Proof Of Concept

Linux Security- Exploitation & Hardening

Task 1: User & Permission Misconfigurations

1. Executive Summary

This PoC demonstrates how incorrect permissions on sensitive system files (e.g., /etc/shadow) can allow low-privileged users to access critical information. The task involves creating users, misconfiguring file permissions, exploiting the misconfiguration, and then mitigating the issue by restoring proper permissions and ownership.

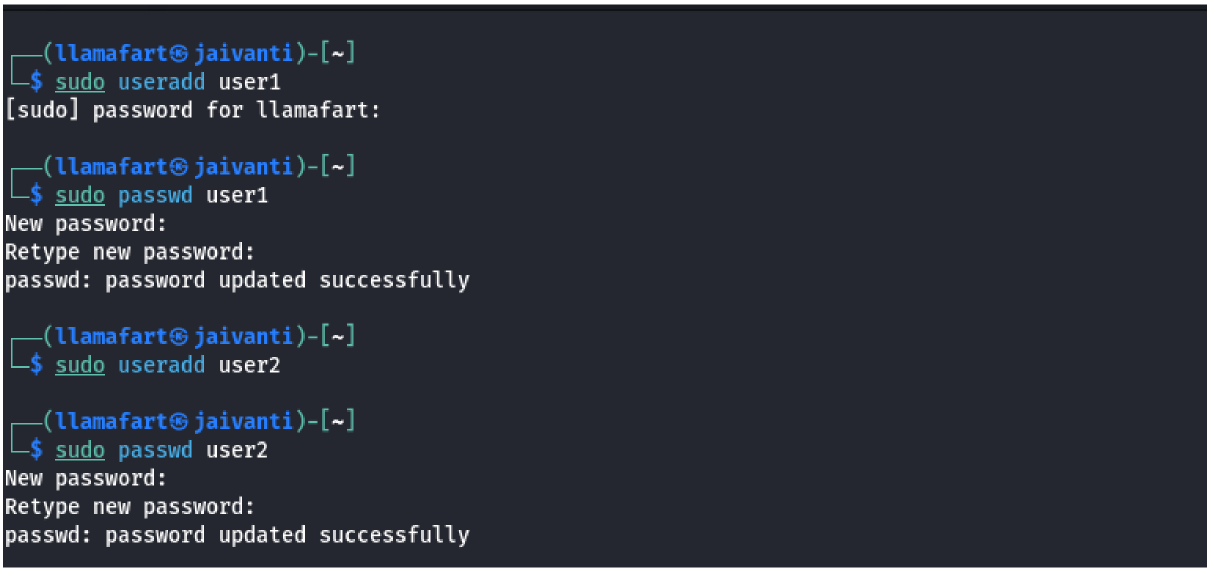
2. Objectives

- Setup: Create users and assign incorrect permissions to sensitive files.
- Exploit: Demonstrate how a low-privileged user can access sensitive files.
- Mitigation: Fix the permission issues and prevent unauthorized access.

3. Setup

3.1. Create Users

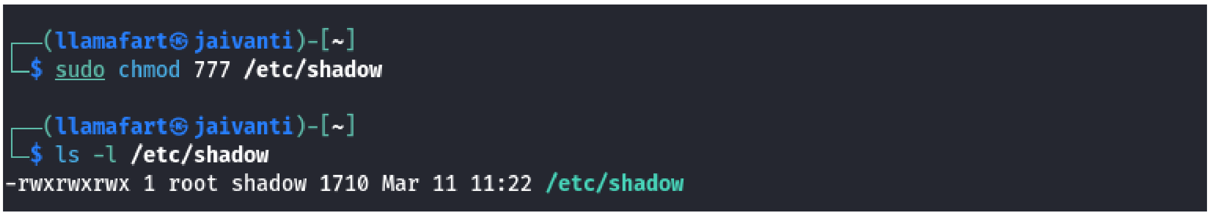
Two users, user1 and user2, were created using the useradd command, and passwords were assigned using the passwd command.



```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo useradd user1
[sudo] password for llamafart:

┌──(llamafart㉿jaivanti)-[~]
└─$ sudo passwd user1
New password:
Retype new password:
passwd: password updated successfully

┌──(llamafart㉿jaivanti)-[~]
└─$ sudo useradd user2

┌──(llamafart㉿jaivanti)-[~]
└─$ sudo passwd user2
New password:
Retype new password:
passwd: password updated successfully
```

3.2. Assign Incorrect Permissions

The permissions for the /etc/shadow file were changed to 777 (read, write, and execute for everyone), making it accessible to all users.

