Lab2: Linux Security Control Simulator

Parishith R.
Index No: 220444K

Functionalities of the Simulator

1. User and Group Management

The simulator provides a complete environment for managing users and groups, similar to a Linux system:

- **User creation and deletion** using `useradd` and `deluser` allows administrators to add new accounts or remove existing ones.
- **Group creation and deletion** using `groupadd` and `delgroup` helps in organizing users into logical units for permission control.
- **Adding users to groups** via `usermod -a -G <group> <user>` enables fine-grained access control by grouping privileges.
- **Login and logout simulation** ensures that commands and file access are tied to the currently active user session.
- All user and group information is **stored persistently** in `users.txt` and `groups.txt` so that the system retains state after restart.

2. Virtual File System (VFS) Operations

The VFS simulates a simplified Linux file system with directory structure and file handling:

- **File creation** (`touch`) and **directory creation** (`mkdir`) replicate basic filesystem operations.
- **Listing contents** with `ls` shows file and folder names, while `ls -l` displays detailed information including permissions, owner, and group.
- **Navigation** through directories using `cd` and displaying the current path with `pwd`.
- **File reading and writing** using `read` and `write` allows storing and retrieving text content.
- **Removal of files and directories** with `rm` (files) and `rm -r` (directories) supports system cleanup.
- **Tree view** (`tree`) provides a hierarchical visual representation of the file system.
- **Persistent storage** via `save` and `load` commands ensures that all files, folders, and metadata are preserved between sessions.

3. Discretionary Access Control (DAC)

The system enforces Linux-like DAC, ensuring that every access request is validated:

- Each file and directory has **owner**, **group**, and **others** permissions in the `rwx` (read, write, execute) model.
- Permissions are stored in **octal format** (e.g., `755` means full access for owner, read+execute for group, read+execute for others).

- Access checks are performed before any file or directory action — for example, a write operation is denied if the user lacks write permission.
- `chmod` allows **changing permission bits**, while `chown` updates the **owner and group** of a file or directory.
- The DAC policy ensures that only authorized users can modify or access files, preventing unauthorized changes.

4. Audit Logging

All significant activities are recorded in an `audit.log` file, mimicking a system security log:

- Every command execution, whether successful or denied, generates a log entry.
- Each log contains:
  - Timestamp of the action
  - User who performed it
  - Command executed
  - Target (file, directory, user, or group)
  - Status (`success`, `denied`, etc.)
- The log provides an **audit trail** for security analysis, helping detect unauthorized attempts or unusual activity.
- Since logs are persistent, administrators can review past activity at any time.