Task 1: User & Permission Misconfigurations

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

1️⃣ **Setup: Creating Users & Misconfiguring Permissions**

Step 1: Create Users

Command:

bash

CopyEdit

sudo useradd -m user1

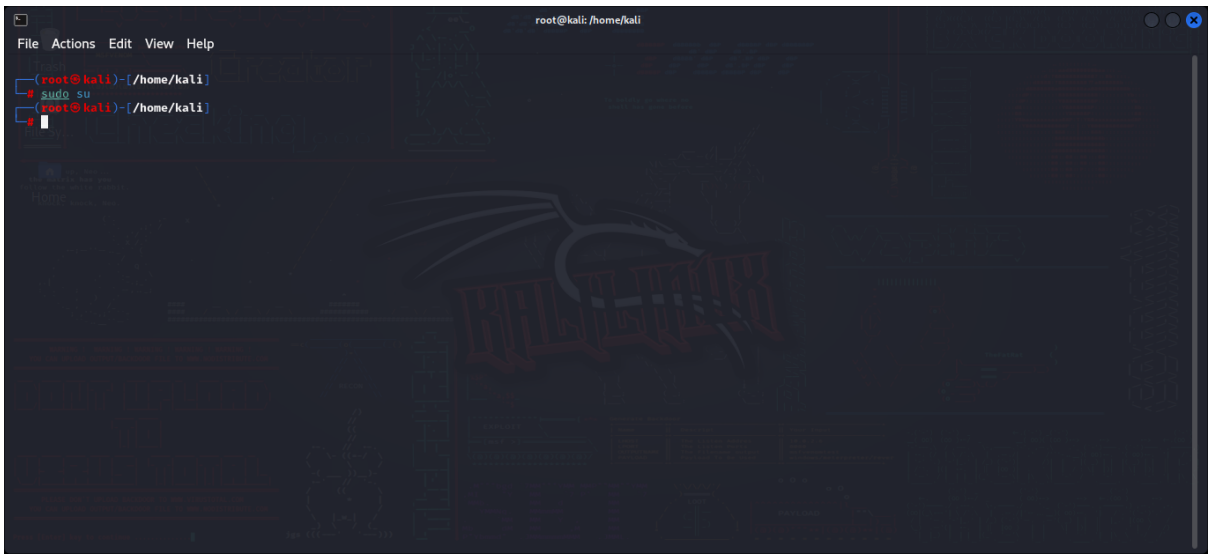
sudo useradd -m user2

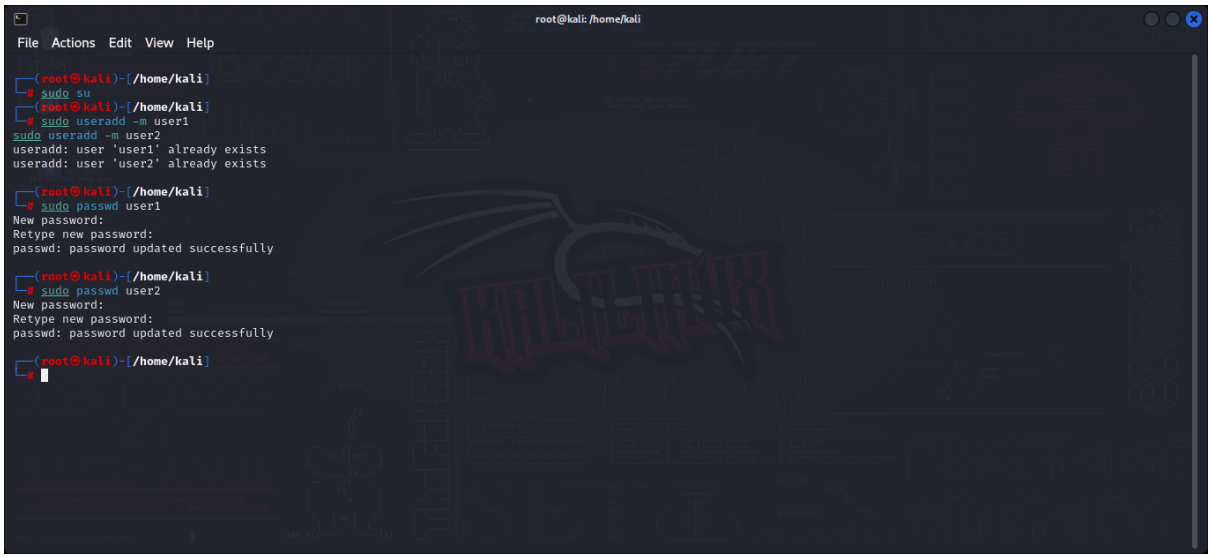
What It Does?