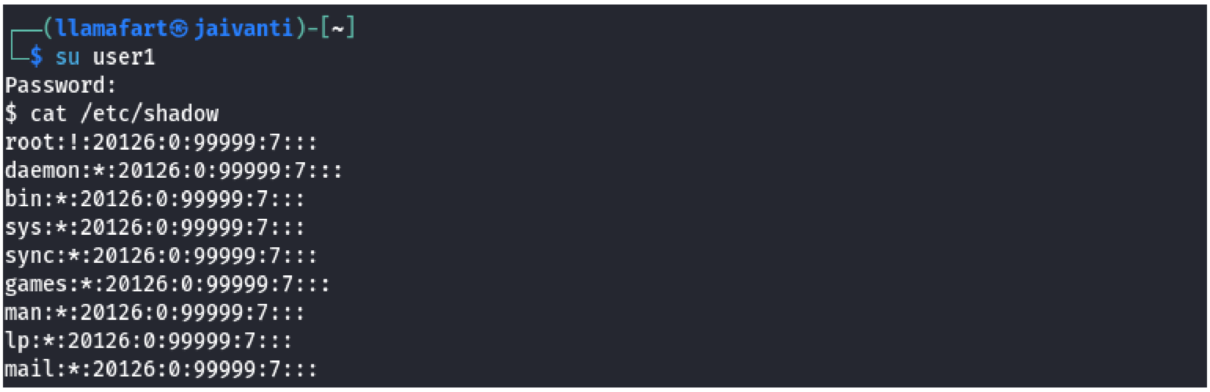


```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo chmod 777 /etc/shadow

┌──(llamafart㉿jaivanti)-[~]
└─$ ls -l /etc/shadow
-rwxrwxrwx 1 root shadow 1710 Mar 11 11:22 /etc/shadow
```

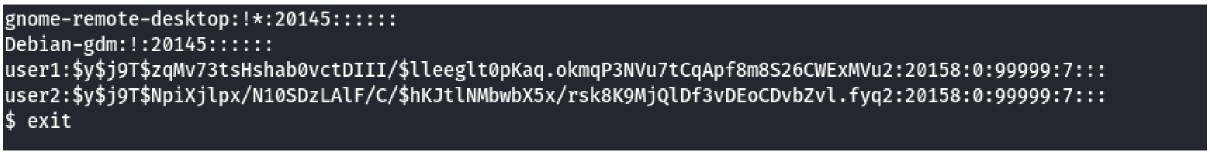
4. Exploitation

4.1. Access Sensitive File as Low-Privileged User

The user user1 was able to switch to their account using the su command and access the /etc/shadow file, which contains encrypted password hashes.



```
┌──(llamafart㉿jaivanti)-[~]
└─$ su user1
Password:
$ cat /etc/shadow
root:!:20126:0:99999:7:::
daemon:*:20126:0:99999:7:::
bin:*:20126:0:99999:7:::
sys:*:20126:0:99999:7:::
sync:*:20126:0:99999:7:::
games:*:20126:0:99999:7:::
man:*:20126:0:99999:7:::
lp:*:20126:0:99999:7:::
mail:*:20126:0:99999:7:::
```

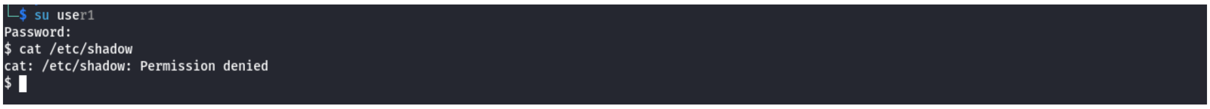


```
gnome-remote-desktop:!*:20145::::::
Debian-gdm:!:20145::::::
user1:$y$j9T$zqMv73tsHshab0vctDIII/$lleeglt0pKaq.okmqP3NVu7tCqApf8m8S26CWExMVu2:20158:0:99999:7:::
user2:$y$j9T$NpiXjlpx/N10SDzLAlF/C/$hKJtlNMbwbX5x/rsk8K9MjQlDf3vDEoCDvbZvl.fyq2:20158:0:99999:7:::
$ exit
```

5. Mitigation

5.1. Restore Proper Permissions

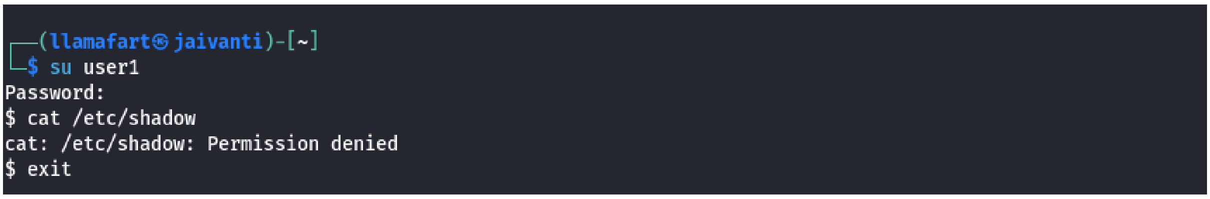
The permissions for /etc/shadow were corrected to 640, and ownership was restored to root:shadow.



```
└─$ su user1
Password:
$ cat /etc/shadow
cat: /etc/shadow: Permission denied
$ █
```

5.2. Verify Mitigation

After fixing the permissions, the user user1 was unable to access
the /etc/shadow file.



```
┌──(llamafart㉿jaivanti)-[~]
└─$ su user1
Password:
$ cat /etc/shadow
cat: /etc/shadow: Permission denied
$ exit
```

6. Recommendations

- Regularly audit file permissions on critical system files.
- Use tools like auditd or tripwire to monitor changes to sensitive files.
- Educate system administrators about the importance of proper file permissions.

Conclusion

This PoC highlights the security risks associated with improper file permissions and demonstrates how to exploit and mitigate such misconfigurations. By following best practices for file permissions and regularly auditing system configurations, administrators can significantly reduce the risk of unauthorized access to sensitive information.