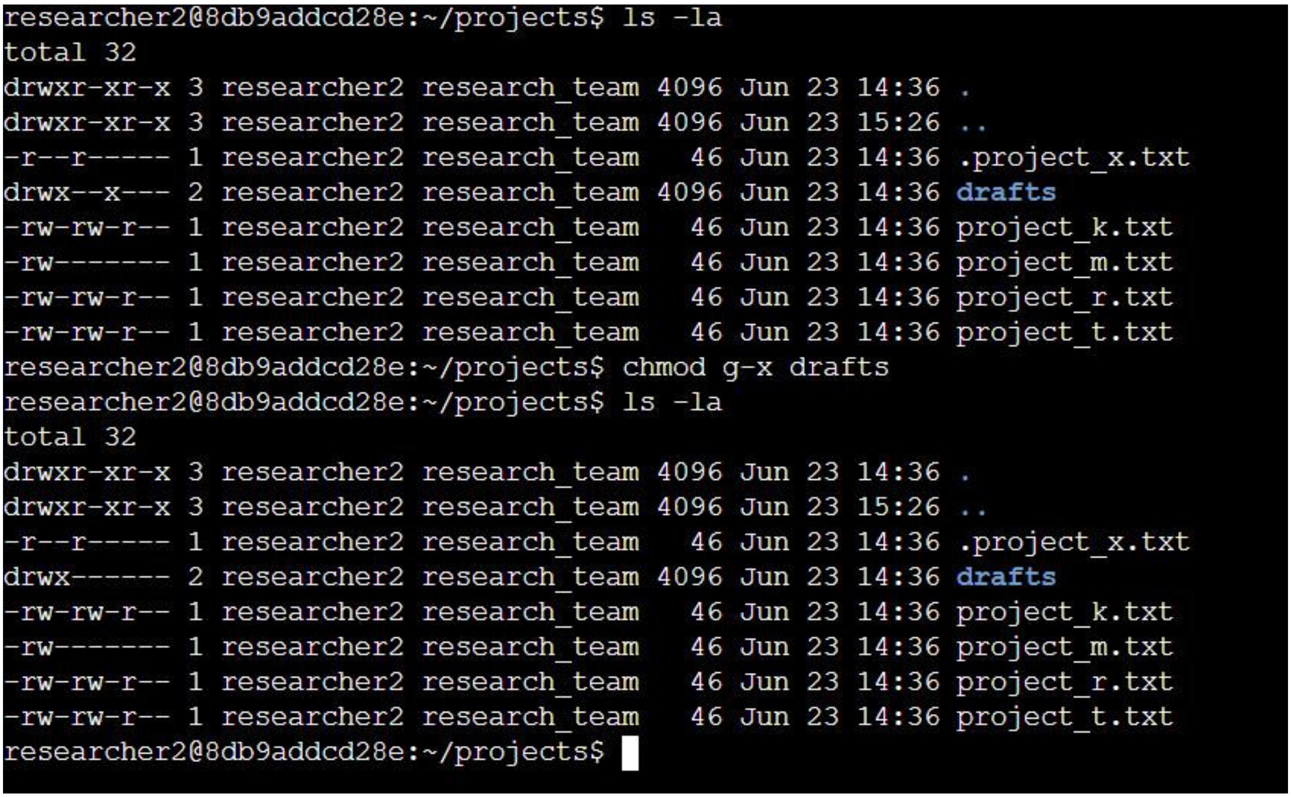


```
researcher2@8db9addcd28e:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@8db9addcd28e:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 23 14:36 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 23 15:26 ..
-r--r----- 1 researcher2 research_team   46 Jun 23 14:36 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 23 14:36 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun 23 14:36 project_k.txt
-rw------- 1 researcher2 research_team   46 Jun 23 14:36 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 23 14:36 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 23 14:36 project_t.txt
researcher2@8db9addcd28e:~/projects$ 
```

The file `.project_x.txt` is a hidden file, that indicated by a leading period (`.`). To locate this file I used the `-la` option with the `ls` command.
In the two lines above, you can see the commands to change permissions of the file and to list the updated results.
The line of the `.project_x.txt` file shows that permissions were changed: write permissions were removed from the user (`u-w`) and group (`g-w`) categories. However, However, read permission was added for the group category (`g+r`).

Change directory permissions

The research team decided to leave access to the `drafts` directory and its inner files only for `researcher2`. The task implies that execute permissions are allowed only for `researcher2`. At the following screenshot I demonstrate how, with the help of Linux commands, I completed this task:



```
researcher2@8db9addcd28e:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 23 14:36 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 23 15:26 ..
-r--r----- 1 researcher2 research_team   46 Jun 23 14:36 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 23 14:36 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun 23 14:36 project_k.txt
-rw------- 1 researcher2 research_team   46 Jun 23 14:36 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 23 14:36 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 23 14:36 project_t.txt
researcher2@8db9addcd28e:~/projects$ chmod g-x drafts
researcher2@8db9addcd28e:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 23 14:36 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 23 15:26 ..
-r--r----- 1 researcher2 research_team   46 Jun 23 14:36 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Jun 23 14:36 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun 23 14:36 project_k.txt
-rw------- 1 researcher2 research_team   46 Jun 23 14:36 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 23 14:36 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 23 14:36 project_t.txt
researcher2@8db9addcd28e:~/projects$ 
```

The screenshot displays the process of changing access mode. Line 7 shows the directory `drafts` with permissions for the user, group, and others categories before and after my changes. I used the `chmod` command to remove execute permissions for the g category: `chmod g-x drafts`. It was not necessary to explicitly add the execute permissions for `researcher2`, because they already existed at the `draft` directory.

Summary

I modified multiple permissions to align with the authorization level required by my organization for the files and directories within the `projects` directory. As a first step, I ran the `ls -la` command to review the current permissions. This helped guide my decisions in the subsequent steps. I then used the `chmod` command several times to adjust the permissions on specific files and directories accordingly.