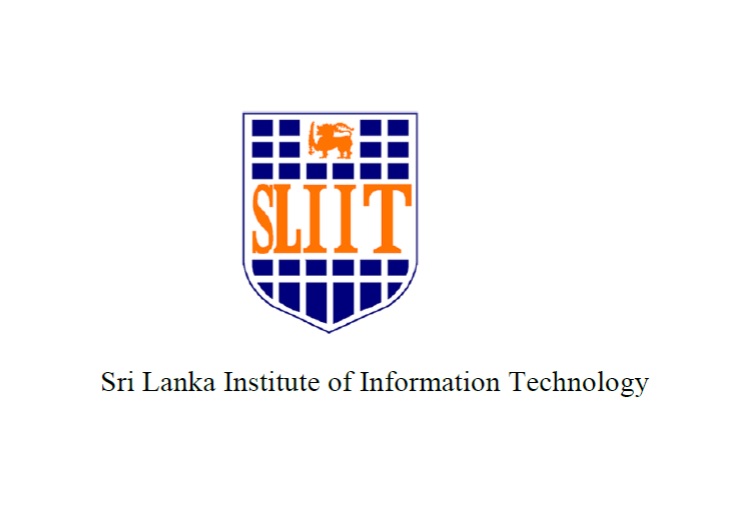
sudo 1.8.27 - Security Bypass

(CVE-2019-14287)

****

G.P.H. Shihani Tharuka

IT19012766

(2 Year 1 Semester)

## Introduction 03

## Referenced on who found the vulnerability 04

## Damage that is can cause 05

## Privilege escalation 06

## Exploit (with screenshots) 07

## 1.method one

## 2.method two

## CONCLUSION 16

## References 16

INTRODUCTION

In this report, I’ll explain in detail on the whole process I went through to complete this security bypass vulnerability exploitation of the system & network programming Assignment for 2nd Year 1 semester Cyber Security students.

I'll be explaining the whole process step by step that I did to exploit this vulnerability, what are the Errors I got in middle of the exploitation and how I solve those things. In the final page, I will be referencing all the resources I used.

In computer security, a vulnerability is a weakness which can be exploited by a threat actor, such as an attacker, to perform unauthorized actions within a computer system. To exploit a vulnerability, an attacker must have at least one applicable tool or technique that can connect to a system weakness.

Security bypass vulnerability, it is mainly based on sudo implemented running commands with random user ID. This security fault could enable a malicious user to execute random commands as root user even in cases where the root access is disallowed.

Sudo stands for "superuser do," is a system command that allows a user to run applications or commands with the privileges of a different user without switching environments—most often, for running commands as the root user.

If a sudoers entry is written to allow the attacker to run a command as any user except root, this fault can be used by the attacker to bypass that control. sudoers file contains the information regarding privileges for different users and groups of our server. It is one of the most important files in the system and needs to be handled with care.

## This security issue discovered by ***Joe Vennix*** of Apple Information Security.

## The title of this exploit is ***sudo1.8.27 – Security Bypass***

## [Common vulnerabilities and exposures](http://cve.mitre.org/) (CVE) ID: ***2019-14287***

## Date : 2019-10-15

## Version : sudo <1.2.28

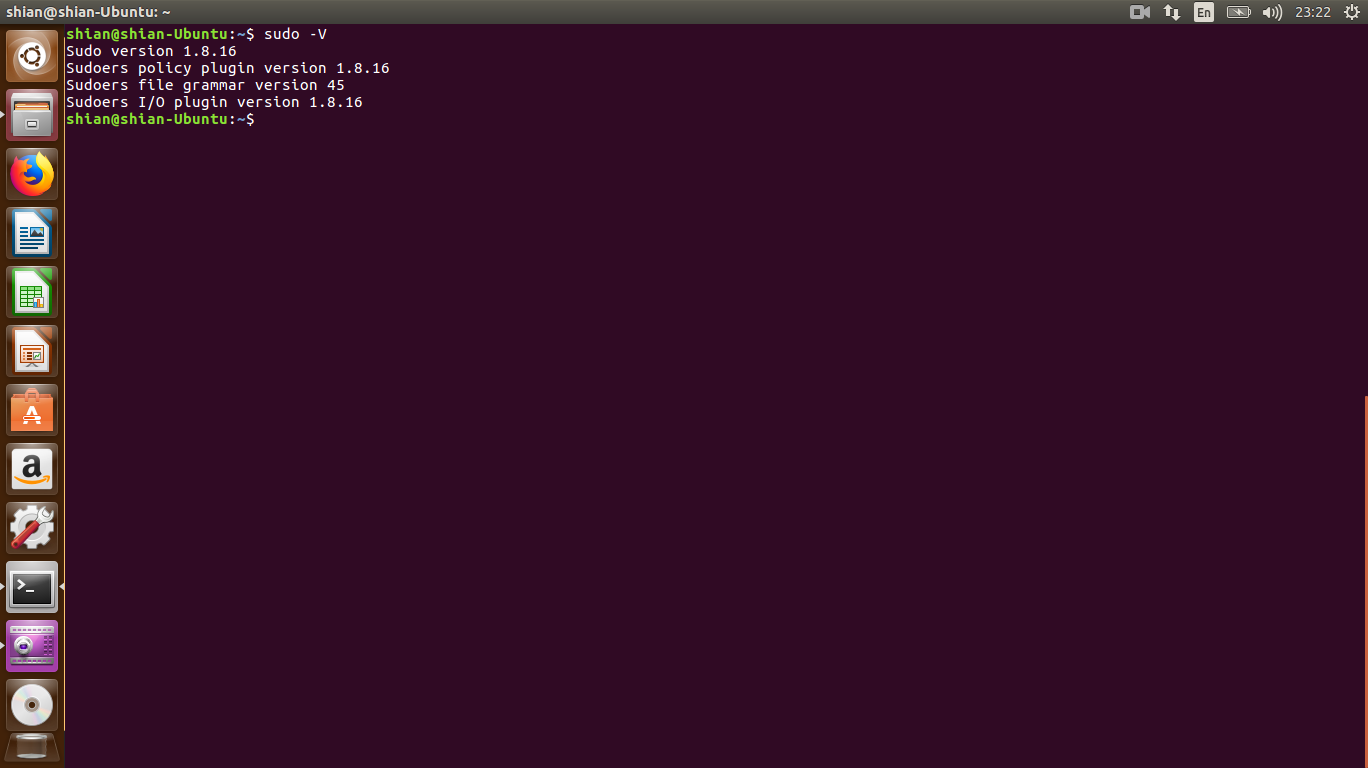
## Platform : Linux

The error might be exploited depends on specific privileges granted in the ***/etc/sudoers*** (Permissions configuration for the sudo command) file. A rule that allows a user to edit files as any user except root, for example, would actually allow that user to edit files as root as well. In this case, the fault could lead to very serious problems.

For a user to exploit the fault, **a user** needs to be assigned privileges in the ***/etc/sudoers***file that allow that user to run commands as some other users, and the error is limited to the command privileges that are assigned in this way.

This problem affects versions prior to 1.8.28.

