

## **DAY 9 — Privilege Escalation Awareness (DEFENSIVE):**

CISO focus: “How does privilege escalation happen, and how do we see it before damage?”

### **LAB SAFETY RULES (READ ONCE)**

- Do NOT install exploit frameworks
- Do NOT modify system binaries
- Do NOT run random sudo commands blindly
- Everything is observe, test safely, and document

## ◆ **PHASE 1 — Identity & Privilege Baseline (Foundation)**

In this first phase we have to see where we are standing .

For foundation first we have check the username ,by typing this (whoami)command we will check the user name ,E.g dawood

After that we will type (id)command which tell us uid(user id) and gid(group id) And the last step of phase one we will type (groups)command ,it will tell us which clubs the user belongs to .

(Why we have to do this )

To know exactly what permissions a user has before an incident occurs.

Proof

```
Oracle VM VirtualBox Manager
Ubuntu Server [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Ubuntu 24.04.3 LTS target-server tty1
target-server login: dawood
Password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-90-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Wed 21 Jan 19:03:00 UTC 2026

System load: 1.8           Memory usage: 11%   Processes:      121
Usage of /:  41.4% of 11.21GB Swap usage:   0%     Users logged in: 0

Expanded Security Maintenance for Applications is not enabled.

65 updates can be applied immediately.
8 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

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings.

dawood@target-server:~$ whoami
dawood
dawood@target-server:~$ id
uid=1000(dawood) gid=1000(dawood) groups=1000(dawood),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),101(lpadmin),134(sambashare)
dawood@target-server:~$ groups
dawood adm cdrom sudo dip plugdev lxd
dawood@target-server:~$
```

## ◆ PHASE 2 — Sudo Configuration Awareness

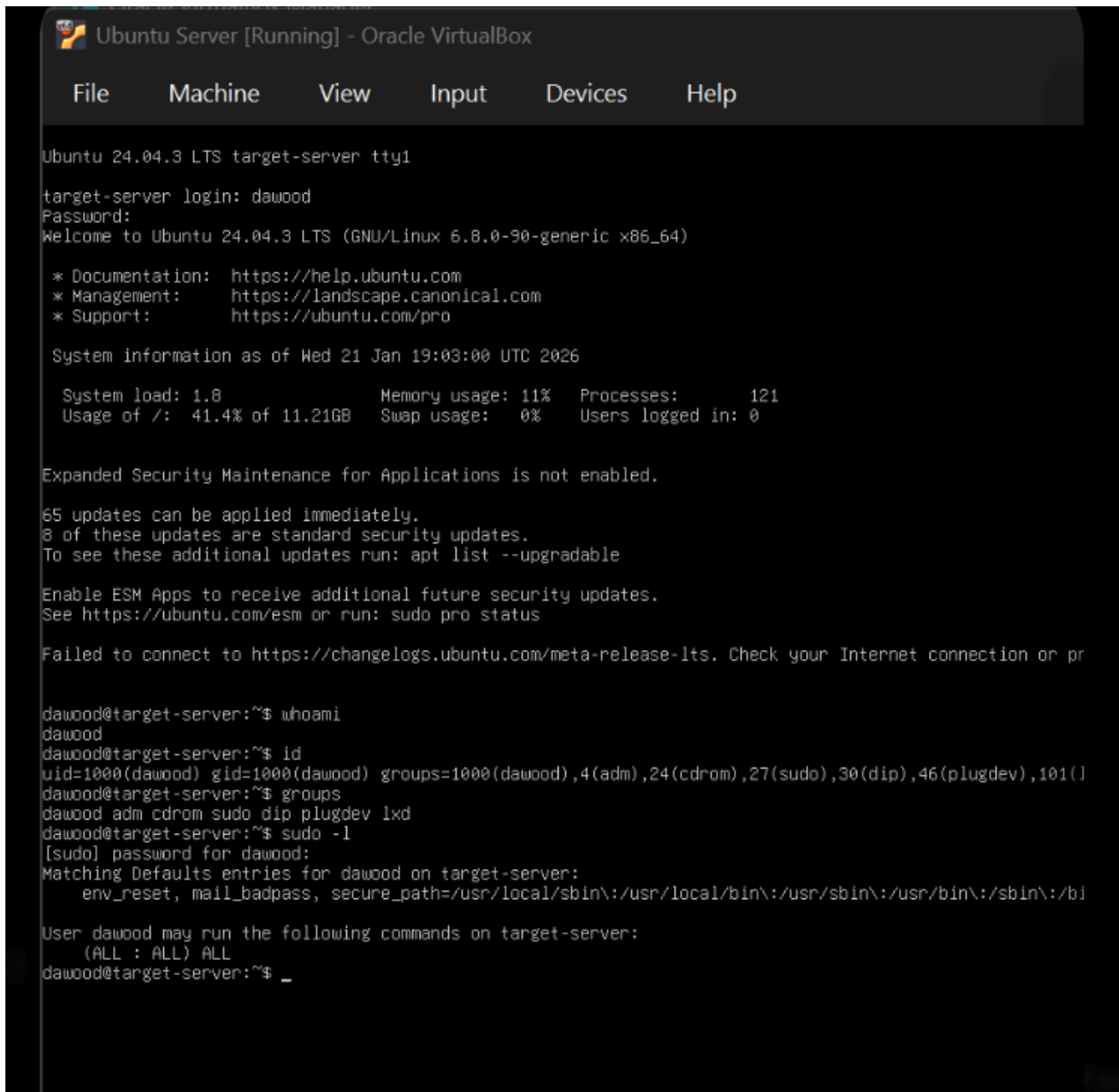
In this step we will check if our internal doors are locked or not .