- 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).





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

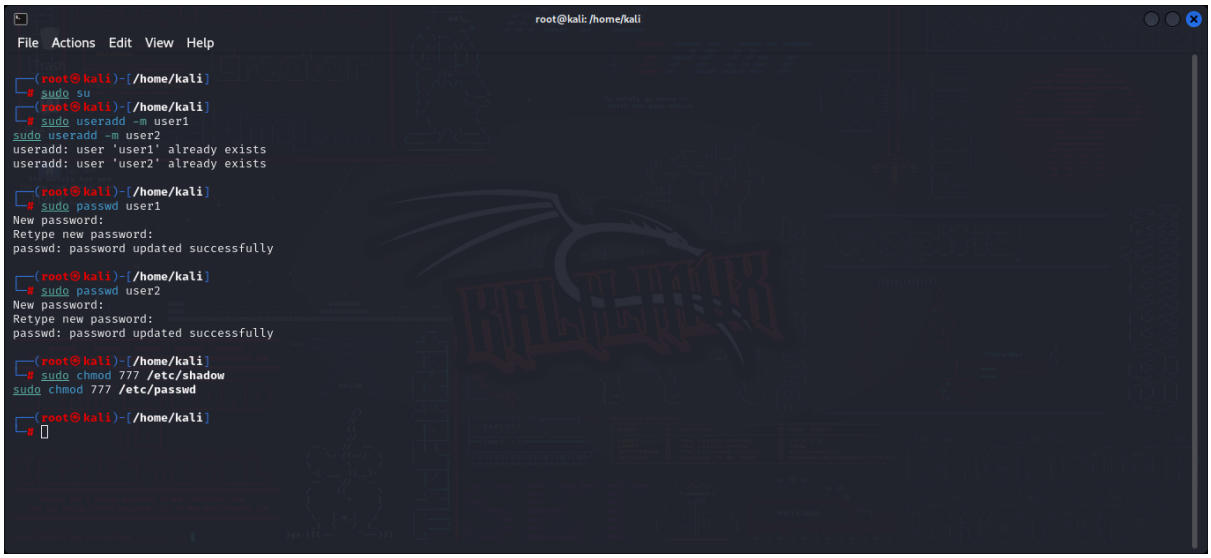
CopyEdit

New password: ******

Retype new password: ******

passwd: password updated successfully

(Repeat the same command for user2.)



Step 3: Assign Incorrect Permissions

Command:

bash

CopyEdit

sudo chmod 777 /etc/shadow

sudo chmod 777 /etc/passwd

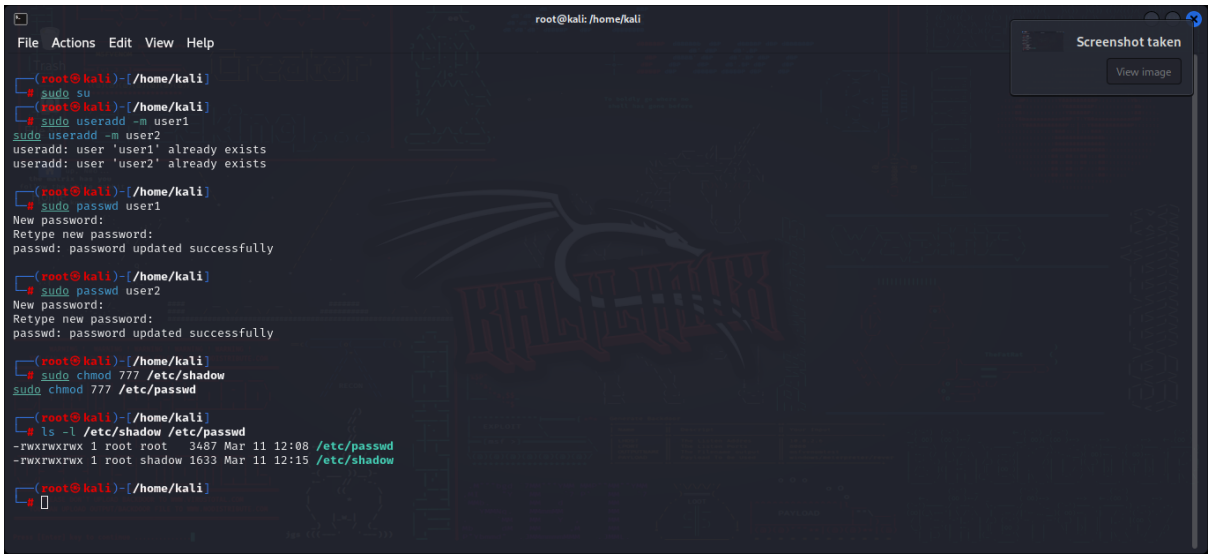
What It Does?

