File permissions in Linux

Project description

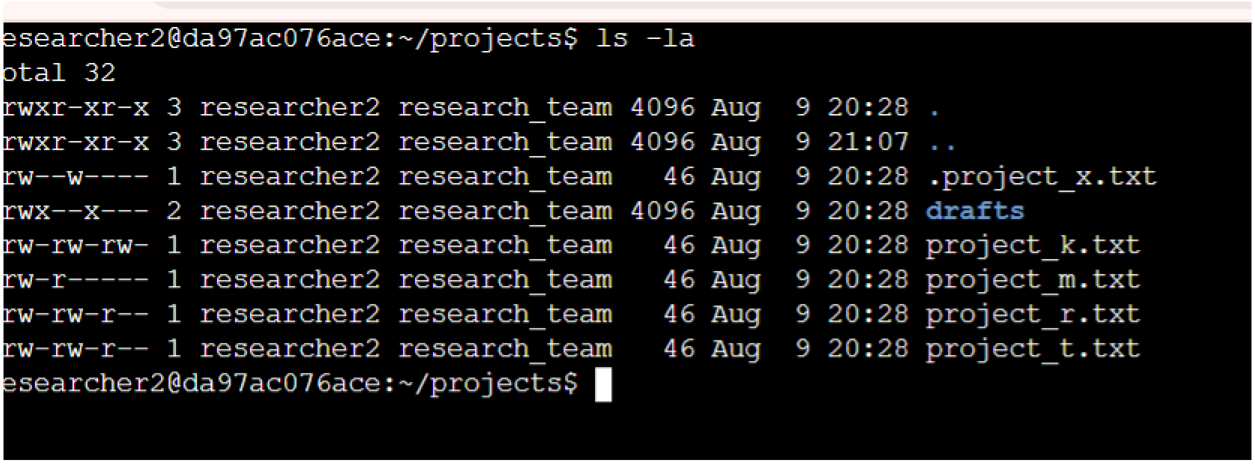
As part of a security review, I examined the permissions for various files and directories located in the `projects` directory. Some permissions were not set according to the organization's security guidelines, which could allow unauthorized modifications. I used a series of Linux commands to verify and adjust these permissions so that they met the correct access requirements.

Checking File and Directory Information

I started by listing all files and directories—including hidden ones—along with their permissions, ownership, and group details. The output revealed:

- A directory named `drafts`.
- A hidden file called `.project_x.txt`.
- Several other project-related files.

The first column displayed a 10-character string that specifies the type of each item and its permissions.



```
esearcher2@da97ac076ace:~/projects$ ls -la
otal 32
rwxr-xr-x 3 researcher2 research_team 4096 Aug  9 20:28 .
rwxr-xr-x 3 researcher2 research_team 4096 Aug  9 21:07 ..
rw--w---- 1 researcher2 research_team   46 Aug  9 20:28 .project_x.txt
rwx--x--- 2 researcher2 research_team 4096 Aug  9 20:28 drafts
rw-rw-rw- 1 researcher2 research_team   46 Aug  9 20:28 project_k.txt
rw-r----- 1 researcher2 research_team   46 Aug  9 20:28 project_m.txt
rw-rw-r-- 1 researcher2 research_team   46 Aug  9 20:28 project_r.txt
rw-rw-r-- 1 researcher2 research_team   46 Aug  9 20:28 project_t.txt
esearcher2@da97ac076ace:~/projects$ 
```

Understanding the Permissions String

This string is broken down as follows:

- The first character indicates the type: `d` for directory, `-` for a regular file.
- Characters 2 to 4 represent the user's permissions (read, write, execute).
- Characters 5 to 7 represent the group's permissions.
- Characters 8 to 10 represent the permissions for all other users.