In this step this(sudo -l) command will tell us about actual scenario

## 🔵 Why do we do this?

Attackers love **"Misconfigurations."** If a lazy Administrator sets a user to **NOPASSWD** so they don't have to type their password every time, they just give a free "Master Key" to any hacker who slips into that account.

Proof



```
Ubuntu Server [Running] - Oracle VirtualBox
File Machine View Input Devices Help

Ubuntu 24.04.3 LTS target-server tty1
target-server login: dawood
Password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-90-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Wed 21 Jan 19:03:00 UTC 2026

System load: 1.8           Memory usage: 11%   Processes:   121
Usage of /:  41.4% of 11.21GB Swap usage:   0%    Users logged in: 0

Expanded Security Maintenance for Applications is not enabled.

65 updates can be applied immediately.
8 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

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or pr

dawood@target-server:~$ whoami
dawood
dawood@target-server:~$ id
uid=1000(dawood) gid=1000(dawood) groups=1000(dawood),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),101(lxd)
dawood@target-server:~$ groups
dawood adm cdrom sudo dip plugdev lxd
dawood@target-server:~$ sudo -l
[sudo] password for dawood:
Matching Defaults entries for dawood on target-server:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/b

User dawood may run the following commands on target-server:
    (ALL : ALL) ALL
dawood@target-server:~$ _
```

## Step 2: View sudo configuration (Read-only)

In this step we will only inspect the logs ,table and master files

The command we will use in this is (sudo cat /etc/sudoers | less)

