Linux File Permissions Activity

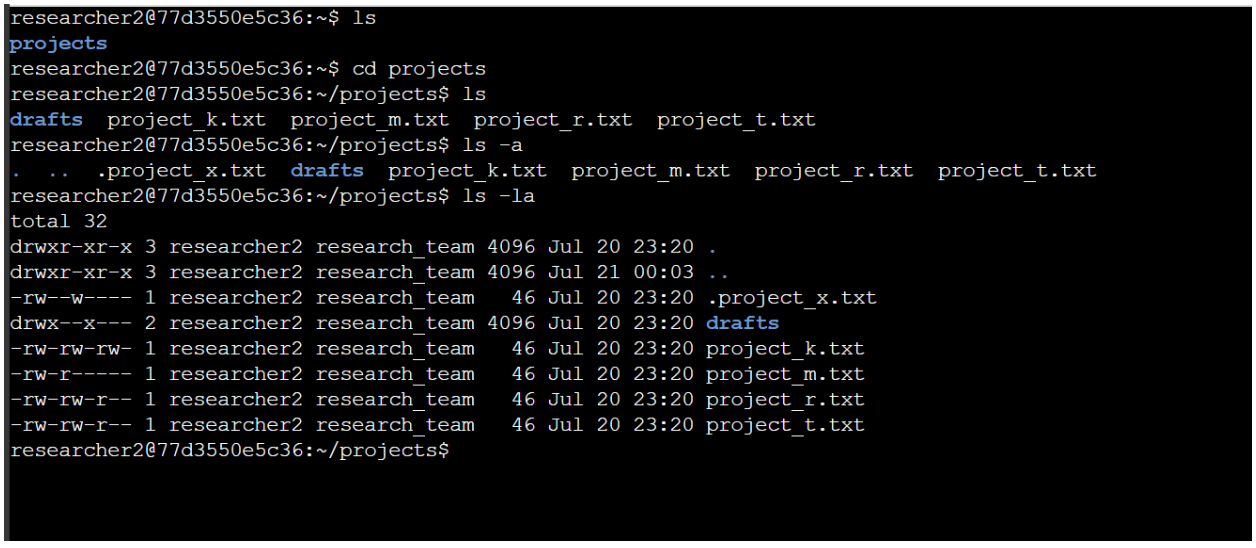
Current file permissions document: [Current file permissions - Google Docs]

Project description

In this scenario, I am a security professional at a large organization and mainly work with the research team. My task is to examine existing permissions in the file system. I need to determine whether they match the authorization that should be given and make any necessary changes. The following sections provide descriptions of the changes that I have made using the linux command line.

Check file and directory details

In the screenshot below, first we view all files and directories using the *ls* command and navigate to the *projects* directory using the *cd projects* command. Once there, we use the *ls* command to check the subdirectories and the files that are not hidden in the *projects* directory. Then, to display all subdirectories and files including those that are hidden, we use the *ls -a* command. Finally, in order to display the permissions of all files and subdirectories in the *projects* directory, we use the *ls -la* command.



Describe the permissions string

When we use the *ls -la* command, we are able to see all of the files and subdirectories contained in the *projects* directory along with their permissions for the user, the group that the user is a part of, and all other users in the system. In the screenshot in the “Check file and

directory details" section above, let's take the file *project_r.txt* as an example. If we find the line containing *project_r.txt* and look all of the way to the left of the screen, we see *-rw-rw-r--.* This 10-character string describes this file's permissions. The "-" at the beginning of the string signifies that the string is describing the permissions for a file. If there was a "d" in place of the "-", this would mean that the string is describing permissions for a directory. An example of this is shown in the permissions for the *drafts* subdirectory in the same screenshot.

The remaining 9 characters of the string can be divided into groups of 3. The first 3 characters are describing the permissions of the user. Since in this case we are the *researcher2* user, these three characters are describing the permissions of the *researcher2* user. The next 3 characters describe the permissions of the group that the user is a part of. In this case, the *researcher2* user is a part of the *research_team* group. This information is shown in the screenshot directly after *researcher2* on each permission line. This means that these 3 characters are describing the permissions of the *research_team* group. Finally, the last 3 characters describe the permissions of all other users who have access to the system outside of the *research_team* group.

