File Permissions in Linux

Using Linux Commands to Manage File Permissions

Wendy Alayon
21/06/2024 **@syszern**

File permissions in Linux

Scenario

A security professional at a large organization mainly works with the research team. Part of the professional's job is to ensure that users on this team are authorized with the appropriate permissions, which helps keep the system secure.

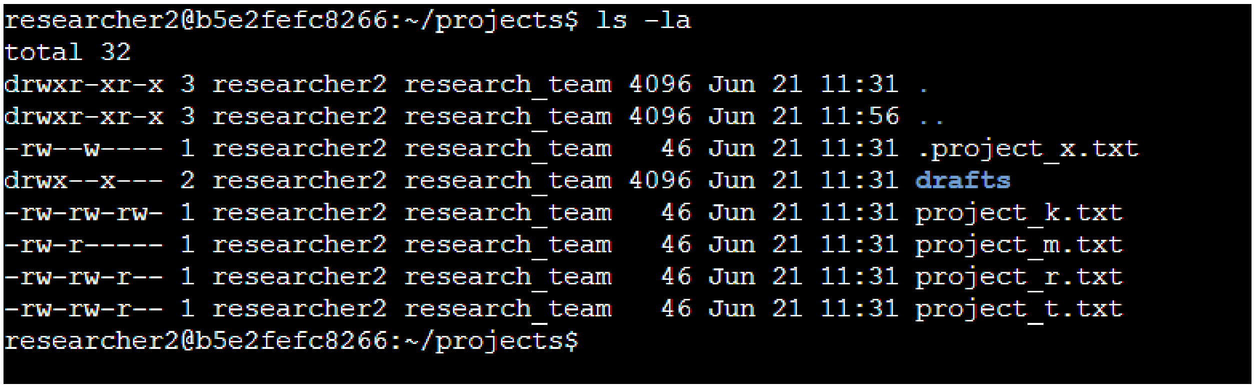
The task for the security professional is to examine existing permissions on the file system. The professional will need to determine if the permissions match the authorization that should be given. If they do not match, the professional will need to modify the permissions to authorize the appropriate users and remove any unauthorized access.

Project description

The research team at the organization needs to update the file permissions for certain files and directories within the `projects` directory. The permissions do not currently reflect the level of authorization that should be given. Checking and updating these permissions will help keep the system secure. To complete this task, the following tasks were performed:

Check file and directory details

The following code demonstrates the use of Linux commands to determine the existing permissions set for a specific directory in the file system.



```
researcher2@b5e2fefc8266:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 21 11:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 21 11:56 ..
-rw--w---- 1 researcher2 research_team   46 Jun 21 11:31 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 21 11:31 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jun 21 11:31 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun 21 11:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 21 11:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 21 11:31 project_t.txt
researcher2@b5e2fefc8266:~/projects$
```

In the screenshot, the command entered is shown on the first line, and the subsequent lines display the output. The code lists all contents of the `projects` directory. The `ls` command with the `-la` option was used to generate a detailed listing of the file contents, including hidden files. The output of the command indicates the presence of one directory named `drafts`, one hidden file named `.project_x.txt`, and five other project files. Each line begins with a 10-character string representing the permissions set on each file or directory.

Describe the permissions string

The 10-character string can be deconstructed to determine who is authorized to access the file and their specific permissions. The characters and what they represent are as follows:

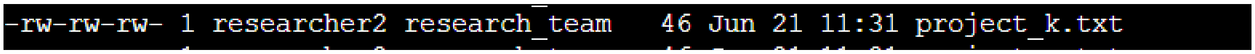
- **1st character:** This character is either a d or hyphen (-) and indicates the file type. If it's a d, it's a directory. If it's a hyphen (-), it's a regular file.

- **2nd-4th characters:** These characters indicate the read (r), write (w), and execute (x) permissions for the user. When one of these characters is a hyphen (-) instead, it indicates that this permission is not granted to the user.

- **5th-7th characters:** These characters indicate the read (r), write (w), and execute (x) permissions for the group. When one of these characters is a hyphen (-) instead, it indicates that this permission is not granted for the group.

- **8th-10th characters:** These characters indicate the read (r), write (w), and execute (x) permissions for other. This owner type consists of all other users on the system apart from the user and the group. When one of these characters is a hyphen (-) instead, that indicates that this permission is not granted for other.

