


Task 1: User & Permission Misconfigurations

 **Objective:** Demonstrate how improper file permissions can lead to security vulnerabilities and how to fix them.

1 Setup: Creating Users & Misconfiguring Permissions

Step 1: Create Users

Command:

bash

CopyEdit

```
sudo useradd -m user1
```

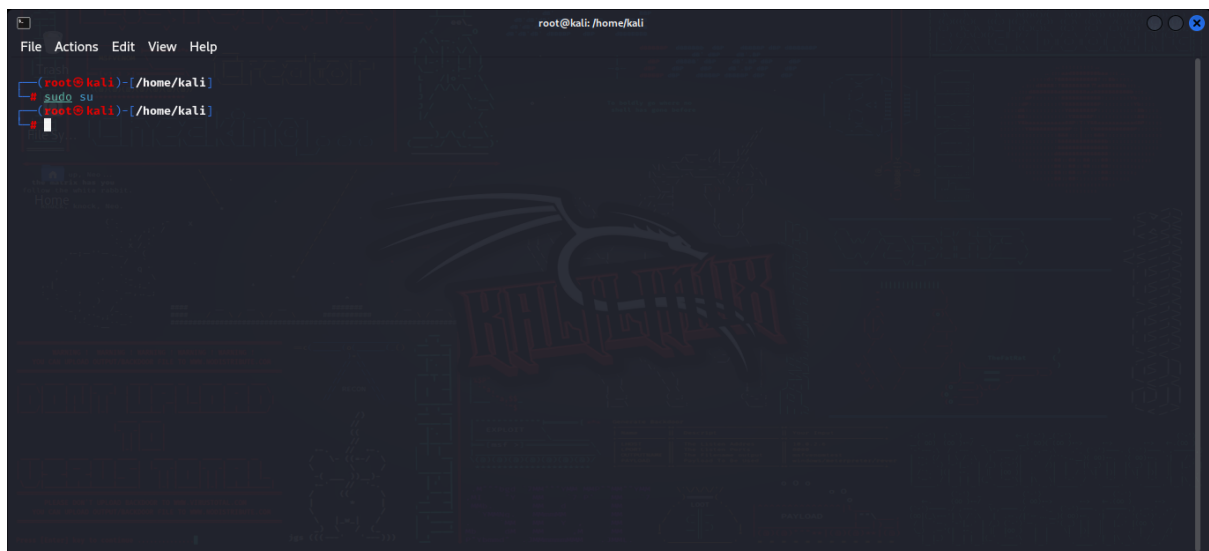
```
sudo useradd -m user2
```

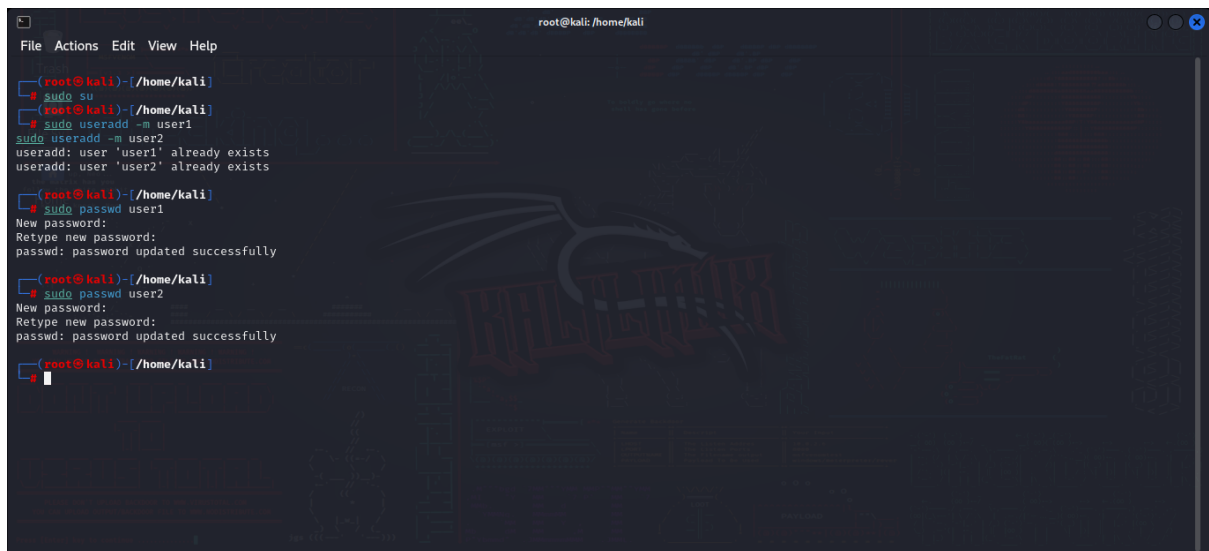
What It Does?