Looking at each group of 3 characters, we have a corresponding character for read, write and execute. *Read* privileges (represented by an "r") represent the ability to read the contents of a file, *write* privileges (represented by a "w") represent the ability to add to or change the contents of a file, and *execute* privileges ("represented by an "x") represent the ability to *execute* the file if it is an executable file. In the *project_r.txt* example, the first 3 characters of the remaining 9 are "rw-". This signifies that the user has *read* and *write* privileges, but not *execute* privileges. Had there been an "x" in place of the "-" in this example, it would signify that the user also has *execute* privileges. The next 3 characters are also "rw-". This signifies that the *research_team* group also has *read* and *write* privileges but not *execute* privileges. The final 3 characters are "r--", meaning that all other users outside of the *research_team* group only have *read* privileges.

Change file permissions

In this example, the organization does not allow other users outside of the group to have write privileges. By looking at the file permissions using the the *ls -la* command in the screenshot below, we can see that there is one file in which other users have write permissions. This is the *project_k.txt* file. In order to edit the permissions of other users, we use the *chmod o-w project_k.txt* command. This removes the *write* privileges from other users in the *project_k.txt* file. There are two arguments in the *chmod* command. The first argument is for any changes that we wish to make to a file's permissions. In this case, we use "o-w" with the "o" representing "other users", the "w" representing "write privileges", and the "-" signifying that we are removing privileges. If we were to have used a "+" instead of a "-", it would signify adding privileges rather than removing them. The second argument is just the file name that we want to edit permissions for. Since we are editing permissions for the file *project_k.txt,* we have written that file in the second argument of our *chmod* command. Finally, we check the file permissions of all files again using the *ls -la* command and are able to see that other users no longer have write privileges for any files.

```
researcher2@77d3550e5c36:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jul 20 23:20 .
drwxr-xr-x 3 researcher2 research_team 4096 Jul 21 00:03 ..
-rw--w---- 1 researcher2 research_team   46 Jul 20 23:20 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jul 20 23:20 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jul 20 23:20 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jul 20 23:20 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 20 23:20 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 20 23:20 project_t.txt
researcher2@77d3550e5c36:~/projects$ chmod o-w project_k.txt
researcher2@77d3550e5c36:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jul 20 23:20 .
drwxr-xr-x 3 researcher2 research_team 4096 Jul 21 00:03 ..
-rw--w---- 1 researcher2 research_team   46 Jul 20 23:20 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jul 20 23:20 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jul 20 23:20 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jul 20 23:20 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 20 23:20 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 20 23:20 project_t.txt
researcher2@77d3550e5c36:~/projects$
```

