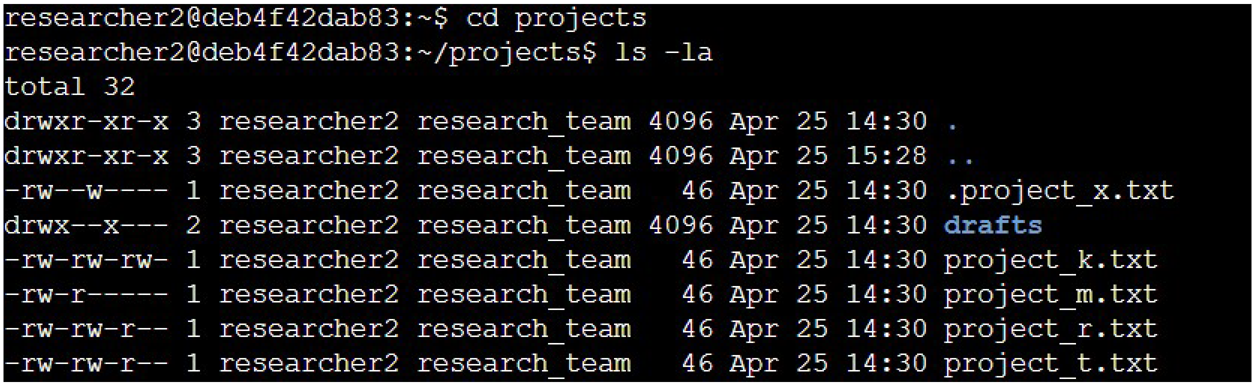
Project description

I am a security professional at a large organization. I mainly work with their research team. Part of my job is to ensure users on this team are authorized with the appropriate permissions. This helps keep the system secure.

My task is to examine existing permissions on the file system. I'll need to determine if the permissions match the authorization that should be given. If they do not match, I'll need to modify the permissions to authorize the appropriate users and remove any unauthorized access. To accomplish this, I performed the following tasks:

Check file and directory details



```
researcher2@deb4f42dab83:~$ cd projects
researcher2@deb4f42dab83:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Apr 25 14:30 .
drwxr-xr-x 3 researcher2 research_team 4096 Apr 25 15:28 ..
-rw--w---- 1 researcher2 research_team   46 Apr 25 14:30 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Apr 25 14:30 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Apr 25 14:30 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Apr 25 14:30 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 25 14:30 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr 25 14:30 project_t.txt
```

The two lines of the screenshot show me navigating to the `projects` directory and then commanding the system to show me all directories and files (even hidden ones) and their permissions. What follows is the output. I accomplished this using the `la` command with the `-la` option, which displays all directories and files, even hidden ones, with their permissions. The output of my command indicates there are two hidden directories, one of which contains the hidden file `.project_x.txt`, there is also the `drafts` directory and 4 other files. The far-left column contains 10-character strings, these represent the permissions for each directory or file.

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:

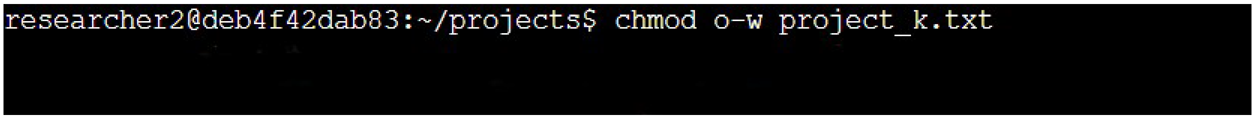
- **1st character**: This character is either a `d` or hyphen (`-`) and indicates the file type. If it's a `d`, it's a directory. If it's a hyphen (`-`), it's a regular file.
- **2nd-4th characters**: These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for the user. When 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.

