# File permissions in Linux

## **Project description**

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

Your task is to examine existing permissions on the file system. You’ll need to determine if the permissions match the authorization that should be given. If they do not match, you’ll need to modify the permissions to authorize the appropriate users and remove any unauthorized access.

**Step 1: 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.

## 

I located the projects sub-directory with pwd and ls commands. Then I viewed permissions of all files and directories within the projects directory with the ls -la commands from the parent directory researcher2

## **Step 2: Describe the permissions string**

For the drafts directory there is the permission string drwx--x---

The 10-character string can be deconstructed to determine who is authorized to access the

file and their specific permissions.

* **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.

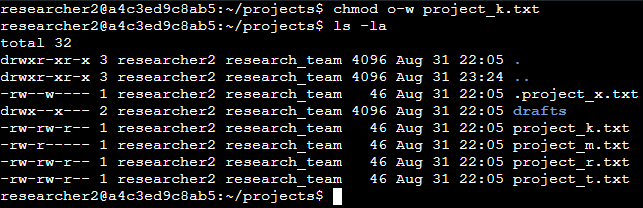
* **5th-7th characters:** These characters indicate the read (r), write (w), and execute (x) permissions for the group. When one of these characters is a hyphen (-) instead, it indicates that this permission is not granted for the group.
* **8th-10th characters:** These characters indicate the read (r), 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.

## **Step 3: Change file permissions**

The organization does not allow “other” to have write access to any files. Based on the permissions established in Step 1, identify which file needs to have its permissions modified. Use a Linux command to modify these permissions.

Based on what permissions that was revealed in step 1 the only permissions that need to be changed was for project\_k.txt the “other” owner had write permissions.



So I changed the permissions using the chmod o-w command.

chmod = changes permission on files and directories

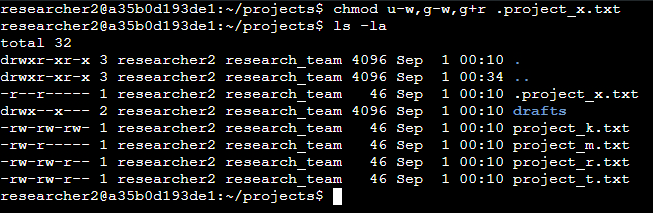
o = other owner

- = to take away

w = write permissions

## **Step 4: 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. Use a Linux command to assign **.project\_x.txt** the appropriate authorization.



Using the chmod u-w,g-w,g+r command I removed the read permissions from both the user and group and gave read permission to the group. I appended the permissions into one command using the , instead of writing three commands.

## **Summary**

As a security professional supporting a research team, my tasks involved verifying and adjusting file system permissions to ensure that only authorized users had appropriate access. I used the ls -la command to check the permissions for files and directories within the projects directory, identified inconsistencies, and modified them using the chmod o-w command to remove unauthorized write access. Additionally, I used the chmod u-w,g-w,g+r command to correct the permissions on a hidden file, ensuring it complied with the organization's security policy.