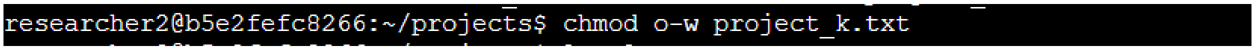
For example, the file permissions for `project_k.txt` are `-rw-rw-rw-`. Since the first character is a hyphen (-), this indicates that `project_k.txt` is a file, not a directory. The second, fifth, and eighth characters are all r, which indicates that user, group, and other all have read permissions. The third, sixth, and ninth characters are w, which indicates that user group, and other all have write permissions. No one has execute permissions for `project_k.txt`.



```
-rw-rw-rw- 1 researcher2 research_team   46 Jun 21 11:31 project_k.txt
```

Change file permissions

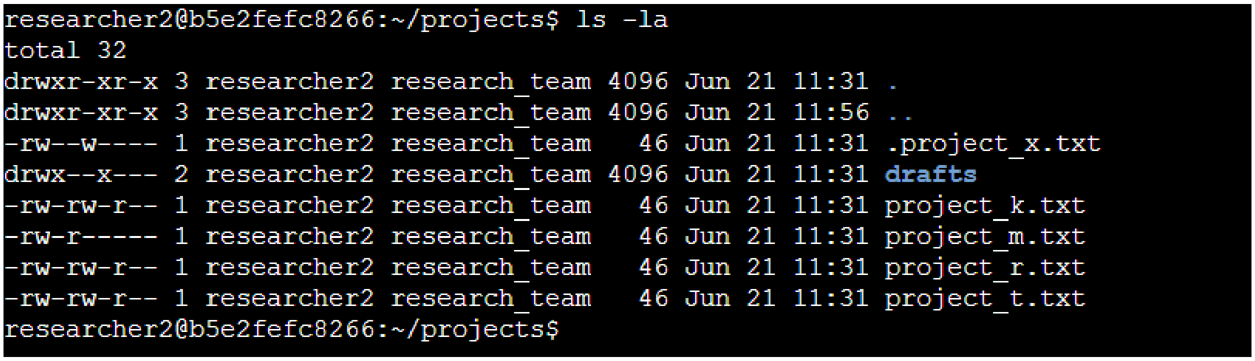
The organization does not allow other to have write access to any files. Based on the current file permissions, the owner type other has `write` permission. The command below removes this permission from the other.



```
researcher2@b5e2fefc8266:~/projects$ chmod o-w project_k.txt
```

The `chmod` command changes permissions on files and directories. The first argument indicates which permissions are being modified, and the second argument specifies the file or directory affected. In this example, write permissions for other were removed from the file `project_k.txt`.

To check if the file permission for the file `project_k.txt` has been updated, `ls -la` was used to review the updates made:

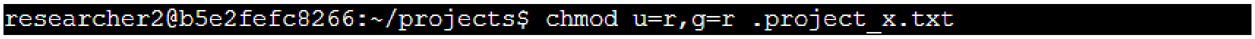


```
researcher2@b5e2fefc8266:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 21 11:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 21 11:56 ..
-rw--w---- 1 researcher2 research_team   46 Jun 21 11:31 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 21 11:31 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun 21 11:31 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun 21 11:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 21 11:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 21 11:31 project_t.txt
researcher2@b5e2fefc8266:~/projects$
```

The write permission for the file `project_k.txt` has been removed from other.

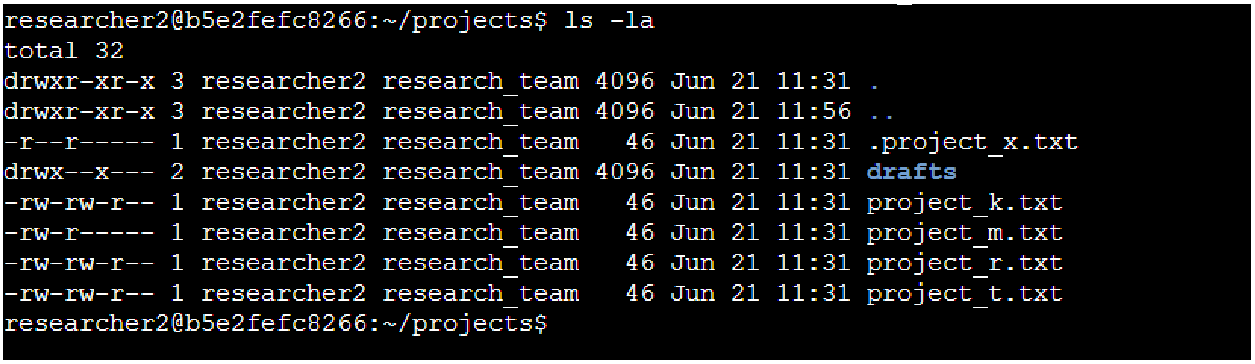
Change file permissions on a hidden file

