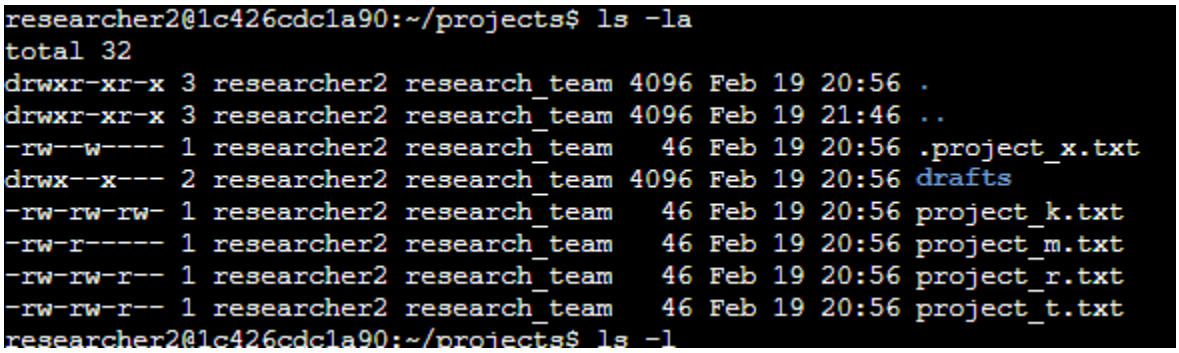
File permissions in Linux

Project description

The research team at my organization wants to ensure robust security mechanisms are implemented by updating file permissions within the `project` directory. The current permissions do not reflect the level of authorization they should be given. Checking and updating file permissions will help keep them secure. To ensure a more tightened security access control, these are the tasks I completed:

Describe the permissions string

The following Linux commands illustrate how I checked the file system's current file permissions in the given directory.



The first line of commands in the screenshot outputs all files and hidden files in the `projects` directory. I combined the command `ls` with the option `-la` to achieve this outcome. The output shows one subdirectory called `drafts`, one hidden file called `.project_x.txt`, and five other subdirectories. On the left, the columns in a string of text illustrate the permissions set for these directories and files.

Describe the permissions string

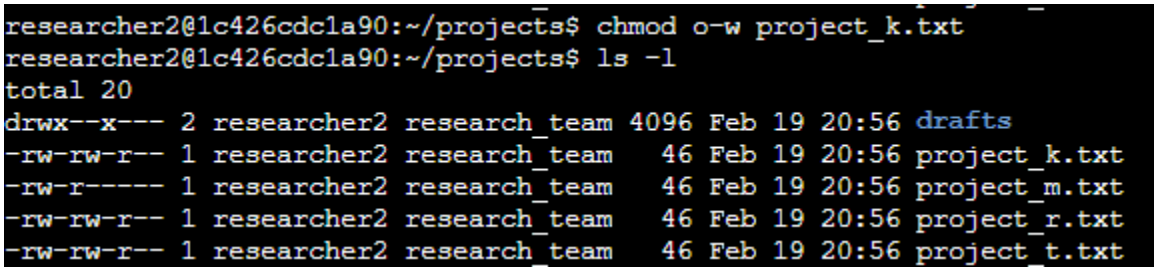
The 10-character string can be deconstructed to determine who is authorized to access the file and their specific permissions. The characters and what they represent are as follows:

- **1st character**: This character is either a `d` or a hyphen (`-`) and indicates the file type. If it's a `d`, it's a directory. If it's a hyphen (`-`), it's a regular file.

- **2nd-4th characters**: These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for the user. When one of these characters is a hyphen (`-`) instead, it indicates that this permission is not granted to the user.
- **5th-7th characters:** These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for the group. When one of these characters is a hyphen (`-`) instead, it indicates that this permission is not granted for the group.
- **8th-10th characters:** These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for other. This owner type consists of all other users on the system apart from the user and the group. When one of these characters is a hyphen (`-`) instead, that indicates that this permission is not granted for other.

Change file permissions

The organization wants to ensure none of the files allow `other` users to write to files. This policy is to revoke unauthorized access to files and strengthen the security posture of the organization.



The commands used to change the following permissions are `chmod` which took two arguments `o-w` which specifies the user and permissions, and `project_k.txt` which is the name of the file in which permissions need to be updated. I removed the write permissions and used the commands `ls -l` to review updates.

Change file permissions on a hidden file

The research team at the organization prefers the hidden file `.project_x.txt` to stay hidden but allows but allow the users and groups to be able to have permission read the context of this file.

```
researcher2@1c426cdc1a90:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb 19 20:56 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb 19 21:46 ..
-rw--w---- 1 researcher2 research_team   46 Feb 19 20:56 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Feb 19 20:56 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 20:56 project_k.txt
-rw------- 1 researcher2 research_team   46 Feb 19 20:56 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 20:56 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 20:56 project_t.txt
researcher2@1c426cdc1a90:~/projects$ chmod u-w,g-w,g+r .project_x.txt
```

To check the current permissions of the hidden file, I used the command `ls -la` which displayed the list of directories and hidden files in the current directory. The results illustrated that the user and group had write permissions to the file. I immediately used the `chmod` command and passed in a few arguments to revoke the group and user permissions from writing to the content of the file and updated them to have read permissions only by using the commands `chmod u-w,g-w,g+r`.

Change directory permissions

The organization wants only `researcher2` to be granted access to the directory drafts and their content. This means only `researcher2` can have executed permissions on this directory and no one else.

```
researcher2@1c426cdc1a90:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Feb 19 20:56 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 20:56 project_k.txt
-rw------- 1 researcher2 research_team   46 Feb 19 20:56 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 20:56 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 19 20:56 project_t.txt
researcher2@1c426cdc1a90:~/projects$ chmod g-x drafts
```

On the first line in the screenshot, the commands I entered were `ls -l` which printed out files and directories in the `projects` directory. The results showed `group` users had execute permissions for the directory. To remove their permissions, I used the command `chmod` together with `g-x` and the name of the directory drafts to change the current permissions of the directory.

Summary

In conclusion, I changed multiple permissions on directories and files to meet the organization's security measures on the `projects` directory. The command `ls` with the option -l displays

files and directories in a directory or sub-directory. Additionally, I used the `chmod` command to update permissions by passing in a few arguments.