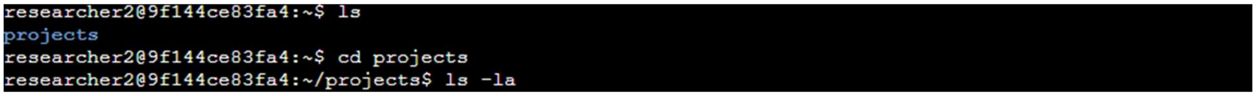
# File permissions in Linux

## Project

The research team of the security division needs to have file permissions updated, so that the organization can maintain a strong security posture by following the Principle of Least Privilege. The Principle of Least Privilege ensures only authorized users whose identity has been verified have the necessary permissions to execute their jobs within the system. Authorization refers to the process of granting access to specific resources within a system. This is crucial because, in the absence of proper authorization, any user could potentially access and modify files that belong to other users or system files. Such unauthorized access would pose a significant security risk.

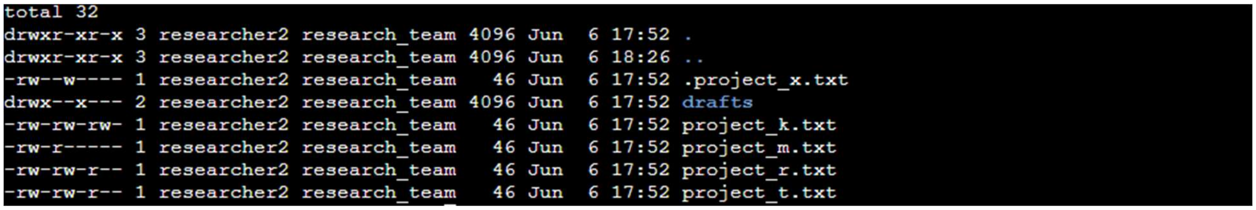
## File and directory details

After navigating to the projects directory, the permissions set for files and subdirectories can be checked by using the Linux command ls -la to display all permissions, including permissions for files that are hidden.



```
researcher2@9f144ce83fa4:~$ ls
projects
researcher2@9f144ce83fa4:~$ cd projects
researcher2@9f144ce83fa4:~/projects$ ls -la
```

## The permissions string



```
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun  6 17:52 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun  6 18:26 ..
-rw--w---- 1 researcher2 research_team   46 Jun  6 17:52 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun  6 17:52 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jun  6 17:52 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun  6 17:52 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun  6 17:52 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun  6 17:52 project_t.txt
```

Current permissions for the three subdirectories and the 5 files (1 of which is a hidden file) of the ~/projects directory which is owned by the research team are as follows:

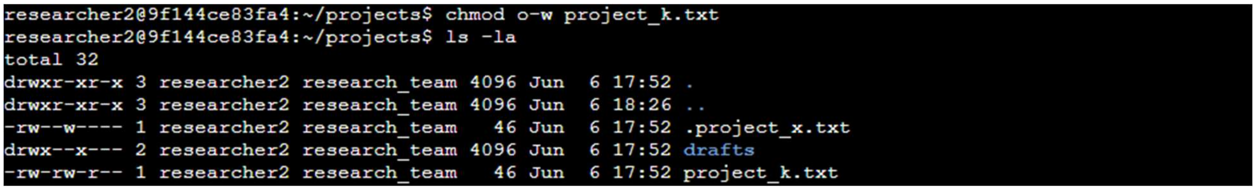
| **Permissions String** | **Subdirectory/File** | **User Permissions** | **Group Permissions** | **Other Permissions** |
|---|---|---|---|---|
| drwxr-xr-x | /. | read, write, execute | read, execute | read, execute |
| drwxr-xr-x | /.. | read, write, execute | read, execute | read, execute |
| -rw-w---- | .project_x.txt | read, write | write | no permissions |
| drwx-x--- | /drafts | read, write, execute | execute | no permissions |
| -rw-rw-rw- | project_k.txt | read, write | read, write | read, write |
| -rw-r----- | project_m.txt | read, write | read | no permissions |

| | | | | |
|---|---|---|---|---|
| -rw-rw-r-- | project_r.txt | read, write | read, write | read |
| -rw-rw-r-- | project_t.txt | read, write | read, write | read |

For example, project_t.txt has the permissions string: - r w - r w - r - -
The 1st character denotes the file type: '-' for regular files.
The 2nd-4th characters indicate user permissions: 'r' for read, 'w' for write, and '-' for no execute permissions. A hyphen (-) means the permission is not granted.
The 5th-7th characters show group permissions: 'r' for read, 'w' for write, and '-' for no execute permissions. A hyphen (-) also indicates the absence of permission.
The 8th-10th characters represent other users' permissions: 'r' for read permissions, '-‘ for absence of write permissions, and ‘-‘ again for absence of execute permissions.