**sudo:** Required because this file is restricted to admins only.

**cat:** Reads the file.

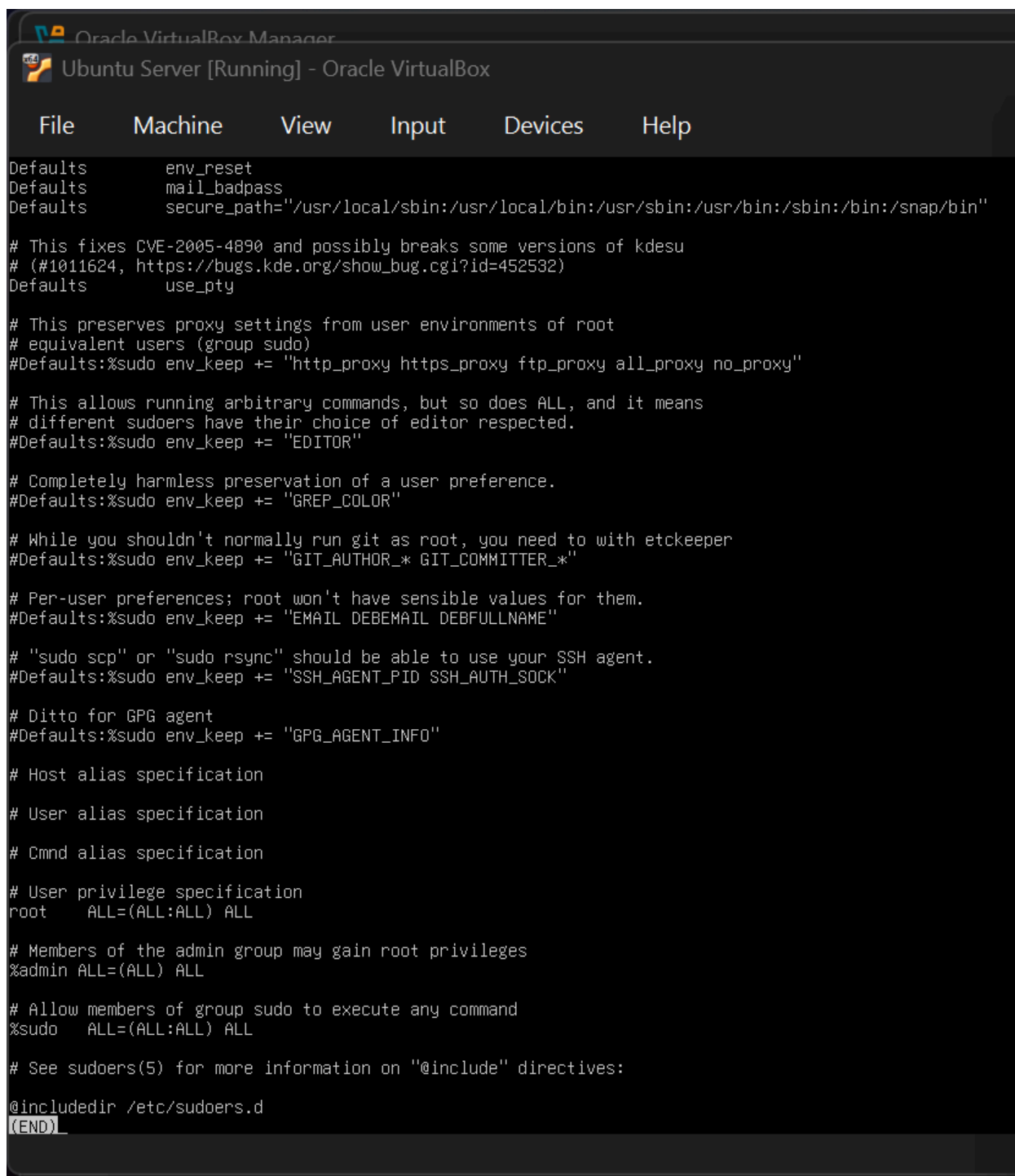
**|:** (The pipe you just fixed!) Send the text to the next tool.

**less:** Opens the file in a "viewer" mode so you don't accidentally change anything.

Why:

In a professional **SOC**, we use a "Look, Don't Touch" policy for audits. If you used a text editor like **nano** or **vi**, a single accidental keystroke could lock **everyone** (including root) out of the **sudo** command. Using **cat | less** is the safe, professional way to observe authority.

