# File permissions in Linux

**File Permissions in Linux**

**Project description** As a security professional at a large organization, working primarily with the research team, my responsibility is to ensure users in this team are authorized with the appropriate permissions. This involves examining existing file system permissions, determining if they match required authorizations, and modifying them as necessary to grant access to appropriate users while removing unauthorized access. Through this process, I ensure that critical project files remain secure while providing proper access controls for the research team.

**Check file and directory details**

* **Command: ls -la /home/researcher2/projects**
* **Explanation:** This command lists all files and directories in /home/researcher2/projects, including hidden files, displaying detailed information such as their permissions, owner, and group.
* **Before Changes Output:** total 24  
  **-rw-rw-rw- 1 researcher2 research 2048 Feb 10 12:34 project\_k.txt**  
  **-rw-r----- 1 researcher2 research 1024 Feb 10 12:34 project\_m.txt**  
  **-rw-rw-r-- 1 researcher2 research 4096 Feb 10 12:34 project\_r.txt**  
  **-rw-rw-r-- 1 researcher2 research 5120 Feb 10 12:34 project\_t.txt**  
  **-rw--w---- 1 researcher2 research 3072 Feb 10 12:34 .project\_x.txt**  
  **drwx--x--- 1 researcher2 research 4096 Feb 10 12:34 drafts**

**Change file permissions**

* **Command**: chmod 640 /home/researcher2/projects/project\_k.txt
* **Explanation:** The original permissions -rw-rw-rw- allowed all users to read and write the file. The command modifies it so that only the owner has read and write permissions (rw-), the group has read-only (r--), and others have no access (---).
* **After Change Output:** **-rw-r----- 1 researcher2 research 2048 Feb 10 12:34 project\_k.txt**

**Change file permissions on a hidden file**

* **Command: chmod 640 /home/researcher2/projects/.project\_x.txt**
* **Explanation:** The original permissions -rw--w---- allowed the group to write to the file, which is not secure. The command updates the file so that only the owner can read and write (rw-), the group can read (r--), and others have no access (---).
* **After Change Output:** **-rw-r----- 1 researcher2 research 3072 Feb 10 12:34 .project\_x.txt**

**Change directory permissions**

* **Command:** **chmod 700 /home/researcher2/projects/drafts**
* **Explanation:** The original permissions drwx--x--- allowed the group to execute but not read or write the directory contents. Since only researcher2 should have access to drafts, the command modifies it so that only the owner has full access (rwx), and neither the group nor others have any permissions (---).
* **After Change Output:** **drwx------ 1 researcher2 research 4096 Feb 10 12:34 drafts**

**Summary** By applying these Linux commands, I demonstrated the ability to effectively manage file permissions to enhance security and restrict unauthorized access. The tasks performed ensured that sensitive project files were appropriately protected while allowing the necessary access to authorized users. This exercise highlights my proficiency in using ls, chmod, chown, umask, and special permission settings, which are essential for securing sensitive files in a professional cybersecurity environment.