

# **Capstone Project Report**

## **Secure File Storage and Access Management on Linux**

**Report Date:** 20<sup>th</sup> Dec 2025

**Report Time:** 12:49 PM

**Reported By:** Harsh Mohile

**Machine:** Kali-Linux, Oracle VirtualBox

### **Real-time scenario:**

Globex Financial faced a security breach where an unauthorized user accessed confidential financial reports, exposing weak access controls and monitoring gaps. To enhance security, the company is implementing a secure access management system with strict permissions and real-time monitoring. The system will enforce user-specific access controls, allowing only file owners to modify or delete files, while others have read-only access. It will also track command history for auditing and log unauthorized access attempts for IT review. A secure web dashboard will enable real-time security monitoring, and all configurations will persist across reboots to ensure continuous protection and compliance.

### **1. Introduction**

In modern IT environments, secure file storage and controlled access to sensitive data are critical to ensure confidentiality, integrity, and accountability. Linux provides robust mechanisms such as user and group permissions, Access Control Lists (ACLs), auditing, and logging to implement strong access control and monitoring.

### **2. Objective**

The objectives of this project are:

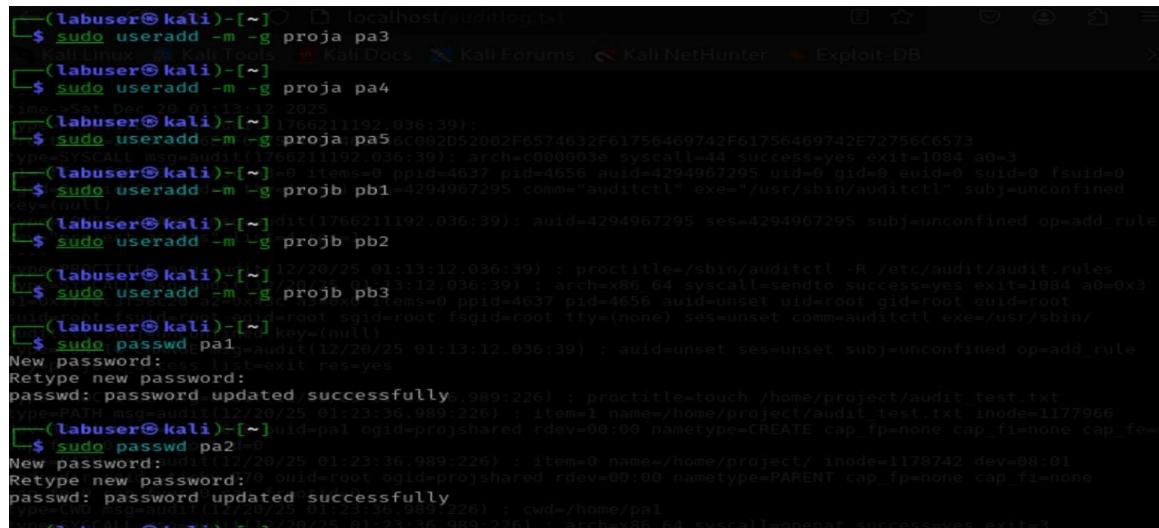
- Create a secure file storage directory on Linux
- Implement role-based access using users and groups
- Enforce explicit permissions and ACLs
- Monitor command execution history
- Log and audit access events
- Provide console and web-based audit reporting

### 3. Tools and Technologies Used

- Kali Linux
- User & Group Management (useradd, groupadd, usermod)
- File Permissions (chmod, chown)
- ACL (setfacl, getfacl)
- auditd
- Apache2
- cron

### 4. User and Group Management

Multiple users and groups were created to simulate project-based access. Users were assigned to respective groups, and a shared group was used for controlled collaboration.



