File permissions in Linux

Project description

As a security analyst at a large organization, my primary responsibility is to work closely with the research team and ensure that proper file permissions are in place to maintain system security. Through thorough examination of the file system's existing permissions, I verify if users within the team possess the appropriate authorizations. If any discrepancies are found, I promptly make the necessary modifications to grant authorized users their required access while removing any unauthorized permissions. This proactive approach ensures that the system remains secure and minimizes potential risks associated with unauthorized access.

Check file and directory details

- As a security analyst when checking file permissions we must make sure that files and directories follow the least amount of privilege format to stay secure. While in the `/home/researcher2/projects` directory for `researcher2` user I would execute command `ls -l` to check permissions for files and directories.

```
researcher2@d1a349d584c4:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Jul 22 17:10 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jul 22 17:10 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jul 22 17:10 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 22 17:10 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 22 17:10 project_t.txt
```

- If there are hidden files I will execute command `ls -la` to check the permissions. `ls -la` allows you to check the permissions of files and directories, including the hidden files.

```
researcher2@d1a349d584c4:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jul 22 17:10 .
drwxr-xr-x 3 researcher2 research_team 4096 Jul 22 17:38 ..
-rw--w---- 1 researcher2 research_team   46 Jul 22 17:10 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jul 22 17:10 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jul 22 17:10 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jul 22 17:10 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 22 17:10 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 22 17:10 project_t.txt
```

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

Permission	Description
Read (r)	Files can be read.
Write (w)	Files allow modifications of content
Execute (x)	Files can be executed if it's an executable file.

- **.project_x.txt**
  - User = read, write
  - Group = write
  - Other = none
- **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

There is also 3 subdirectories inside the projects directory with the following names and permissions:

- **.**
  - User = read, write, execute
  - Group = read, execute
  - Other = read, execute
- **..**
  - User = read, write, execute
  - Group = read, execute

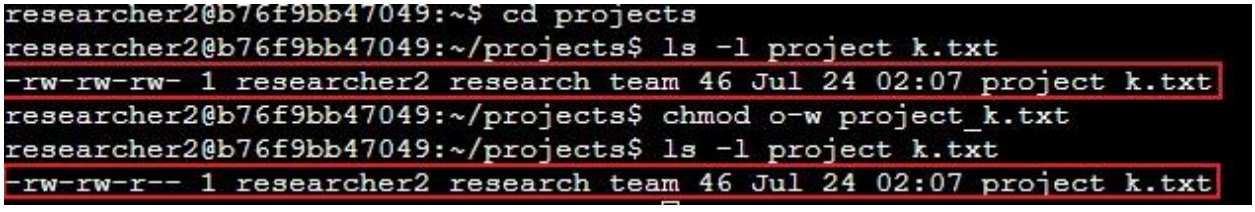
    - Other = read, execute
- **drafts**
    - User = read, write, execute
    - Group = execute
    - Other = none

Describe the permissions string

- **`Drafts` = dr,w,x-,-,x-,-,-**
    - Owner (User): has read, write, and execute (drwx--x--) privileges, which means they can view, edit, and access files within the directory.
    - Group: has execute privileges (--x--x--), which allows members of the group to access files within the directory but not read or write to them.
    - Other: has no permissions (-------), which means they have no access to the directory or its files.
        - Explanation:
            - The owner has full control over the directory, allowing them to read, write, and execute files. This is usually the person who created the directory and is responsible for managing its contents.
            - The group only has execute permissions, indicating that members of the group can access files in the directory without being able to modify or read them. This setup is suitable when multiple users belong to a specific group and require shared access to the directory's content.
            - Other users, who do not fall under the owner or group categories, have no permissions at all. They cannot access, read, or modify any files within the directory. This could be appropriate for maintaining confidentiality or restricting access to sensitive information to authorized individuals only.

Change file permissions

- **`project_k.txt`**



    - To remove write permissions from other users for the file **`project_k.txt`**, you can use the **`chmod`** command in the shell. After executing the command, other

users (users who are not the owner of the file and not part of the file's group) will no longer have permission to modify `project_k.txt`. They will still be able to read the file, but any attempt to modify or write to the file will be denied.

