Linux IAM & Hardening Project Report

Student Name: Mohammad Kaif

Date:04-11-2025

Lab Environment: Kali Linux VM

Table of Contents

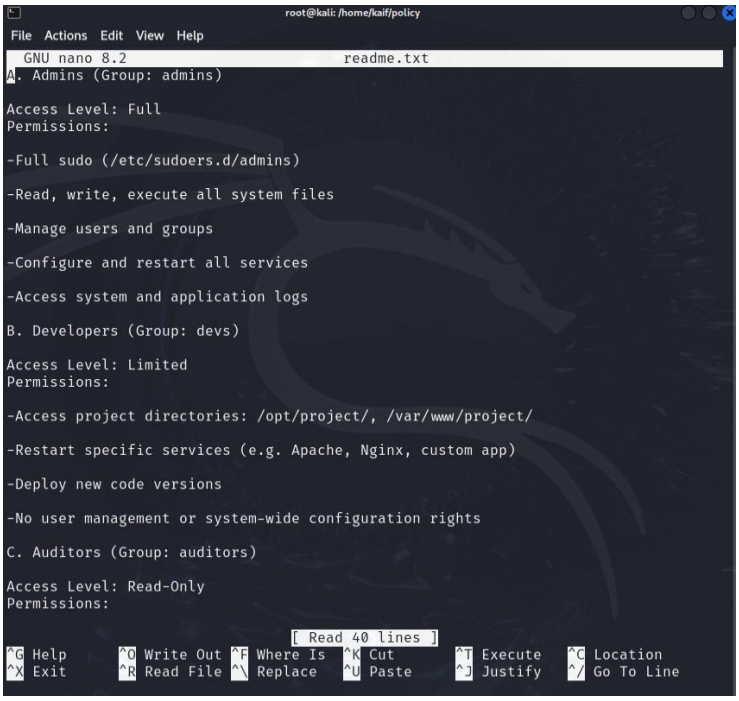
---

1. Baseline Policy Document

Define user roles, sudo needs, file access levels.

- **Admin**
  - Username: alice
  - Full sudo rights
- **Developer**
  - Username: bob
  - Limited sudo (only apt update, systemctl restart apache2)
- **Auditor**
  - Username: carol