- `useradd -m user1`: Creates a new user named user1 with a home directory (/home/user1).
- `useradd -m user2`: Creates a new user named user2 with a home directory (/home/user2).

Expected Output:

No output (Success is silent).



A screenshot of a Kali Linux terminal window. The window title is 'root@kali: /home/kali'. The terminal shows a series of commands and their outputs. The user is in the root shell. They run 'sudo su', then 'sudo useradd -m user1', and 'sudo useradd -m user2'. The output for the last two commands is 'useradd: user 'user1' already exists' and 'useradd: user 'user2' already exists'. Then they run 'sudo passwd user1', which prompts for a new password and retypes it, resulting in 'passwd: password updated successfully'. Finally, they run 'sudo passwd user2', which also prompts for a new password and retypes it, resulting in 'passwd: password updated successfully'. The terminal has a dark background with a faint 'KALI' logo in the center.

```
root@kali: /home/kali
File Actions Edit View Help

root@kali: /home/kali
# sudo su
root@kali: /home/kali
# sudo useradd -m user1
useradd: user 'user1' already exists
# sudo useradd -m user2
useradd: user 'user2' already exists

root@kali: /home/kali
# sudo passwd user1
New password:
Retype new password:
passwd: password updated successfully

root@kali: /home/kali
# sudo passwd user2
New password:
Retype new password:
passwd: password updated successfully

root@kali: /home/kali
#
```

Step 2: Set Passwords

Command:

bash

CopyEdit

sudo passwd user1

What It Does?

- Prompts to enter and confirm a password for user1.

Expected Output:

plaintext

CopyEdit

New password: *****

Retype new password: *****

passwd: password updated successfully

(Repeat the same command for user2.)

```
root@kali: /home/kali
File Actions Edit View Help

(root@kali)-[/home/kali]
# sudo su
# (root@kali)-[/home/kali]
# sudo useradd -m user1
# sudo useradd -m user2
useradd: user 'user1' already exists
useradd: user 'user2' already exists

(root@kali)-[/home/kali]
# sudo passwd user1
New password:
Retype new password:
passwd: password updated successfully

(root@kali)-[/home/kali]
# sudo passwd user2
New password:
Retype new password:
passwd: password updated successfully

(root@kali)-[/home/kali]
# sudo chmod 777 /etc/shadow
# sudo chmod 777 /etc/passwd

(root@kali)-[/home/kali]
#
```

Step 3: Assign Incorrect Permissions

Command:

bash


CopyEdit

```
sudo chmod 777 /etc/shadow
```

```
sudo chmod 777 /etc/passwd
```

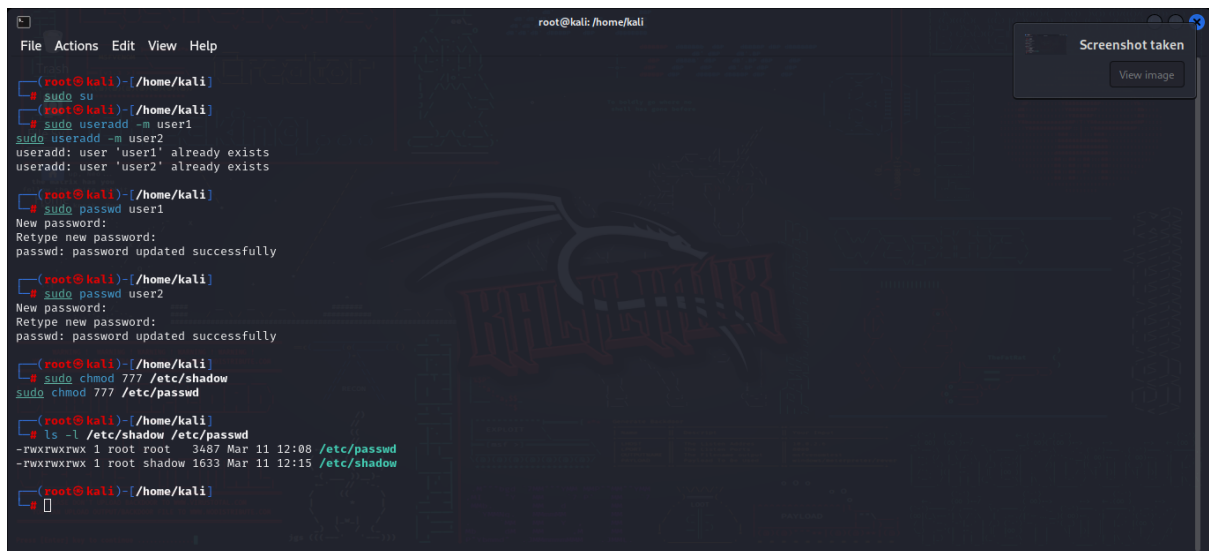
What It Does?

- `chmod 777 /etc/shadow`: Makes the password file (`/etc/shadow`) **readable, writable, and executable by all users**.
- `chmod 777 /etc/passwd`: Makes the user information file (`/etc/passwd`) **fully accessible to everyone**.

 **Security Risk:** Any user can read and modify these files!

Expected Output:

No output (Success is silent).



```
(root@kali)~/home/kali
# sudo su
(root@kali)~/home/kali
# sudo useradd -m user1
useradd: user 'user1' already exists
# sudo useradd -m user2
useradd: user 'user2' already exists

(root@kali)~/home/kali
# sudo passwd user1
New password:
Retype new password:
passwd: password updated successfully

(root@kali)~/home/kali
# sudo passwd user2
New password:
Retype new password:
passwd: password updated successfully

(root@kali)~/home/kali
# sudo chmod 777 /etc/shadow
# sudo chmod 777 /etc/passwd

(root@kali)~/home/kali
# ls -l /etc/shadow /etc/passwd
-rwxrwxrwx 1 root root 3487 Mar 11 12:08 /etc/passwd
-rwxrwxrwx 1 root shadow 1633 Mar 11 12:15 /etc/shadow

(root@kali)~/home/kali
#
```

Step 4: Verify Permissions

Command:

bash

CopyEdit

```
ls -l /etc/shadow /etc/passwd
```

What It Does?

- Lists the **detailed file permissions** for /etc/shadow and /etc/passwd.

Expected Output:

plaintext

CopyEdit

```
-rwxrwxrwx 1 root shadow <date> /etc/shadow
```

```
-rwxrwxrwx 1 root root <date> /etc/passwd
```

 **Issue:** 777 means **anyone** can modify these critical files.

```
root@kali: /home/kali
File Actions Edit View Help

(root@kali)-[/home/kali]
# sudo passwd user1
New password:
Retype new password:
passwd: password updated successfully

(root@kali)-[/home/kali]
# sudo passwd user2
New password:
Retype new password:
passwd: password updated successfully

(root@kali)-[/home/kali]
# sudo chmod 777 /etc/shadow
sudo chmod 777 /etc/passwd

(root@kali)-[/home/kali]
# ls -l /etc/shadow /etc/passwd
-rwxrwxrwx 1 root root 3487 Mar 11 12:08 /etc/passwd
-rwxrwxrwx 1 root shadow 1633 Mar 11 12:15 /etc/shadow

(root@kali)-[/home/kali]
# su - user1
$ cat /etc/shadow
root:*:19500:0:99999:7:::
daemon:*:19500:0:99999:7:::
bin:*:19500:0:99999:7:::
sys:*:19500:0:99999:7:::
sync:*:19500:0:99999:7:::
games:*:19500:0:99999:7:::
man:*:19500:0:99999:7:::
lp:*:19500:0:99999:7:::
mail:*:19500:0:99999:7:::
```

2 Exploitation: Accessing Sensitive Files as a Low-Privilege User

Step 5: Switch to Non-Root User

Command:

bash

CopyEdit

su - user1

What It Does?

