File permissions in Linux

Project description

The purpose of this project is to use the CLI to check the permissions for various users and groups have concerning certain files and directories and correct them as needed to what is appropriate for the context. Inappropriate permissions may allow unauthorized individuals and processes to access, modify, replace and exfiltrate sensitive information which should otherwise remain private or inaccessible.

Check file and directory details

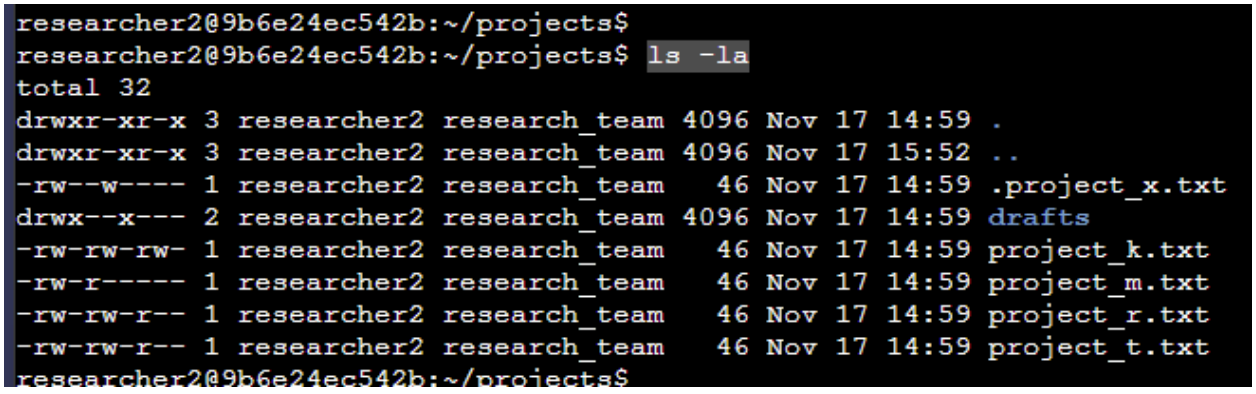
******Project Instructions:*** *The organization does not allow Other to have write access to any files. Identify which file needs to have its permissions modified.*

This screenshot shows me checking the current permissions of the files in the Projects directory with the *ls* command and *-la* flags.

The *ls* command displays the contents of the current directory. In addition, the *-l* flag shows the long form information for each file (name, date, owner, etc.) and the *-a* flag shows all the files in a directory including hidden files.

For this project commands I use are highlighted in gray

Fig. 1



Describe the permissions string

The first section in a file listing is the permissions string which shows who may act on a file and what they are allowed to do with it. The first entry in the string is either a '*d*' (for a directory) or a dash (which indicates a file).

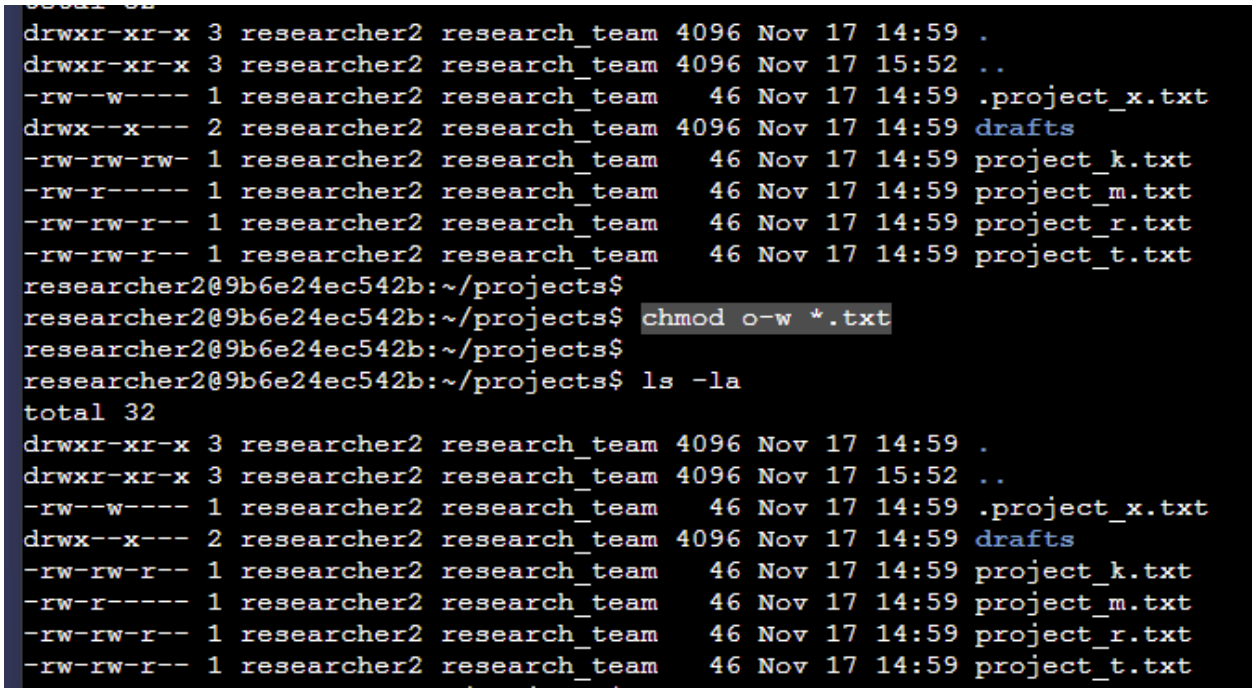
The next three entries show which permissions the User/Owner (in this case, researcher2) has over the file. The second three entries show which permissions Group (in this case, research_team) has over the file and the final three entries show the permissions which Others have over the file.