Change file permissions on a hidden file

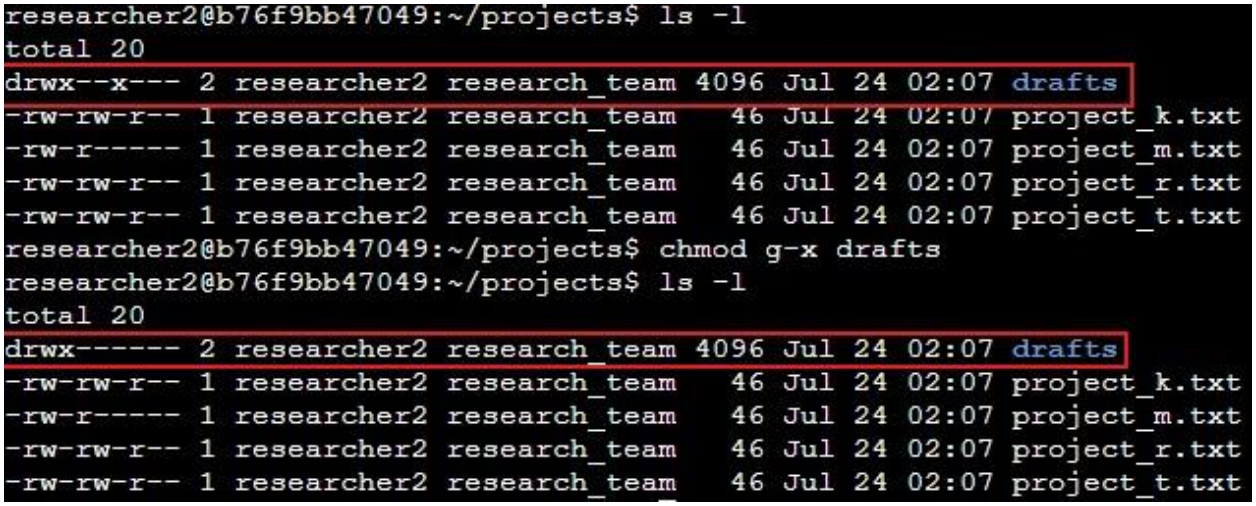
- `.project_x.txt`

```
researcher2@b76f9bb47049:~/projects$ ls -la .project_x.txt
-rw--w---- 1 researcher2 research_team 46 Jul 24 02:07 .project_x.txt
researcher2@b76f9bb47049:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@b76f9bb47049:~/projects$ ls -la .project_x.txt
-r--r----- 1 researcher2 research_team 46 Jul 24 02:07 .project_x.txt
```

  - To change the file permissions on hidden file `.project_x.txt` while ensuring that only the user/owner and a specific group have read authorization, you can use the `chmod` command along with appropriate options.
    - First, identify the group that should have access to the file.
    - Next, set the permissions using the `chmod` command as follows:
      - `chmod`: The command used to change file permissions.
      - `u-w`: This removes the write permission from the user.
      - `g-w`: This removed the write permission from the specific group.
      - `g+r`: This adds the read permission to the specific group.
      - `.project_x.txt`: The second argument that is specifying the file that needs the permissions changed.
  - After executing the command, only the file owner and specific group will have read-only permissions and all other users will have no permissions to the `.project_x.txt` file.

Change directory permissions

- `Draft` directory



- To change the permissions for directory **`drafts`** while ensuring that only the user/owner has full read,write, and execute authorization, and group and other users have no permissions you can use the **`chmod`** command along with appropriate options.
    - First, identify the group that should have access to the file.
    - Next, set the permissions using the **`chmod`** command as follows:
        - **`chmod`**: The command used to change file permissions.
        - **`g-x`**: This removes the execute permission from the specific group.
        - **`drafts`**: The second argument that is specifying the file that needs the permissions changed.
- After executing the command, only the file owner will have read, write and execute permissions and group and all other users will have no permissions to the **`draft`** directory.
-

Summary

As a security analyst for the research team, I conducted a comprehensive review of file permissions in the system and identified areas for improvement. To enhance security, several modifications were implemented. For the “project_k.txt,” write access was revoked from users not belonging to the research team, allowing them read-only access. The hidden file “.project_x.txt” had all write permissions removed for both users and a specific group, restricting access to read-only. Additionally, the “draft” directory now grants full permissions solely to the user/owner, limiting access to unauthorized users. These changes were enacted to bolster system security, reduce the risk of unauthorized access, and mitigate potential threats and attacks.