- 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).



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:

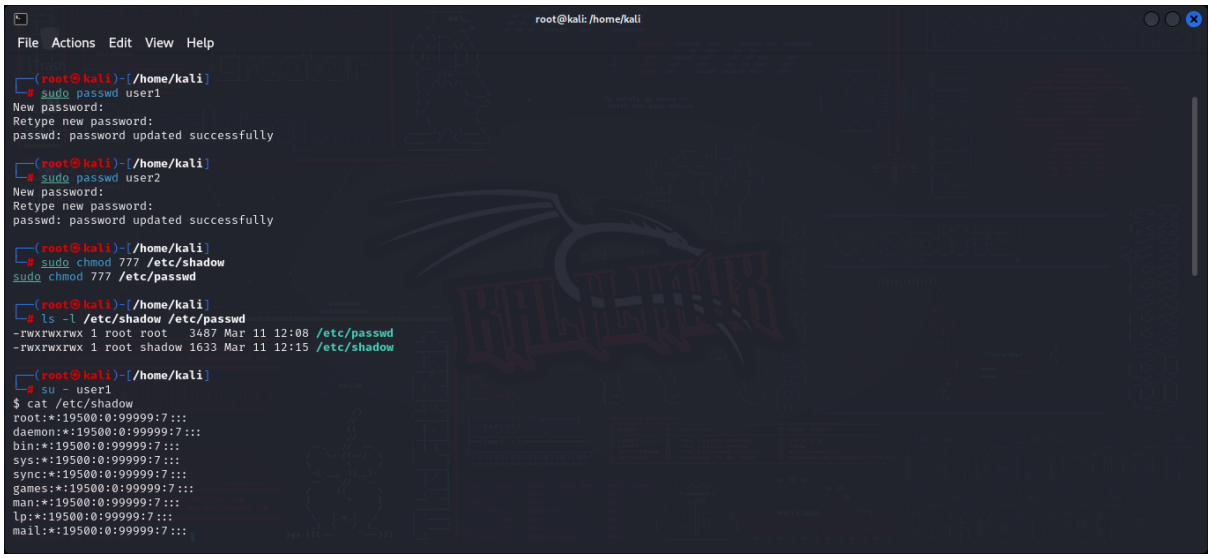
plaintext

CopyEdit

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

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

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



2️⃣ 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!):

