File permissions in Linux

Project description

Through linux commands, the permissions of files and directories in a mock system were viewed and edited to fit the requirements set forth by the fictional employer.

Check file and directory details

Used **ls -la** to display all files/directories and their permissions.

- project_k.txt -rw-rw-rw-
- Project_m.txt -rw-r-----
- Project_r.txt -rw-rw-r--
- Project_t.txt -rw-rw-r--
- .project_x.txt -rw--w----
- Drafts drwx--x---

Describe the permissions string

The permission string describes what permissions different people have with the according file/directory. Projext_X as an example. Displays a **-** in the first slot since it is a file. The next three characters are the permissions for the current user. In this case **rw-** means that the current user can read and write the file, but not execute it (if possible). The next three are for the group the object belongs to. So in this case **-w-** means that the group that owns Projext_x can only write to the file. The last three are for anyone else. **---** means they have no permissions.

Change file permissions

No file can be written to by the “other” group. Therefore, write permissions were removed from project_k,txt for the others group with the command **chmod o-w project_k.txt**.

Change file permissions on a hidden file

Projext_x should not be able to be written to by anyone, but user and group should be able to read it. Thus, write permissions were removed for both user and group and read permissions were added for group with the command **chmod u=r, g=r .project_x.txt**.

Change directory permissions

Only the user **researcher** should be able to do anything with the directory titled **drafts**. So execute permissions were removed for group with the command **chmod g-x drafts**.

Summary

Permissions of all files and directories were viewed with **ls -a**. They were then modified accordingly based on requirements placed fourth and the principle of least privilege in action.