- Switches to the user1 account.

Expected Output:

plaintext

CopyEdit

user1@hostname:~\$

(The shell prompt changes, indicating you're now user1.)

Step 6: Attempt to Read Sensitive Files

Command:

bash

CopyEdit

cat /etc/shadow

What It Does?

- Displays the contents of /etc/shadow, which should be **restricted to root**.

Expected Output (If the vulnerability exists!):

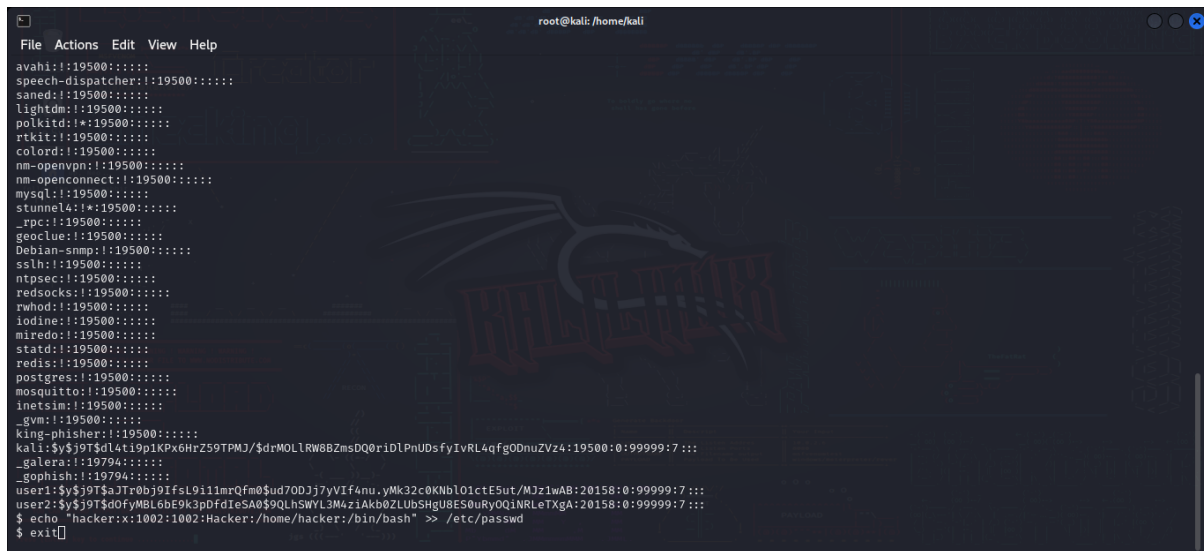
plaintext

CopyEdit

```
root:$6$randomhash:19000:0:99999:7:::
```

```
user1:$6$randomhash:19000:0:99999:7:::
```

```
user2:$6$randomhash:19000:0:99999:7:::
```



```
root@kali: /home/kali
File Actions Edit View Help
avahi:::19500:::
speech-dispatcher:::19500:::
saned:::19500:::
lightdm:::19500:::
polkitd:::19500:::
rtkit:::19500:::
colord:::19500:::
nm-openvpn:::19500:::
nm-openconnect:::19500:::
mysql:::19500:::
stunnel4:::19500:::
_rpc:::19500:::
geoclue:::19500:::
Debian-snmpp:::19500:::
sshd:::19500:::
ntpsvc:::19500:::
redsocks:::19500:::
rwhod:::19500:::
iodine:::19500:::
miredo:::19500:::
statd:::19500:::
redis:::19500:::
postgres:::19500:::
mosquitto:::19500:::
inetsim:::19500:::
_gvm:::19500:::
king-phisher:::19500:::
kali:$y$9T$dL4ti9p1KPx6HzZ59TPMJ/$drMOLLRW8BZmsDQ0r1DLPnUDsfyIVRL4qfg0DnuZVz4:19500:0:99999:7:::
_galera:::19794:::
_gophish:::19794:::
user1:$y$9T$zJr0b9JIfcL9i1mrQfm0$ud70DJj7yVif4nu.yMk32c0KNb101ctE5ut/MJz1wAB:20158:0:99999:7:::
user2:$y$9T$d0fyMBL6bE9k3pDfdIeSA0$9QLhSWYL3M4ziAkB0ZLUBSHgU8ES0uRyOQ1NRLeTXgA:20158:0:99999:7:::
$ echo "hacker:x:1002:1002:Hacker:/home/hacker:/bin/bash" >> /etc/passwd
$ exit
```