### Before exploiting the fault, you need to check your sudo version, using this command:

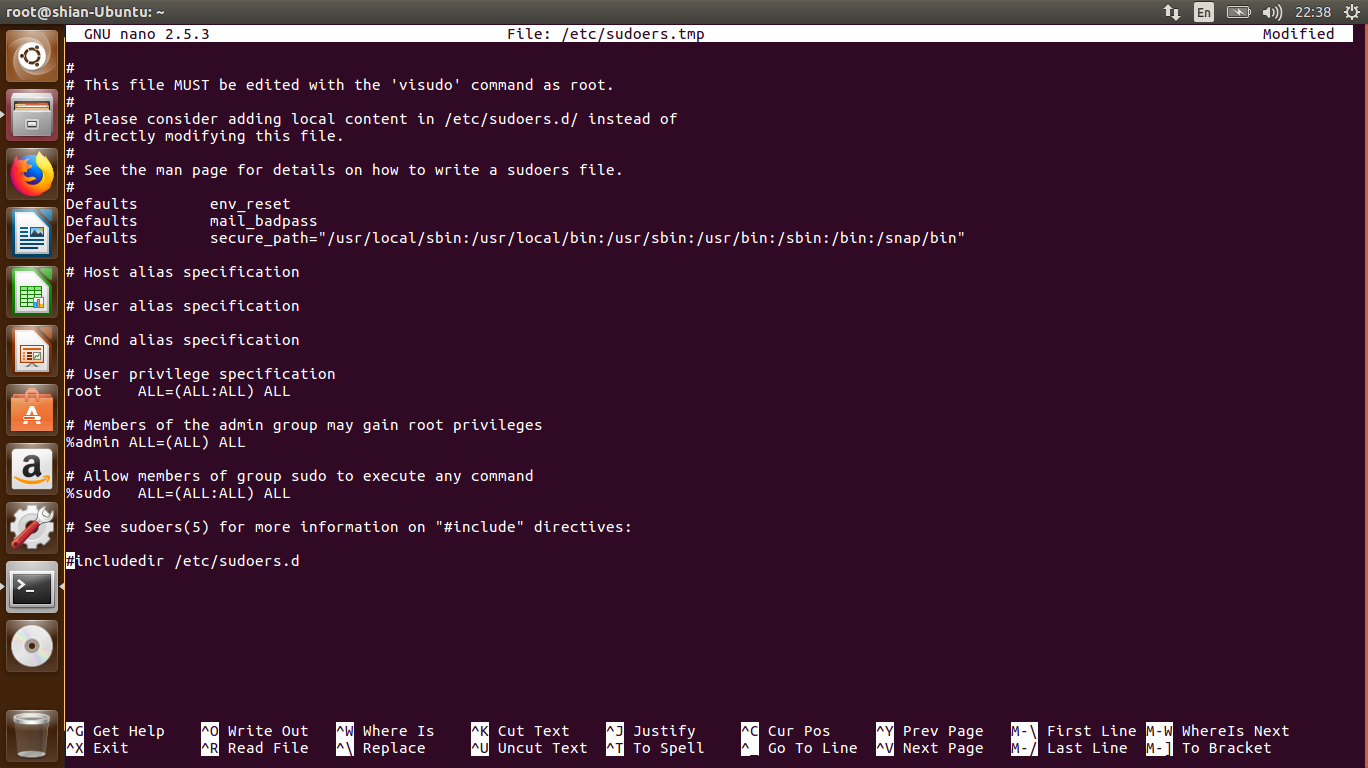


If sudo version less than 1.8.27, then you can exploit. In my case it’s 1.8.16

Updating sudo to version 1.8.28 should address the problem, and Linux admins are encouraged to do so as soon as possible.

**The Damage that it can cause**

The risk that it can cause is that any user who has been given the ability to run even a single command as an arbitrary user may be able to escape the restrictions and run that command as root – even if the specified privilege is written to disallow running the command as root.



**Privilege Escalation**

The exploit type is a ***privilege Escalation***. Privilege escalation is the act of exploiting a bug, design flaw or configuration oversight in an operating system or software application to gain elevated access to resources that are normally protected from an application or user.

The sudoers policy plugin determines a user’s sudo privileges. It is the default sudo policy plugin. The policy is driven by the /etc/sudoers file or, optionally in LDAP.

