File permissions in Linux

Project description

The security team and I were assigned to check the permissions for all files in the `projects` directory, including any hidden files, to make sure that permissions align with the authorization that should be given. When it doesn't, we must change the permissions. Within the `projects` directory there are specific files and directories that need to be modified.

Check file and directory details

The code demonstrating how I used Linux commands to determine the existing permissions set for all the files/directories within the `projects` directory is as follows.

```
researcher2@9cd9244e0ac7:~$ cd projects
researcher2@9cd9244e0ac7:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Mar 17 04:04 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Mar 17 04:04 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Mar 17 04:04 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_t.txt
```

The first line of the screenshot shows the command `cd projects`. This command navigates me to the `projects` directory which is the directory we are intending to examine. The second line of the screenshot displays the `ls` command with the `-l` option to display a more detailed list of the file contents. The output of my `ls -l` command shows one directory named `drafts` and four project text files. Each file listed in the output is displayed in rows and columns. The first column shows a 10-character string (drwx--x---) which represents the permissions set on the corresponding file (`drafts`).

Describe the permissions string

The 10-character permissions string is a line of 10 characters constructed to indicate who can access a file and the type of permissions they have in regards to that specific file.

The 1st character in the string indicates if the file is a directory or normal file. It will have a letter D (`d`) if the file type is a directory, or a hyphen (`-`) if the file type is a normal file.

The 2nd - 4th characters consist of the user's permissions.

- The 2nd is the user's permission to "read" indicated by the letter R (`r`) . If the user is not authorized for the "read" permission then a hyphen (`-`) will take the place of the (`r`).
- The 3rd is the user's permission to "write" indicated by the letter W (`w`). If the user is not authorized for the "write" permission then a hyphen (`-`) will take the place of the (`w`).
- The 4th is the user's permission to "execute" indicated by the letter X (`x`). If the user is not authorized for the "execute" permission then a hyphen (`-`) will take the place of the (`x`).

The 5th - 7th characters consist of the group's permissions.

- The 5th is the group's permission to "read" indicated by the letter R (`r`) . If the group is not authorized for the "read" permission then a hyphen (`-`) will take the place of the (`r`).
- The 6th is the group's permission to "write" indicated by the letter W (`w`). If the group is not authorized for the "write" permission then a hyphen (`-`) will take the place of the (`w`).
- The 7th is the group's permission to "execute" indicated by the letter X (`x`). If the group is not authorized for the "execute" permission then a hyphen (`-`) will take the place of the (`x`).

The 8th - 10th characters consist of the other's permissions.

- The 8th is the other's permission to "read" indicated by the letter R (`r`) . If other is not authorized for the "read" permission then a hyphen (`-`) will take the place of the (`r`).
- The 9th is the other's permission to "write" indicated by the letter W (`w`). If other is not authorized for the "write" permission then a hyphen (`-`) will take the place of the (`w`).
- The 10th is the other's permission to "execute" indicated by the letter X (`x`). If other is not authorized for the "execute" permission then a hyphen (`-`) will take the place of the (`x`).

Change file permissions

At this time I have been informed that the user should be the only one to have access to "read" permissions on the `project_m.txt` file. To complete this task, I decided to remove the group's privilege to "read" permissions on the `project_m.txt` file. The code beneath shows how I used Linux commands to accomplish this task.

```
drwx--x--- 2 researcher2 research_team 4096 Mar 17 04:04 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Mar 17 04:04 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_t.txt
researcher2@9cd9244e0ac7:~/projects$ chmod g-r project_m.txt
researcher2@9cd9244e0ac7:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Mar 17 04:04 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_k.txt
-rw------- 1 researcher2 research_team   46 Mar 17 04:04 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_t.txt
```

The first 5 lines of the screenshot show the current state of permissions of the files in the `projects` directory.
The 6th line is the command I used to remove the group's "read" permission on the `project_m.txt` file. The `chmod` command allows me to modify the permissions of a specified file or directory.
The 7th line is the `ls -l` command. This is used to display the current state of permissions and make sure the modification was successful.

Change file permissions on a hidden file

The security team has been instructed to remove all "write" permissions for the `.project_x.txt` file. The following code demonstrates how I used Linux commands to change file permissions on a hidden file.

```
researcher2@9cd9244e0ac7:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Mar 17 04:04 .
drwxr-xr-x 3 researcher2 research_team 4096 Mar 17 04:11 ..
-rw--w---- 1 researcher2 research_team   46 Mar 17 04:04 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Mar 17 04:04 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_k.txt
-rw------- 1 researcher2 research_team   46 Mar 17 04:04 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_t.txt
researcher2@9cd9244e0ac7:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@9cd9244e0ac7:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Mar 17 04:04 .
drwxr-xr-x 3 researcher2 research_team 4096 Mar 17 04:11 ..
-r--r----- 1 researcher2 research_team   46 Mar 17 04:04 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Mar 17 04:04 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_k.txt
-rw------- 1 researcher2 research_team   46 Mar 17 04:04 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_t.txt
```

The first line of the screenshot shows the `ls -la` command used to display the current state of permissions for all the files in the `projects` directory. In order to view the hidden files you have to add the `-la` option to the command. Using only the `-l` option to the command will not reveal any hidden files. The 5th line of the screenshot shows the permissions for the `.project_x.txt` hidden file. The 11th line of the screenshot shows the `chmod` command used to change the permissions of the `.project_x.txt` hidden file.

Change directory permissions

The organization has informed the security team that the `drafts` subdirectory has some permissions that need to be modified. The `researcher2` user is the only user to have access

to the `drafts` subdirectory and the files inside.

```
researcher2@9cd9244e0ac7:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Mar 17 04:04 .
drwxr-xr-x 3 researcher2 research_team 4096 Mar 17 04:11 ..
-r--r----- 1 researcher2 research_team   46 Mar 17 04:04 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Mar 17 04:04 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_k.txt
-rw------- 1 researcher2 research_team   46 Mar 17 04:04 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_t.txt
researcher2@9cd9244e0ac7:~/projects$ chmod g-x drafts
researcher2@9cd9244e0ac7:~/projects$ ls -l
total 20
drwx------ 2 researcher2 research_team 4096 Mar 17 04:04 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_k.txt
-rw------- 1 researcher2 research_team   46 Mar 17 04:04 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar 17 04:04 project_t.txt
researcher2@9cd9244e0ac7:~/projects$
```

The 6th line of the screenshot shows the current state of permissions for the `drafts` subdirectory. The 11th line of the screenshot shows the `chmod` command used to change the permissions of the `drafts` subdirectory specified in the commands argument. I adjusted the subdirectory by removing its “execute” permissions using the `g-x` command. The 12th line of the screenshot shows the `ls -l` command used to confirm that our permission modification was successful.

Summary

In order to accomplish the organization’s assigned task, I used Linux commands to modify multiple permissions and give the organization proper authorization for the `projects` directory and its contents. Implementing least of privilege is an extremely important step into improving our security posture. Maintaining proper permissions for the corresponding files and directories provides a better stance in protecting the confidentiality and integrity of the organization's data.