Project Title: Linux IAM & Hardening — User, Group & Permission Management

📄 Linux IAM & Hardening Mini 📄

🔹 Objective

The objective of this project is to design and implement a secure user, group, and permission model on an Ubuntu system.

Additionally, we identify and fix three security misconfigurations related to Identity and Access Management (IAM) and file permissions.

🔹 Tools & Environment

Tool / Component	Description
Ubuntu 22.04 VM	Target (lab) system
Kali Linux	Attacker/test system
sudo, useradd, visudo, chmod	Used for configuration
auditd	For tracking system changes
VirtualBox / VMware	Virtualization environment

TASK: ***Create a baseline policy document: who needs sudo, group roles, and file access requirements for a small team (e.g., admin, dev, auditor).***

🧩 Baseline Policy

Objective:

Design a secure Identity and Access Management (IAM) setup on Ubuntu to ensure least privilege, proper auditing, and controlled file access.

Team Roles & Access Policy:

Role	Group Name	Purpose/ Responsibility	Access/Privileges
Admin	admin	Manage sustem,users,and critical services	Limited sudo access for user management, service control, and package installs (no NOPASSWD)
Developer	dev	Work on project files and app restarts	Can restart app service only (systemctl restart myapp.service), write access to project folder
Auditor	auditor	Review configurations and logs	Read-only access, no sudo permissions
Project Team	project-team	Common collaboration group	Write access to shared folder /srv/project, others read-only

Sudo Policy:

- Only members of admin group get sudo privileges.
- Allowed commands (via visudo):/usr/sbin/useradd, /usr/sbin/usermod, /usr/sbin/userdel, /usr/bin/apt-get, /bin/systemctl restart myapp.service.
- Developers (dev) allowed only systemctl restart myapp.service.

- Auditors: No sudo permissions.
- NOPASSWD entries are not allowed.

File & Folder Policy:

Folder	Ower:Group	Permissions	Notes
/srv/project	root:project-team	2775(rwxrwxr-x)	Setgid bit ensures files inherit project-team group
/etc/sudoers, /etc/passwd	root:root	rw-r--r--	Monitored by auditd
/home/*	user:user	700	Private home directories

Auditing Policy

- auditd enabled and running.
- Files monitored: /etc/sudoers, /etc/passwd
- Rules added to /etc/audit/rules.d/local.rules:
- [ ] -w /etc/sudoers -p wa -k sudoers_changes
- [ ] -w /etc/passwd -p wa -k passwd_changes
- Logs reviewed using ausearch -k sudoers_changes and aureport -f.

Password & Account Policy

- Passwords must be at least 8 chars, with uppercase, lowercase, number, and special symbol.
- Root SSH login disabled (/etc/ssh/sshd_config → PermitRootLogin no).
- Inactive accounts reviewed every 30 days.

🧩 Slide Remediation Checklist & Summary

🔧 Common Issues Found & Fixes

Issue	Risk	Fix Applied
World-writable file in /etc/cron.d/	Could allow privilege escalation	Set correct permission: chmod o-w /etc/cron.d/somefile
NOPASSWD: ALL in sudoers	Allowed password-less root access	Edited /etc/sudoers → restricted commands only
/etc/shadow readable by others	Password hashes exposed	Fixed with chmod 640 /etc/shadow

✅ **Ongoing Security Practices**

Use visudo to edit sudoers safely.

Enable auditd to log config changes.

Review find / -perm -o+w monthly for world-writable files.

Regularly backup /etc/sudoers, /etc/passwd, and /srv/project.

