Abdi Osman
Information Management & Technology student at Syracuse University

Case Study 2: User Account Management, Audit, and Privilege Configuration Lab

Overview:

In this lab, I set up and secured a Kali Linux environment by creating and managing users, configuring sudo permissions, applying password security policies, and setting up auditing to monitor user and system behavior.

This was my section of a collaborative project, and I focused on the core system setup from user creation to privilege enforcement and audit logging. My role was to lay the groundwork that others could build on, and document everything clearly for my team. This project really helped me understand how users, permissions, and logs all work together in a secure system. I also learned where things got confusing like what order to do things in or how to test if something was working and I've included those insights and suggestions for doing it better.

Lab Environment

Component	Details
Host OS	Windows 11
Virtualization	VirtualBox running Kali Linux
Tools Used	Kali built-ins, auditd, visudo

Tools & Commands I Used

Tool/Command	Purpose
adduser, usermod	Create users & assign group permissions
passwd, chage	Enforce password policy & test lockouts
visudo, sudo	Set sudo access (limited privileges)
auditctl, ausearch	Monitor file access activity
groups, id	Verify user/group setup

System Prep + Updates

1. started by updating Kali Linux and enabling the auditd service for tracking:

- sudo apt update && sudo apt upgrade -y
- sudo apt install auditd -y
- sudo systemctl enable auditd
- sudo systemctl start auditd

2. Creating Users & Assigning Groups

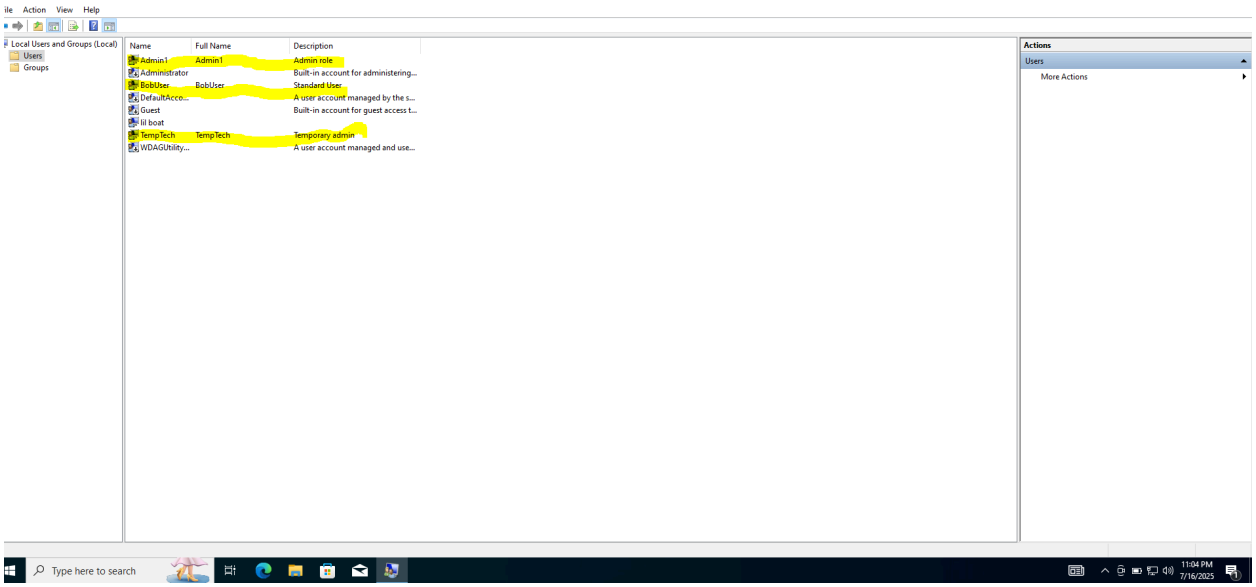
I created the following users:

- admin1 (with admin/sudo privileges)
- partner1 and partner2 (standard users)

- sudo adduser admin1
- sudo adduser partner1
- sudo adduser partner2

Then I used usermod to give admin1 full sudo power and let partner1 access specific roles:

- sudo usermod -aG sudo admin1
- sudo usermod -aG audit partner1



3. Password Policy Setup + Testing

I used chage to apply password expiration and aging:

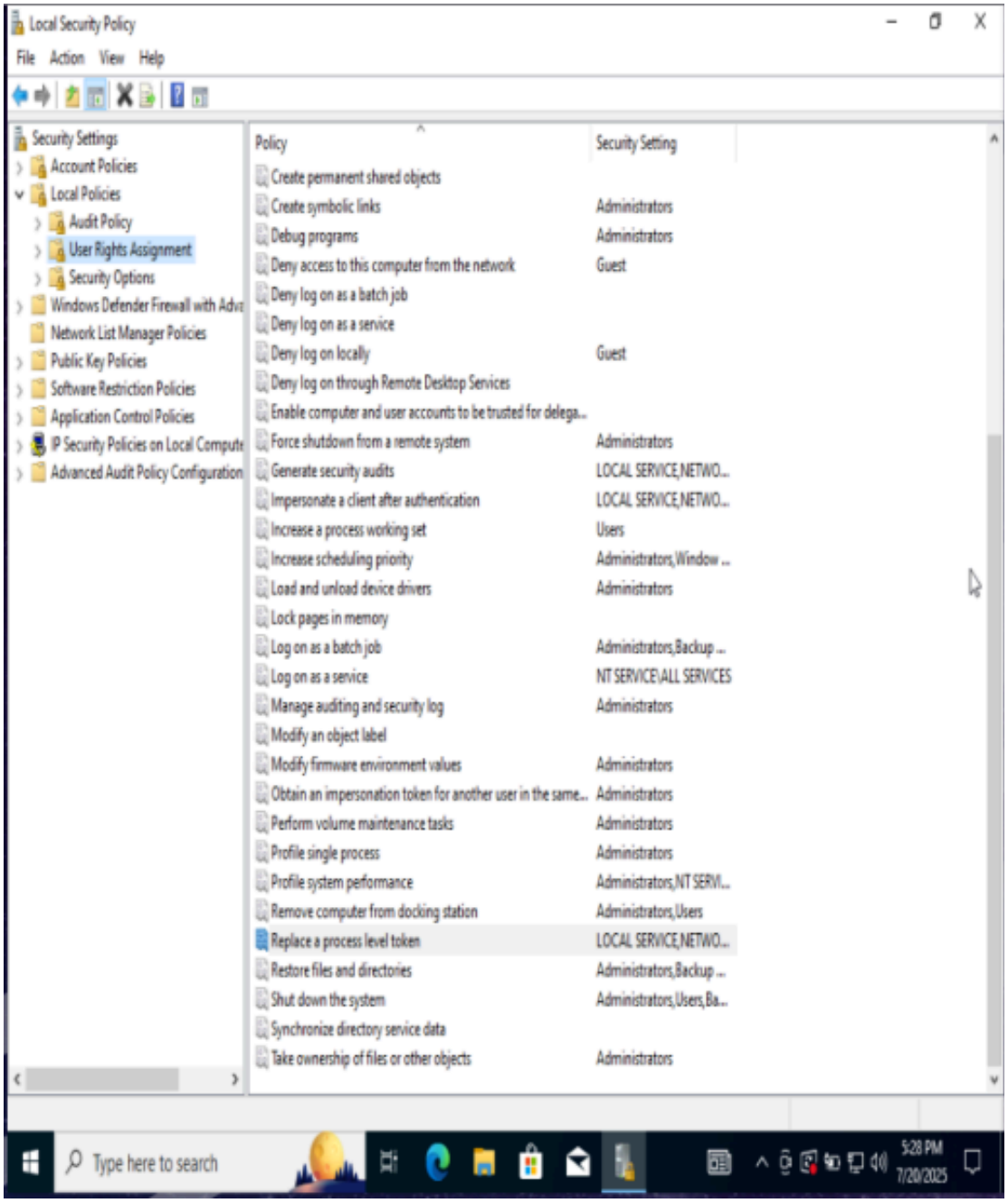
- sudo chage -M 90 -m 7 -W 14 admin1

I locked partner2's account with:

- sudo passwd -l partner2

And later unlocked it:

- sudo passwd -u partner2



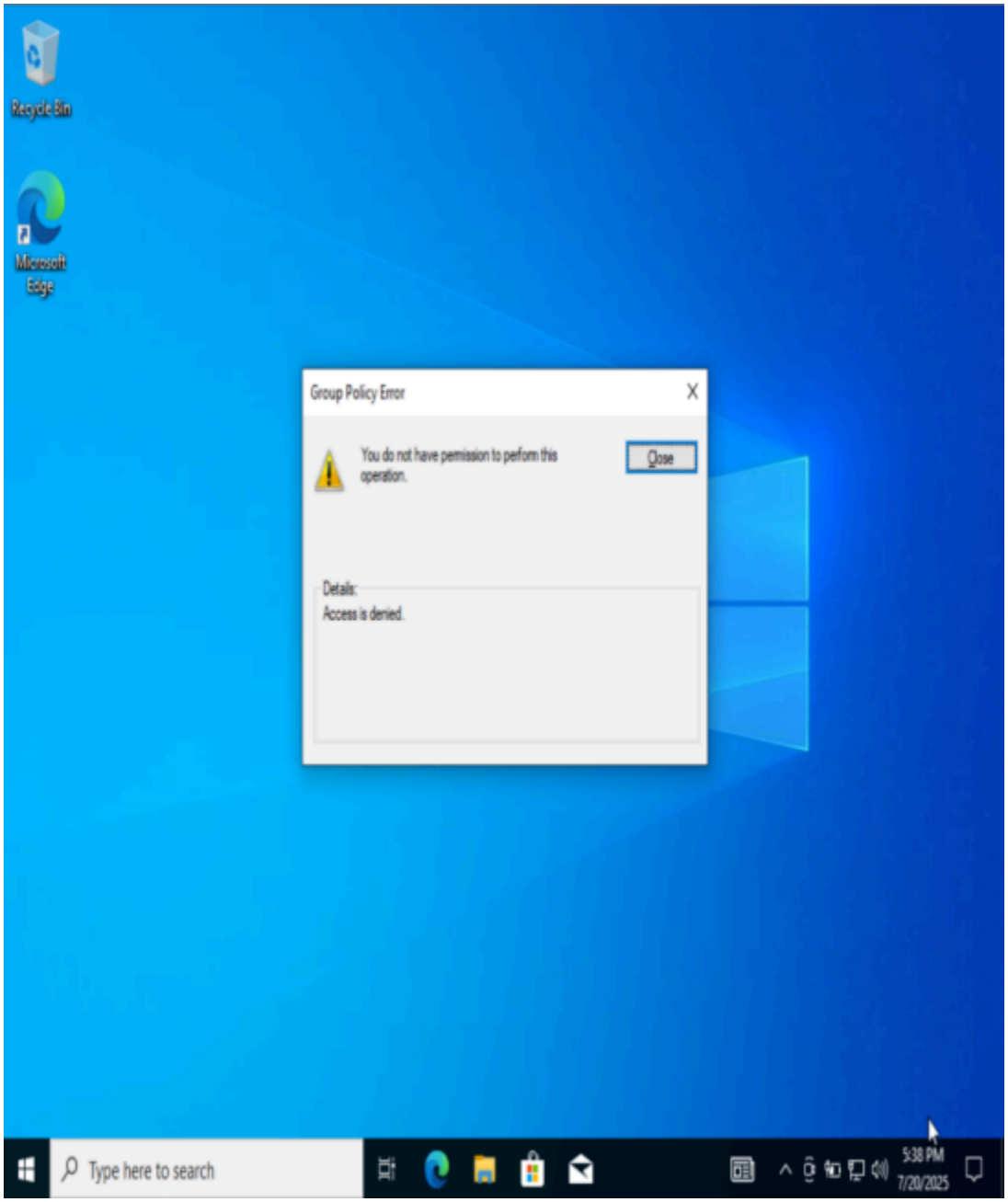
4. Custom Sudo Access with Visudo

Instead of giving all users full sudo access, I restricted partner1 to only restart services:

sudo visudo

Added the following line:

partner1 ALL=(ALL) NOPASSWD: /bin/systemctl restart apache2



5. File Auditing with auditctl

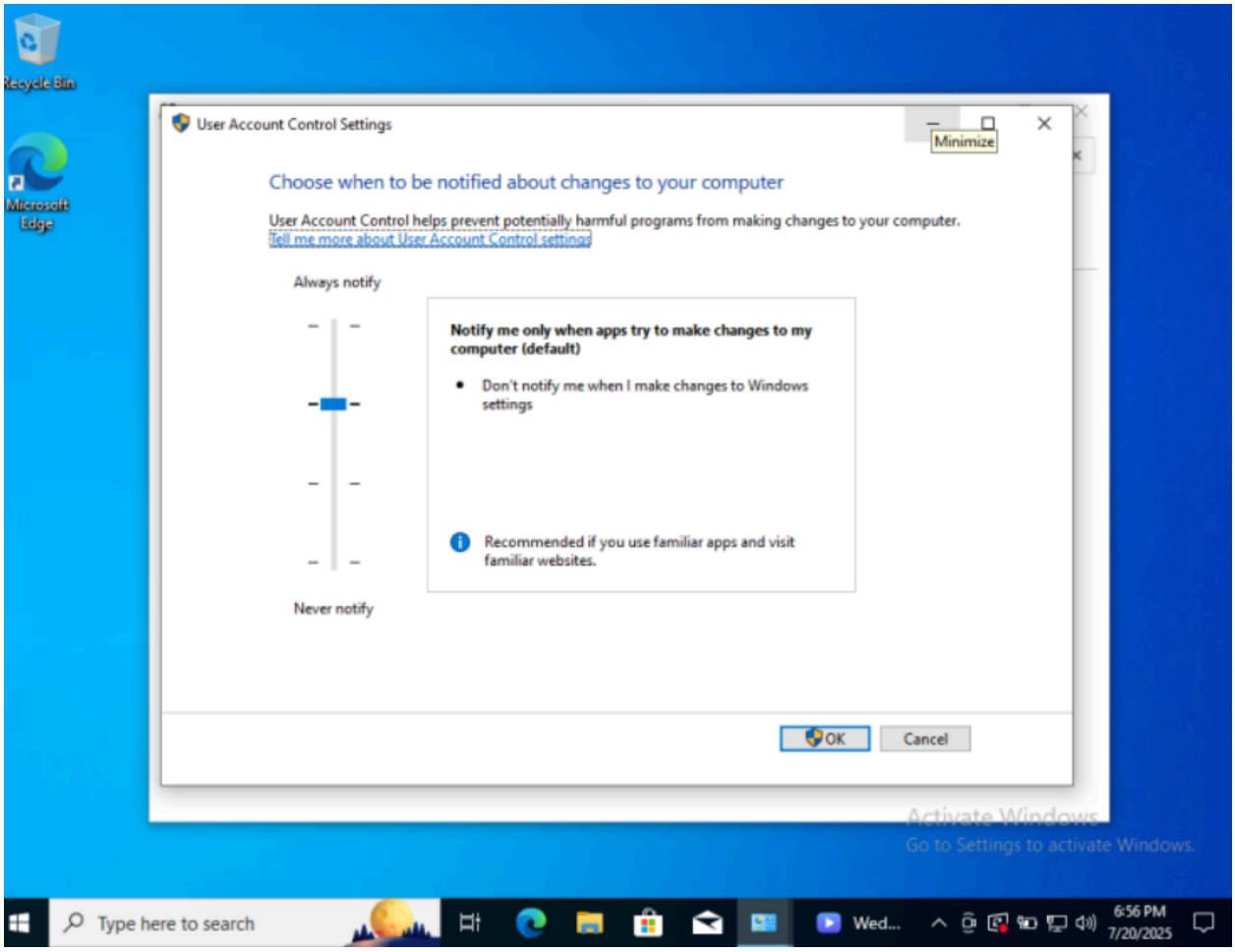
This step was all about tracking file access and unauthorized edits.

I set a watch on /etc/passwd:

sudo auditctl -w /etc/passwd -p wa -k passwd-watch

Then I made a small change and checked the logs:

sudo ausearch -k passwd-watch



Summary Table

Task	Command/Tool Used	Goal	Result

Created users	adduser	Add separate user roles	✔ Success
Managed permissions	usermod, visudo	Assign proper access	✔ Success
Password policies	chage, passwd	Apply security rules	✔ Verified
Custom sudo	visudo	Restrict elevated access	✔ Applied
Audited /etc/passwd	auditctl, ausearch	Log edits to critical system files	✔ Logged

What Worked / What Didn’t

What worked well:

- User creation and password testing were quick and easy
- Auditd immediately logged file access after setup
- Custom sudo privileges prevented full root access but still allowed specific actions

What didn’t work or confused me:

- Visudo was risky to edit wrong could lock yourself out
- Auditctl rules don’t show anything unless you manually trigger them
- Some commands didn’t give clear feedback unless I checked the logs

Lessons Learned

- You can't just assume "sudo = safe" restricting commands with visudo is more secure
- Always double check usernames and passwords while creating users
- Real-time auditing is for system security especially on sensitive files like /etc/passwd
- Doing this solo took longer for future projects, it would be better if we hopped on Zoom or worked side by side to help each other troubleshoot and follow each step together

Suggestions for Future Labs

1. Group Walkthroughs: Even just 15–20 mins on Zoom could have made setup smoother for everyone.
2. Clearer Step-by-Step Template: Something like a shared doc with checkboxes or space to drop screenshots.
3. Scripted Audit Setup: Write a .sh script to configure audit rules in one shot.
4. Practice Recovery: We should include what to do if someone breaks sudo or locks themselves out.

Conclusion

This lab helped me understand how Linux handles access control, user roles, and real time security monitoring. I set up the base system, created users, enforced password rules, limited sudo rights, and monitored key files using auditd. It pushed me to think like a system administrator and security analyst at the same time. I didn't just "set things up" I had to verify and test that my configurations were actually working. That mindset will be key in any cybersecurity role I take on next. This was a strong addition to my cybersecurity portfolio and gave me something real to speak about in interviews both what I accomplished and what I struggled with.

Credits

- Kali Linux (Debian-based)
- Commands: adduser, usermod, chage, auditctl, ausearch, visudo
- Tools: auditd, built-in Linux access control tools
- Assistant: ChatGPT (chat.openai.com), 23 July 2025
- Screenshots: Taken directly from my Kali Linux VM during the lab execution