Proof



```
Oracle VM VirtualBox Manager
Ubuntu Server [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Defaults env_reset
Defaults mail_badpass
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# This fixes CVE-2005-4890 and possibly breaks some versions of kdesu
# (#1011624, https://bugs.kde.org/show_bug.cgi?id=452532)
Defaults use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
Defaults:%sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# 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:

@include /etc/sudoers.d
(END)
```

## ◆ PHASE 3 — SUID Awareness (Critical Concept)

### Step 3: Find SUID binaries

In step we will visualize everything

In this step the command we are using is (find / -perm -4000 -type f

2>/dev/null)

**find** /: Look everywhere on the entire server.

**-perm -4000**: This is the technical code for **SUID**. It tells Linux "show me programs that run with the owner's power."

**-type f**: Only look for files (not folders).

**2>/dev/null**: This hides "Permission Denied" errors so your screen stays clean.

### **Why:**

By running this command, you are performing a "**Vulnerability Assessment.**"

You are making sure that the only programs with extra power are the ones that actually *need* it to keep the system running.

Proof

```
Oracle VM VirtualBox Manager
Ubuntu Server [Running] - Oracle VirtualBox

File Machine View Input Devices Help

# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# 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:

@includedir /etc/sudoers.d
dawood@target-server:~$ find / -perm -4000 -type f 2>/dev/null
/usr/bin/fusermount3
/usr/bin/sudo
/usr/bin/mount
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/umount
/usr/bin/su
/usr/bin/passwd
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
dawood@target-server:~$
```

## PHASE 4 — Log-Based Detection (MOST IMPORTANT)

### Step 4: Monitor sudo usage

In this step we will move deeper in sudo with the help of auth.log we will monitor live logs

In this step the command we are using is (sudo tail -f /var/log/auth.log).

In the next step we have to open tty2 (second terminal )for observing live logs  
We will type (alt+F2) for tty2 then enter (sudo ls).

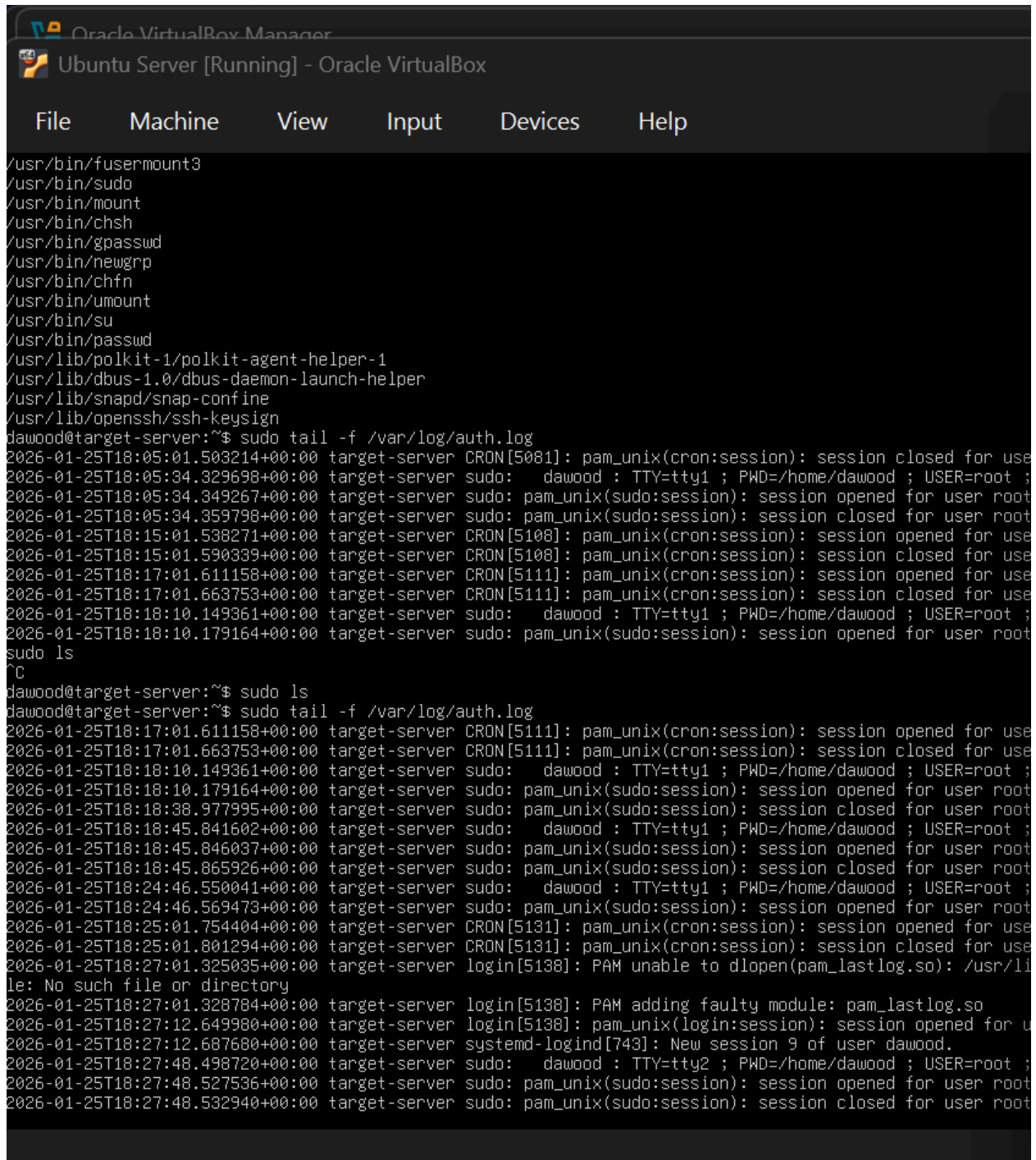
Why:

If an attacker steals your password and runs `sudo cat /etc/shadow` to steal everyone's passwords, the **only** way the company will know is by looking at this log.

By watching `auth.log`, you are turning the server's memory into a weapon against the attacker.



## Proof



```
Oracle VM VirtualBox Manager
Ubuntu Server [Running] - Oracle VirtualBox

File Machine View Input Devices Help

/usr/bin/fusermount3
/usr/bin/sudo
/usr/bin/mount
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/umount
/usr/bin/su
/usr/bin/passwd
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
dawood@target-server:~$ sudo tail -f /var/log/auth.log
2026-01-25T18:05:01.503214+00:00 target-server CRON[5081]: pam_unix(cron:session): session closed for use
2026-01-25T18:05:34.329698+00:00 target-server sudo: dawood : TTY=ttty1 ; PWD=/home/dawood ; USER=root ;
2026-01-25T18:05:34.349267+00:00 target-server sudo: pam_unix(sudo:session): session opened for user root
2026-01-25T18:05:34.359798+00:00 target-server sudo: pam_unix(sudo:session): session closed for user root
2026-01-25T18:15:01.538271+00:00 target-server CRON[5108]: pam_unix(cron:session): session opened for use
2026-01-25T18:15:01.590339+00:00 target-server CRON[5108]: pam_unix(cron:session): session closed for use
2026-01-25T18:17:01.611158+00:00 target-server CRON[5111]: pam_unix(cron:session): session opened for use
2026-01-25T18:17:01.663753+00:00 target-server CRON[5111]: pam_unix(cron:session): session closed for use
2026-01-25T18:18:10.149361+00:00 target-server sudo: dawood : TTY=ttty1 ; PWD=/home/dawood ; USER=root ;
2026-01-25T18:18:10.179164+00:00 target-server sudo: pam_unix(sudo:session): session opened for user root
dawood@target-server:~$ sudo ls
^C
dawood@target-server:~$ sudo ls
dawood@target-server:~$ sudo tail -f /var/log/auth.log
2026-01-25T18:17:01.611158+00:00 target-server CRON[5111]: pam_unix(cron:session): session opened for use
2026-01-25T18:17:01.663753+00:00 target-server CRON[5111]: pam_unix(cron:session): session closed for use
2026-01-25T18:18:10.149361+00:00 target-server sudo: dawood : TTY=ttty1 ; PWD=/home/dawood ; USER=root ;
2026-01-25T18:18:10.179164+00:00 target-server sudo: pam_unix(sudo:session): session opened for user root
2026-01-25T18:18:38.977995+00:00 target-server sudo: pam_unix(sudo:session): session closed for user root
2026-01-25T18:18:45.841602+00:00 target-server sudo: dawood : TTY=ttty1 ; PWD=/home/dawood ; USER=root ;
2026-01-25T18:18:45.846037+00:00 target-server sudo: pam_unix(sudo:session): session opened for user root
2026-01-25T18:18:45.865926+00:00 target-server sudo: pam_unix(sudo:session): session closed for user root