For example, directory permissions for the `.` directory are `drwxr-xr-x`. The first character is a `d`, which designates this as a directory rather than a file. The second, fifth and eighth character is an `r` which means that the user, group and other all have read permissions, The third character is a `w`, which means that the user has write permissions, but the sixth and ninth character is a (`-`) which means the group and other do not have write permissions. The fourth, seventh, and tenth character is an `x`, which means that the user, group, and other all have the execute permissions.

Change file permissions

The organization determined that others should not have write access to any files. This means that `project_k.txt` needed to have its permissions modified. I accomplished this as follows:

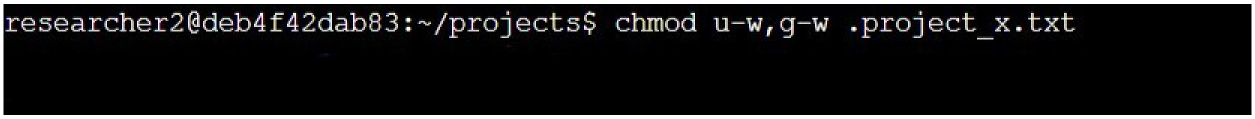


```
researcher2@deb4f42dab83:~/projects$ chmod o-w project_k.txt
```

I used the `chmod` command to change the permissions of files and directories. I used the argument `o-w` which means that other (`o`) should have their write permissions (`w`) removed (`-`).

Change file permissions on a hidden file

The research team had archived `.project_x.txt`, which is why it's a hidden file. Because of this, they wanted write permissions removed for anyone, but wanted the user and group to able to still read the file. I used the appropriate Linux command, `chmod`, to assign `.project_x.txt` the appropriate authorization.

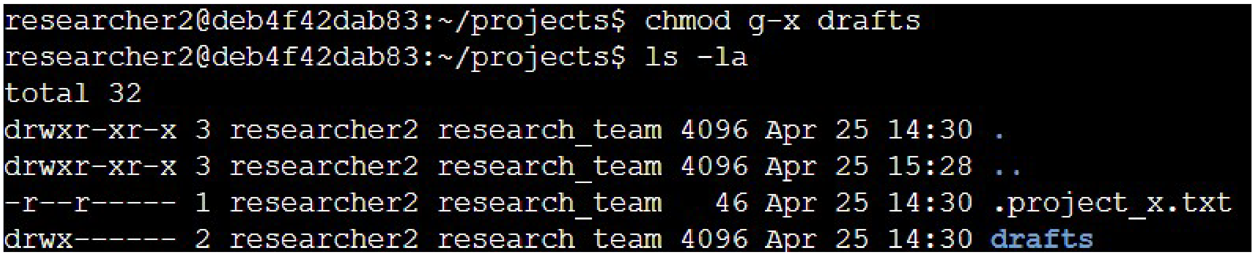


```
researcher2@deb4f42dab83:~/projects$ chmod u-w,g-w .project_x.txt
```

Both the user and the group still had write permissions. By using the `chmod` command, I removed the write permissions for the user and the group for only the `.project_x.txt` file.

Change directory permissions

The files and directories in the `projects` directory belong to the `researcher2` user. Only `researcher2` should be allowed to access the `drafts` directory and its contents. I used the appropriate Linux command, `chmod`, to modify the permissions accordingly.



```
researcher2@deb4f42dab83:~/projects$ chmod g-x drafts
researcher2@deb4f42dab83:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Apr 25 14:30 .
drwxr-xr-x 3 researcher2 research_team 4096 Apr 25 15:28 ..
-r--r----- 1 researcher2 research_team   46 Apr 25 14:30 .project_x.txt
drwx------ 2 researcher2 research team 4096 Apr 25 14:30 drafts
```

In the first line I remove the execute permissions from the group for the `drafts` directory. I then used the `ls` command with the `-la` option to verify my input.

Summary

Per my instructions, I examined the directories, files, and their permissions in the `projects` directory of the system. I then modified the permissions to the given specifications. I changed file permissions, including the permissions of a hidden file and I modified the permissions of the `draft` directory.