 Critical Risk: user1 can read password hashes, which an attacker could crack.

Another Exploit - Modify /etc/passwd

Command:

```
bash
```

Copy

Edit

```
echo "hacker:x:1002:1002:Hacker:/home/hacker:/bin/bash" >> /etc/passwd
```

What It Does?

Adds a fake user (hacker) with a shell, allowing privilege escalation.

Expected Output:

(No error = Vulnerability present!)

 Mitigation: Fixing Permission Issues

Step 7: Exit User1 Session

Command:

bash

Copy

Edit

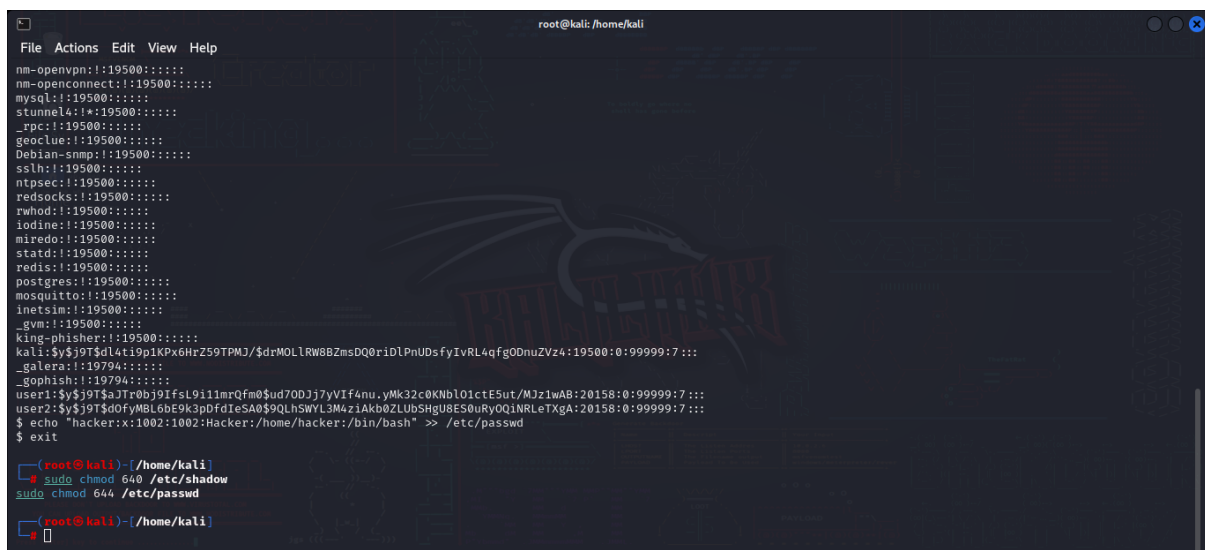
exit

What It Does?

Returns to the root user session.

