Project Description: Managing File Permissions Using Linux Commands

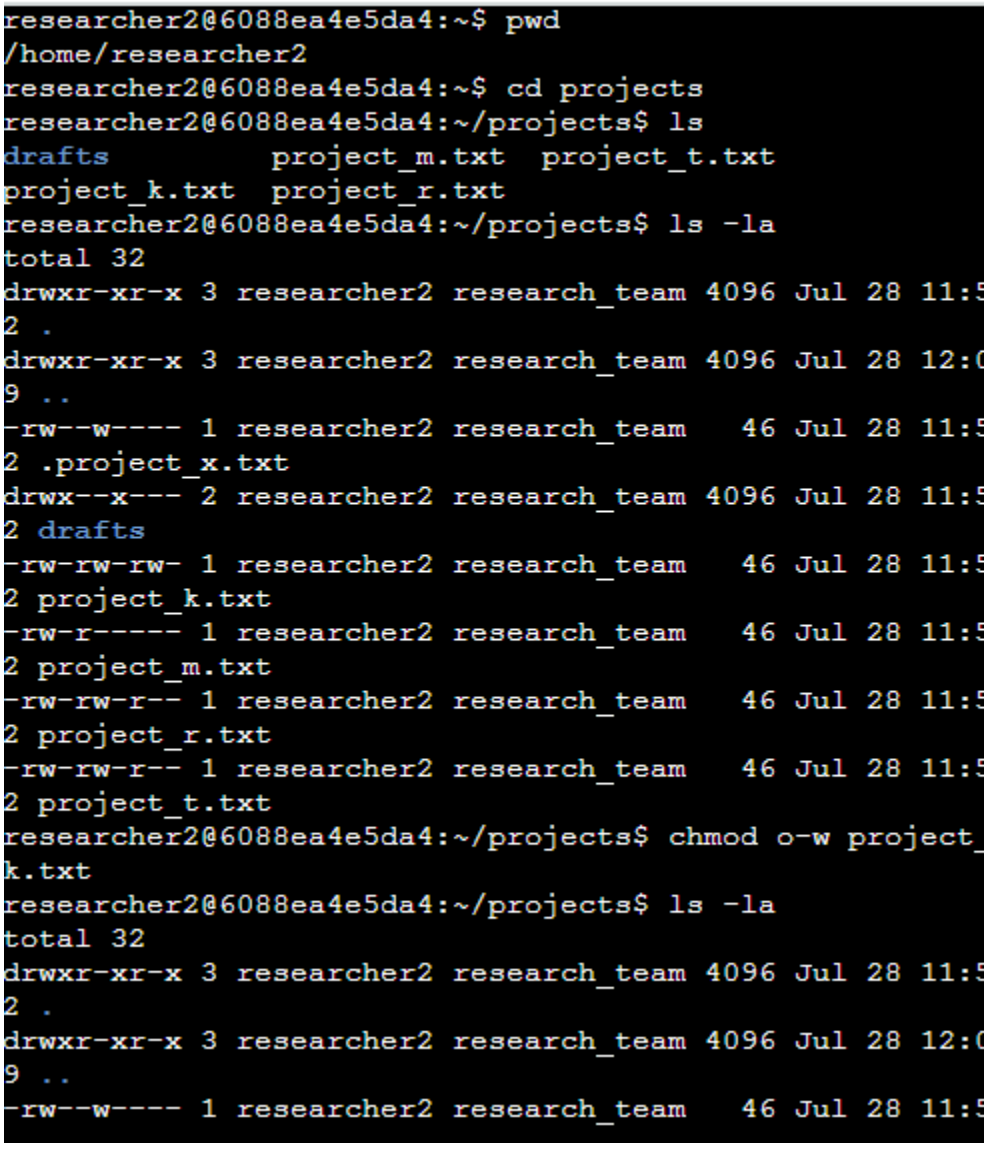
This project involved managing file permissions and directory access within a Qwiklab environment. The primary task was to authenticate and control file authorization through the use of various Linux commands.

Key activities included:

- Assigning and modifying permissions for different users using the Bash shell commands in a project file named `Project_x.txt`.
- Identifying and uncovering hidden files, as well as verifying and removing unauthorized users to secure access to critical files.

These tasks are essential for safeguarding organizational data and ensuring that sensitive information remains protected. The Linux Bash shell commands utilized in this project, such as `chmod`, `ls -la`, `ls`, `cd`, and `nano`, are detailed below.