Each of those entries has four possibilities: read (*r*), write (*w*), execute (*x*) or none (-).

Change file permissions

As seen in *Fig. 2* the Other category has write permissions over the *project_k.txt file*, which should not be allowed by organizational policies. To correct this I use the *chmod* command to remove write permissions from Others (*o-w*) along with a wildcard (*.txt) to ensure the change applies to all files in the directory. Then I use the *ls -al* command to check the new permissions (*Fig. 2*)

Fig. 2

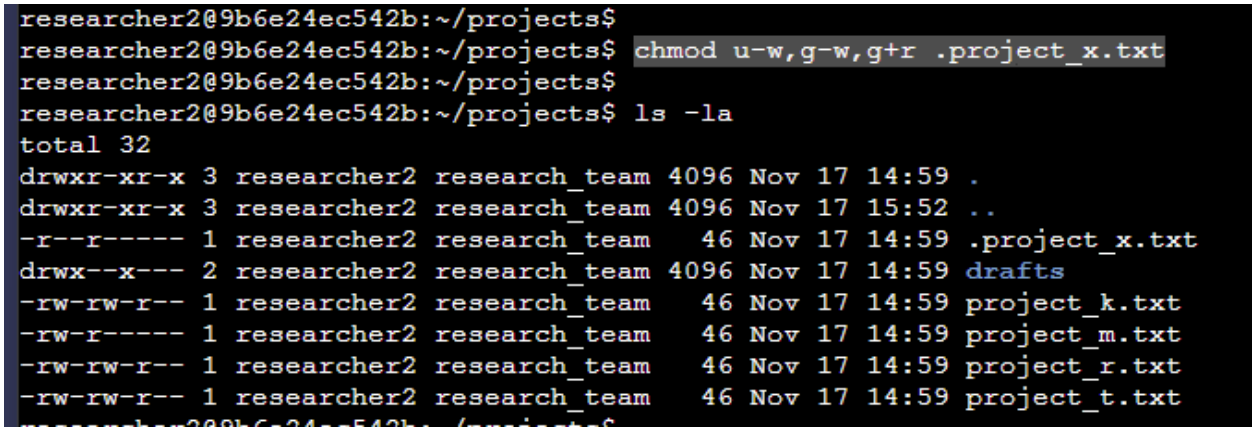


Change file permissions on a hidden file

******Project Instructions:*** *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.*

As can be seen in *Fig. 2*, User and Group have write permissions over the hidden file *.project_x.txt*, and Group doesn't have read permissions. I use *chmod* to grant the appropriate access rights, using *ls* afterwards to check the new permission rights (*Fig. 3*)

Fig. 3



Change directory permissions

******Project Instructions:*** *The files and directories in the projects directory belong to the researcher2 user. Only researcher2 should be allowed to access the drafts directory.*

As seen in *Fig. 3*, the Group has execute permissions for the drafts directory. Again I use *chmod* to assign the appropriate permissions to Group and check to assure the change was made correctly (*Fig. 4, below*)

Fig. 4