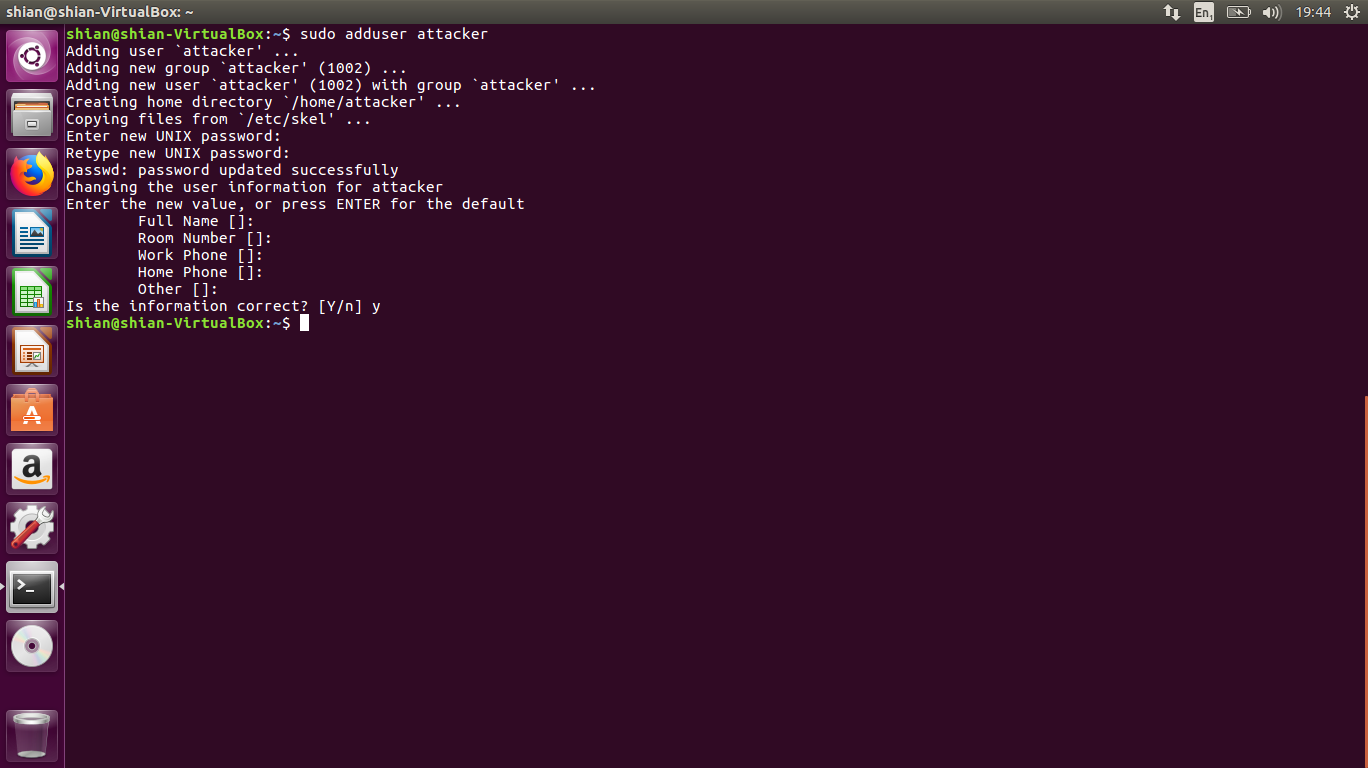
A close up of a logo

Description automatically generated

**Exploit**

**Method 01:**

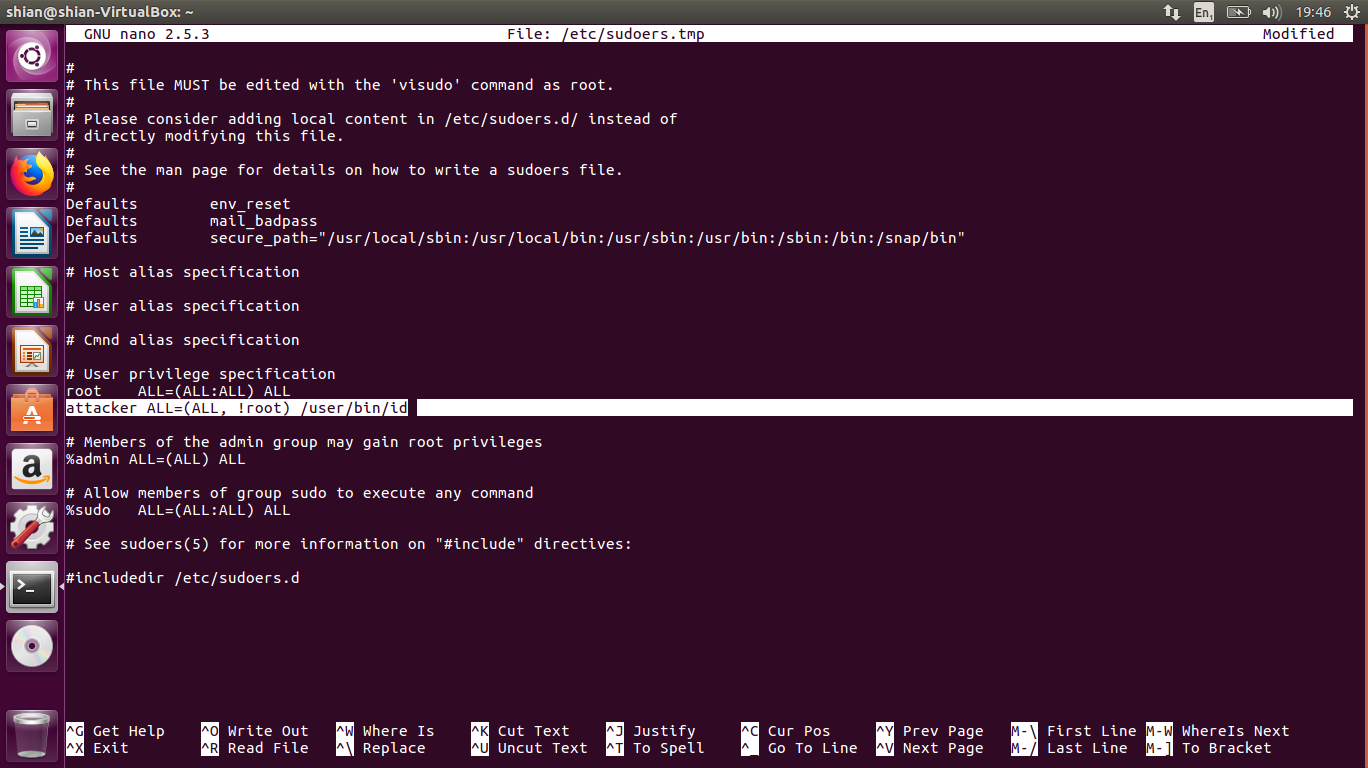
* After checking sudo version, I have added another user called attacker with a suitable password



* Then go to the visudo edits the sudoers file, which defines the users and groups with administrator rights.



File: /etc/sudoers.tmp



Here, "**root ALL=(ALL:ALL) ALL**" states that the user **root**, logged into any [hostname](https://www.computerhope.com/jargon/h/hostname.htm), may run, as any user or group, any command. The general form of this directive is:

*user* *hostname***=(***runas-user***:***runas-group***)** *command*

The special word **ALL** may be used for any of these values, and means that any are allowed.

If *user* begins with a **%**, it's interpreted as the name of a group, and the directive applies to all users in that group. So the line "**%admin ALL=(ALL) ALL**" allows any user belonging to the group **admin** to run any command as any user or group. The same permissions are defined for members of group **sudo**, three lines later.

By default on most Linux distributions, the ALL keyword in Run As specification in /etc/sudoers file, as shown in the screenshot, allows all users in the admin or sudo groups to run any command as any valid user on the system.

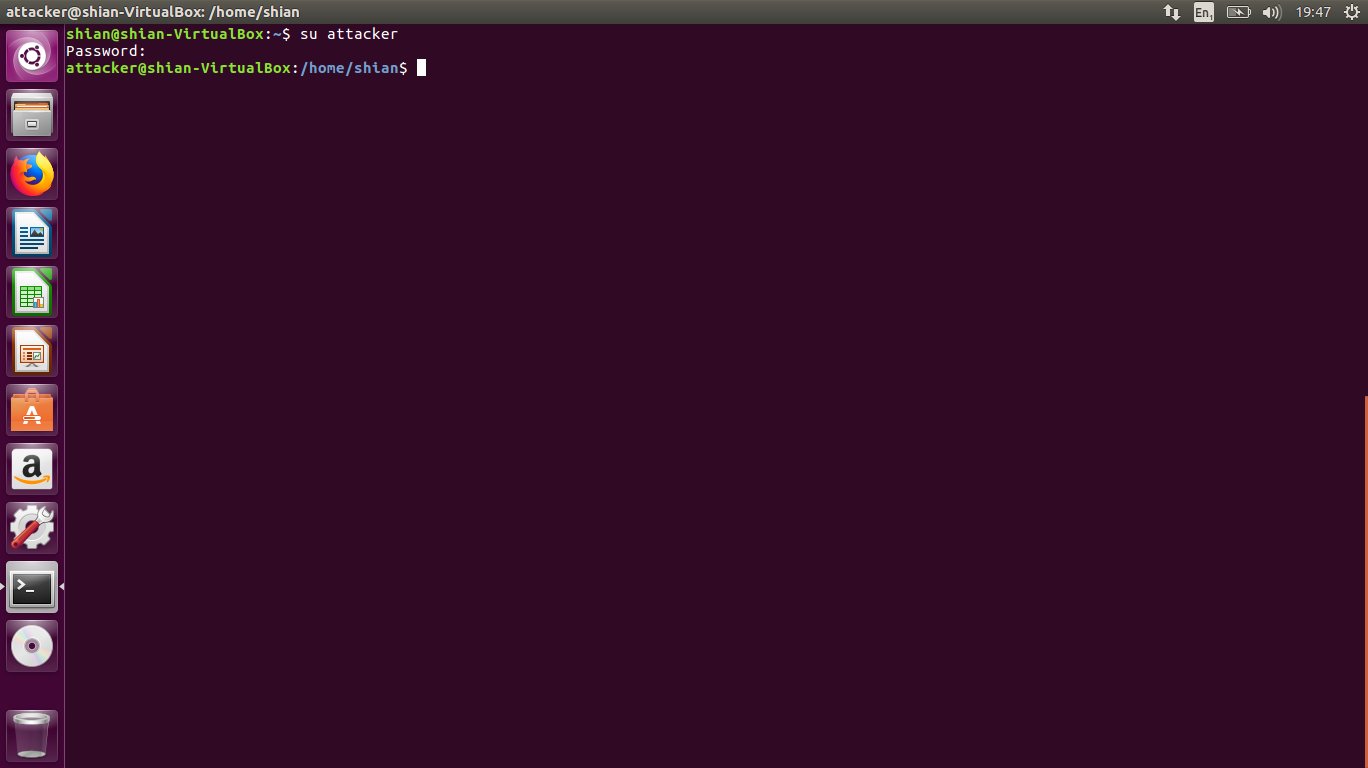
* In my case I have Changed sudoers file as below,