2026-01-25T18:24:46.550041+00:00 target-server sudo: dawood : TTY=ttty1 ; PWD=/home/dawood ; USER=root ;
2026-01-25T18:24:46.569473+00:00 target-server sudo: pam_unix(sudo:session): session opened for user root
2026-01-25T18:25:01.754404+00:00 target-server CRON[5131]: pam_unix(cron:session): session opened for use
2026-01-25T18:25:01.801294+00:00 target-server CRON[5131]: pam_unix(cron:session): session closed for use
2026-01-25T18:27:01.325035+00:00 target-server login[5138]: PAM unable to dlopen(pam_lastlog.so): /usr/li
le: No such file or directory
2026-01-25T18:27:01.328784+00:00 target-server login[5138]: PAM adding faulty module: pam_lastlog.so
2026-01-25T18:27:12.649980+00:00 target-server login[5138]: pam_unix(login:session): session opened for u
2026-01-25T18:27:12.687680+00:00 target-server systemd-logind[743]: New session 9 of user dawood.
2026-01-25T18:27:48.498720+00:00 target-server sudo: dawood : TTY=ttty2 ; PWD=/home/dawood ; USER=root ;
2026-01-25T18:27:48.527536+00:00 target-server sudo: pam_unix(sudo:session): session opened for user root
2026-01-25T18:27:48.532940+00:00 target-server sudo: pam_unix(sudo:session): session closed for user root
```

## ◆ PHASE 5 — Attack Simulation (Safe & Minimal)

In this step we will run(`ssh root@ubuntu_ip`) this command ,and it is expected to be failure.

When Kali asks for a password, type `password123` (or any wrong password).

Do this **3 times**. SSH will eventually kick you out.

### **Why:**

By seeing these logs, you have proven that:

**Visibility:** You have a "camera" on your most sensitive account (root).

**Attribution:** You know the attacker's IP is `10.0.0.1`.

**Prevention:** Because you saw these failures, you could now block that IP before they ever get a chance to try a privilege escalation trick like `sudo` or exploiting a **SUID** binary.

To prove the **SOC (Security Operations Center)** can detect Brute Force attacks before an attacker gets in.

Proof

kali@kali: ~

Session Actions Edit View Help

```
command 'tda' from deb devtodo
command 'adr' from deb adr-tools
command 'aa' from deb astronomical-almanac
command 'agda' from deb agda-bin
command 'ad' from deb netatalk-tools
command 'ava' from deb ava
command 'mda' from deb mailutils-mda
command 'sada' from deb plc-utils-extra
command 'aha' from deb aha
command 'adb' from deb adb
command 'adb' from deb google-android-platform-tools-installer
command 'pda' from deb speech-tools
```

Try: `sudo apt install <deb name>`

(kali@kali)-[~]

\$ ssh root@10.0.0.2

root@10.0.0.2's password:

adad

adad

adPermission denied, please try again.

root@10.0.0.2's password:

adda

adaPermission denied, please try again.

root@10.0.0.2's password:

root@10.0.0.2: Permission denied (publickey,password).