The screenshot shows a terminal session on a Kali Linux desktop. The user is creating multiple users and groups. It starts with creating a user 'projja' in group 'pa3'. Then it creates two more users, 'projb', under two different group IDs (pb1 and pb2). For each user creation, it specifies '-m' for the user ID and '-g' for the group ID. After creating the users, it adds them to a shared group 'projshared' using 'usermod -aG projshared'. Finally, it changes the password for all three users using 'passwd'.

```
(labuser㉿kali)-[~] $ sudo useradd -m -g projja pa3
(labuser㉿kali)-[~] $ sudo useradd -m -g projb pb1
(labuser㉿kali)-[~] $ sudo useradd -m -g projb pb2
(labuser㉿kali)-[~] $ sudo usermod -aG projshared projja
(labuser㉿kali)-[~] $ sudo usermod -aG projshared projb
(labuser㉿kali)-[~] $ sudo usermod -aG projshared pb1
(labuser㉿kali)-[~] $ passwd projja
New password:
Re-type new password:
passwd: password updated successfully.
(labuser㉿kali)-[~] $ passwd projb
New password:
Re-type new password:
passwd: password updated successfully.
(labuser㉿kali)-[~] $ passwd pb1
New password:
Re-type new password:
passwd: password updated successfully.
```

Instead of granting individual users direct ownership or exclusive access to the project directory, a **shared folder access model** was implemented. A dedicated shared group (projshared) was created, and authorized users from different project teams were added to this group. The project directory (/home/project) was assigned group ownership to this shared group, ensuring controlled collaboration while maintaining security.

This approach avoids user-specific dependency, simplifies access management, and follows **best practices of role-based access control (RBAC)**. Additionally, the use of **setgid** ensures that all newly created files inherit the shared group ownership, while the **sticky bit**

prevents unauthorized deletion of files by other users. This design enhances scalability, security, and accountability in multi-user environments.

## 5. Secure Project Directory Configuration

A secure directory (/home/project) was created with restricted permissions, setgid and sticky bit enabled, and ACLs applied to ensure fine-grained access control.

```
(labuser㉿kali)-[~]
$ sudo chmod +t /home/project
[labuser㉿kali)-[~]
$ sudo setfacl -m g:projshared:rwx /home/project
[labuser㉿kali)-[~] 564997463746C002D51002f6574612F61756469742F61756469742E72756C6573
$ sudo setfacl -m g:projshared:rwx /home/project syscalls=44 success=yes exits=1084 so=3
[labuser㉿kali)-[~] 564997463746C002D51002f6574612F61756469742F61756469742E72756C6573
$ sudo setfacl -d -m g:projshared:rwx /home/project syscalls=44 success=yes exits=1084 so=3
[labuser㉿kali)-[~] 564997463746C002D51002f6574612F61756469742F61756469742E72756C6573
$ sudo setfacl -d -m g:projshared:rwx /home/project syscalls=44 success=yes exits=1084 so=3
[labuser㉿kali)-[~] 564997463746C002D51002f6574612F61756469742F61756469742E72756C6573
$ sudo setfacl -d -m g:projshared:rwx /home/project syscalls=44 success=yes exits=1084 so=3
[labuser㉿kali)-[~] 564997463746C002D51002f6574612F61756469742F61756469742E72756C6573
$ sudo setfacl -d -m g:projshared:rwx /home/project syscalls=44 success=yes exits=1084 so=3
[labuser㉿kali)-[~] 564997463746C002D51002f6574612F61756469742F61756469742E72756C6573
$ sudo chsh -s /bin/bash pal
[labuser㉿kali)-[~] 564997463746C002D51002f6574612F61756469742F61756469742E72756C6573
$ sudo chsh -s /bin/bash pa2
[labuser㉿kali)-[~] 564997463746C002D51002f6574612F61756469742F61756469742E72756C6573
$ sudo chsh -s /bin/bash pa3 egid=projshared rdev=00:00 nametype=CREATE cap_fp=none cap_fi=none cap_fe=none
[labuser㉿kali)-[~] 564997463746C002D51002f6574612F61756469742F61756469742E72756C6573
$ sudo chsh -s /bin/bash pa4 egid=projshared rdev=00:00 nametype=PARENT cap_fp=none cap_fi=none cap_fe=none
[labuser㉿kali)-[~] 564997463746C002D51002f6574612F61756469742F61756469742E72756C6573
```

## 6. Command History Monitoring

Command history limits were applied for selected users using HISTSIZE configuration to monitor recent activities.