attacker ALL=(ALL, !root) /user/bin/id

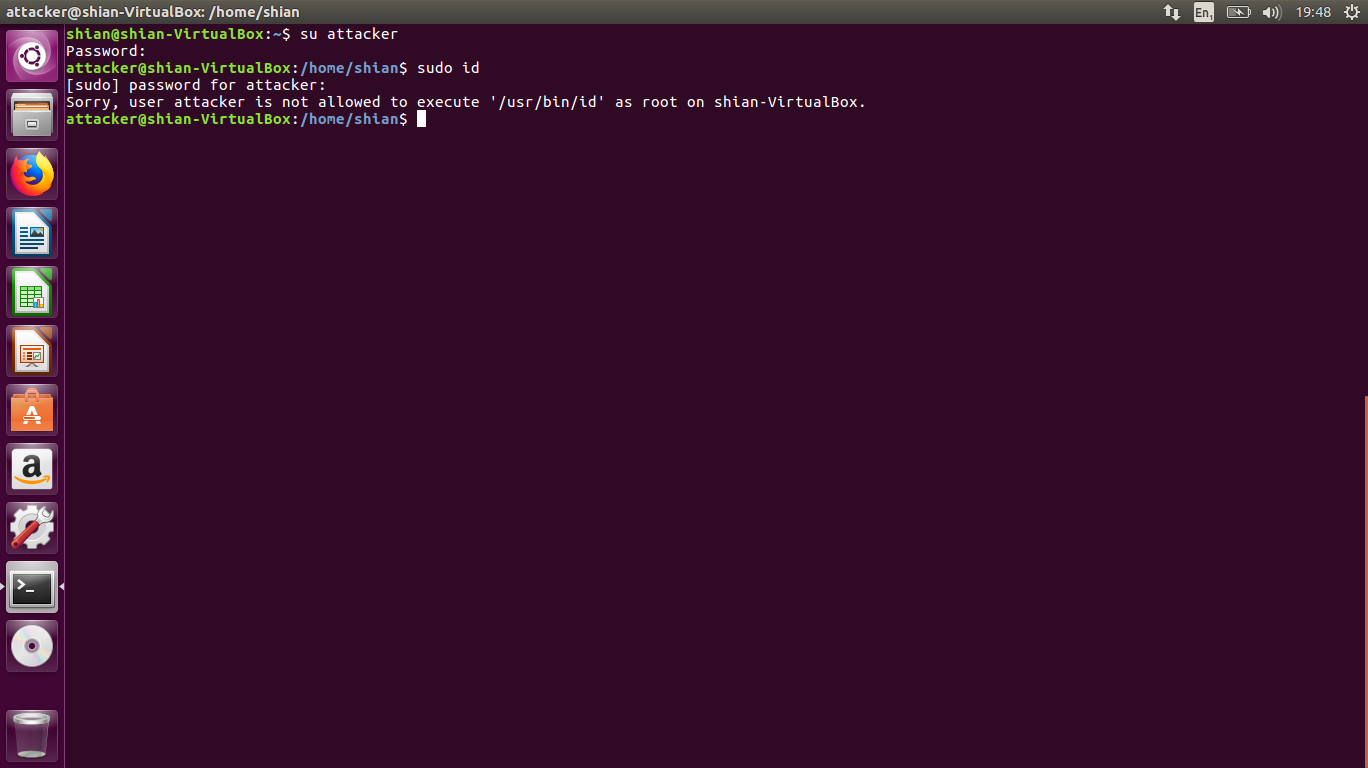
(!root means Not root)

Then saved the file and exit the text editor launched by visudo.

* Switch user to attacker entering attacker’s user password



* Next I have tried to know my id, but it didn’t allowed me. So it means still I can’t get root privileges. It says that I can’t execute those commands which only root can execute.

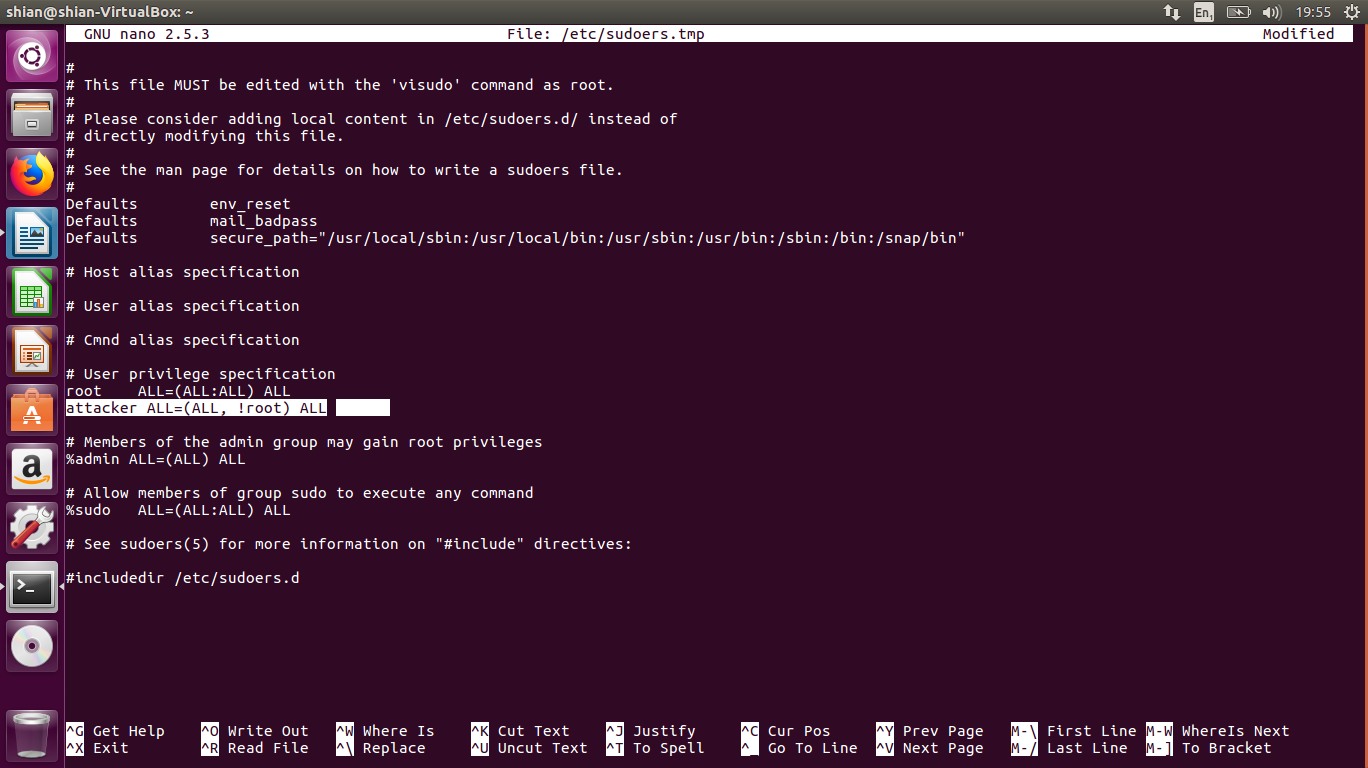


* After that , again I went to the sudoers file.

