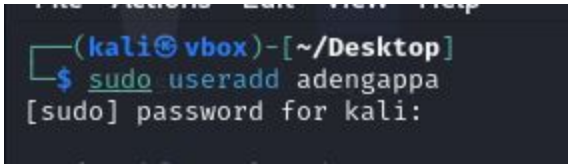
Task 1 :

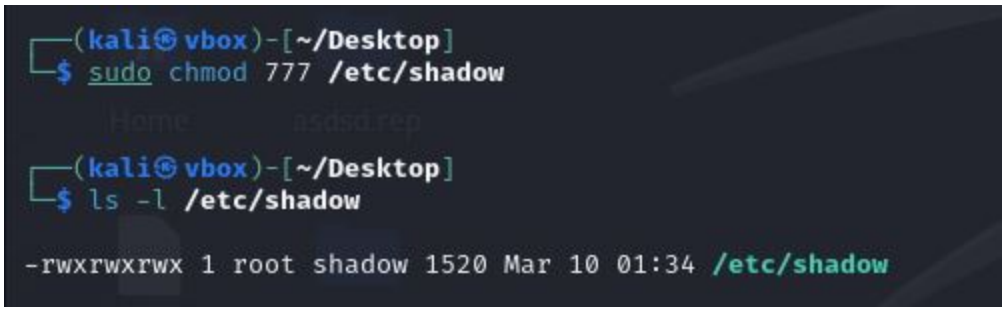
User permission and system misconfigurations ⚙️ :



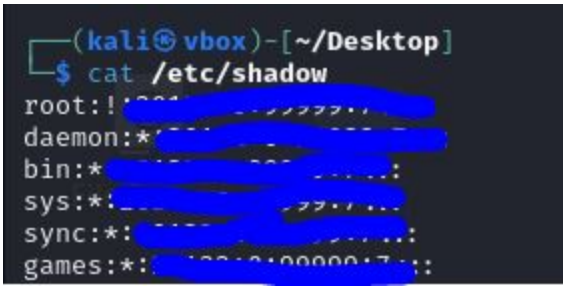
1. First, we create a user named “**adengappa**” using the `sudo useradd <username>` command.

2. We assign the password “**1234**” by using the `echo` command to write it into the password file `chpassword`, with elevated privileges via `sudo`.

3. We examine the permissions of the password file to identify and exploit any misconfigurations.



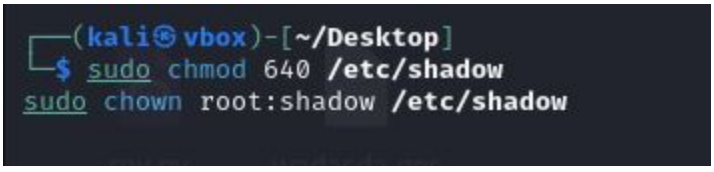
4. We modify the permissions of the shadow file using the `sudo chmod 777` command to grant full access. Then, we verify the updated permissions to confirm the ability to view the file.



5. As observed, we can now view the contents of the `/etc/shadow` file, which contains hashed passwords, even with normal user privileges.

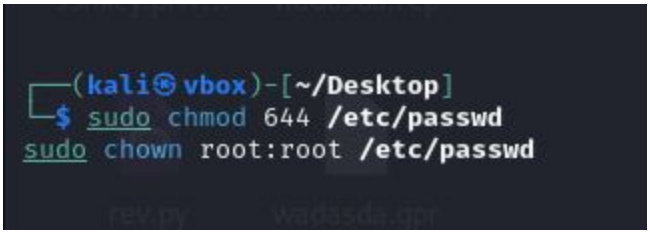
6. We have successfully configured `/etc/shadow` to be accessible by normal users.

Securing permissions 🔒:



1. We secure the password file by setting its permissions to `640` using the `chmod` command. This ensures that only the root user and members of the shadow group

can access it. The root user's password remains viewable only under superuser privileges.



2. We modify the permissions of the `/etc/passwd` file using `sudo chmod 644` and set its ownership to `root:root` with `sudo chown root:root`. This ensures that regular users can read the file but cannot modify it.

3. Finally we use sudo visudo to check permissions.

📌 Summary of Steps

Step	Command	Purpose
Create Users	`sudo useradd user1`	Add new users
Set Passwords	`echo "user1:pass" | sudo chpasswd`	
Break Security	`sudo chmod 777 /etc/shadow`	Make shadow file world-readable (BAD)
Exploit	`su user1 && cat /etc/shadow`	Access passwords as normal user
Fix Permissions	`sudo chmod 640 /etc/shadow`	Secure shadow file
Secure `/etc/passwd`	`sudo chmod 644 /etc/passwd`	Prevent unauthorized edits
Fix `sudo` Privileges	`sudo visudo`	Restrict `sudo` access