- No sudo; read-only access to /var/log and project file



---

2. User and Group Creation

Commands Used

sudo groupadd devs

sudo groupadd auditors

sudo useradd -m -s /bin/bash alice

sudo usermod -aG sudo alice

sudo useradd -m -s /bin/bash bob

sudo usermod -aG devs bob

sudo useradd -m -s /bin/bash carol

sudo usermod -aG auditors carol

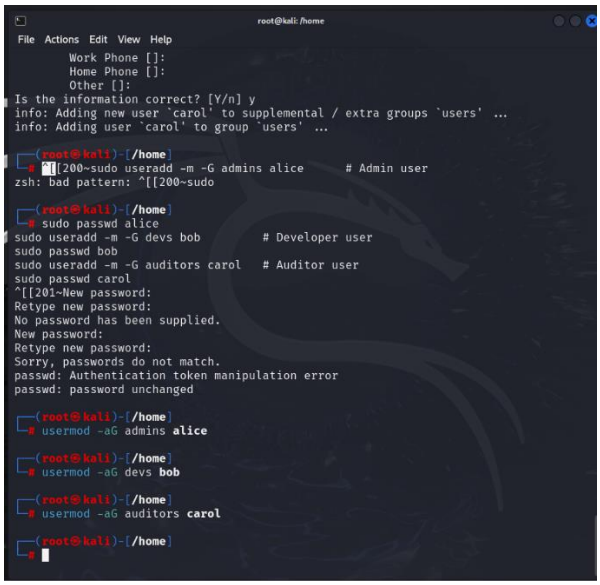
✅ **Verification**

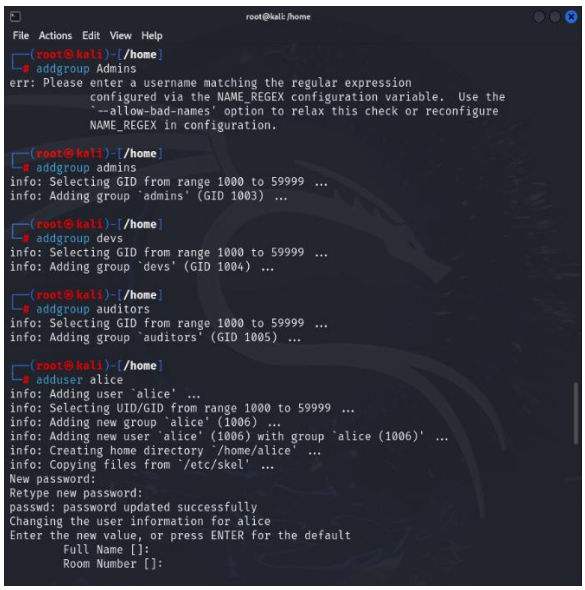
id alice

id bob

id carol

📸 *Screenshot: Terminal output showing group memberships.*





---

3. Sudoers Configuration

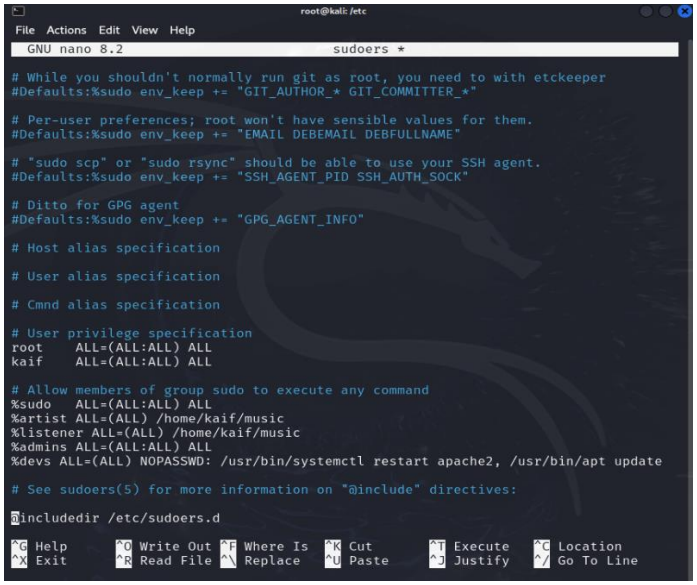
Files Edited

- /etc/sudoers.d/devs

%devs ALL=(ALL) NOPASSWD: /usr/bin/systemctl restart apache2, /usr/bin/apt update

- /etc/sudoers.d/admins *(optional if using sudo group)*

📸 *Screenshot: Content of sudoers files*



✅ sudo -l output for each user to confirm their permissions.

---

4. Shared Project Directory with ACLs

Setup Commands

sudo mkdir /policy/project

sudo chown root:devs /policy/project

sudo chmod 2775 /policy/project

sudo setfacl -m g:devs:rwx /policy/project

sudo setfacl -m o:rx /policy/project

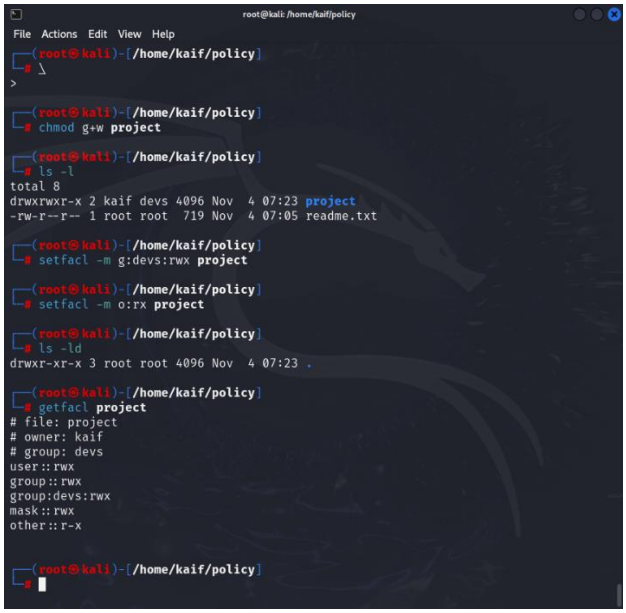
sudo setfacl -d -m g:devs:rwx /policy/project

