# File permissions in Linux

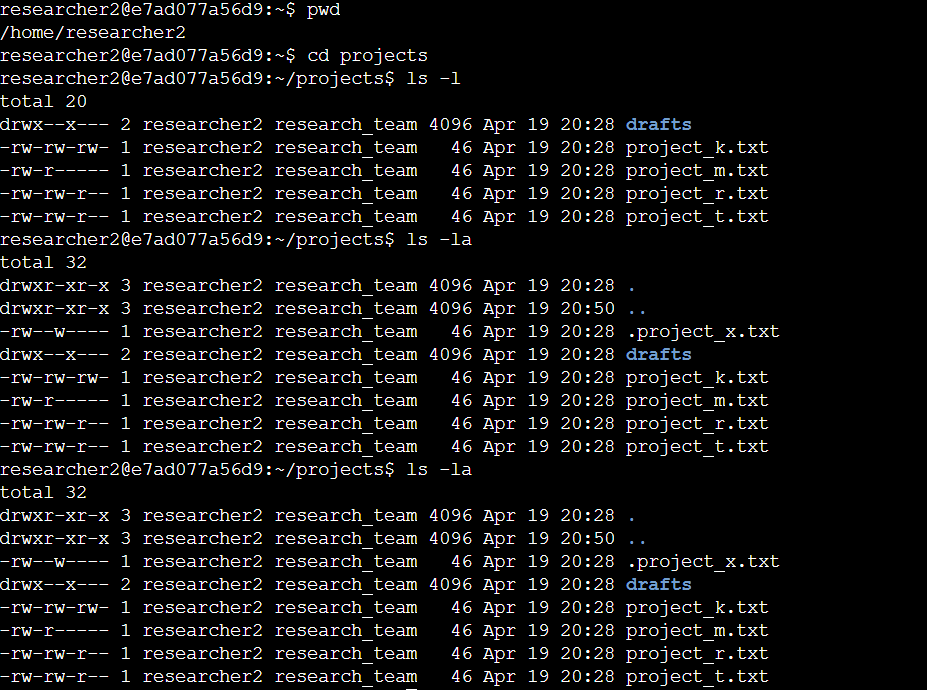
## Project description

The research team at my organization needs investigating and updating existing permissions on the file system. If the permissions do not match authorization, there is a need to modify the permissions to authorize the appropriate users and remove any unauthorized access. This helps to keep the system secure.

## Check file and directory details

Check the permissions set for files and subdirectories in the projects directory. Make sure you display all permissions, including hidden files.

ls -la



ls command can be used to display a list of files in the specified directory. Here it is used in combination of -la command. The second argument which is used in compound with ls command, -la, use to display permission of files, hidden filles and subdirectories. Here -l command displays the permissions of files and subdirectories while a displays hidden files and its permissions.

## Describe the permissions string

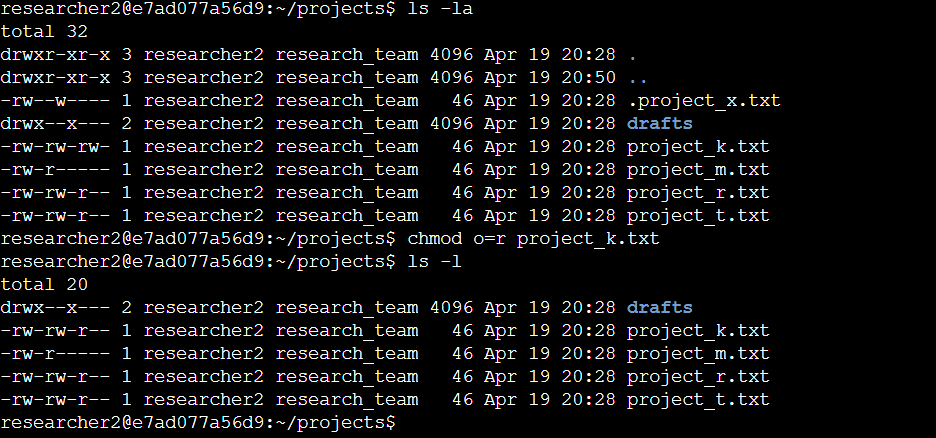
 For a directory with permissions for all owner types would be drwxr-xr-x (the same case as it can be seen in previous step):

* The 1st character indicates the file type. The d indicates it’s a directory. When this character is a hyphen (-), it's a regular file.
* The 2nd-4th 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.
* The 5th-7th characters indicate the read (r) and execute (x) permissions for the group. While one of these characters is a hyphen (-) instead, it indicates that write(w) permission is not granted for the group.
* The 8th-10th characters indicate the read (r) and execute (x) permissions for the owner type of other. While one of these characters is a hyphen (-) which indicates that write (w) permission is not granted for other. This owner type consists of all other users on the system apart from the user and the group.