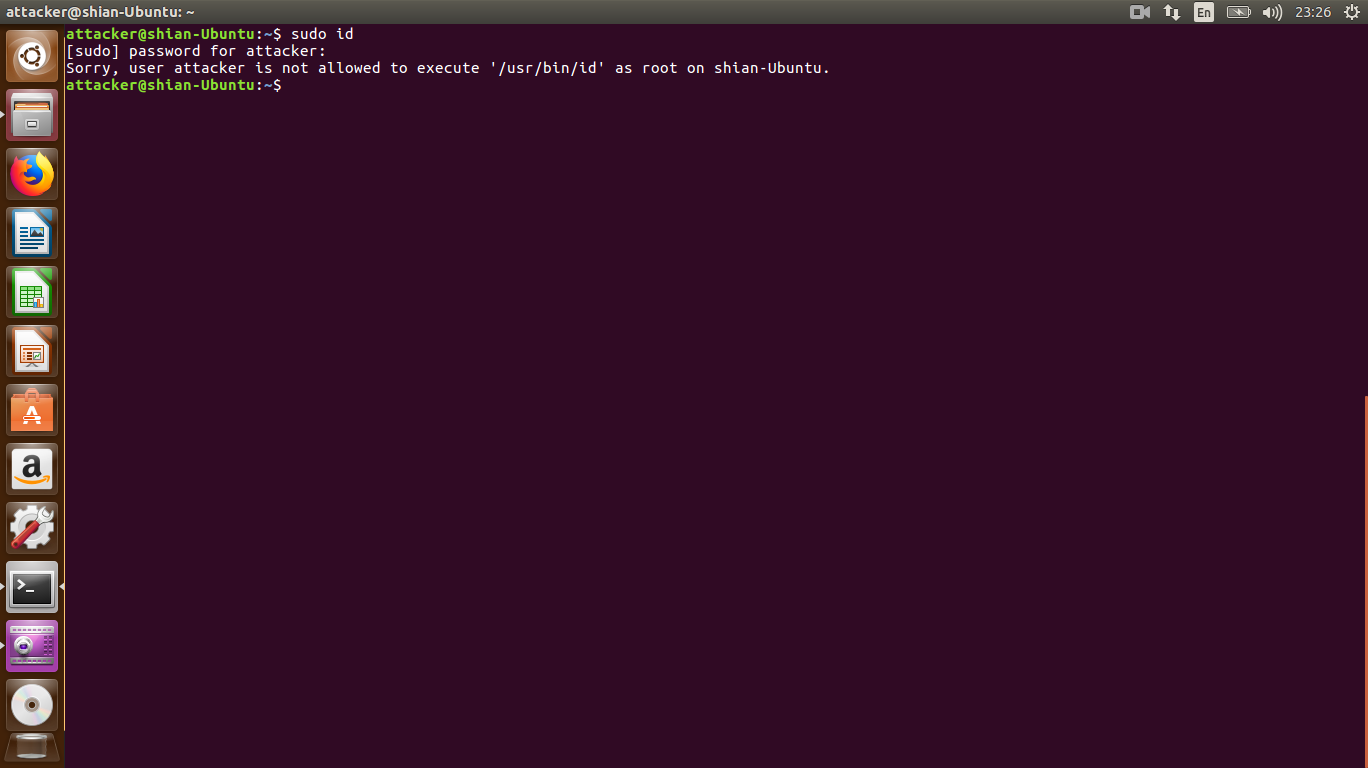
Changed it to,

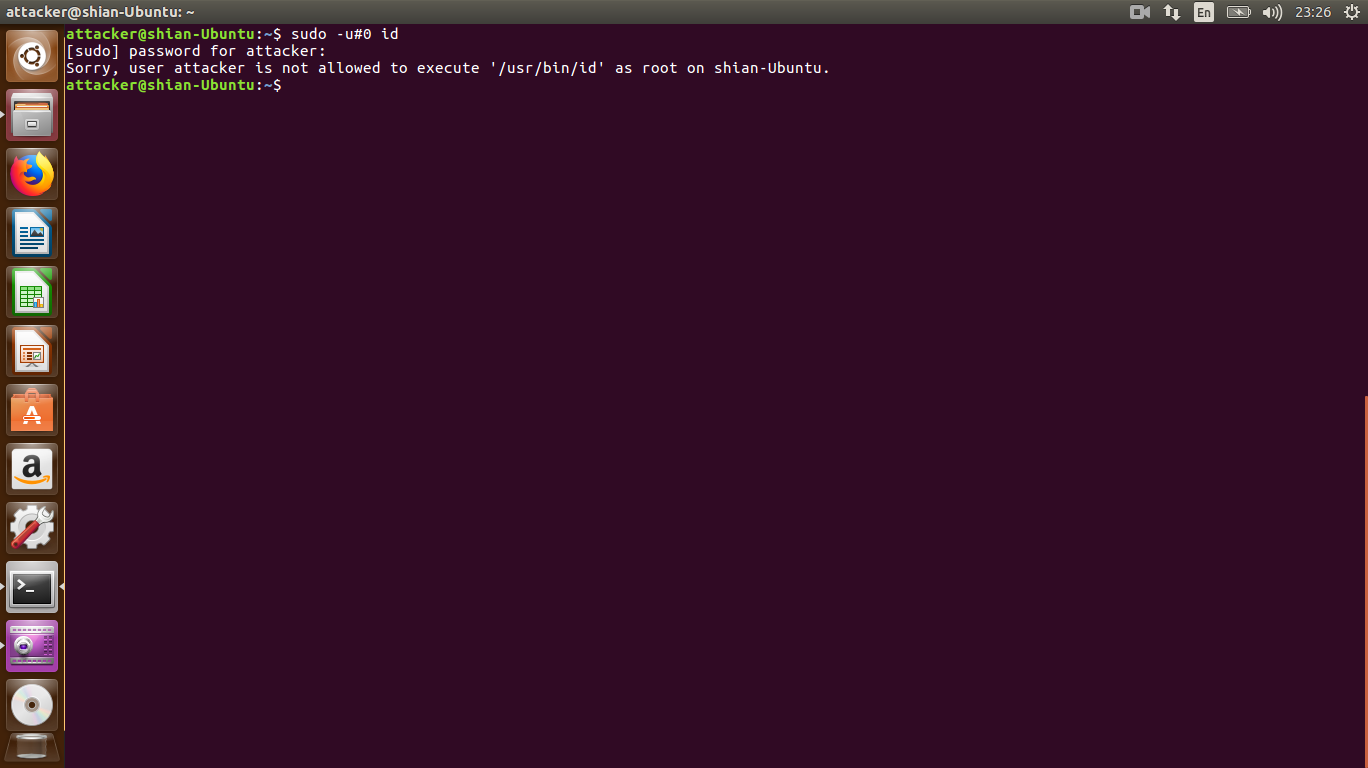
attacker ALL=(ALL, !root) ALL

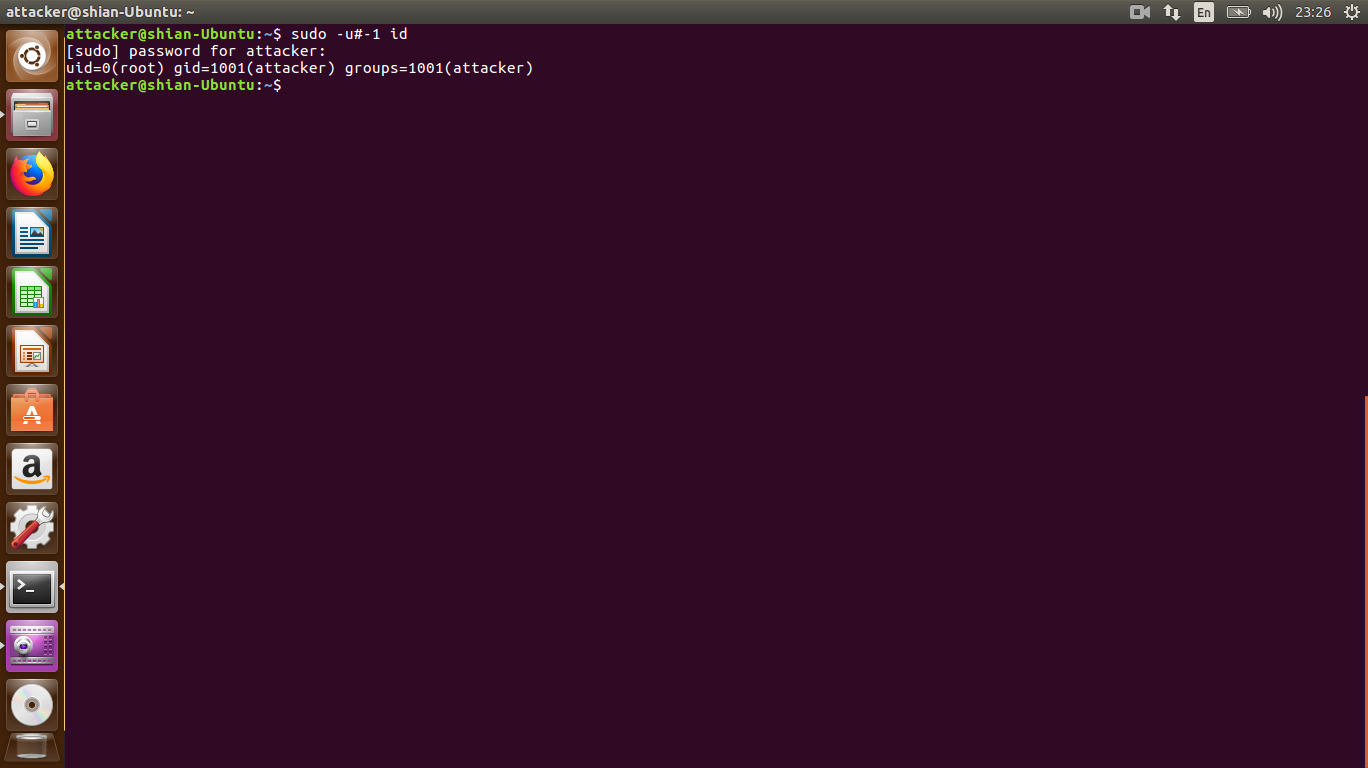
File: /etc/sudoers.tmp



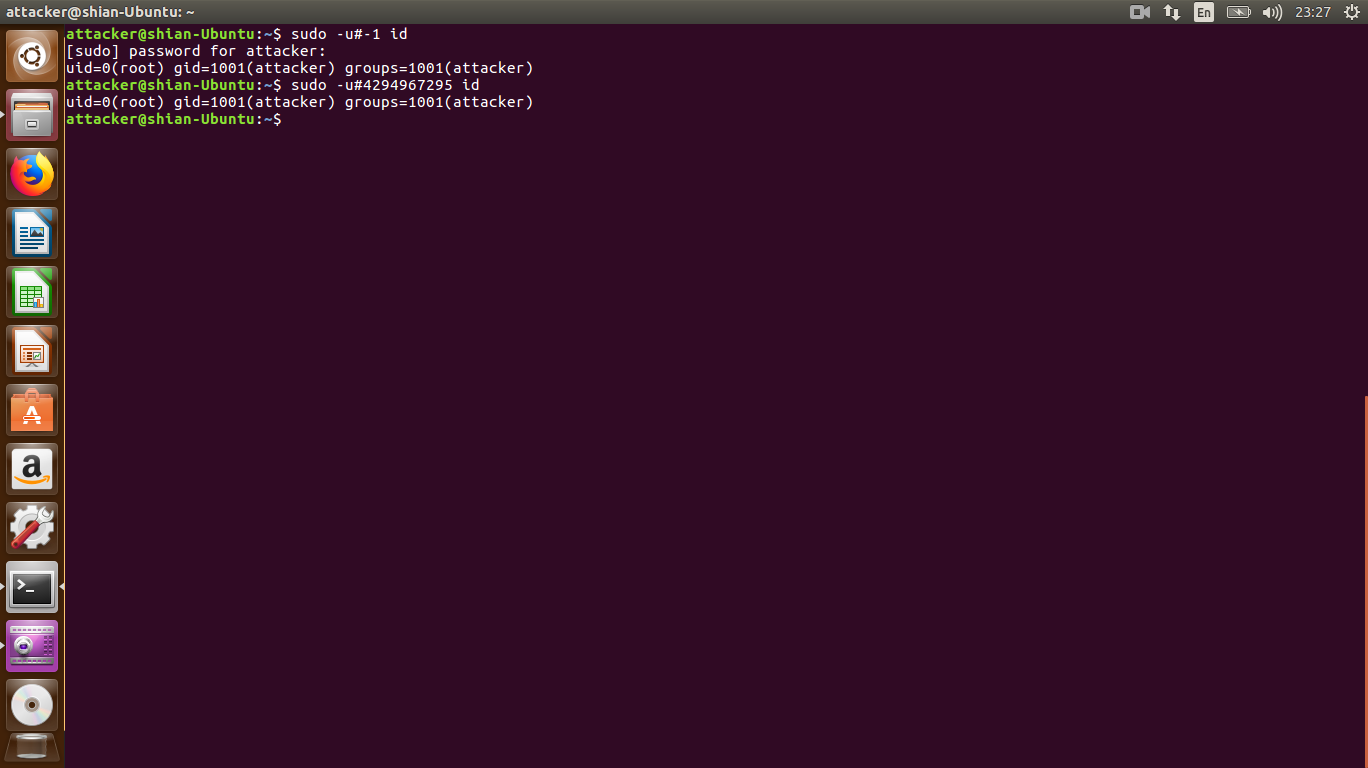
* Then I run id command, it also gave the same error message.



* I have tried to use 0 because root’s uid=0. But I got that error message again.



* The error can be exploited by an attacker to run commands as root by specifying the user ID "-1". The response of "1" demonstrates that the command is being run as root (showing root's user ID).