sudo setfacl -d -m o:rx /policy/project

✅ **Verification**

getfacl /policy/project

📸 *Screenshot: getfacl and ls -ld /policy/project output.*



---

5. Audit Configuration

Installation & Service

sudo apt install auditd

sudo systemctl enable --now auditd

Rules Added

sudo auditctl -w /etc/passwd -p wa -k user-modify

sudo auditctl -w /etc/sudoers -p wa -k sudoers-mod

📸 *Screenshot: Output of sudo auditctl -l*

---

6. Vulnerable Snapshot: Misconfigurations Found and Fixed

🔧 Issue 1: World-Writable /etc/cron.d

- **Before:**

ls -ld /etc/cron.d

(e.g. drwxrwxrwx)

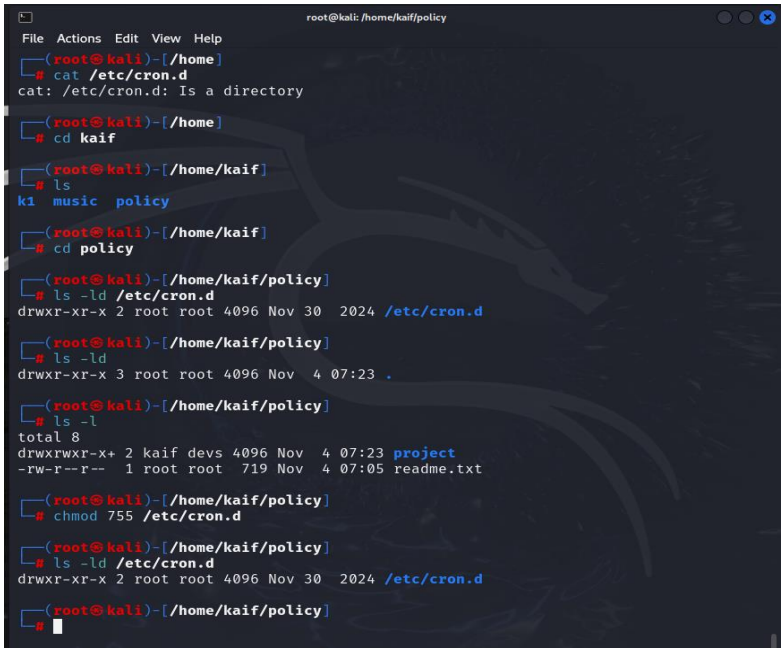
- **Fix:**

sudo chmod 755 /etc/cron.d

- **After:**

ls -ld /etc/cron.d

📸 *Screenshots before & after*



🔗 Issue 2: Unrestricted NOPASSWD in sudoers

- **Before:**

sudo grep -R NOPASSWD /etc/sudoers*

(e.g. user1 ALL=(ALL) NOPASSWD: ALL)

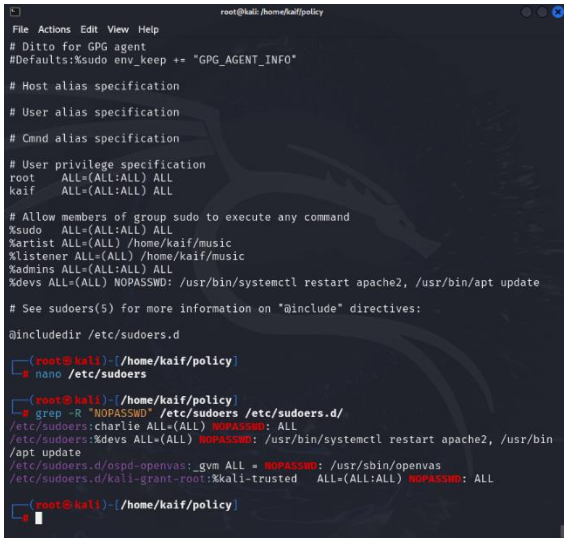
- **Fix:**
  Edit file via sudo visudo or sudo nano /etc/sudoers.d/user1, replace with:

user1 ALL=(ALL) /usr/bin/apt

- **After:**
  Confirm with:

sudo -l -U user1

📸 *Screenshots before & after*



🔗 Issue 3: Weak Permissions on /etc/shadow

- **Before:**

ls -l /etc/shadow

(e.g. -rw-r--r--)