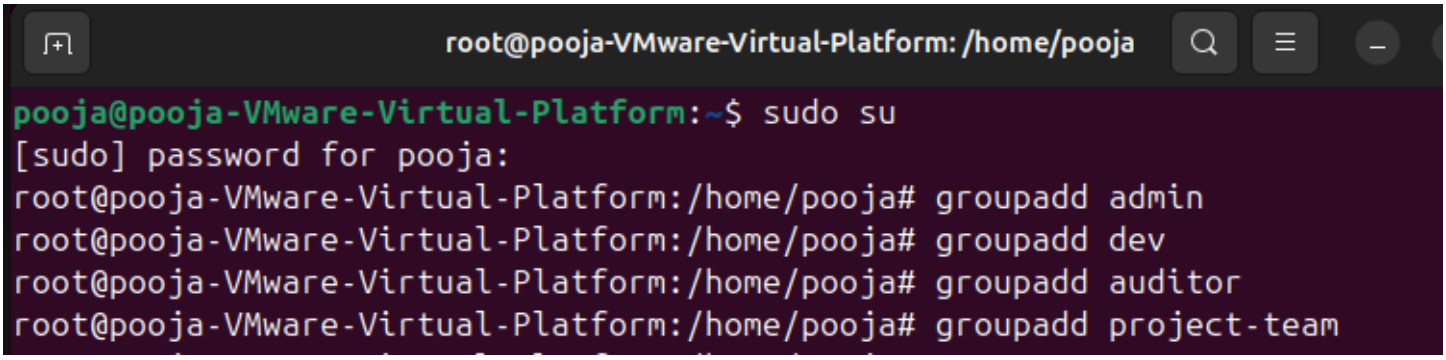
Remove inactive or test accounts monthly.

Enforce password expiration every 90 days.

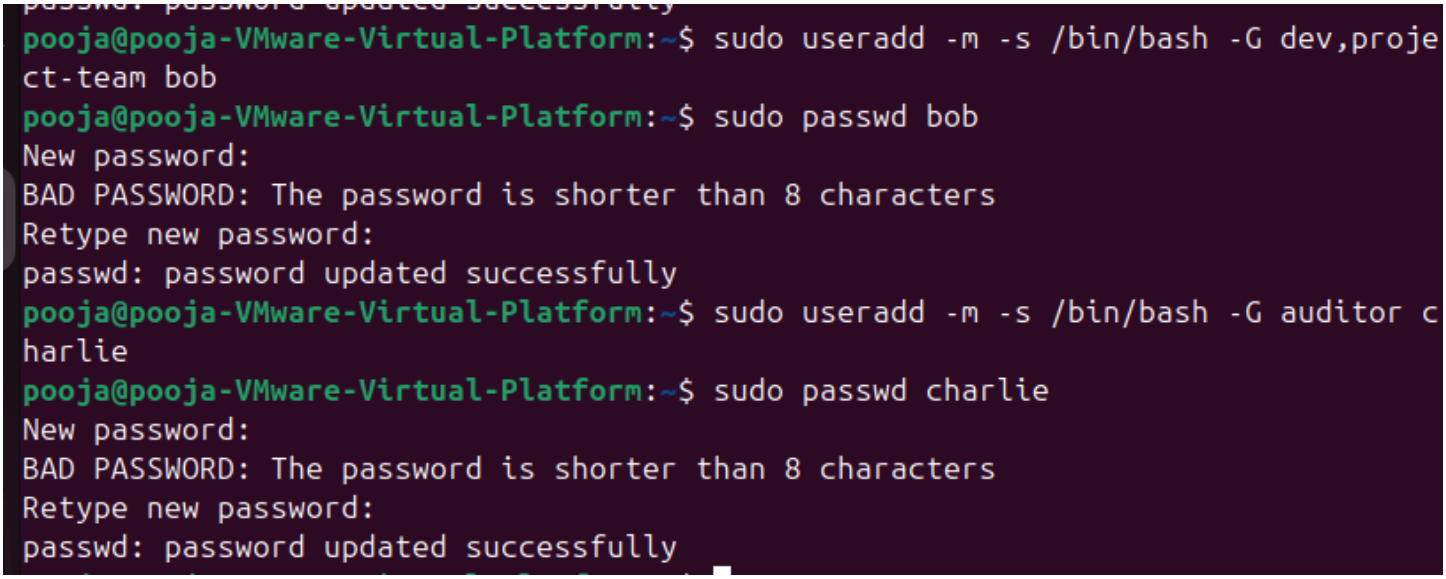
TASK : ***On an Ubuntu VM, create users and groups according to the policy (use useradd, groupadd).***

On an Ubuntu VM, we use useradd to create new users and groupadd to create groups as per the security policy. This helps organize users by roles and set proper permissions for each group, improving system management and security.