* The same thing happens in entering code “4294967295” to “-1” position.
* Sudo doesn't check for the existence of the specified user id and executes the with arbitrary user id with the sudo privilege -u#-1 returns as 0 which is root's id and “/bin/bash” is executed with root permission.
* Then I have created “openMe.txt” file into root directory.

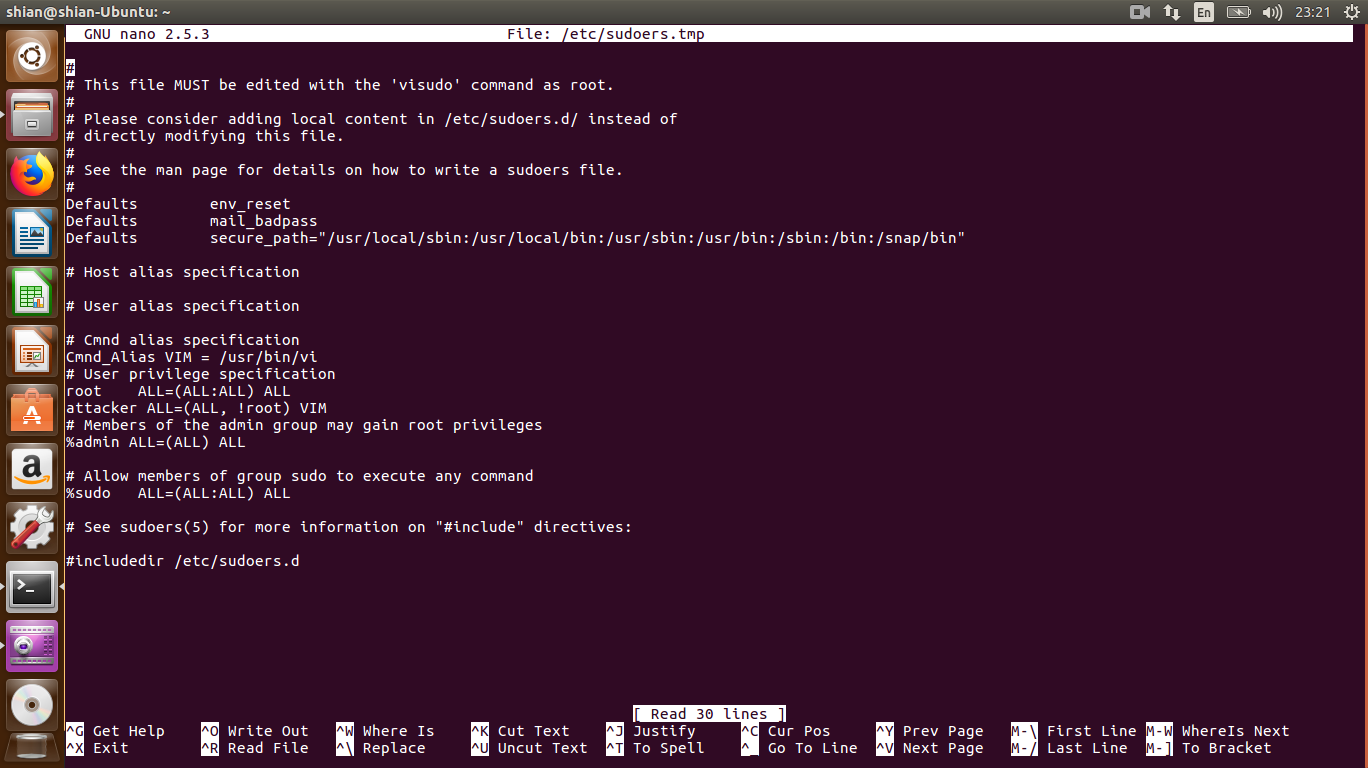


The first method of exploiting this vulnerability was successful.

