Linux Permissions Portfolio Project

Google Cybersecurity Professional Certification

Linux Permissions Portfolio Project 0
Table of 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 8
1. Examining Current Permissions with ls -la 9
2. Modifying Permissions with chmod 9

Project description

In this project, I utilized the Linux Bash shell to manage file and directory permissions to ensure secure access and prevent unauthorized users from viewing, modifying, or executing sensitive data. This process involved identifying files and directories with improper permissions, analyzing the required access levels for specific users or groups, and modifying permissions using standard Linux commands.

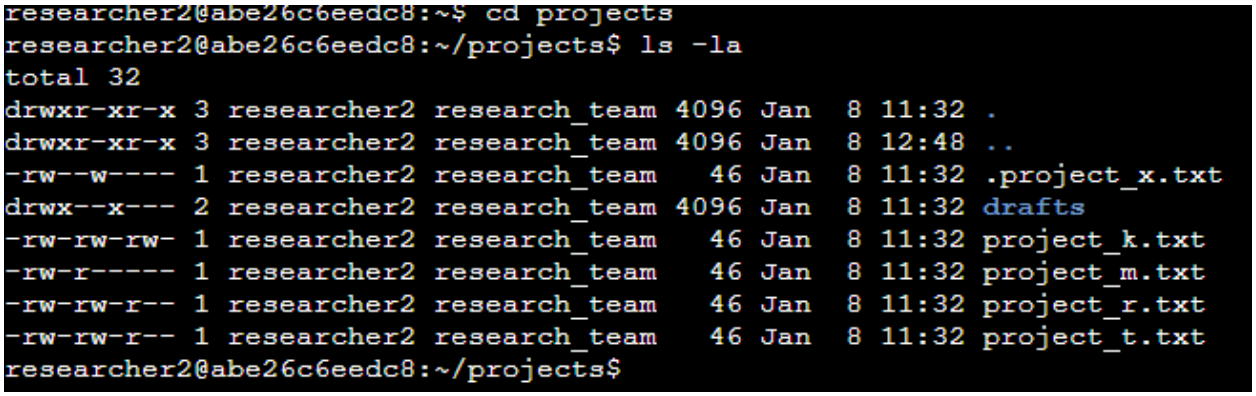
Ensuring files and directories have proper access permissions is critical to maintaining the overall security of a system. Unauthorized access can lead to breaches, privacy violations, and even system exploitation.

To complete this task, I performed the following steps:

Check file and directory details

This document displays the structure of the /home/researcher2/projects directory and the permissions of the files and subdirectory it contains.

In this directory, there are five files with the following names and permissions:



To run this step, I used the `ls -la` command in the Shell terminal. This shows the files and directories, including the hidden ones.

Describe the permissions string

Permissions are represented with a 10-character string. Permissions include: read, write and execute. These permissions are given to the following owners: user, group and other.

- The first character indicates the file type. The d indicates that it is a directory and the hyphen (-) describes 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, it indicates that the permission is not granted.
- Similarly, the 5th to 7th character describes permissions for the group
- Lastly, the 8th to 10th characters describes permission for other

Taking the project_k.txt file as an example, the user and group and other have permissions to read and write, but not execute.

Change file permissions

The organization does not allow others to have write access to any files. Based on the permissions established in Step 3, identify which file needs to have its permissions modified. Use a Linux command to modify these permissions.

```
researcher2@abe26c6eedc8:~/projects$ chmod o-w project_k.txt
researcher2@abe26c6eedc8:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jan  8 11:32 .
drwxr-xr-x 3 researcher2 research_team 4096 Jan  8 12:48 ..
-rw--w---- 1 researcher2 research_team   46 Jan  8 11:32 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jan  8 11:32 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jan  8 11:32 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jan  8 11:32 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan  8 11:32 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan  8 11:32 project_t.txt
researcher2@abe26c6eedc8:~/projects$
```

Based on the information provided, I used the `chmod` command to change permissions of the project_k.txt file so that the group does not have write permissions.

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.

```
researcher2@7ee068e708e3:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jan  8 10:55 .
drwxr-xr-x 3 researcher2 research_team 4096 Jan  8 11:32 ..
-rw--w---- 1 researcher2 research_team   46 Jan  8 10:55 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jan  8 10:55 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jan  8 10:55 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jan  8 10:55 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan  8 10:55 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan  8 10:55 project_t.txt
researcher2@7ee068e708e3:~/projects$ chmod u-w .project_x.txt
researcher2@7ee068e708e3:~/projects$ chmod g+r-w .project_x.txt
researcher2@7ee068e708e3:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jan  8 10:55 .
drwxr-xr-x 3 researcher2 research_team 4096 Jan  8 11:32 ..
-r--r----- 1 researcher2 research_team   46 Jan  8 10:55 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jan  8 10:55 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jan  8 10:55 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jan  8 10:55 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan  8 10:55 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan  8 10:55 project_t.txt
researcher2@7ee068e708e3:~/projects$
```

By using the `chmod` command, I removed the write permission from the file from both user and group while adding the read permission to the group

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. Use a Linux command to modify the permissions accordingly.

```
researcher2@7ee068e708e3:~/projects$ pwd
/home/researcher2/projects
researcher2@7ee068e708e3:~/projects$ chmod g-x drafts
researcher2@7ee068e708e3:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jan  8 10:55 .
drwxr-xr-x 3 researcher2 research_team 4096 Jan  8 11:32 ..
-r--r----- 1 researcher2 research_team   46 Jan  8 10:55 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Jan  8 10:55 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jan  8 10:55 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jan  8 10:55 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan  8 10:55 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan  8 10:55 project_t.txt
researcher2@7ee068e708e3:~/projects$
```

With this command, I removed the execute command for the group from the drafts directory

Summary

In this project, I used Linux commands to examine and modify the file and directory permissions within my organization to ensure the appropriate level of access. Here's an overview of the steps I followed:

---

1. Examining Current Permissions with `ls -la`

- I used the `ls -la` command to display detailed information about files and directories, including their permissions, ownership, and size.
- This allowed me to identify:
    - Files or directories with overly permissive or restrictive permissions.
    - Ownership mismatches that might indicate potential security risks.
    - Hidden configuration files that required stricter access control.

2. Modifying Permissions with `chmod`

- I used the `chmod` command to change permissions for files and directories, ensuring they were aligned with organizational security policies.
- Permissions were adjusted based on the principle of least privilege:
    - Granting only the necessary access to the owner, group, and others.
    - Restricting unauthorized read, write, or execute access.

Key Outcomes:

- Ensured that sensitive files, such as payroll data and configuration files, were protected from unauthorized access.
- Adjusted permissions to meet internal security policies and maintain confidentiality.
- Prevented potential breaches caused by excessive permissions, such as files being writable or executable by unauthorized users.

Using `ls -la` for auditing and `chmod` for modifying permissions provided a simple yet powerful method to enhance file and directory security across the organization.