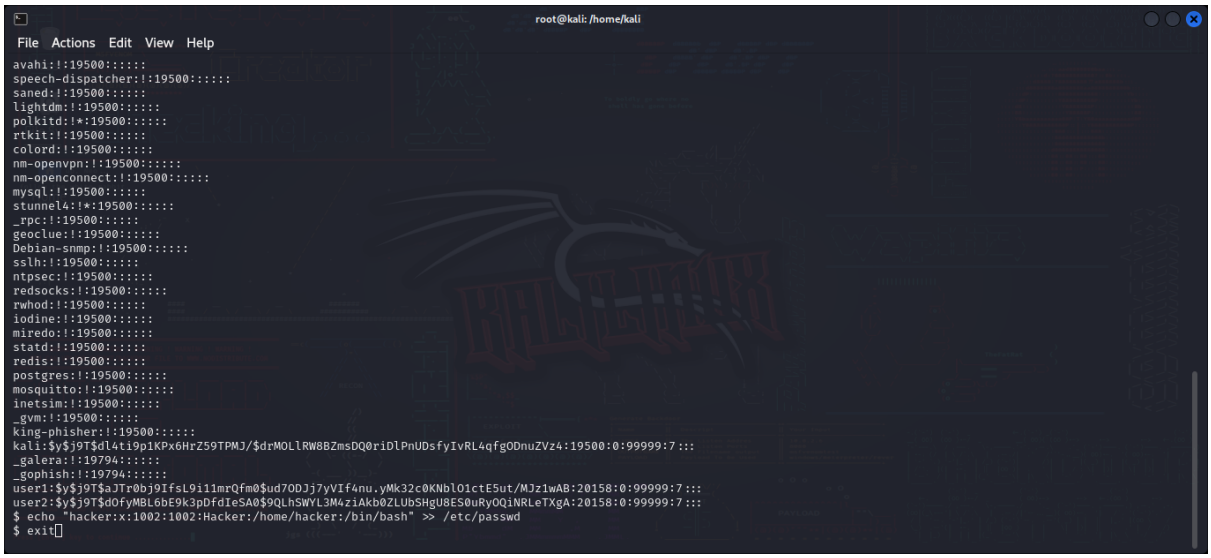
plaintext

CopyEdit

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

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

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



🔴 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!)

3️⃣ Mitigation: Fixing Permission Issues

Step 7: Exit User1 Session

Command:

bash

Copy

Edit

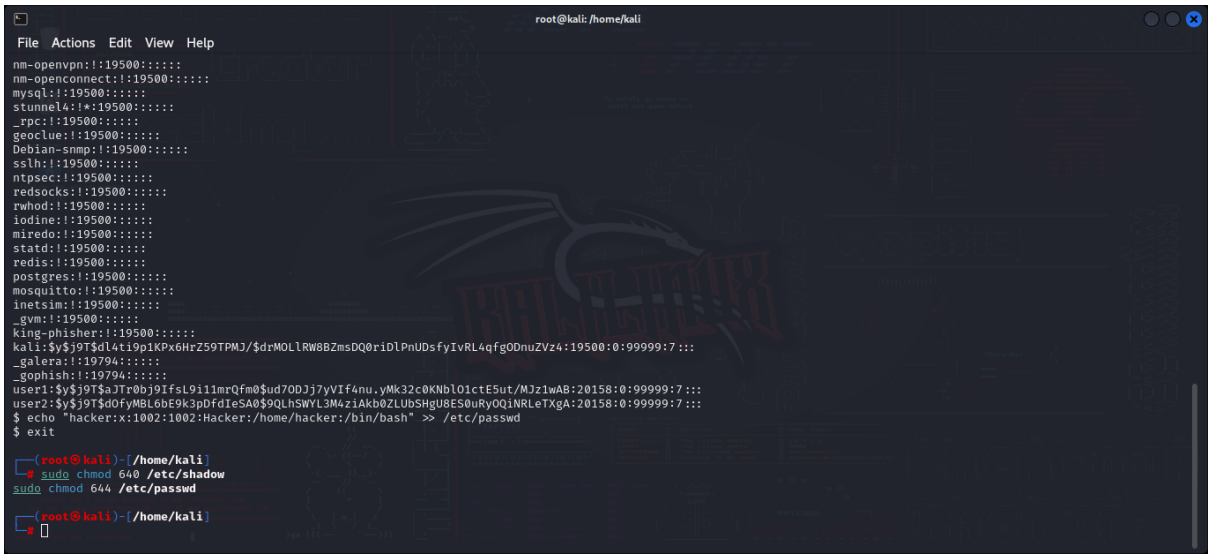
exit

What It Does?

Returns to the root user session.