CREATE GROUPS



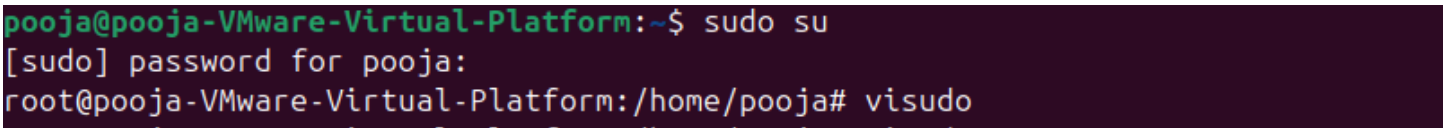
CREATE USERS, SET HOME, ADD TO GROUPS



TASK: ***Assign minimal privileges: configure sudoers using any text editor with least-privilege rules, set sudo rules for specific commands only.***

Configure the sudoers file to give users only the specific commands they need using visudo. This ensures minimal privileges, preventing full root access and improving system security.

CONFIGURE SUDOERS(LEAST PRIVILEGE)



```
  GNU nano 7.2                        /etc/sudoers.tmp
# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
#%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:
%admin ALL=(ALL) /usr/sbin/useradd, /usr/sbin/userdel, /usr/sbin/usermod, /bin/>

%dev ALL=(root) /bin/systemctl restart myapp.service
@includedir /etc/sudoers.d

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

TASK: ***Use POSIX permissions + ACLs for a shared project folder so that only the intended group has write access; others have read-only.***

Set **POSIX permissions** and **ACLs** on the shared folder so only the specific group can write, while others have read-only access. This controls data modification and maintains security in shared environments.

```
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo mkdir -p /srv/project
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo chown root:project-team /sr
v/project
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo chmod 2775 /srv/project
root@pooja-VMware-Virtual-Platform:/home/pooja# ls -ld /srv/project
drwxrwsr-x 2 root project-team 4096 Nov  5 10:43 /srv/project
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo setfacl -R -m g:project-tea
m:rwx /srv/project
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo setfacl -R -m o::r-x /srv/p
roject
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo setfacl -d -m g:project-tea
m:rwx /srv/project
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo setfacl -d -m o::r-x /srv/p
roject
```

CHECK

```
root@pooja-VMware-Virtual-Platform:/home/pooja# getfacl /srv/project
getfacl: Removing leading '/' from absolute path names
# file: srv/project
# owner: root
# group: project-team
# flags: -s-
user::rwx
group::rwx
group:project-team:rwx
mask::rwx
other::r-x
default:user::rwx
default:group::rwx
default:group:project-team:rwx
default:mask::rwx
default:other::r-x
```

TASK:*Enable simple auditing (auditd) to log changes to /etc/sudoers and /etc/passwd.*

Enable **auditd** to monitor and log any changes made to critical files like /etc/sudoers and /etc/passwd, helping track unauthorized modifications and enhance system security.

ENABLE SIMPLE AUDITING(AUDITD) TO WATCH /etc/passwd

sudo apt update

sudo apt install -y auditd audispd-plugins

```
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo bash -c 'echo "-w /etc/sudoers -p wa -k sudoers_changes" >> /etc/audit/rules.d/local.rules'
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo bash -c 'echo "-w /etc/sudoers -p wa -k passwd_changes" >> /etc/audit/rules.d/local.rules'
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo systemctl restart auditd
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo auditctl -l
-w /etc/sudoers -p wa -k sudoers_changes
-w /etc/sudoers -p wa -k passwd_changes
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo ausearch -k sudoers_changes --start recent
----
time->Wed Nov  5 11:21:28 2025
type=PROCTITLE msg=audit(1762321888.483:162): proctitle=2F7362696E2F617564697463746C002D52002F6574632F61756469742F61756469742E72756C6573
type=SYSCALL msg=audit(1762321888.483:162): arch=c000003e syscall=44 success=yes exit=1084 a0=3 a1=7fff81989580 a2=43c a3=0 items=0 ppid=5033 pid=5048 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(1762321888.483:162): auid=4294967295 ses=4294967295 subj=unconfined op=add_rule key="sudoers_changes" list=4 res=1
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo ausearch -k passwd_changes --start today
----
time->Wed Nov  5 11:21:28 2025
type=PROCTITLE msg=audit(1762321888.483:163): proctitle=2F7362696E2F617564697463746C002D52002F6574632F61756469742F61756469742E72756C6573
type=PATH msg=audit(1762321888.483:163): item=0 name="/etc/" inode=1835009 dev=08:02 mode=040755 ouid=0 ogid=0 rdev=00:00 nametype=PARENT cap_fp=0 cap_fi=0 cap_fe=0 cap_f
er=0 cap_frootid=0
type=CWD msg=audit(1762321888.483:163): cwd="/"
type=SOCKADDR msg=audit(1762321888.483:163): saddr=100000000000000000000000
type=SYSCALL msg=audit(1762321888.483:163): arch=c000003e syscall=44 success=yes exit=1084 a0=3 a1=7fff81989580 a2=43c a3=0 items=1 ppid=5033 pid=5048 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(1762321888.483:163): auid=4294967295 ses=4294967295 subj=unconfined op=add_rule key="passwd_changes" list=4 res=1
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo aureport -f