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. To fix this, the `chmod` command used below assigns only the read permission for both user and group.



```
researcher2@b5e2fefc8266:~/projects$ chmod u=r,g=r .project_x.txt
```

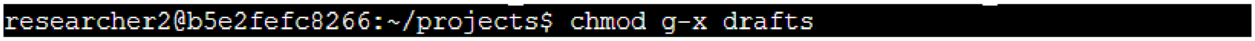
To check if the file permission for the hidden file `.project_x.txt` has been updated:



```
researcher2@b5e2fefc8266:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 21 11:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 21 11:56 ..
-r--r----- 1 researcher2 research_team   46 Jun 21 11:31 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 21 11:31 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun 21 11:31 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun 21 11:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 21 11:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 21 11:31 project_t.txt
researcher2@b5e2fefc8266:~/projects$
```

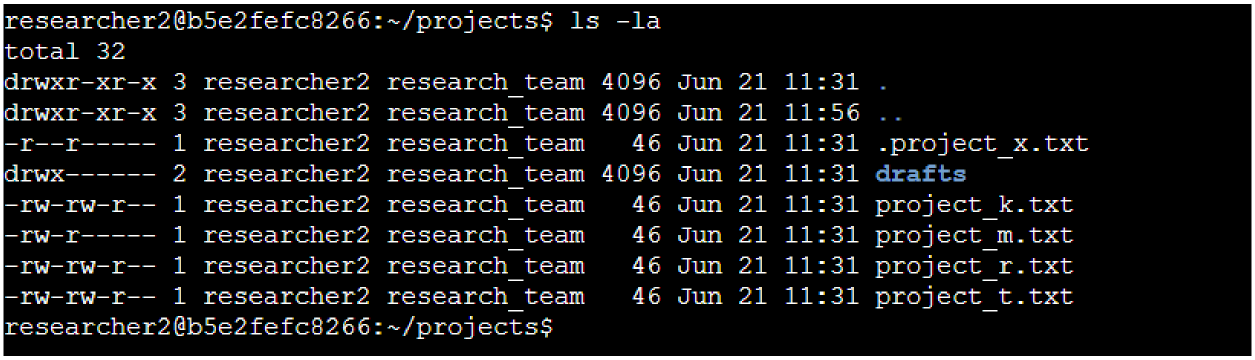
Change directory permissions

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. The Linux command below modifies the permissions accordingly.



```
researcher2@b5e2fefc8266:~/projects$ chmod g-x drafts
```

To check:



```
researcher2@b5e2fefc8266:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 21 11:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 21 11:56 ..
-r--r----- 1 researcher2 research_team   46 Jun 21 11:31 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Jun 21 11:31 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun 21 11:31 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun 21 11:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 21 11:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 21 11:31 project_t.txt
researcher2@b5e2fefc8266:~/projects$
```

The output displays the permission listing for several files and directories:

- Line 1: Current directory (`projects`)
- Line 2: Parent directory (`home`)
- Line 3: Regular file named `.project_x.txt`
- Line 4: Directory named `drafts` with restricted permissions, where only `researcher2` has execute permissions.

It was previously identified that the group had execute permissions on the `drafts` directory. Using the `chmod` command, these permissions were removed. Since `researcher2` already had execute permissions, no changes were necessary for them.

Summary

Multiple permissions were adjusted to align with the desired level of authorization for files and directories in the `projects` directory. The initial step involved using `ls -la` to examine the current permissions of the directory, which guided subsequent decisions. The `chmod` command was then utilized multiple times to modify permissions on files and directories as necessary.