## 7. Auditing and Logging

auditd was enabled and configured to monitor read, write, execute, and delete operations on the project directory.

```
(labuser㉿kali)-[~]
$ sudo systemctl status auditd
● auditd.service - Security Audit Logging Service
  Loaded: loaded (/usr/lib/systemd/system/auditd.service; enabled; preset→
  Active: active (running) since Sat 2025-12-20 01:06:47 EST; 15s ago
    Invocation: fc1e7871a5944f619e9ab3b2c205e9ec
      Docs: man:auditd(8)
             https://github.com/linux-audit/audit-documentation
   Process: 4569 ExecStart=/usr/sbin/auditd (code=exited, status=0/SUCCESS)
 Main PID: 4570 (auditd)
    Tasks: 2 (limit: 4548)
   Memory: 676K (peak: 1.7M)
     CPU: 22ms
    CGroup: /system.slice/auditd.service
              └─4570 /usr/sbin/auditd

Dec 20 01:06:47 kali systemd[1]: Starting auditd.service - Security Audit Lo→
Dec 20 01:06:47 kali auditd[4570]: No plugins found, not dispatching events
Dec 20 01:06:47 kali auditd[4570]: Init complete, auditd 4.1.2 listening for→
Dec 20 01:06:47 kali systemd[1]: Started auditd.service - Security Audit Log→
lines 1-18/18 (END)
```

## 8. Access Monitoring

All file operations such as create, read, modify, and delete were successfully logged with user attribution.

## 9. Console-Based Audit Reporting

Audit logs were extracted using ausearch and stored in a centralized log file.

```
File Actions Edit View Help
~ ~ trash
~ ~
~ ~
~ ~
~ ~
~ ~ File System
└── (labuser㉿kali)-[~]
$ sudo nano /etc/audit/rules.d/audit.rules
└── (labuser㉿kali)-[~]
$ sudo systemctl restart auditd
└── (labuser㉿kali)-[~]
$ sudo ausearch -k project_access
time→Sat Dec 20 01:13:12 2005
type=PROCTITLE msg=audit(1766211192.036:39): proctitle=2F7362696E2F617564697463746C002D52002F657
4632F61756469742F61756469742E72756C6573
type=SYSCALL msg=audit(1766211192.036:39): arch=c000003e syscall=44 success=yes exit=1084 a0=3 a1=7ffec3f58c20 a2=43c a3=0 items=0 ppid=4637 pid=4656 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="auditctl" exe="/usr/sbin/auditctl"
subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1766211192.036:39): auid=4294967295 ses=4294967295 subj=unconfined op=add_rule key="project_access" list=4 res=1
└── (labuser㉿kali)-[~]
$
```