- **Fix:**

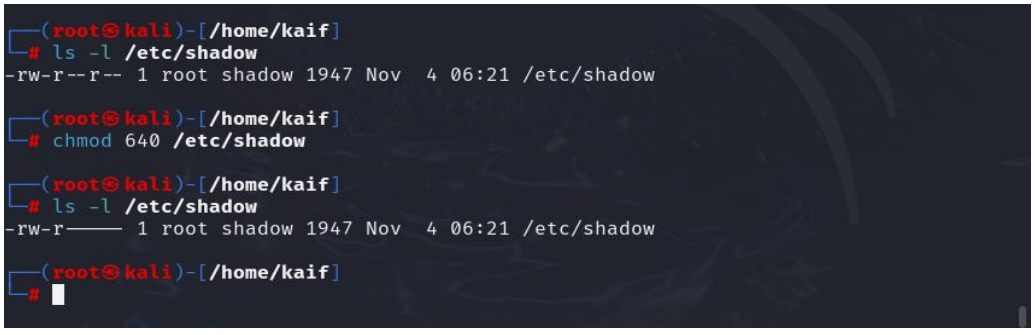
sudo chown root:shadow /etc/shadow

sudo chmod 640 /etc/shadow

- **After:**

ls -l /etc/shadow

📸 *Screenshots before & after*



---

7. Remediation Checklist ✅

Checkpoint	Status
All users in correct groups	✅
Sudo access limited via /etc/sudoers.d/	✅
/etc/sudoers protected and audited	✅
/policy/project writeable only by devs	✅
Audit rules log changes to passwd/sudoers	✅
Cron directories not world-writable	✅
No NOPASSWD:ALL in sudoers	✅
Sensitive files (e.g. /etc/shadow) secured	✅

---

8. Audit Logs and Final State Summary

Audit Log Examples

sudo ausearch -k user-modify

sudo ausearch -k sudoers-mod

📸 *ausearch logs showing modifications*

Final System State

Users created with appropriate permissions

- Project directory ACL verified
- Audit logs enabled and working

- All found vulnerabilities remediated

Major Learnings

Through this project, I gained practical insights into **Linux Identity and Access Management (IAM)** and **system hardening** practices that form the foundation of cybersecurity and ethical hacking. Key learnings include:

1. **Role-Based Access Control (RBAC):**
   I learned how to design and enforce a structured access hierarchy using groups such as admins, devs, and auditors — ensuring each role has only the privileges needed to perform their tasks.
2. **Principle of Least Privilege:**
   Configuring limited sudo permissions (via /etc/sudoers.d/) helped me understand how privilege minimization can significantly reduce attack surfaces and prevent privilege escalation.
3. **Secure File Access Management:**
   Implementing **POSIX permissions** and **Access Control Lists (ACLs)** for shared directories (/policy/project) showed how to control read/write access granularly across teams.
4. **System Auditing and Monitoring:**
   Setting up **auditd** rules provided hands-on experience in tracking changes to critical files like /etc/passwd and /etc/sudoers, strengthening accountability and traceability in system operations.
5. **Vulnerability Identification & Remediation:**
   Discovering real-world misconfigurations (e.g., world-writable cron directories, weak permissions on /etc/shadow, and unrestricted NOPASSWD rules) reinforced how small oversights can lead to significant security risks — and how to remediate them properly.
6. **Hardening Mindset:**
   The project cultivated a security-oriented mindset, emphasizing continuous verification (sudo -l, getfacl, ausearch) and post-fix validation to maintain a hardened environment.