File Report
===============================================
# date time file syscall success exe auid event
===============================================
1. 11/05/2025 11:21:28 /etc/ 44 yes /usr/sbin/auditctl -1 163
```

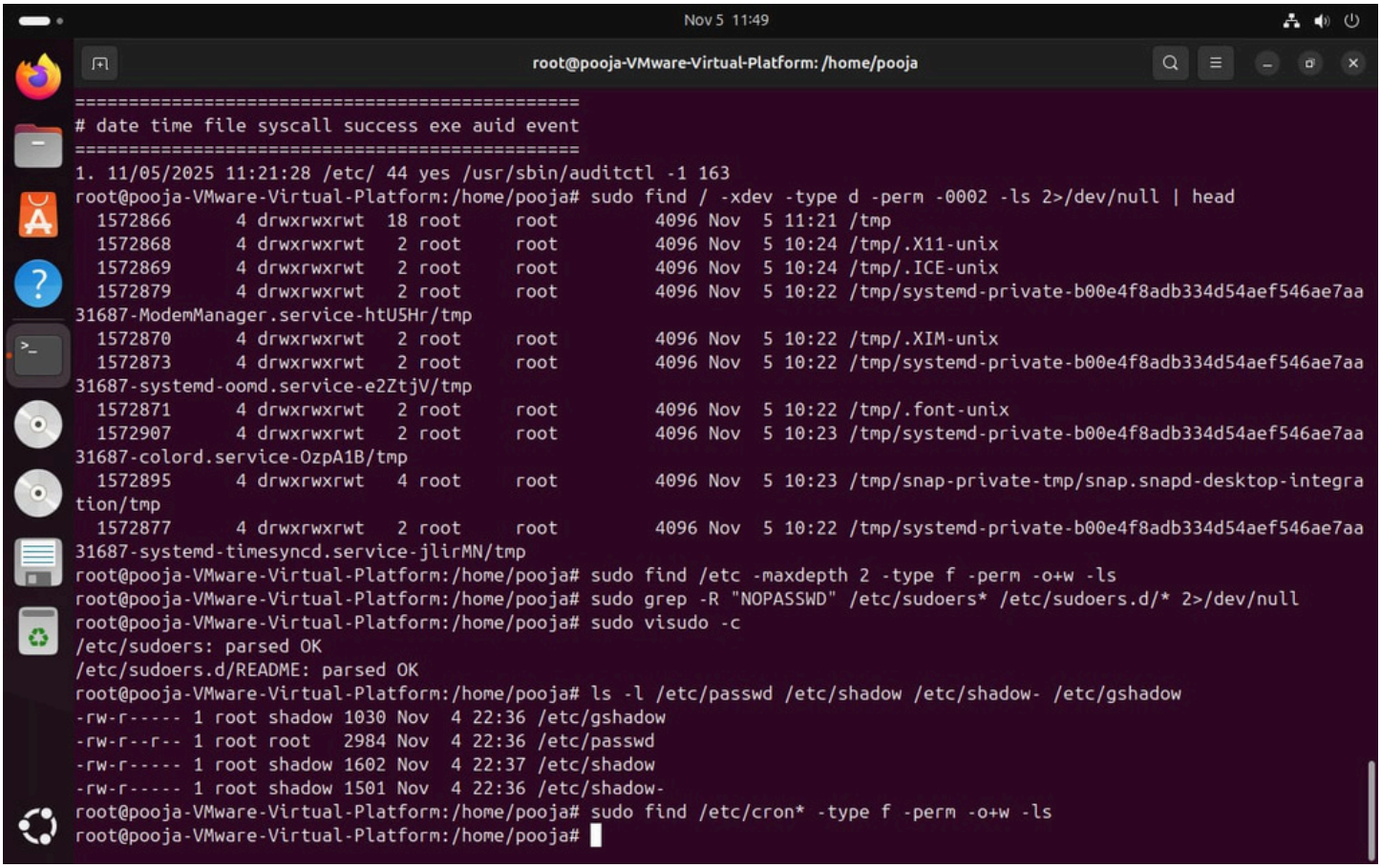
TASK:*In a separate vulnerable snapshot (instructor provides or students intentionally misconfigure), identify three misconfigurations (e.g., world-writable /etc/cron.d, unrestricted sudo NOPASSWD, weak file permissions on sensitive files).*

Find three misconfigurations in the vulnerable snapshot (intentional or instructor-provided) and fix them.

Examples you might find:

- ***World-writable cron file (/etc/cron.d): detected with ls -l; fix with chmod o-w /etc/cron.d/<file>.***
- ***Unrestricted NOPASSWD sudo for a user: detected by checking /etc/sudoers or sudo -l; fix by removing NOPASSWD entry via visudo.***
- ***Weak permissions on sensitive files (e.g., /etc/passwd or /etc/shadow): detected with ls -l; fix with chmod 644 /etc/passwd and chmod 640 /etc/shadow.***

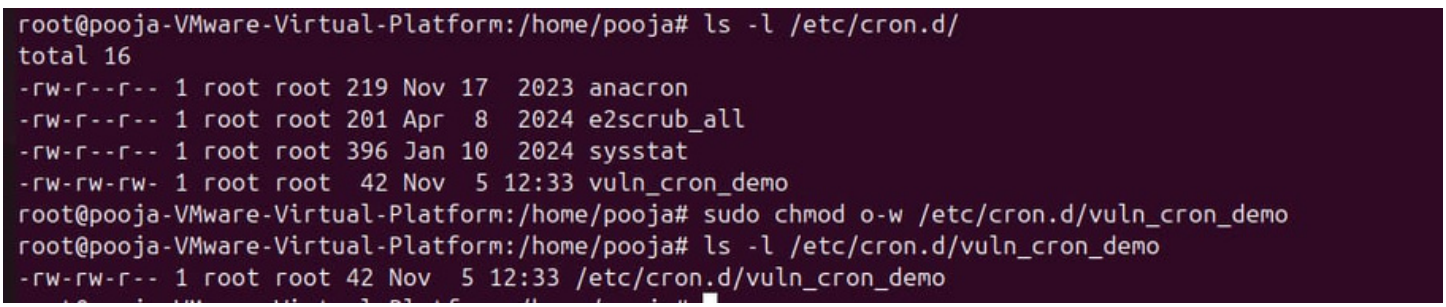
1.Misconfig 1 — World-writable /etc/cron.d/somefile



