Auditing file permissions in Linux

Audit description

As a security professional at a large organization, I work with the research team to ensure that users have the appropriate permissions to access files. This allows them to do their regular duties, but also helps to keep the file system secure.

As part of the job, I regularly audit the file system permissions to make sure they match the authorization that should be given. If they do not match, I modify the permissions to authorize the appropriate users and remove any unauthorized access.

- **Audit Goal:** To ensure that users on the research team have the appropriate permissions to access files on the file system.
- **Audit Scope:** The audit will include the following tasks:
    - Examination of existing file system permissions.
    - Determination of whether permissions match authorization.
    - Modification of permissions to authorize appropriate users and remove unauthorized access.

By usingLinux commands, you can audit file permissions and modify them as needed to ensure that only authorized users have access to files. This helps to keep the file system secure and protect the confidentiality, integrity, and availability of data.

Check file and directory details

```
researcher2@f372295f2a44:~$ pwd
/home/researcher2
researcher2@f372295f2a44:~$ ls
projects
researcher2@f372295f2a44:~$ cd projects
researcher2@f372295f2a44:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jul 17 21:51 .
drwxr-xr-x 3 researcher2 research_team 4096 Jul 17 22:35 ..
-rw--w---- 1 researcher2 research_team   46 Jul 17 21:51 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jul 17 21:51 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jul 17 21:51 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jul 17 21:51 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 17 21:51 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 17 21:51 project_t.txt
researcher2@f372295f2a44:~/projects$
```

1. To start off I like to confirm that I am in the directory where I need to do my work in. By typing the command `pwd`, we are asking the system to show us the working directory with this command.
2. Next, I'll type the `ls` command to list the contents of a directory. We can see a directory called projects. This is a directory that we need to audit.
3. From this point, we will use the `cd` command to change the directory. After typing in this command, we will now be working within the appropriate directory
4. Using the `ls` command combined with the `-la` command, we start to dive deeper into the directory and are able to view contents and file permissions. Using `-la` tells the system to return information about files and is a powerful command because it allows you to see all of the files in a directory, including hidden files.

Describe the permissions string

The permissions string is a combination of three characters for each of the three user categories: owner, group, and others. The characters represent the following permissions:

- **r** - read
- **w** - write
- **x** - execute

In the `/home/researcher2/projects` directory, there are five files with the following names and permissions:

- `project_k.txt`
    - User = read, write,
    - Group = read, write
    - Other = read, write
- `project_m.txt`
    - User = read, write
    - Group = read
    - Other = none
- `project_r.txt`
    - User= read, write
    - Group = read, write
    - Other = read
- `project_t.txt`
    - User = read, write
    - Group = read, write
    - Other = read
- `.project_x.txt`

    - User = read, write
    - Group = write
    - Other = none

There is also one subdirectory inside the `projects` directory named `drafts`. The permissions on `drafts` are:

- User = read, write, execute
- Group = execute
- Other = none

Let's take a look at the output string associated with a hidden file we discovered: `.project_x.txt`

```
-rw--w---- 1 researcher2 research_team   46 Jul 17 21:51 .project_x.txt
```

- **-rw--w----** is the permissions string for the file. It indicates that the owner has read and write permissions, the group has write permissions, and others have no permissions.
- **1** is the number of hard links to the file.
- **researcher2** is the owner of the file.
- **research_team** is the group of the file.
- **46** is the size of the file in bytes.
- **Jul 17 21:51** is the date and time the file was last modified.
- **.project_x.txt** is the name of the file.

Change file permissions

I noticed that incorrect permissions were set on the other category, in the `project_k.txt` file. We need to remove this, because the organization does not allow other to have write access to any files. Here's what happens when we type in `chmod o-w project_k.txt`

```
researcher2@8b9d566e0adf:~$ cd projects
researcher2@8b9d566e0adf:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Jul 17 22:40 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jul 17 22:40 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jul 17 22:40 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 17 22:40 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 17 22:40 project_t.txt
researcher2@8b9d566e0adf:~/projects$ chmod o-w project_k.txt
researcher2@8b9d566e0adf:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Jul 17 22:40 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jul 17 22:40 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jul 17 22:40 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 17 22:40 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 17 22:40 project_t.txt
researcher2@8b9d566e0adf:~/projects$
```

The `chmod` command will remove the write permissions for others on the file. Since the other category has been removed from write permission, only the user and group has the right permissions, per our audit instructions. The owner and group can still write to the file, but others cannot write to the file.

Change file permissions on a hidden file

```
-rw--w---- 1 researcher2 research_team   46 Jul 17 22:40 .project_x.txt
```

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. Let's assign the appropriate authorization now and then we will see what the correct permissions should actually be.

By typing in the command `chmod u-w .project_x.txt` and `chmod g-w .project_x.txt` we will be able to remove the user and group write permissions.

```
researcher2@8b9d566e0adf:~/projects$ chmod u-w .project_x.txt
researcher2@8b9d566e0adf:~/projects$ chmod g-w .project_x.txt
researcher2@8b9d566e0adf:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jul 17 22:40 .
drwxr-xr-x 3 researcher2 research_team 4096 Jul 17 23:35 ..
-r-------- 1 researcher2 research_team   46 Jul 17 22:40 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jul 17 22:40 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jul 17 22:40 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jul 17 22:40 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 17 22:40 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 17 22:40 project_t.txt
researcher2@8b9d566e0adf:~/projects$
```

Viola! A quick confirmation using the `ls -la` command confirms that neither the user, nor the group can write. We now have the correct authorization set.

Change directory permissions

The last thing we need to audit are the files and directories in the projects directory. They belong to the *researcher2* user, so only *researcher2* should be allowed to access the *drafts* directory and its contents. Let's check the permissions and change them if necessary.

```
researcher2@8b9d566e0adf:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Jul 17 22:40 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jul 17 22:40 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jul 17 22:40 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 17 22:40 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 17 22:40 project_t.txt
researcher2@8b9d566e0adf:~/projects$
```

Upon further inspection, we found that the research_team group does indeed have execute permissions that they should not have. At this time, we need to make sure that execute permissions are removed.

```
researcher2@8b9d566e0adf:~/projects$ chmod g-x drafts
researcher2@8b9d566e0adf:~/projects$ ls -l
total 20
drwx------ 2 researcher2 research_team 4096 Jul 17 22:40 drafts
```

By typing in the command `chmod g-x drafts` we were able to remove the group execute permissions. We are now able to confirm that the audit has been finished and permissions have been set according to the organization's needs.

Summary

This audit was to ensure that users on the research team have the appropriate permissions to access files on the file system by:

- Examining existing file system permissions, determining whether permissions match authorization.
- Modifying permissions to authorize appropriate users.
- Removing unauthorized access.

It is important to regularly audit file system permissions to make sure they match authorization. The chmod command can be used to modify file permissions to authorize appropriate users and remove unauthorized access. Only authorized users should have access to files, as this helps to keep the file system secure.