## Change file permissions

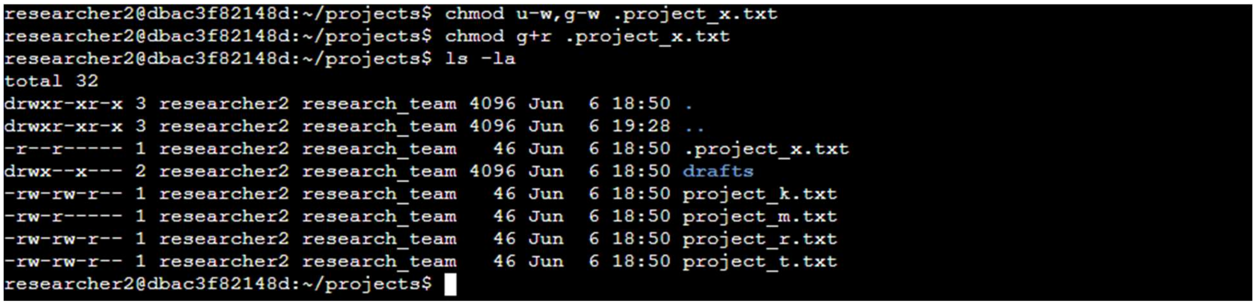
The organization does not allow others to have write access to any files. Based on the permissions established above, this means that project_k.txt needs to have the write permissions for Other revoked. This can be done by using the command “chmod o-w project_k.txt” (the chmod command changes permissions on files and directories and requires 2 arguments. The first argument being how to change the permissions, and the second being the file we want to change the permissions on. In this case the first argument “o-w” indicates that we want to subtract w (write permissions) from o (Other), and the second argument indicates that we want those changes to happen for file “project_k.txt”) which gives us the output “researcher2@dbac3f82148d:~/projects$”. Since there were no errors, it can be assumed that the command succeeded, but we must check to make sure so we input the “ls -la” command to review the change in permissions.



```
researcher2@9f144ce83fa4:~/projects$ chmod o-w project_k.txt
researcher2@9f144ce83fa4:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun  6 17:52 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun  6 18:26 ..
-rw--w---- 1 researcher2 research_team   46 Jun  6 17:52 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun  6 17:52 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun  6 17:52 project_k.txt
```

# Change file permissions on a hidden file

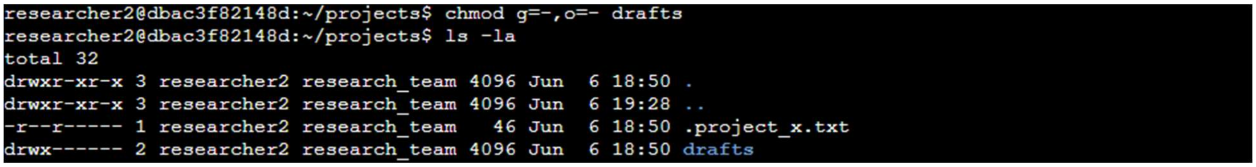
The research team has archived .project_x.txt, which is why it’s a hidden file. This file should not have write permissions for anyone, but the user and group should be able to read the file. So, to revoke all write permissions, we use the Linux command: “chmod u-w,g-w .project_x.txt”. Again, the “chmod” command allows us to make changes to permissions on a file or directory and accepts 2 arguments. In this case, we want to remove the write permissions for user and group, so the first argument is “u-w,g-w” where “u-w” indicates we want to revoke write permissions for user, and “g-w” indicates that we would like to revoke the write permissions for group, followed by the second argument being “.project_x.txt” which is the file we want to have the permissions changed for. Then, since the user already has read permissions for this file, we use the Linux command: “chmod g+r .project_x.txt” to add read permissions for group (instructed by using the first argument “g+r” for the chmod command, followed by the second argument “.project_x.txt” indicating the location where we want these changes to happen . For both of these changes we receive the output in the shell “researcher2@dbac3f82148d:~/projects$”. Since we didn’t receive any errors, we can conclude that the changes were made successfully however we must review the permissions changes with the Linux command “ls -la” to be certain.



```
researcher2@dbac3f82148d:~/projects$ chmod u-w,g-w .project_x.txt
researcher2@dbac3f82148d:~/projects$ chmod g+r .project_x.txt
researcher2@dbac3f82148d:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun  6 18:50 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun  6 19:28 ..
-r--r----- 1 researcher2 research_team   46 Jun  6 18:50 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun  6 18:50 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun  6 18:50 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun  6 18:50 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun  6 18:50 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun  6 18:50 project_t.txt
researcher2@dbac3f82148d:~/projects$ 
```

# 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. To ensure that only researcher2 (user) has access to the draft directory and its contents, we will use the Linux command: “chmod g=-,o=- drafts”. The first argument assigns permissions to ‘g’ as ‘=’ (equal to) ‘-‘ (no permissions), and the same for other, then for the second argument we input the location “drafts” as where we want these changes to happen. The output should again be “researcher2@dbac3f82148d:~/projects$”, meaning that there were no errors in executing the command and we can check this by using the Linux command “ls -la” to review the permissions.



```
researcher2@dbac3f82148d:~/projects$ chmod g=-,o=- drafts
researcher2@dbac3f82148d:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun  6 18:50 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun  6 19:28 ..
-r--r----- 1 researcher2 research_team   46 Jun  6 18:50 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Jun  6 18:50 drafts
```

## Summary

To conclude, after reviewing the priveleges of the research team, we found that project_k.txt extended write privileges to other, and the organization does not allow others to have write access to any files, so we revoked the write privileges for other for project_k.txt. Then we found that the research team archived a file .project_x.txt, which is why it is a hidden file, but the file still had write permission extended to user and group, when this file should not have write permissions for anyone, so these permissions were revoked. We still wanted read permissions for user and group on the .project_x.txt file so after reviewing the permissions, only group was missing read permissions so we added them. Finally, since only researcher2 should have any access to the drafts directory, we made sure that group, and other had no permissions for the directory.