World-writable cron file — Not vulnerable (fixed).

The cron file was secured by removing the world-write bit so others cannot modify scheduled jobs; verified with ls -l.

Fix applied (example): sudo chmod o-w /etc/cron.d/vuln_cron_demo — now permissions prevent unauthorized writes.

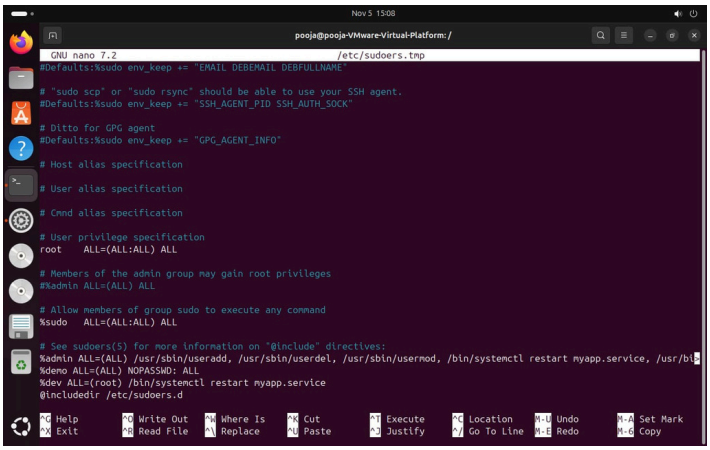


Misconfig 2 — NOPASSWD: ALL in sudoers

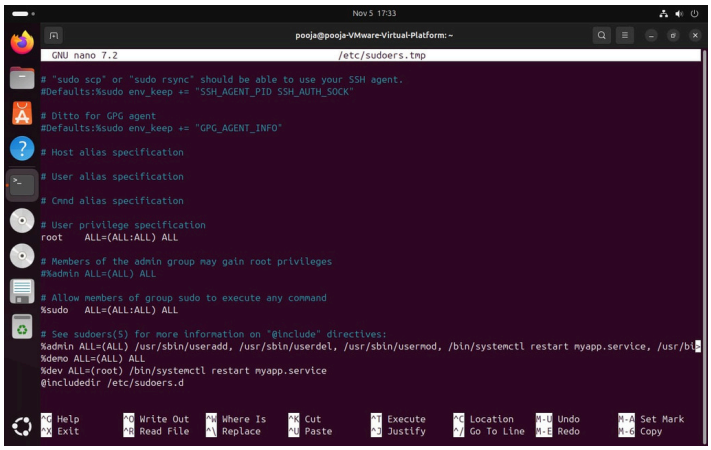
```
pooja@pooja-VMware-Virtual-Platform:~$ sudo ls -l /etc/sudoers.d/
total 8
-r--r----- 1 root root   30 Nov  5 16:35 demo-nopass
-r--r----- 1 root root 1068 Jan 29  2024 README
pooja@pooja-VMware-Virtual-Platform:~$ sudo grep -R "NOPASSWD" /etc/sudoers* /etc/sudoers.d/* 2>/dev/null || echo "no NO
PASSWD found"
/etc/sudoers.d/demo-nopass:%demo ALL=(ALL) NOPASSWD: ALL
/etc/sudoers.d/demo-nopass:%demo ALL=(ALL) NOPASSWD: ALL
pooja@pooja-VMware-Virtual-Platform:~$ sudo rm -f /etc/sudoers.d/demo-nopass
pooja@pooja-VMware-Virtual-Platform:~$ sudo visudo -c
/etc/sudoers: parsed OK
/etc/sudoers.d/README: parsed OK
```