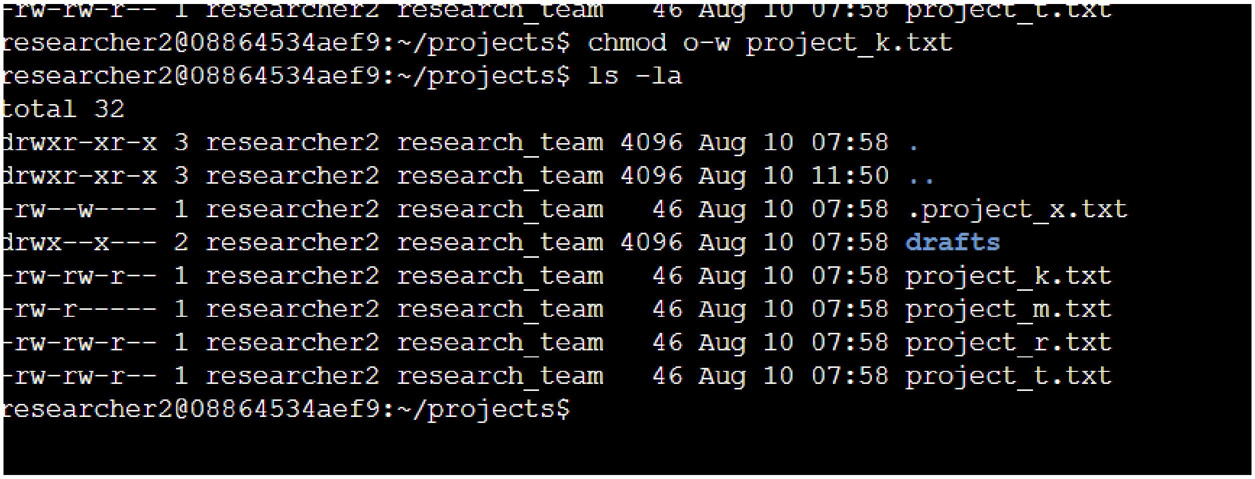
For example, the permission string `-rw-rw-r--` means:

- The user has read and write permissions.
- The group has read and write permissions.
- Others have read-only access.

A close-up of the permissions string explanation, or highlighting a specific file's permissions.

Modifying File Permissions

I noticed that the file `project_k.txt` allowed users outside the owner and group to write to it, which is against policy. I removed write permission for others to restrict modification rights solely to the owner and group, while still allowing others to read the file.

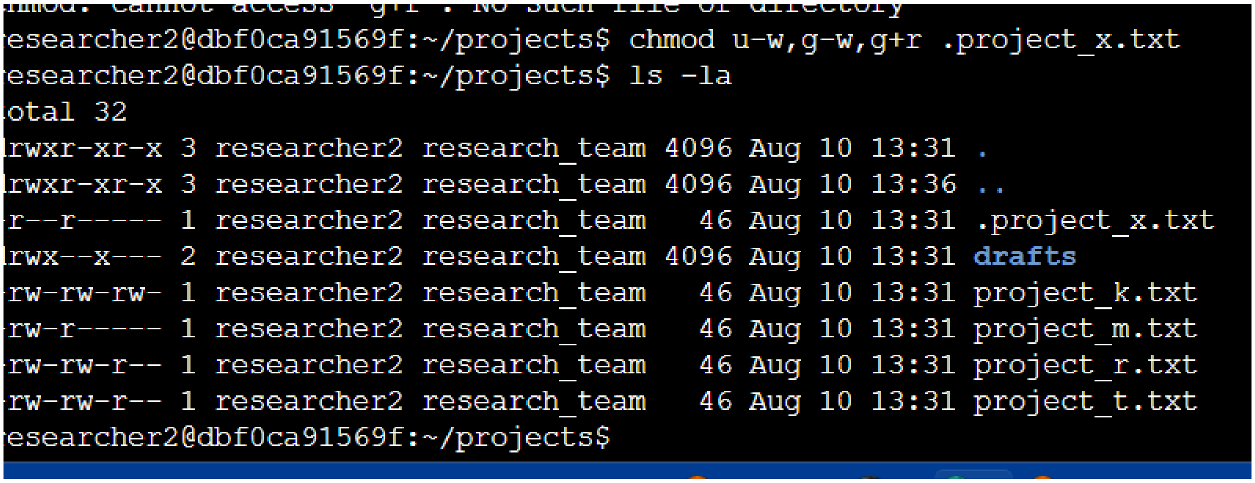


```
-rw-rw-r-- 1 researcher2 research_team   46 Aug 10 07:58 project_t.txt
researcher2@08864534aef9:~/projects$ chmod o-w project_k.txt
researcher2@08864534aef9:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 10 07:58 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 10 11:50 ..
-rw--w---- 1 researcher2 research_team   46 Aug 10 07:58 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug 10 07:58 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Aug 10 07:58 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 10 07:58 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 10 07:58 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 10 07:58 project_t.txt
researcher2@08864534aef9:~/projects$
```

- Screenshot showing the command used to remove write permission from others on `project_k.txt` and the output verifying the change.

