Innovate Solutions Project Access Management

Learning Objectives

By completing this exercise, you will:

- Set up a Linux environment for user/group and permission-based access control
- Create and manage user accounts and groups
- Configure group-based directory permissions
- Verify and test user access scenarios

Scenario Overview

You are the newly hired System Administrator at **Innovate Solutions**. The company has two teams:

- The **Alpha Development Team**: needs **full control** over their project directory.
- The **Beta Testing Team**: needs **read-only access** to shared reports.

You will create appropriate users, groups, and access controls to reflect this structure. Additionally, the **development lead** should have read access to Beta reports.

Instructions

Step 1: Create Project Groups and Users

1. Set up two new groups named after the two teams: one for developers and one for testers.
2. Create the following user accounts:
    - `dev_lead` and `dev_junior` for the development team
    - `tester_lead` and `tester_junior` for the testing team
      `guest_user` who is not part of any team
3. Set a temporary password for each user (e.g., `password123` for lab use only).
4. Confirm that the new users and groups have been correctly added by inspecting the relevant system files.

Checkpoint Question: What group is assigned to new users by default if no group is specified?

Step 2: Assign Users to Groups

1. Add `dev_lead` and `dev_junior` to the development team group.
2. Add `tester_lead` and `tester_junior` to the testing team group.
3. Assign `dev_lead` to both groups (developer and tester) so they can read testing reports.

Checkpoint Question: After adding `dev_lead` to both groups, use a command to list their group memberships. What do you observe?

Step 3: Set Up Project Directories

1. Create a parent directory called `/projects`, if it doesn't exist.
2. Inside `/projects`, create:
   - A directory for the Alpha developers (`/projects/alpha`)
   - A directory for Beta reports (`/projects/beta_reports`)
3. Assign group ownership of each directory:
   - The Alpha directory should belong to the development group.
   - The Beta reports directory should belong to the testing group.
4. Verify that group ownership has been applied correctly.

Checkpoint Question: What is the default owner and group of a directory created by the root user?

Step 4: Set Up Directory Permissions

1. For the **Alpha project** directory:
   - Ensure that **owner** and **group** have full access (read/write/execute)
   - Ensure that **others** have no access
2. For the **Beta reports** directory:
   - Ensure that the **owner** has full access
   - Ensure that the **group** has read and execute permissions, but **not write**
   - Ensure that **others** have no access

Checkpoint Question: Translate these permission needs into octal notation. What should the three-digit permissions be for each directory?

Step 5: Test Access to the Alpha Directory

1. Log in as `dev_junior` and try to:
   - Navigate into `/projects/alpha`
   - Create a file called `dev_file.txt`
   - List the directory contents
2. Log in as `guest_user` and try to:
   - Navigate into `/projects/alpha`

Checkpoint Question: Why is `dev_junior` able to write to the directory? Why is `guest_user` blocked?

Step 6: Test Access to Beta Reports Directory

1. Log in as `tester_junior` and attempt to:
   - Navigate to `/projects/beta_reports`
   - Create a new file
2. As root, create a file called `sample_report.txt` inside the Beta directory with sample text.
3. Log in as `dev_lead` and try to:
   - Read the contents of `sample_report.txt`
   - Delete `sample_report.txt`

Checkpoint Questions:

- Was `tester_junior` able to write to the directory? Why or why not?
- Could `dev_lead` read or delete the file? Explain based on group memberships and permission settings.

Final Reflection

Answer the following questions to summarize what you've learned:

1. What do directory execute (`x`) permissions allow users to do?

2. What are the consequences of setting group permissions without also configuring the correct group ownership?
3. What command would you use to safely assign a user to multiple groups without removing existing memberships?

Extension Challenge (Optional)

Modify the Alpha directory so that team members can write files but **cannot delete each other’s files**.

Submission Format

Submit the file student_id.zip with a markdown file (student_id.md) and its exported PDF file (student_id.pdf) that includes:

- Your command for each task
- Short explanation
- Output snippets (or screenshots if needed)