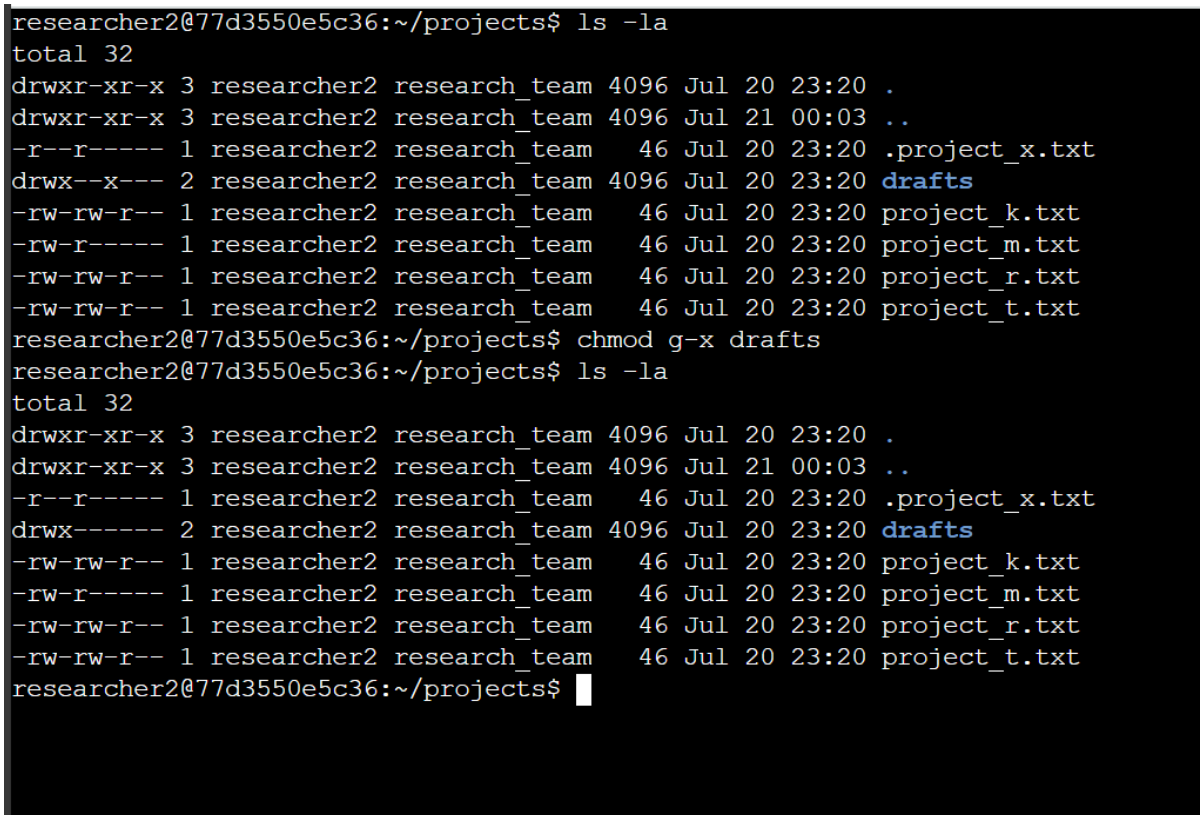
Change file permissions on a hidden file

Next, the research team has archived *.project_x.txt* which is why it is a hidden file. Because of this, no users should be able to write on the file. However, the user in this case, *researcher2* and the *research_team* group should have *read* privileges on this file. Looking at the *.project_x.txt* file permissions in the screenshot below after using the `ls -la` command, we can see that both the user and group have *write* privileges. Additionally, the group does not have any *read* privileges. In this case, we need to take away *write* privileges from both the user and group and give *read* privileges to the group. In order to do so, we use the `chmod u-w,g+r,g-w .project_x.txt`. After executing this command, we use the *ls -la* command to view all privileges again. In doing so, we can see that the *write* privileges have been removed from both the user and the group and both now only have *read* privileges.

```
researcher2@77d3550e5c36:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jul 20 23:20 .
drwxr-xr-x 3 researcher2 research_team 4096 Jul 21 00:03 ..
-rw--w---- 1 researcher2 research_team   46 Jul 20 23:20 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jul 20 23:20 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jul 20 23:20 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jul 20 23:20 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 20 23:20 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 20 23:20 project_t.txt
researcher2@77d3550e5c36:~/projects$ chmod u-w,g+r,g-w .project_x.txt
researcher2@77d3550e5c36:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jul 20 23:20 .
drwxr-xr-x 3 researcher2 research_team 4096 Jul 21 00:03 ..
-r--r----- 1 researcher2 research_team   46 Jul 20 23:20 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jul 20 23:20 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jul 20 23:20 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jul 20 23:20 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 20 23:20 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 20 23:20 project_t.txt
researcher2@77d3550e5c36:~/projects$
```

Change directory permissions

Finally, because all of the files and subdirectories in the *projects* directory belong to the *researcher2* user, only the user should be able to access the *drafts* subdirectory and its contents. This means that only the user should have *execute* privileges on the *drafts* subdirectory. After using the *ls -la* command in the screenshot below, we can see that both the user and the group have *execute* privileges on the *drafts* subdirectory. Since only the user should have *execute* privileges, we need to remove these privileges from the group. In order to do so, we use the *chmod g-x drafts* command. After, we once again use the *ls -la* command and see that the *execute* privileges have been removed from the group and that the user is now the only one that has *execute* privileges on the *drafts* subdirectory.



Summary

Overall, there were several changes to file permissions that needed to be made in order to follow the principle of least privilege and organizational guidelines. These changes have been implemented using the commands that I provided in each section above.