NOPASSWD: ALL lets a user run *any* sudo command without entering their password. This defeats accountability and makes it easy for attackers (or a compromised account) to gain full root control.

BEFORE FIX:



AFTER FIX:



Before (vulnerable) — user can run *anything* as root without a password.

Risk: removes accountability and lets a compromised account gain full root access.

After — require password (safer) ,least-privilege alternative (restrict commands).

Misconfig 3 — /etc/shadow readable (should be only root)

Misconfiguration 3 — /etc/shadow is world-readable (should be root-only):
/etc/shadow stores hashed user passwords and account expiry data. If it is readable by non-root users, an attacker can copy the hashes and perform offline cracking (e.g., with john/hashcat) to recover passwords and escalate privileges. Check with ls -l /etc/shadow — correct: -rw-r----- 1 root shadow .... Fix by restoring ownership and permissions: sudo chown root:shadow /etc/shadow and sudo chmod 0640 /etc/shadow.

BEFORE FIX:

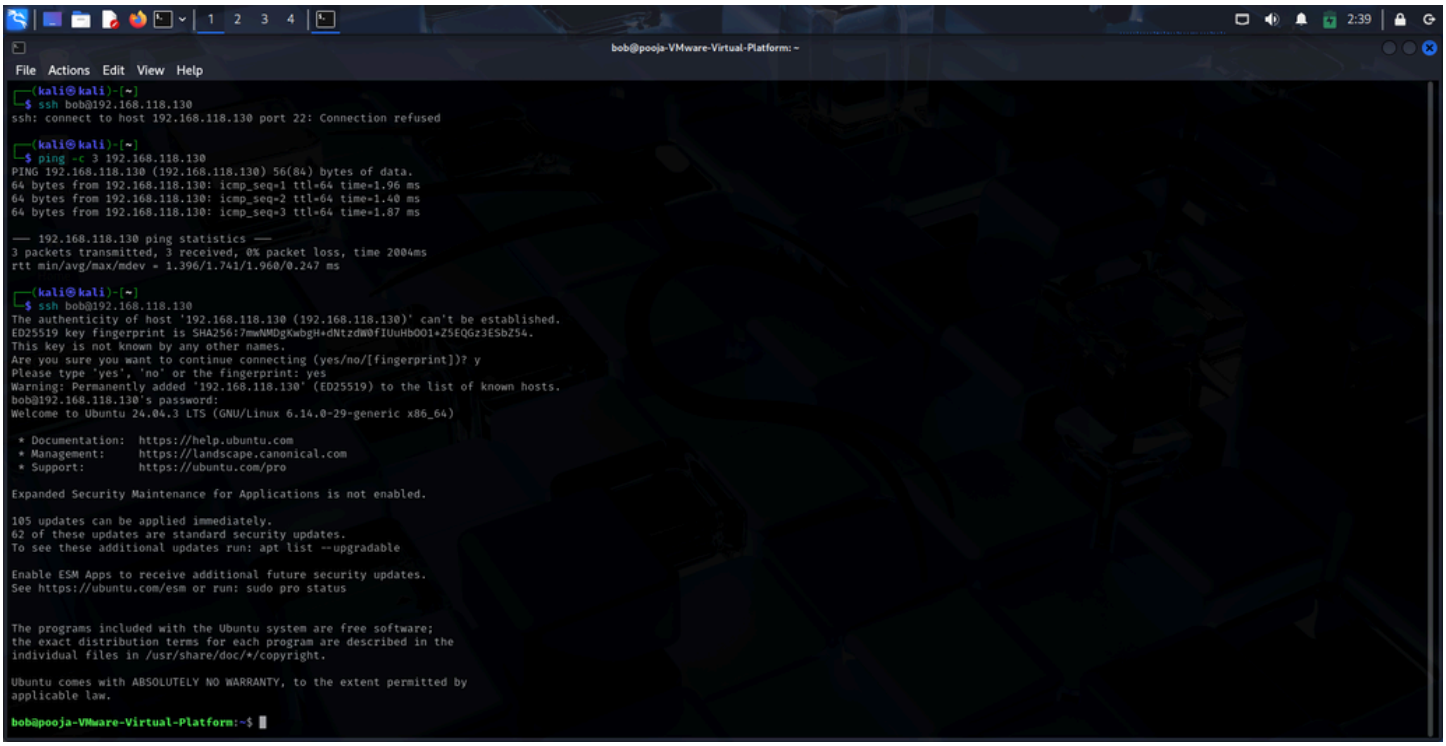
```
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo chown root:shadow /etc/shad
ow /etc/gshadow
root@pooja-VMware-Virtual-Platform:/home/pooja# sudo chmod 0640 /etc/shadow /etc