Expected Output:

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



Step 8: Secure File Permissions

Command:

bash

CopyEdit

sudo chmod 640 /etc/shadow

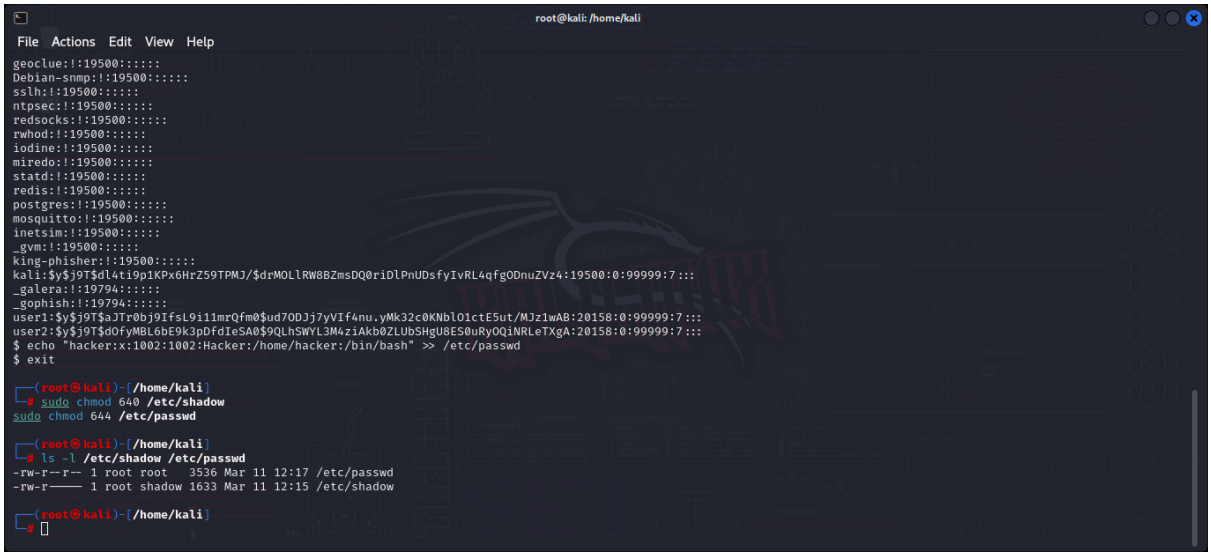
sudo chmod 644 /etc/passwd

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)



Step 9: Verify Correct Permissions

Command:

bash

CopyEdit

ls -l /etc/shadow /etc/passwd

Expected Output:

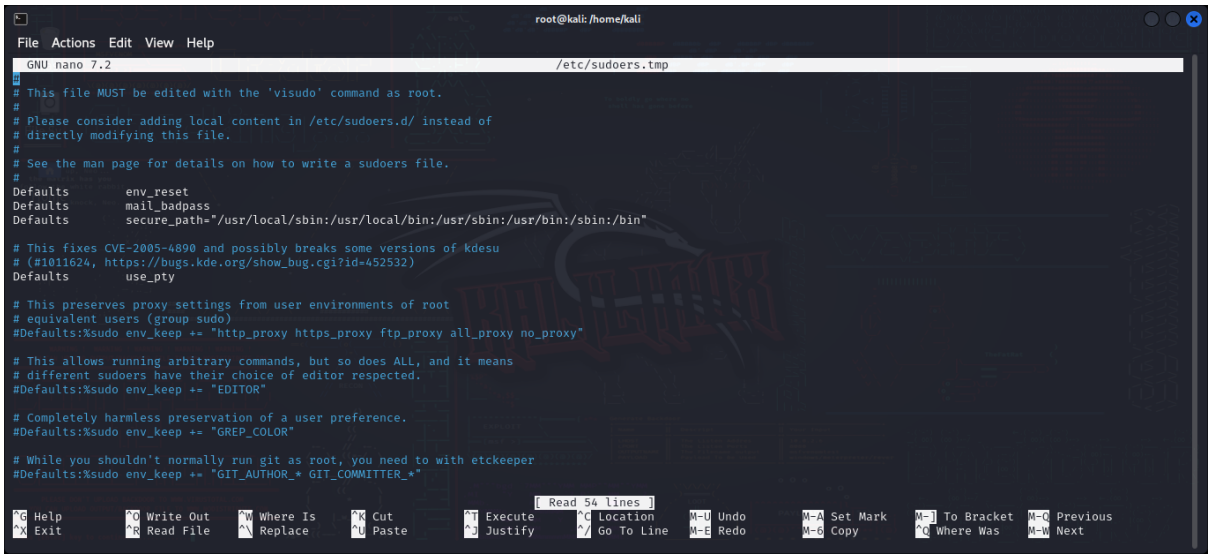
plaintext

CopyEdit

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

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

✅ **Permissions Fixed!**



4️⃣ 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).