Linux File Permission

Google Cybersecurity Professional Certification

Francisco Raul Ambriz

Portfolio Project

Project Portfolio

Title	Using-Linux-Commands-To-File-Permissions
Version	2.3
Date Issued	2/8/23
Status	Finished
Document Owner	Francisco Raul Ambriz
Creator Name	Francisco Raul Ambriz
Creator Organization Name	Google Coursera
Subject Category	Cybersecurity - Linux - Permissions

Contents

1. Project Description
2. Check file and directory details
3. Describe the permissions string
4. Change file permissions
5. Change file permissions on a hidden file
6. Change directory permissions
7. Summary

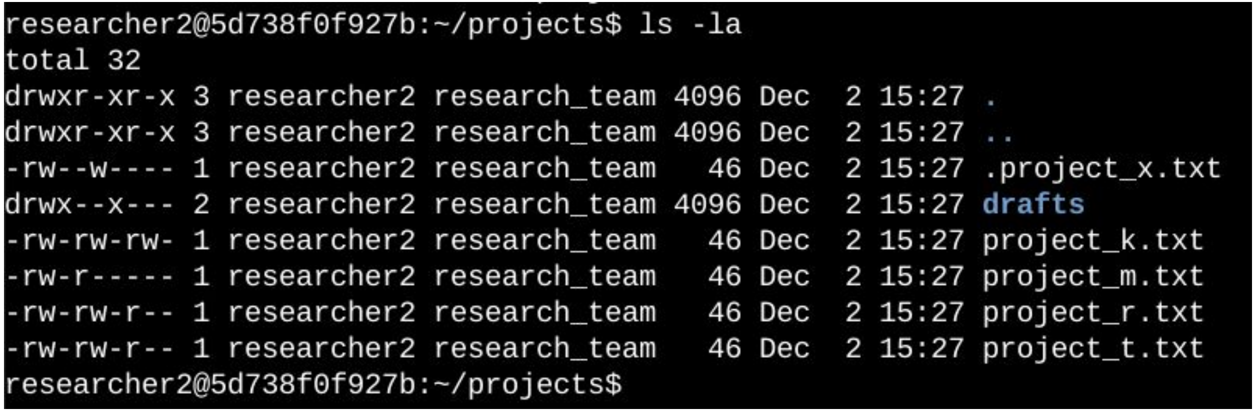
1. Project Description

Authorization involves granting access to particular resources within a system. This is crucial for maintaining security, as without proper authorization, any user could potentially access and modify files belonging to other users or system files, posing a significant security risk. In Linux, file and directory permissions are utilized to determine who has access to specific files and directories. However, the current permissions might not accurately reflect the level of authorization that should be granted. Therefore, regularly checking and updating these permissions is essential to maintaining system security.

To address this, I undertook the following tasks:

2. Check File and Directory Details

This document provides an overview of the file structure within the */home/researcher2/projects* directory, including the permissions of both files and subdirectories it encompasses. Within the */home/researcher2/projects* directory, there exist five files with specific names and corresponding permissions. To verify this information, I executed the **ls -la** command within the Shell terminal.



```
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-rw--w---- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

I use the First, ls -la displays permissions to files and directories. This shows the files and directories, including the hidden ones.

3. Describe the Permission String

Understanding the 10-character string helps determine who has access to a file and what permissions they possess. Here's what each character represents:

1. 1st character:
   - A "d" signifies a directory.
   - A hyphen "-" signifies a regular file.

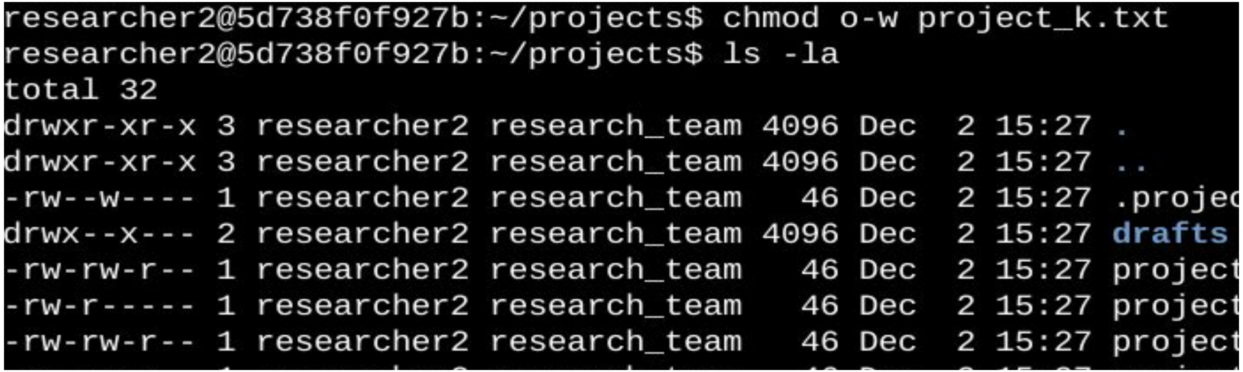
2. 2nd-4th characters:
   - These represent read (r), write (w), and execute (x) permissions for the user.
   - A hyphen "-" indicates the absence of a specific permission for the user.

3. 5th-7th characters:
   - These represent read (r), write (w), and execute (x) permissions for the group.
   - A hyphen "-" indicates the absence of a specific permission for the group.

4. 8th-10th characters:
   - These represent read (r), write (w), and execute (x) permissions for others, i.e., users who are neither the owner nor part of the group.
   - A hyphen "-" indicates the absence of a specific permission for others.

4. Change File Permissions

The organization has decided to restrict write access for "others" on their files. To comply with this decision, I reviewed the file permissions I previously obtained. I identified that the file "project_k.txt" needed to have its write access revoked for "others."
Below is the code snippet demonstrating how I executed this using Linux commands:



```
researcher2@5d738f0f927b:~/projects$ chmod o-w project_k.txt
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-rw--w---- 1 researcher2 research_team   46 Dec  2 15:27 .projec
drwx--x--- 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project
```

The screenshot shows the commands I entered followed by their output. I identified ".project_x.txt" as a hidden file because its name starts with a period (.). In this example, I modified permissions by:

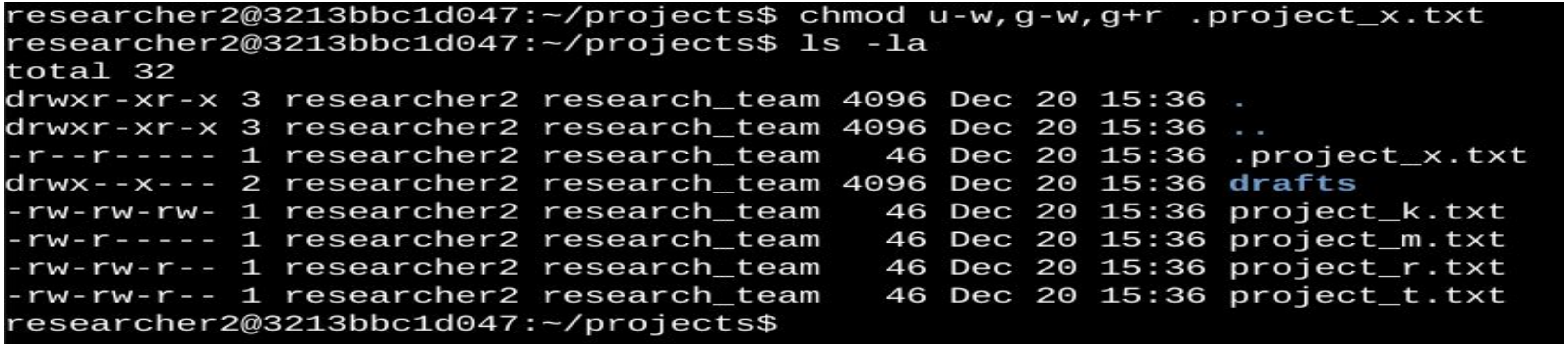
- Removing write permissions from the user and group.
- Adding read permissions to the group.

I achieved this by using the commands "u-w" to remove write permissions from the user, "g-w" to remove write permissions from the group, and "g+r" to add read permissions to the group.

Change File Permissions on Hidden File

The research team at my organization has recently archived "project_x.txt." They want to restrict write access to this project while allowing both the user and group to have read access.

Below is the code snippet showing how I modified the permissions using Linux commands:



```
researcher2@3213bbc1d047:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@3213bbc1d047:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec 20 15:36 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec 20 15:36 ..
-r--r----- 1 researcher2 research_team   46 Dec 20 15:36 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec 20 15:36 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Dec 20 15:36 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec 20 15:36 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec 20 15:36 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec 20 15:36 project_t.txt
researcher2@3213bbc1d047:~/projects$
```

The screenshot shows the commands I entered followed by their output. I recognized ".project_x.txt" as a hidden file because its name starts with a period (.). In this case, I adjusted permissions by:

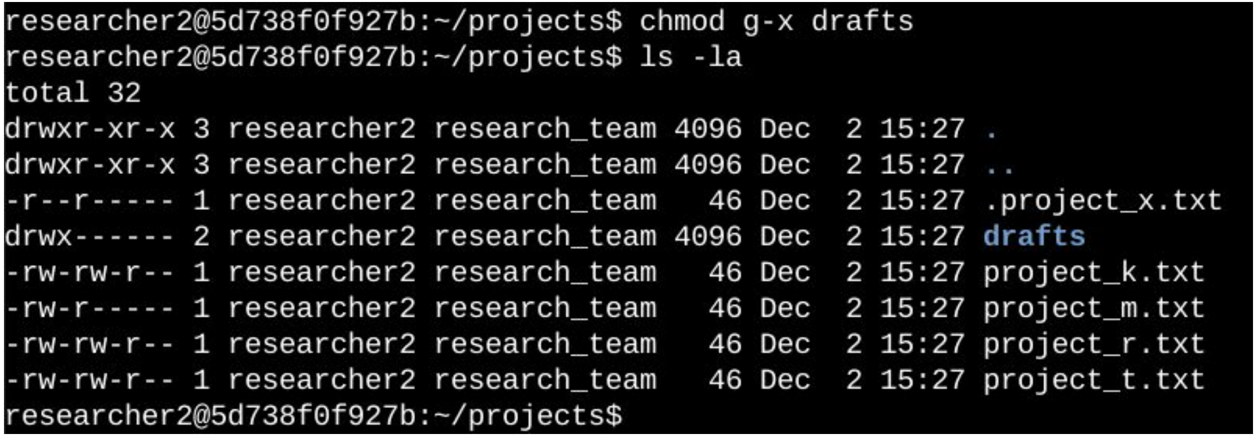
- Revoking write permissions from both the user and group.
- Granting read permissions to the group.

I achieved this by using the commands "u-w" to remove write permissions from the user, "g-w" to remove write permissions from the group, and "g+r" to add read permissions to the group.

Change Directory Permissions

My organization requires that only the "researcher2" user has access to the "drafts" directory and its contents. This entails ensuring that no other user besides "researcher2" has execute permissions.

Below is the code snippet illustrating how I adjusted the permissions using Linux commands:



```
researcher2@5d738f0f927b:~/projects$ chmod g-x drafts
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-r--r----- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

The screenshot shows the commands I typed in, followed by their outputs. I had earlier found out that the group had execute permissions, so I utilized the chmod command to eliminate them. Since the researcher2 user already had execute permissions, there was no need to add them.

Summary

I adjusted various permissions to align with the authorization level desired by my organization for files and directories within the "projects" directory. Initially, I checked the permissions for the directory using "ls -la", which guided my subsequent actions. I proceeded to use the "chmod" command multiple times to modify permissions on files and directories accordingly.