File permissions in Linux

Cameron Haley

Project description

This is a quick example of my competence in file permission management in Linux. Though I am capable of much more in-depth operations and scripting, it's a good example of some core competencies and also a required project for the google cybersecurity certification ;-)

Checking file and directory details

```
researcher2@74af64454eab:~$ cd projects
researcher2@74af64454eab:~/projects$ ls
drafts  project_k.txt  project_m.txt  project_r.txt  project_t.txt
researcher2@74af64454eab:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 25 12:28 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 25 13:16 ..
-rw--w---- 1 researcher2 research_team   46 Aug 25 12:28 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug 25 12:28 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Aug 25 12:28 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 25 12:28 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 12:28 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 12:28 project_t.txt
researcher2@74af64454eab:~/projects$
```

As you can see above, I entered the 'projects' directory, took a quick glance at the files in the directory, and then took a more in-depth look at all of the files in the directory and their permissions using ls with 'l' (long) and 'a' (all) flags.

The permissions strings of each file are shown first in each row of the long list (ls -l) and tell us the type of file, and who has what level of permissions for each file. The first character shows us which files are directories and which are regular files, denoted by either a 'd' or a '-'. The next three sets of three characters reflect the permissions of the user, group, and 'other'. As we can see in the fourth row, 'drafts' is a directory that the current user (researcher2) has read, write, and execute privileges of. The group, research_team, has execute privileges, but not read/write. Finally, 'other' has no permissions on this file.

Change file permissions

Below is an example of examining the contents of a directory using **ls**, then changing permissions using the **chmod** command.

```
drwxr-xr-x 3 researcher2 research_team 4096 Aug 25 15:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 25 16:09 ..
-rw--w---- 1 researcher2 research_team   46 Aug 25 15:31 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug 25 15:31 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Aug 25 15:31 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 25 15:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 15:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 15:31 project_t.txt
researcher2@bcba617fbc5b:~/projects$ chmod g-rw,o-rw project_k.txt
researcher2@bcba617fbc5b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 25 15:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 25 16:09 ..
-rw--w---- 1 researcher2 research_team   46 Aug 25 15:31 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug 25 15:31 drafts
-rw------- 1 researcher2 research_team   46 Aug 25 15:31 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 25 15:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 15:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 15:31 project_t.txt
researcher2@bcba617fbc5b:~/projects$
```

To use **chmod** the first argument passed indicates what permissions you want to change and for whom. The second argument is the file that you are changing permissions of. An easy example would be:

`chmod u-r project_k.txt.`

This command removes read permissions (u-r) of the project k file. In the screenshot above, I removed read and write permissions of the group and other by passing the second argument:

`g-rw,o-rw`

After entering this command, I checked the file permissions again and as you can see in the permissions string, the command successfully changed the permissions of project k.

Summary

Though this is a simple example, understanding file permissions is a key component of being an effective cyber security analyst/engineer. When I am playing with self-hosted linux boxes running Ubuntu or playing around on HTB with Parrot, I have found that my systems programming class from my undergrad has been hugely beneficial in helping me with cybersecurity projects. I am comfortable with bash/zsh and accepting of all different flavors of text editor. Even nano.