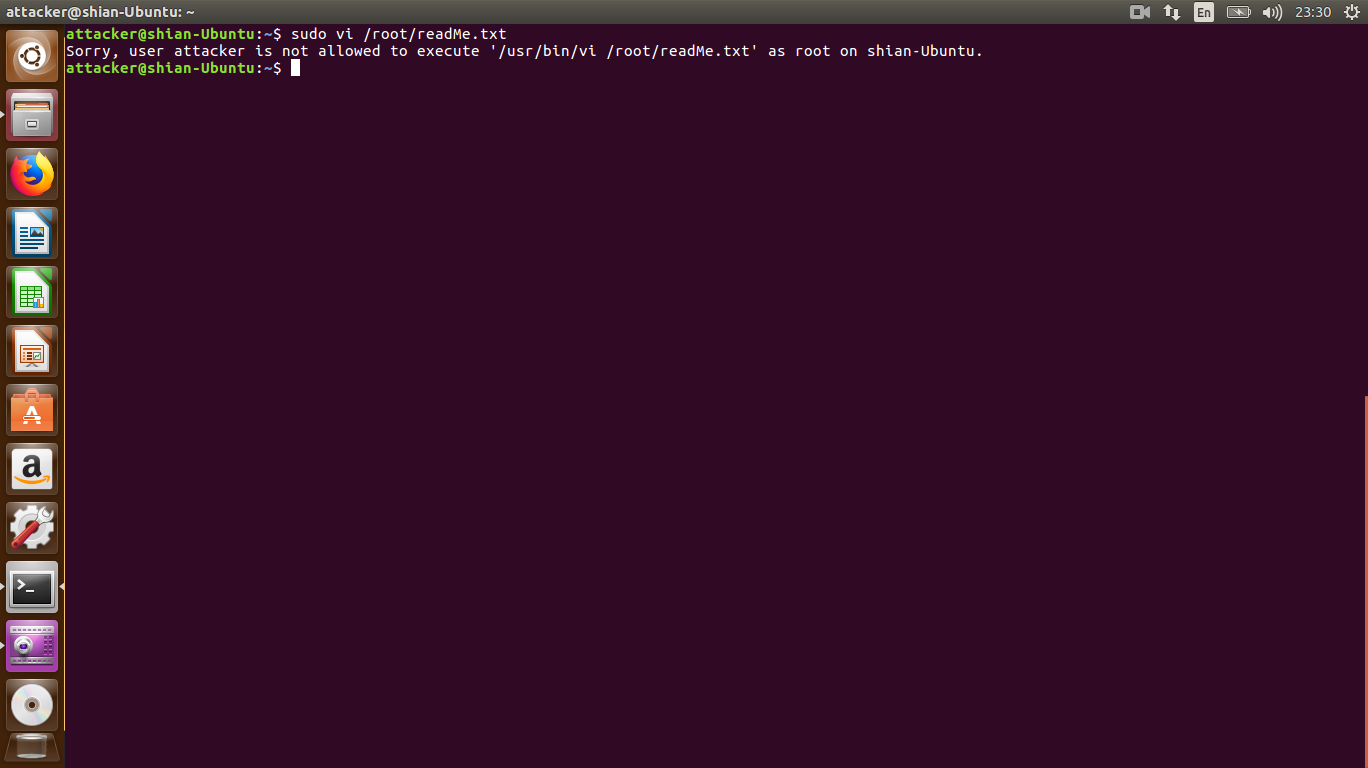
Lets see method number two.

**Method 02:**

* First I went to the visudo file and it edit like this,



* Then I have switched to the attacker tried to create a file called “readMe.txt” in root directory. But it didn’t allow me. Because I didn’t specify root’s ID.



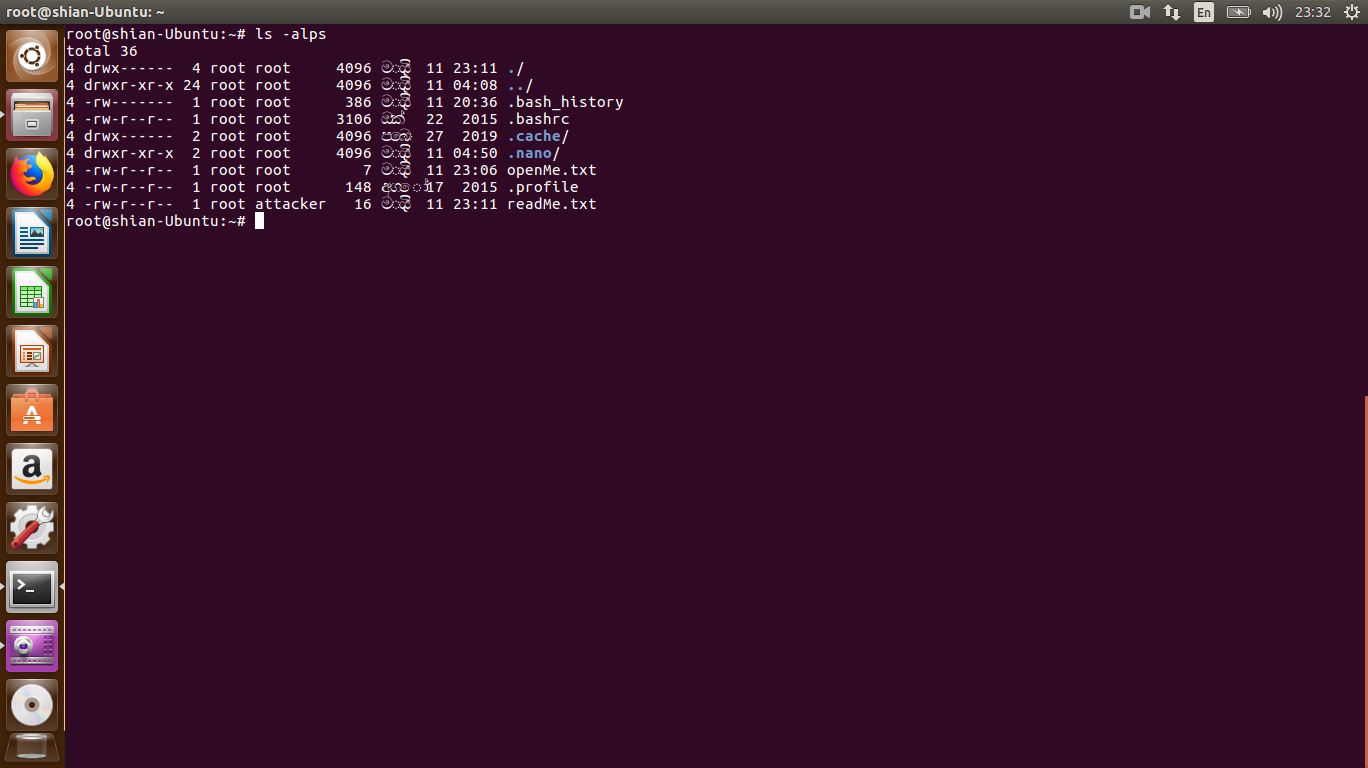
* Then I have entered that command using -u#-1 . Then it works.



