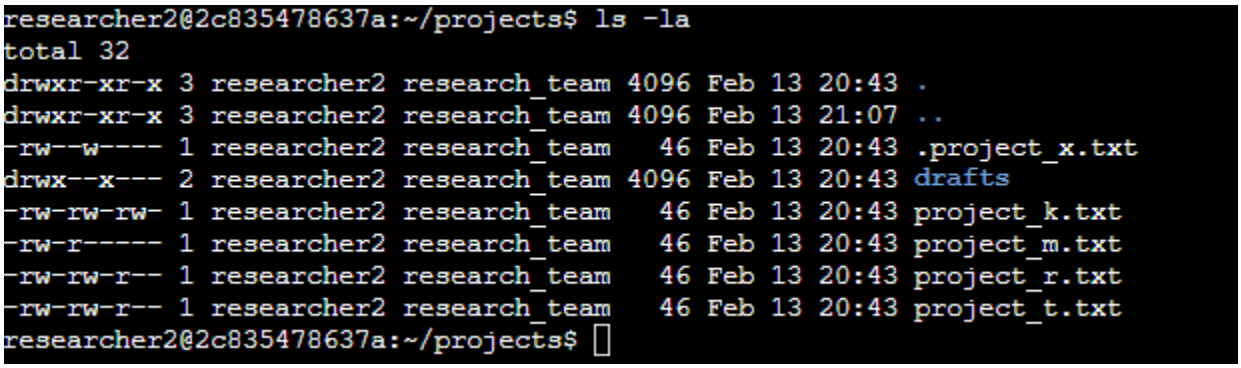
File permissions in Linux

Project description

As a security professional, I analyzed file and directory permissions, identified those that did not match, and updated them to match the appropriate user permissions. I used Linux commands in the shell to check and modify permissions. Checking and updating these permissions helped keep the system secure. To complete this task, I performed the following steps:

Check file and directory details

To check the existing permissions in the projects directory, I used Linux commands in the Shell.



The first line of the screenshot shows the executed command, while the following lines show the generated output. The executed command lists all the contents of the projects directory. I used the ls -la command to display a detailed list of the contents, including hidden files. The output indicates a drafts directory and a hidden .project_x.txt file, plus 4 other project files. A 10-character string in the first column represents the permissions set for each file and repository.

Describe the permissions string

The 10-character sequence represents the file type and the permissions assigned to determine who is authorized to access the file and their specific permissions. The characters mean:

- First Character: This character indicates the file type, either “d” or a hyphen (-). If it is “d”, it means it is a directory. If it is a (-), it is a regular file.

- 2nd - 4th Character: This sequence of 3 characters defines the user's permissions: read (r), write (w), and execute (x). If one of these is a hyphen (-), it indicates that the permission is not granted to the user.

- 5th - 7th Character: This sequence of 3 characters defines the group's permissions: read (r), write (w), and execute (x). If one of these is a hyphen (-), it indicates that the permission is not granted to the group.

- 8th - 10th Character: This sequence of 3 characters defines the permissions for others: read (r), write (w), and execute (x). This type of user consists of all system users except the user and the group. If one of these is a hyphen (-), it indicates that the permission is not granted to others.

For example, in the file project_r.txt the sequence is -rw-rw-r--. The first character indicates that project_r.txt is a file. From the 2nd to the 4th character, it means that the user can read and edit the file; the group from the 5th to the 7th character indicates that they can also read and edit; and from the 8th to the 10th character indicates that others can only read. No one has permission to execute the project_r.txt file.

Change file permissions

The organization determined that no "other" user should have write permission on any of the files. To comply with this security guideline, I reviewed the previously identified permissions. The file project_k.txt was not complying with the policy, so its permission had to be removed.

Below is the Linux command used to update the file permissions:

```
researcher2@2c835478637a:~/projects$ chmod o-w project_k.txt
researcher2@2c835478637a:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb 13 20:43 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb 13 21:07 ..
-rw--w---- 1 researcher2 research_team   46 Feb 13 20:43 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Feb 13 20:43 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Feb 13 20:43 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Feb 13 20:43 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 13 20:43 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 13 20:43 project_t.txt
researcher2@2c835478637a:~/projects$
```

The first two lines show the commands I entered; the following lines show the output of the second command. The chmod command changes file and directory permissions. The first argument indicates which permissions should be added or removed, and the second specifies which file or directory. In this example, I removed write permission for "other" from the

project_k.txt file using the o-w command. After that, I checked with the ls -la command to see if the permission had changed. The modification impacted the third block of the permissions string, corresponding to the "others" category. The write character (w) was replaced by a hyphen (-), changing the sequence from rw- to r--.

Change file permissions on a hidden file

The file .project_x.txt is hidden, and it was requested that only the user and group have read permission.

The following commands demonstrate how the permissions were changed:

```
researcher2@2c835478637a:~/projects$ chmod u=r,g=r .project_x.txt
researcher2@2c835478637a:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb 13 20:43 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb 13 21:07 ..
-r--r----- 1 researcher2 research_team   46 Feb 13 20:43 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Feb 13 20:43 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Feb 13 20:43 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Feb 13 20:43 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 13 20:43 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 13 20:43 project_t.txt
researcher2@2c835478637a:~/projects$
```

The first two lines are the input commands, and the remaining lines show the output of the second command. Files whose name begins with a period (.) are treated as hidden files in Linux. In this example, I defined that only the user and group have read permission on the .project_x.txt file. The command u=r defines that the user can only read the file. The command g=r defines that the group can only read the file.

Change directory permissions

It had been requested that only the user researcher2 should have access to the drafts directory. This means that no other user category should have permissions over the directory.

The following code demonstrates how I used Linux commands to change the permissions:

```
researcher2@2c835478637a:~/projects$ chmod g-x drafts
researcher2@2c835478637a:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb 13 20:43 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb 13 21:07 ..
-r--r----- 1 researcher2 research_team   46 Feb 13 20:43 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Feb 13 20:43 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Feb 13 20:43 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Feb 13 20:43 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 13 20:43 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 13 20:43 project_t.txt
researcher2@2c835478637a:~/projects$
```

The first two lines are the input commands, and the remaining lines show the output of the second command. I had already determined that the group could execute the directory, so I used the chmod command and removed the execute permission from the group category. The owning user (researcher2) already had full permissions (rwx), so it was not necessary to change them.

Summary

I modified multiple file and directory permissions within the project to align them with the authorization levels established by the organization. The first step involved displaying all files and directories along with their respective permissions to assess the current configuration. Based on this analysis, I identified inconsistencies and applied the necessary corrections using the chmod command to adjust specific access rights. These actions ensured that only authorized users maintained appropriate access, reinforcing the principle of least privilege and strengthening overall system security.