## 10. Web-Based Audit Reporting

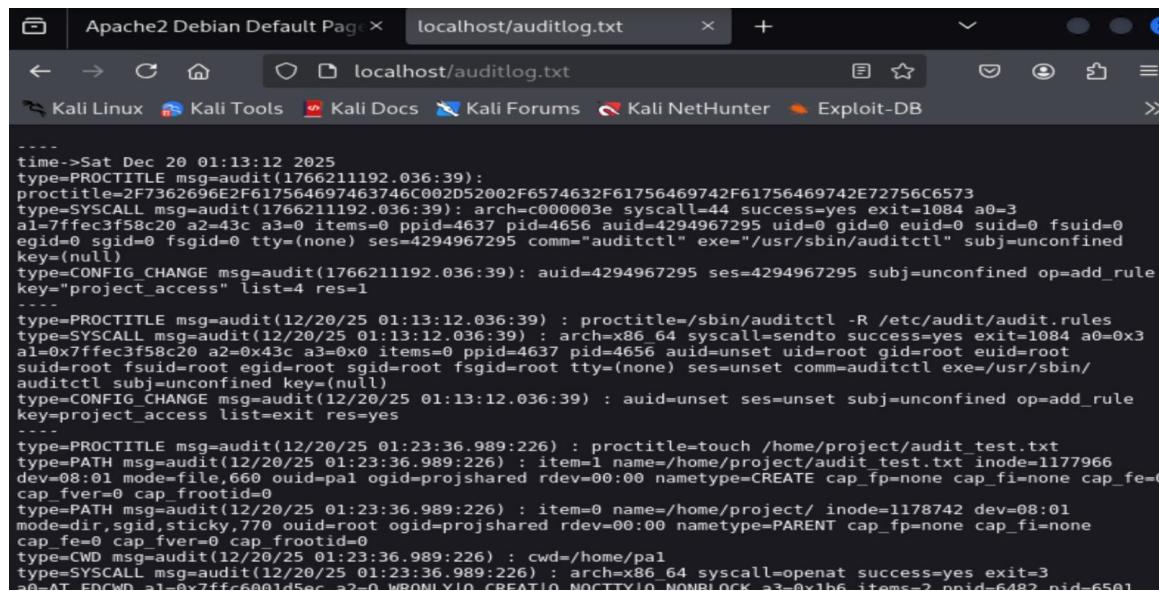
```
-----  
time->Sat Dec 20 01:13:12 2025  
type=PROCTITLE msg=audit(1766211192.036:39):  
proctitle=2F7362696E2F617564697463746C002D52002F6574632F61756469742F61756469742E72756C6573  
type=SYSCALL msg=audit(1766211192.036:39): arch=c000003e syscall=44 success=yes exit=1084 a0=3  
a1=7ffec3f58c20 a2=43c a3=0 items=0 ppid=4637 pid=4656 auid=4294967295 uid=0 gid=0 euid=0 suid=0  
egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="auditctl" exe="/usr/sbin/auditctl" subj=unconfined  
key=(null)  
type=CONFIG_CHANGE msg=audit(1766211192.036:39): auid=4294967295 ses=4294967295 subj=unconfined op=add_rule  
key="project_access" list=4 res=1  
-----  
type=PROCTITLE msg=audit(12/20/25 01:13:12.036:39) : proctitle=/sbin/auditctl -R /etc/audit/audit.rules  
type=SYSCALL msg=audit(12/20/25 01:13:12.036:39) : arch=x86_64 syscall=sendto success=yes exit=1084 a0=0x3  
a1=0x7ffec3f58c20 a2=0x43c a3=0x0 items=0 ppid=4637 pid=4656 auid=unset uid=root gid=root euid=root  
suid=root fsuid=root egid=0 sgid=0 fsgid=0 tty=(none) ses=unset comm="auditctl" exe="/usr/sbin/  
auditctl" subj=unconfined key=(null)  
type=CONFIG_CHANGE msg=audit(12/20/25 01:13:12.036:39) : auid=unset ses=unset subj=unconfined op=add_rule  
key=project_access list=exit res=yes  
-----  
type=PROCTITLE msg=audit(12/20/25 01:23:36.989:226) : proctitle=touch /home/project/audit_test.txt  
type=PATH msg=audit(12/20/25 01:23:36.989:226) : item=1 name=/home/project/audit_test.txt inode=1177966  
dev=08:01 mode=file,660 ouid=0pal ogid=projshared rdev=00:00 nametype=CREATE cap_fp=none cap_fi=none cap_fe=  
cap_fver=0 cap_frootid=0  
type=PATH msg=audit(12/20/25 01:23:36.989:226) : item=0 name=/home/project/ inode=1178742 dev=08:01  
mode=dir,sgid,sticky,770 ouid=root ogid=projshared rdev=00:00 nametype=PARENT cap_fp=none cap_fi=none  
cap_fe=0 cap_fver=0 cap_frootid=0  
type=CWD msg=audit(12/20/25 01:23:36.989:226) : cwd=/home/pal  
type=SYSCALL msg=audit(12/20/25 01:23:36.989:226) : arch=x86_64 syscall=openat success=yes exit=3  
a0=AT_FDCWD a1=0x7ffec6001d5e a2=0x4001000000000000 a3=0x1b6 items=2 ppid=6482 pid=6501
```

Apache2 was used to serve audit logs via browser access at <http://localhost/auditlog.txt>.

## 11. Automation

A cron job was configured to periodically update audit logs automatically.

## Final Output



## 12. Conclusion

This project demonstrates a secure and auditable Linux file storage system suitable for enterprise environments.