/gshadow
root@pooja-VMware-Virtual-Platform:/home/pooja# ls -l /etc/passwd /etc/shadow /e
tc/gshadow
-rw-r----- 1 root shadow 1060 Nov  5 14:21 /etc/gshadow
-rw-r--r-- 1 root root   3077 Nov  5 14:21 /etc/passwd
-rw-r----- 1 root shadow 1651 Nov  5 14:21 /etc/shadow
```

AFTER FIX:

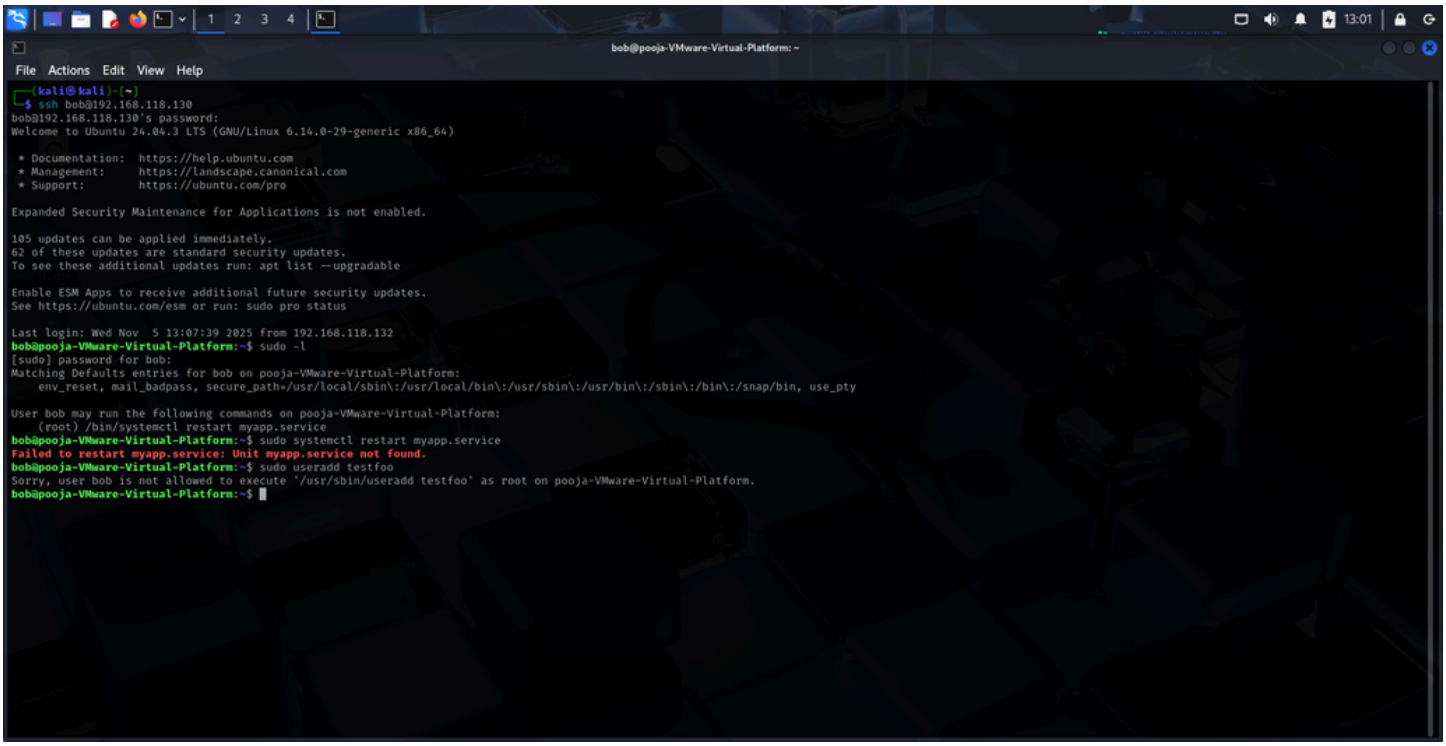
```
root@pooja-VMware-Virtual-Platform:/home/pooja# ls -l /etc/passwd /etc/shadow /e
tc/gshadow
-rw-r----- 1 root shadow 1060 Nov  5 14:21 /etc/gshadow
-rw-r--r-- 1 root root   3077 Nov  5 14:21 /etc/passwd
-rw-r--r-- 1 root shadow 1651 Nov  5 14:21 /etc/shadow
```

TASH:Testing / Verification — Attacker VM (Kali) & Target VM,Use Attacker VM to simulate tests against the target

SSH attempt initially refused, then after confirming the host key and successful ping, the attacker successfully SSH'd into the target as user bob (Ubuntu 24.04.3 LTS).



A terminal screen shows a user (bob) attempting and failing to restart a myapp.service using sudo systemctl restart myapp.service on an Ubuntu system, followed by an error when trying to run a testfoo command.


1 2 3 4
13:01
bob@pooja-VMware-Virtual-Platform: ~
File Actions Edit View Help
┌──(kali㉿kali)-[~]
└─$ ssh bob@192.168.118.130
bob@192.168.118.130's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-29-generic x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
Expanded Security Maintenance for Applications is not enabled.
105 updates can be applied immediately.
62 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
Last login: Wed Nov 5 13:07:39 2025 from 192.168.118.132
bob@pooja-VMware-Virtual-Platform:~$ sudo -l
[sudo] password for bob:
Matching Defaults entries for bob on pooja-VMware-Virtual-Platform:
env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty
User bob may run the following commands on pooja-VMware-Virtual-Platform:
(root) /bin/systemctl restart myapp.service
bob@pooja-VMware-Virtual-Platform:~$ sudo systemctl restart myapp.service
Failed to restart myapp.service: Unit myapp.service not found.
bob@pooja-VMware-Virtual-Platform:~$ sudo useradd testfoo
Sorry, user bob is not allowed to execute '/usr/sbin/useradd testfoo' as root on pooja-VMware-Virtual-Platform.
bob@pooja-VMware-Virtual-Platform:~$