Title: Security Analysis of File Permissions and User Access in Linux Operating Systems

1. Experiment aim:

The aim of this experiment is to investigate the effectiveness of different user groups and file access permissions in securing files on a Linux system.  
   
2. Theoretical background:

The experiment is based on the concept of user groups and file access permissions in Linux systems. User groups are collections of users that share common permissions for files and directories. File access permissions determine who can read, write, or execute a file.

**Experiment Design**

The experiment involves three scenarios:

* Exploring chmod Commands Scenario: This scenario focuses on setting different permissions for three text files in a directory.
* User Groups and File Access Scenario: This scenario involves creating a new user and adding it to a group, ensuring that the group has access to all files in a directory, and testing the user's ability to modify a file.
* Logging and Monitoring File Access Scenario: This scenario explores the use of file access logging to record all file access events and tests the effectiveness of this method in monitoring file access.

**Experiment Results**

The results of the experiment show that:

* Chmod Commands Scenario: Different chmod commands were used to set specific permissions for three text files.
* User Groups and File Access Scenario: "TestUser" was added to the "SecureGroup" and was unable to modify "file3.txt" because the group permission for this file was set to read, write, and execute only for the owner and the group.
* Logging and Monitoring File Access Scenario: File access logging was enabled for the "SecureFiles" directory, and a log file was created to record all file access events. When accessing "file1.txt" from another user account, information logged included details such as the user who accessed the file, the time of access, and the type of access.

3. Research:

**Ex. 1. Exploring chmod Commands Scenario:**

1. Create a new directory named "SecureFiles" in your home directory.

2. Inside "SecureFiles," create three text files: "file1.txt," "file2.txt," and "file3.txt."

3. Set the following permissions:

* "file1.txt" should be readable, writable, and executable by the owner, and readable by others.
* "file2.txt" should be readable and writable by the owner only.
* "file3.txt" should be readable, writable, and executable by the owner and the group.

# Create a new directory named "SecureFiles" in the home directory

mkdir ~/SecureFiles

# Inside "SecureFiles," create three text files: "file1.txt," "file2.txt," and "file3.txt"

touch ~/SecureFiles/file1.txt

touch ~/SecureFiles/file2.txt

touch ~/SecureFiles/file3.txt

# Set permissions for "file1.txt"

chmod 744 ~/SecureFiles/file1.txt

# Set permissions for "file2.txt"

chmod 600 ~/SecureFiles/file2.txt

# Set permissions for "file3.txt"

chmod 770 ~/SecureFiles/file3.txt

**Questions:**

What chmod commands did you use to set the specified permissions?

For "file1.txt": chmod 744 ~/SecureFiles/file1.txt

For "file2.txt": chmod 600 ~/SecureFiles/file2.txt

For "file3.txt": chmod 770 ~/SecureFiles/file3.txt

How do the permissions of each file affect user access?

For "file1.txt":

* Owner: Read, Write, Execute
* Group: Read
* Others: Read

For "file2.txt":

* Owner: Read, Write
* Group: No access
* Others: No access

For "file3.txt":

* Owner: Read, Write, Execute
* Group: Read, Write, Execute
* Others: No access

**Ex. 2.  User Groups and File Access Scenario:**

1. Create a new user named "TestUser" on your Linux system.
2. Add "TestUser" to a group named "SecureGroup."
3. Ensure that "SecureGroup" has read and write access to all files inside the "SecureFiles" directory.
4. Log in as "TestUser" and attempt to modify "file3.txt" inside the "SecureFiles" directory.

# Add a new user named "TestUser" to the Linux system

sudo adduser TestUser

# Add "TestUser" to a group named "SecureGroup"

sudo addgroup SecureGroup

sudo usermod -aG SecureGroup TestUser

# Ensure that "SecureGroup" has read and write access to all files inside the "SecureFiles" directory

chmod -R 660 ~/SecureFiles

# Log in as "TestUser" and attempt to modify "file3.txt" inside the "SecureFiles" directory

su - TestUser

nano ~/SecureFiles/file3.txt

**Questions:**

How did you add "TestUser" to the "SecureGroup"?

Added "TestUser" to the existing group "SecureGroup" using the command: sudo usermod -aG SecureGroup TestUser

Why was "TestUser" unable to modify "file3.txt"? What permissions were missing?

For "file1.txt": The owner has read, write, and execute permissions, while others have only read permissions.

For "file2.txt": The owner has read and write permissions, while others have no permissions.

For "file3.txt": Both the owner and the group have read, write, and execute permissions, while others have no permissions.

**Ex. 3. Logging and Monitoring File Access Scenario:**

1. Enable file access logging for the "SecureFiles" directory.
2. Create a log file to record all file access events.
3. Access "file1.txt" from another user account, and check the log file for the recorded event.

# Enable file access logging for the "SecureFiles" directory

sudo nano /etc/audit/audit.rules

# Add the following line to the audit rules file:

-w /home/username/SecureFiles -p wa

# Restart the audit daemon

sudo systemctl restart auditd

# Create a log file to record all file access events

touch /var/log/file\_access.log

# Access "file1.txt" from another user account and check the log file for the recorded event

cat ~/SecureFiles/file1.txt

cat /var/log/file\_access.log

**Questions:**

How did you enable file access logging for the directory?

File access logging was enabled by adding a rule to the audit rules file specifying monitoring of the "SecureFiles" directory for write and attribute changes.

What information is logged when accessing "file1.txt" from another user account?

When accessing "file1.txt" from another user account, information logged includes details such as the user who accessed the file, the time of access, and the type of access (read, write, etc.).

4. Conclusions:

The experiment demonstrated the importance of user groups and file access permissions in securing files on a Linux system. It also highlighted the effectiveness of file access logging in monitoring file access. The experiment provided insights into the best practices for setting file permissions and the importance of user groups in managing file access.