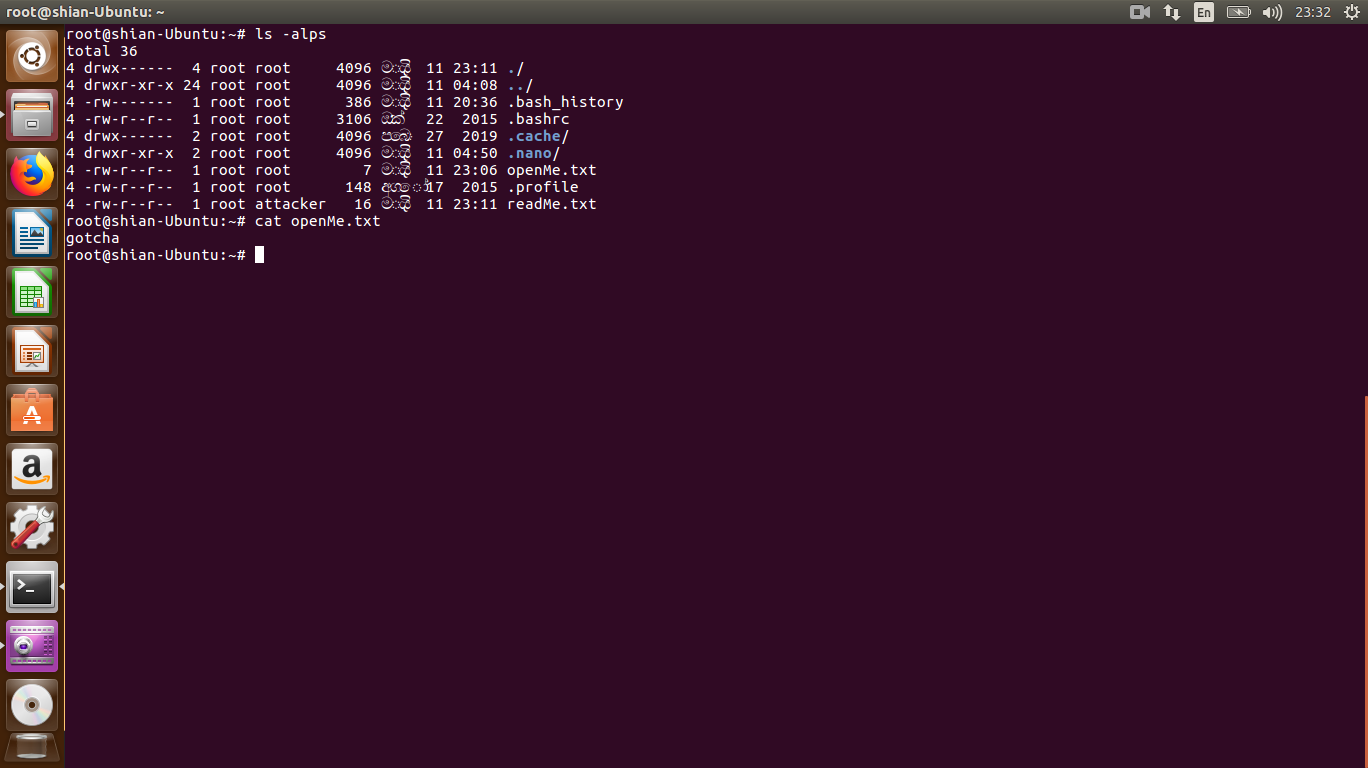
Because of that vulnerability , if someone want to they can insert virus files into root directory which can damage the operating system by opening that file. This is a very serious issue.

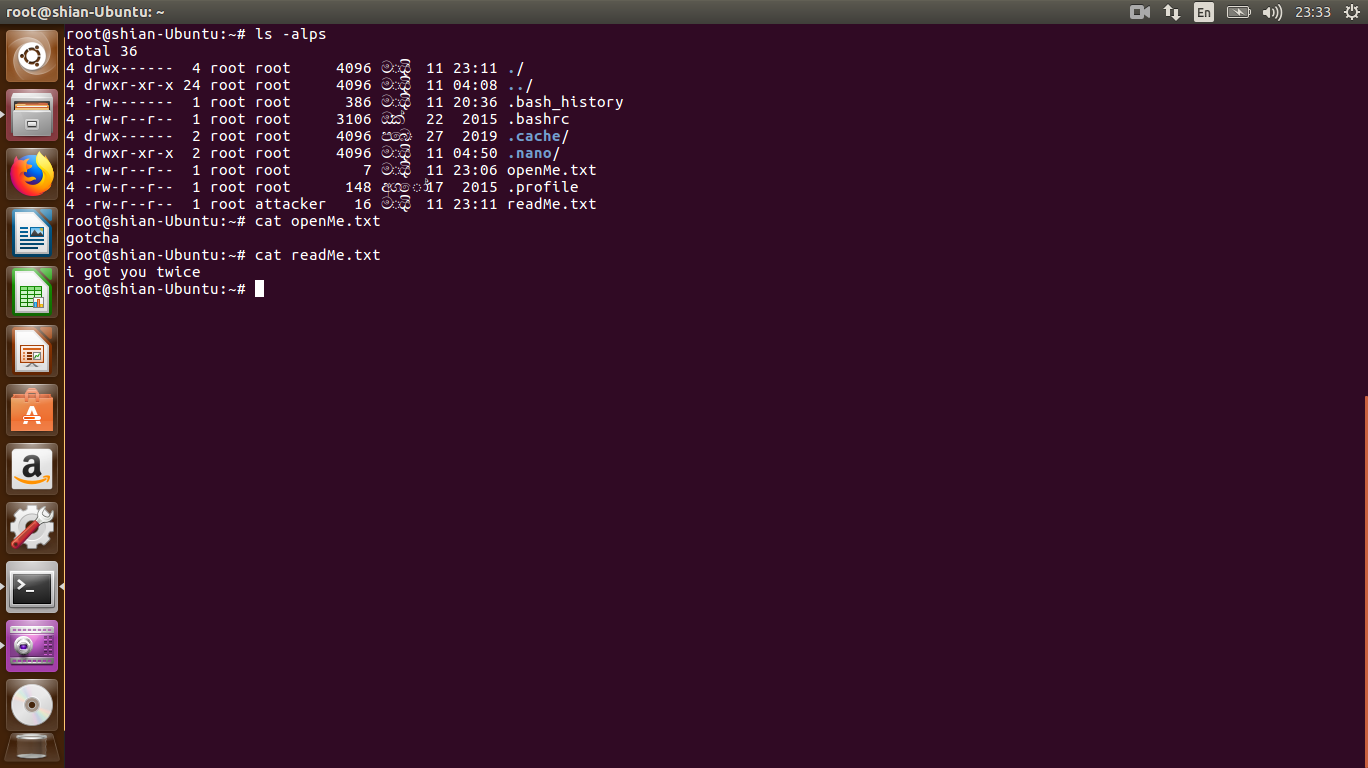
To know that exploit is success or not, I have accessed into my root directory.

* Switch user to root



* Now we can see that openMe.txt file & readMe.txt file in root directory which I created earlier.
* The cd command, also known as **chdir** (**change directory**), is a command-line shell command used to ***change the current working directory in various operating systems***. It can be used in shell scripts and batch files.
* The **ls command** is a **command**-line utility for listing the contents of a directory or directories given to it via standard input. It writes results to standard output. The **ls command** supports showing a variety of information about files, sorting on a range of options and recursive listing.
* The **cat command** (short for “concatenate “) is one of the most frequently used **command** in Linux/Unix, Apple Mac OS X operating systems. **cat command** allows us to create single or multiple files, view contain of file, concatenate files and redirect output in terminal or files.
* Typing “cat *filename*.txt”, Now I can see what is inside that file. So, with this prove you can accept that fact I have exploited sudo security bypass vulnerability successfully.





**Conclusion**

Detecting exploitation attempts of this vulnerability is fundamental. If you cannot patch your systems immediately you can detect and potentially prevent the attack. If you have already upgraded your system and containers affected by the vulnerability, it is still extremely interesting to detect any attempt so we can catch malicious activity in our environment.

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.

It allows users to writing a custom rule to match the exploit behaviour pattern, to then alert regarding the malicious activity across our hosts and containers. Sysdig Secure 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://sysdig.com/blog/detecting-cve-2019-14287/>

<https://www.youtube.com/watch?v=YCXnFEz_Qq8>

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

<https://www.sudo.ws/alerts/minus_1_uid.html>

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

My video - https://youtu.be/QuFYhT7Drvw