Jake Murray

Google Cybersecurity Certification

File permissions in Linux

Project description

For this project, I am a security professional at a fictional organization. My job is to update the permissions for a `projects` directory for the research team, as they do not currently reflect the correct levels of authorization. This is how I completed the project.

Check file and directory details

I started by navigating to the `/home/researcher2/projects directory` and using the `li` command with `-la` options to list all of the child directories and files along with their permissions.

```
researcher2@db569db84b65:~$ cd projects/
researcher2@db569db84b65:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb 27 02:10 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb 27 02:57 ..
-rw--w---- 1 researcher2 research_team   46 Feb 27 02:10 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Feb 27 02:10 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Feb 27 02:10 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Feb 27 02:10 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 27 02:10 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 27 02:10 project_t.txt
```

There is 1 `drafts` directory and 5 project files, one of which is hidden.

Describe the permissions string

The permissions string is a 10 letter string broken up into 4 section.
The first letter (in this example `d` or `-`) indicate whether you are looking at a directory or file.
The next 9 characters are 3 sequences of the same 3 characters. Each sequence can have the letters `rwx`. Each character stands for a specific permission; `r` for read, `w` for write, and `x` for execute. If there is a dash instead of a letter, it means that that permission is not granted to that user or group.
The first sequence of these characters references the permissions of the user (researcher2), the second sequence references the permissions of the group (research_team) that the user

belongs to, and the third sequence references users from other parts of the system outside of the user and group.

So, for example, if we look at the `drafts` directory:
The permissions string for this directory are `drwx--x---`

`d` tells us that we are looking at a directory (folder) and not a file.

`rwx` is the first sequence of permissions, so it tells us that the user has read, write, and execute permissions.

`--x` is the second sequence and tells us that the group has only execute permissions.

`---` is the final sequence, telling us that users outside of researcher2 and research_team have no permissions at all to access this directory.

Change file permissions

It was determined that none of the files should allow other users to write. So my first task was to use the `chmod o-w project_k.txt` command to remove the write permission for others from `project_k.txt`.

Here is the result:

```
drwxr-xr-x 3 researcher2 research_team 4096 Feb 27 02:10 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb 27 02:57 ..
-rw--w---- 1 researcher2 research_team   46 Feb 27 02:10 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Feb 27 02:10 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Feb 27 02:10 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Feb 27 02:10 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 27 02:10 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 27 02:10 project_t.txt
```

Next, the group should have no permissions for `project_m.txt`.
I used `chmod g-r project_m.txt` to complete this.

Change file permissions on a hidden file

The file `.project_x.txt` is a hidden file that has been archived and should not be written to by anyone. (The user and group should still be able to read this file.)

I fixed this with the command `chmod g-w,u-w,g+r .project_x.txt`.

Change directory permissions

Only the `researcher2` user should be allowed to access the `drafts` directory and its contents. (This means that only `researcher2` should have execute privileges.)

I used `chmod g-x drafts` to fix this permission.

Summary

Here is the end result.

```
researcher2@60d73cf3cab7:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb 27 03:20 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb 27 04:01 ..
-r--r----- 1 researcher2 research_team   46 Feb 27 03:20 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Feb 27 03:20 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Feb 27 03:20 project_k.txt
-rw------- 1 researcher2 research_team   46 Feb 27 03:20 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 27 03:20 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Feb 27 03:20 project_t.txt
researcher2@60d73cf3cab7:~/projects$
```

I used the `chmod` command several times according to authorizations that were provided to me. This fixed file and directory permissions, increasing security by allowing only those who needed specific permissions to read, write, and execute files and directories.