Adjusting Permissions for a Hidden File

The archived hidden file `.project_x.txt` should be readable by the owner and group but should not be writable by anyone. I updated its permissions accordingly, ensuring write permissions were removed for both user and group, and read permission was granted to the group if not already set. I verified these changes through a permission listing.

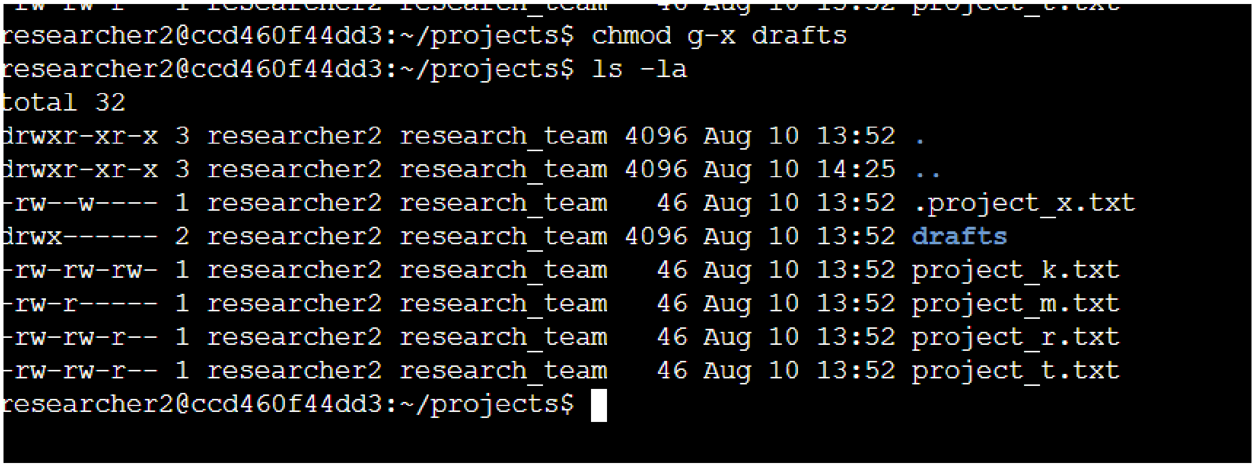


```
hmod: cannot access 'g+r': No such file or directory
esearcher2@dbf0ca91569f:~/projects$ chmod u-w,g-w,g+r .project_x.txt
esearcher2@dbf0ca91569f:~/projects$ ls -la
otal 32
rwxr-xr-x 3 researcher2 research_team 4096 Aug 10 13:31 .
rwxr-xr-x 3 researcher2 research_team 4096 Aug 10 13:36 ..
r--r----- 1 researcher2 research_team   46 Aug 10 13:31 .project_x.txt
rwx--x--- 2 researcher2 research_team 4096 Aug 10 13:31 drafts
rw-rw-rw- 1 researcher2 research_team   46 Aug 10 13:31 project_k.txt
rw-r----- 1 researcher2 research_team   46 Aug 10 13:31 project_m.txt
rw-rw-r-- 1 researcher2 research_team   46 Aug 10 13:31 project_r.txt
rw-rw-r-- 1 researcher2 research_team   46 Aug 10 13:31 project_t.txt
esearcher2@dbf0ca91569f:~/projects$
```

- Screenshot showing commands used to adjust `.project_x.txt` permissions and the resulting permission output.

Restricting Access to a Directory

Access to the `drafts` directory was limited to the owner, `researcher2`. Previously, the group had execute permission, which would allow them to access the directory contents. I revoked this permission so that only `researcher2` can enter or list files within the directory.



```
-rw-rw-r-- 1 researcher2 research_team   46 Aug 10 13:52 project_t.txt
researcher2@ccd460f44dd3:~/projects$ chmod g-x drafts
researcher2@ccd460f44dd3:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 10 13:52 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 10 14:25 ..
-rw--w---- 1 researcher2 research_team   46 Aug 10 13:52 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Aug 10 13:52 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Aug 10 13:52 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 10 13:52 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 10 13:52 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 10 13:52 project_t.txt
researcher2@ccd460f44dd3:~/projects$ 
```

- Screenshot showing the command that removed execute permission from the group on the `drafts` directory and its updated permissions output.

Summary

By carefully examining and updating file and directory permissions, I aligned access controls within the `projects` folder to comply with organizational security standards. These adjustments help prevent unauthorized changes and secure sensitive data.