Below are screenshots of the lab activities conducted using Bash shell commands in a Linux environment.



```
9 ..
-rw--w---- 1 researcher2 research_team   46 Jul 28 11:5
2 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jul 28 11:5
2 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jul 28 11:5
2 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jul 28 11:5
2 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 28 11:5
2 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 28 11:5
2 project_t.txt
researcher2@6088ea4e5da4:~/projects$ chmod g-r project_
m.txt
researcher2@6088ea4e5da4:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jul 28 11:5
2 .
drwxr-xr-x 3 researcher2 research_team 4096 Jul 28 12:0
9 ..
-rw--w---- 1 researcher2 research_team   46 Jul 28 11:5
2 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jul 28 11:5
2 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jul 28 11:5
2 project_k.txt
-rw------- 1 researcher2 research_team   46 Jul 28 11:5
2 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 28 11:5
2 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 28 11:5
2 project_t.txt
```

```
9 ..
-r--r----- 1 researcher2 research_team   46 Jul 28 11:5
2 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jul 28 11:5
2 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jul 28 11:5
2 project_k.txt
-rw------- 1 researcher2 research_team   46 Jul 28 11:5
2 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 28 11:5
2 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 28 11:5
2 project_t.txt
researcher2@6088ea4e5da4:~/projects$ chmod g-x drafts/
researcher2@6088ea4e5da4:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jul 28 11:5
2 .
drwxr-xr-x 3 researcher2 research_team 4096 Jul 28 12:0
9 ..
-r--r----- 1 researcher2 research_team   46 Jul 28 11:5
2 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Jul 28 11:5
2 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jul 28 11:5
2 project_k.txt
-rw------- 1 researcher2 research_team   46 Jul 28 11:5
2 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 28 11:5
2 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 28 11:5
2 project_t.txt
researcher2@6088ea4e5da4:~/projects$
```

Explanation of the chmod Command for Updating File Permissions

In the Bash shell environment, we utilized the **`chmod`** command to manage file permissions for hidden files, specifically targeting the file named **`.Project_x.txt`**. Our objectives were to verify and adjust unauthorized write permissions and read permissions for users and groups.

The process included:

1. Identifying the File: We identified the file named **`.Project_x.txt`** within the project directory.

Modifying Permissions: We removed write permissions for other users and read permissions for groups from `.Project_x.txt` using the following command:

```
chmod o-w,g-r .Project_x.txt
```

2. This command ensures that other users cannot write to the file, and the group cannot read it.

Adjusting Permissions for Users and Groups: After discovering hidden files, we also adjusted permissions for both users and groups. To manage these permissions, we used:

```
chmod u-w,g-rwx .Project_x.txt
```

3. This command removes write permissions for the original users and read, write, and execute permissions for the group.

These actions were critical in securing file access and ensuring proper authorization within the project environment.

Details on Checking File Permissions with `ls -la`

After updating file permissions using the `chmod` command, we employed the `ls -la` command to verify that the changes were correctly applied to the file `Project_x.txt`.

The `ls -la` command, when executed, provides a detailed listing of file permissions, ownership, and other attributes. This allows us to confirm that the intended permissions for `Project_x.txt` were successfully removed for other users and groups.

The command was executed as follows:

```
ls -la Project_x.txt
```

This command displays the file's permissions in the detailed list, allowing us to review and ensure that the modifications were correctly implemented.

Details on Interpreting the 10-Character String for File Permissions

The 10-character string representing file permissions, such as `drwxrwxrwx`, provides a detailed view of the file's access settings. Upon interpreting this string in the Linux Bash shell, we observed that certain unauthorized users had been granted access permissions to the file. Additionally, we identified hidden files, including `.project_x.txt`, that were subject to the same permission settings.

The string is structured as follows:

- The first character indicates the file type.
- The next nine characters are grouped into three sets of three, representing permissions for the owner, group, and others, respectively.

This detailed view enabled us to assess and manage file access accurately.

Details on Hidden Files and Directories

In this activity, we focused on the hidden file `.project_x.txt` located within the project directories. This file was initially configured with read-only permissions for both users and groups, effectively restricting other types of access.

The permission string for the file was represented as `-r--r-----`, indicating read permissions for the owner and group, while other permissions were denied.

To secure the file appropriately, we utilized the chmod command to modify the permissions for both users and groups. This was achieved by applying the relevant 10-character permission string to update access settings accordingly.

Project Summary: Managing File Permissions

This project focused on managing file permissions to ensure proper access control within the file and directory structure. As part of the cybersecurity analysis, we authorized permissions for legitimate users and discovered a hidden file named .project_x.txt in the project directory.

Using the chmod command, we updated the file permissions to ensure that only authorized users could access the file. The permissions were managed using the 10-character permission string, which allowed us to grant read and write access to the appropriate users while removing unauthorized access.

This process was crucial for protecting confidential information and ensuring that permissions were accurately assigned to authorized individuals.