Expected Output:

Returns to root shell (root@hostname:~#).



```
root@kali: /home/kali
File Actions Edit View Help
nm-openvpn:!:19500:!!!!
nm-openconnect:!:19500:!!!!
mysql:!:19500:!!!!
stunnel4:!*:19500:!!!!
_rpc:!:19500:!!!!
geoclue:!:19500:!!!!
Debian-snmpp:!:19500:!!!!
ssllh:!:19500:!!!!
ntpsec:!:19500:!!!!
redsocks:!:19500:!!!!
rwhod:!:19500:!!!!
iodine:!:19500:!!!!
miredo:!:19500:!!!!
statd:!:19500:!!!!
redis:!:19500:!!!!
postgres:!:19500:!!!!
mosquitto:!:19500:!!!!
inetsim:!:19500:!!!!
gvm:!:19500:!!!!
king-phisher:!:19500:!!!!
kali:!:19500:!!!!
_galera:!:19794:!!!!
_gophish:!:19794:!!!!
user1:!:19500:!!!!
user2:!:19500:!!!!
$ echo "hacker:x:1002:1002:Hacker:/home/hacker:/bin/bash" >> /etc/passwd
$ exit
root@kali:~#
root@kali:~# sudo chmod 640 /etc/shadow
root@kali:~# sudo chmod 644 /etc/passwd
root@kali:~#
```

Step 8: Secure File Permissions

Command:

bash

Copy

sudo chmod 640 /etc/shadow

sudo chmod 644 /etc/passwd

What It Does?

- chmod 640 /etc/shadow: Only **root** and the **shadow group** can access /etc/shadow.
- chmod 644 /etc/passwd: Only **root** can edit /etc/passwd, but all users can read it.

Expected Output:

(No output = Success)

```
root@kali: /home/kali
File Actions Edit View Help
geoclue::19500:~::~:
Debian-snmpp::19500:~::~:
ssllh::19500:~::~:
ntpsec::19500:~::~:
redsocks::19500:~::~:
rwhod::19500:~::~:
iodine::19500:~::~:
miredo::19500:~::~:
statd::19500:~::~:
redis::19500:~::~:
postgres::19500:~::~:
mosquitto::19500:~::~:
inetsim::19500:~::~:
gvm::19500:~::~:
king-phisher::19500:~::~:
kali:$y$9T$dl4ti9p1KPx6HzZ59TPMJ/$drMOLLRW8BZmsDQ0r1dLPnUDsfyIVRL4qfg0DnuZVz4:19500:0:99999:7:::
_galera::19794:~::~:
_gophish::19794:~::~:
user1:$y$9T$3a7r0b9j9f5L9i11mrQfm0$ud70Dj7yVif4nu.yMk32c0KNb101ctE5ut/Mj21wAB:20158:0:99999:7:::
user2:$y$9T$3d0fYMBL6bE9k3p0fdIe5A0$9QLhSWYL3M4ziAk60ZLUBSHgUE50uRyOQiNRLeTXgA:20158:0:99999:7:::
$ echo "hacker:x:1002:1002:Hacker:/home/hacker:/bin/bash" >> /etc/passwd
$ exit

(root@kali)-[/home/kali]
# sudo chmod 640 /etc/shadow
sudo chmod 644 /etc/passwd

(root@kali)-[/home/kali]
# ls -l /etc/shadow /etc/passwd
-rw-r--r-- 1 root root 3536 Mar 11 12:17 /etc/passwd
-rw-r----- 1 root shadow 1633 Mar 11 12:15 /etc/shadow

(root@kali)-[/home/kali]
#
```

Step 9: Verify Correct Permissions

Command:

bash

CopyEdit

ls -l /etc/shadow /etc/passwd

Expected Output:

plaintext

CopyEdit

-rw-r----- 1 root shadow <date> /etc/shadow

-rw-r--r-- 1 root root <date> /etc/passwd

✅ Permissions Fixed!


```
root@kali: /home/kali
File Actions Edit View Help
GNU nano 7.2 /etc/sudoers.tmp
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/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_*"
[ Read 54 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  ^U Undo      ^M Set Mark  ^= To Bracket ^_ Previous
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify   ^/ Go To Line ^E Redo      ^B Copy      ^_ Where Was ^W Next
```

4 Additional Security Hardening

Step 10: Secure sudo Access

Command:

bash

CopyEdit

sudo visudo

What It Does?

- Opens the sudoers configuration file for editing.
- Ensure only **trusted users** have sudo access.

Expected Output:

Opens the **sudo configuration file** (no text output).