(kali@kali)-[~]

\$

```

kali-linux-2025.4-virtualbox-amd64 (Running) - Oracle VM VirtualBox
Ubuntu Server [Running] - Oracle VM VirtualBox

File      Machine      View      Input      Devices      Help

2026-01-25T18:05:34.329698+00:00 target-server sudo: dawood : TTY=ttty1 ; PWD=/home/dawood ; USER=root
2026-01-25T18:05:34.349267+00:00 target-server sudo: pam_unix(sudo:session): session opened for user root
2026-01-25T18:05:34.359798+00:00 target-server sudo: pam_unix(sudo:session): session closed for user root
2026-01-25T18:15:01.538271+00:00 target-server CRON[5108]: pam_unix(cron:session): session opened for user root
2026-01-25T18:15:01.590339+00:00 target-server CRON[5108]: pam_unix(cron:session): session closed for user root
2026-01-25T18:17:01.611158+00:00 target-server CRON[5111]: pam_unix(cron:session): session opened for user root
2026-01-25T18:17:01.663753+00:00 target-server CRON[5111]: pam_unix(cron:session): session closed for user root
2026-01-25T18:18:10.149361+00:00 target-server sudo: dawood : TTY=ttty1 ; PWD=/home/dawood ; USER=root
2026-01-25T18:18:10.179164+00:00 target-server sudo: pam_unix(sudo:session): session opened for user root
sudo ls
^C
dawood@target-server:~$ sudo ls
dawood@target-server:~$ sudo tail -f /var/log/auth.log
2026-01-25T18:17:01.611158+00:00 target-server CRON[5111]: pam_unix(cron:session): session opened for user root
2026-01-25T18:17:01.663753+00:00 target-server CRON[5111]: pam_unix(cron:session): session closed for user root
2026-01-25T18:18:10.149361+00:00 target-server sudo: dawood : TTY=ttty1 ; PWD=/home/dawood ; USER=root
2026-01-25T18:18:10.179164+00:00 target-server sudo: pam_unix(sudo:session): session opened for user root
2026-01-25T18:18:38.977995+00:00 target-server sudo: pam_unix(sudo:session): session closed for user root
2026-01-25T18:18:45.841602+00:00 target-server sudo: dawood : TTY=ttty1 ; PWD=/home/dawood ; USER=root
2026-01-25T18:18:45.846037+00:00 target-server sudo: pam_unix(sudo:session): session opened for user root
2026-01-25T18:18:45.865926+00:00 target-server sudo: pam_unix(sudo:session): session closed for user root
2026-01-25T18:24:46.550041+00:00 target-server sudo: dawood : TTY=ttty1 ; PWD=/home/dawood ; USER=root
2026-01-25T18:24:46.569473+00:00 target-server sudo: pam_unix(sudo:session): session opened for user root
2026-01-25T18:25:01.754404+00:00 target-server CRON[5131]: pam_unix(cron:session): session opened for user root
2026-01-25T18:25:01.801294+00:00 target-server CRON[5131]: pam_unix(cron:session): session closed for user root
2026-01-25T18:27:01.325035+00:00 target-server login[5138]: PAM unable to dlopen(pam_lastlog.so): /usr/lib
le: No such file or directory
2026-01-25T18:27:01.328784+00:00 target-server login[5138]: PAM adding faulty module: pam_lastlog.so
2026-01-25T18:27:12.649980+00:00 target-server login[5138]: pam_unix(login:session): session opened for user root
2026-01-25T18:27:12.687680+00:00 target-server systemd-logind[743]: New session 9 of user dawood.
2026-01-25T18:27:48.498720+00:00 target-server sudo: dawood : TTY=ttty2 ; PWD=/home/dawood ; USER=root
2026-01-25T18:27:48.527536+00:00 target-server sudo: pam_unix(sudo:session): session opened for user root
2026-01-25T18:27:48.532940+00:00 target-server sudo: pam_unix(sudo:session): session closed for user root
2026-01-25T18:35:01.868443+00:00 target-server CRON[5237]: pam_unix(cron:session): session opened for user root
2026-01-25T18:35:01.923712+00:00 target-server CRON[5237]: pam_unix(cron:session): session closed for user root
2026-01-25T18:37:02.144538+00:00 target-server sshd[5242]: Server listening on 0.0.0.0 port 22.
2026-01-25T18:37:02.151145+00:00 target-server sshd[5242]: Server listening on :: port 22.
2026-01-25T18:37:13.665201+00:00 target-server sshd[5243]: pam_unix(sshd:auth): authentication failure;
r=root
2026-01-25T18:37:15.994340+00:00 target-server sshd[5243]: Failed password for root from 10.0.0.1 port 50
2026-01-25T18:37:23.942324+00:00 target-server sshd[5243]: message repeated 2 times: [ Failed password fo
2026-01-25T18:37:24.132924+00:00 target-server sshd[5243]: Connection closed by authenticating user root
2026-01-25T18:37:24.145325+00:00 target-server sshd[5243]: PAM 2 more authentication failures; logname= u
2026-01-25T18:37:57.274128+00:00 target-server sshd[5247]: pam_unix(sshd:auth): authentication failure;
r=root
2026-01-25T18:37:58.970509+00:00 target-server sshd[5247]: Failed password for root from 10.0.0.1 port 40
2026-01-25T18:38:02.327927+00:00 target-server sshd[5247]: message repeated 2 times: [ Failed password fo
2026-01-25T18:38:02.345850+00:00 target-server sshd[5247]: Connection closed by authenticating user root
2026-01-25T18:38:02.346181+00:00 target-server sshd[5247]: PAM 1 more authentication failure; logname= u