## Change file permissions

Chmod o=r project\_k.txt

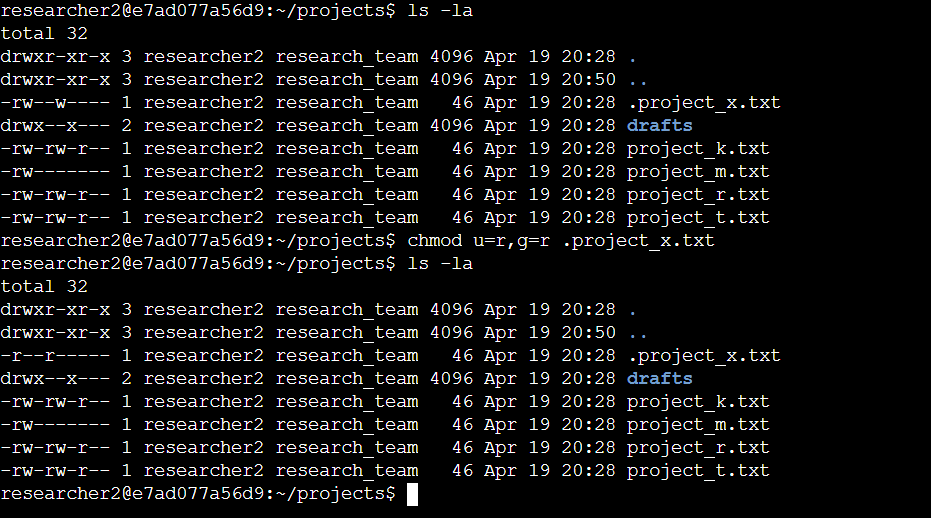
In the above-mentioned commands, I changed the permission of a file using change mode command. chmod command which is used to manage permission of files or directories. Here the output result displays our command function as now the others do not have permission to write or modify a file project\_k.txt. Also, the first line in the screenshot represents the current directory and second line represents the home directory.



## Change file permissions on a hidden file

chmod u=r,g=r .project\_x.txt

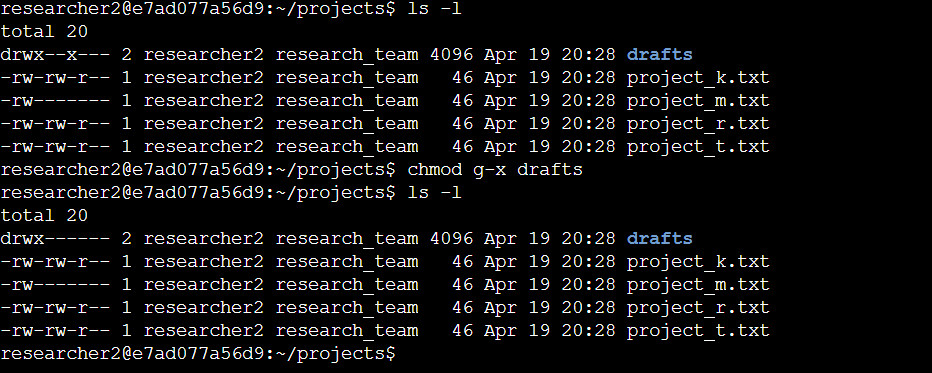
Here chmod command used for changing command set as a second argument we used = operator to overwrite the permissions for user and group. And finally, the third argument represents a name of hidden file with dot in its start in which we required to make modifications. Output displays that the hidden files don’t have permissions to authorize user and group to write on a file while still they can read.



## Change directory permissions

chmod g-x drafts

In this command first we used change mode to manage permissions and then specified a group permission and used hyphen with execute(x) to indicate the elemination of execute authorization to mentioned file for the group. Command g-x appends the permissions means unlike = operator it does not replaces all other permissions as well, like read or write,



## Summary

File permission to authorize users if not managed properly could lead to serious security issues. To avoid such cases file permissions should be managed in a way to give the least privilege to the users to complete the task. For the above-mentioned case I used Linux commands like ls, -la and chmod to manage file permission as per defined criteria to avoid security risks.