```

## ◆ **PHASE 6 — Defensive Thinking (WRITE THIS)**

**1)What misconfigurations enable privilege escalation?**

ans) Giving users admin power without needing a password (**NOPASSWD**) or leaving "Master Keys" (SUID) on files that don't need them.

***2) What log entries indicate escalation attempts?***

ans) Repeated "Failed password" messages followed by someone successfully becoming "root" in the logs.

***3) How would Fail2Ban or SIEM detect this?***

ans) They act like a "Digital Security Guard" that counts failures and locks the door (blocks the IP) after too many bad guesses.

***4) How does this map to cloud IAM misuse?***

ans) It is also the same thing that we are doing here, just the difference is it will be on cloud.

***conclusion:***

when a "bad guy" (Kali) tried to guess a password, your "Security Camera" (Ubuntu) caught them immediately.

**Day 9: Command Table**

Command	What it does (Easy Words)	Why we use it (Recruiter)

<b>id</b>	Shows your "Identity" number.	To see if you are a normal user or Root.
<b>sudo -l</b>	Lists your "Admin Powers."	To find security gaps in the rulebook.
<b>sudo cat /etc/sudoers</b>	Reads the "Master Law File."	To check if the system is locked properly.
<b>find / -perm -4000</b>	Searches for "Borrowed Keys."	To find hidden backdoors in programs.
<b>tail -f /var/log/auth.log</b>	Watches "Security Logs" live.	To catch attackers while they are moving.

### Day 9: Full Form Keys

Short Form	Full Form	Easy Meaning

<b>UID</b>	<b>User IDentifier</b>	Your unique digital "ID Card" number.
<b>GID</b>	<b>Group IDentifier</b>	The ID for your "Team" (like the Sudo team).
<b>SU DO</b>	<b>SuperUser DO</b>	Acting like the "Big Boss" (Root) for one task.
<b>SUI D</b>	<b>Set User ID</b>	A file that "borrows" Root's power to run.
<b>IAM</b>	<b>Identity &amp; Access Management</b>	The rulebook for who gets to touch what.
<b>SO C</b>	<b>Security Operations Center</b>	The "Security Room" where we watch the logs.