**Sudo Vulnerability (CVE – 14287)**

****

IT19026794

Jayasuriya K.R.P.B

SUDO or superuser do lets a normal user access functions that only a superuser or root user is allowed to access but the degree of access to sudo functions can be limited by the administrator using the visudo function and it will be changed in the sudoers file but this vulnerability lets a user by pass these limitations to access root functions and even get access and act as the root user. This bug had become a huge threat of privilege escalation.

Where SUDO is used:-

* For a normal user to access a function of the root user (super user) but the system administrator can limit the functions they can access through SUDO
* Control the users access to commands in each and every host
* Log which users used which command.
* Using timestamp files to control the amount of times a user has to enter commands after they have entered the system using their password and granted the deserved privillages.

Privilege escalation is a user using a bug to increase their privileges than they were supposed to have in the hierarchy of the system . There are 2 kinds of privilege escalations:-

* Horizontal Privilege Escalation – Where the user gets access to accounts in the same level of the hierarchy as he/she is but those accounts will have different privileges assigned to them and the attacker will be able to access each and ever one of those privileges.
* Vertical Privilege Escalation – Where the user access accounts and privileges of users above him in the hierarchy

Attackers can use these privileges to steal confidential data run, administrative commands or to deploy malware which can do serious harm to a computer network or a server.

Privilege escalation is usually done as a subpart of a much larger attack which allows attackers to enter a malicious payload or run unauthorized code in the targeted system.

For example other than the sudo exploitation centered in this document Dirty Cow is another hugely talked about privilege escalation exploit this exploit gave attackers the privilege to write and edit read only memory mappings and if it is combined with other attacks it can give the attacker root access to the system.

**Person Who Discovered The Vulnerability**

The vulnerability was discovered by, Security expert and hacker from apple cooperation Joe Vennix.

**When Was The Vulnerability Discovered**

14th October 2019

**When Was The Vulnerability Fixed**

The vulnerability was fixed in SUDO version 1.8.28.

**Impact caused from the vulnerability**

* Data Integrity is lost.
* Confidential data and functions can be accessed with vertical or horizontal privilege escalation.
* Access can be granted to root user which can lead to complete chaos.

**Explaining the bug**

Sudo is a program that is assigned to operating systems that are used to delegate privileges and are Unix like such as the LINUX operating system. Joe Vennix of Apple information security found a new security flaw in the Sudo program versions that came before version 1.8.28 and on October 14th 2019, the Sudo the program team issued a security alert about this new found flaw which was named as CVE-2019-14287. This newly established security issue has the ability to make a hostile user to pose as a root use and carry out arbitrary commands even when root access is rejected. This security issue could be a high vulnerability to Sudo as it is a widely used program by Linux users.

The Sudo program has a security policy known as sudoers. This enables only the specific authorized users to use certain privileges that is, the privileges are restricted to these users. The security issues CVE-2019-14287 is pertinent to this certain configuration.

To further explain the bug, if a sysadmin enters an entry in the sudoers file such as,

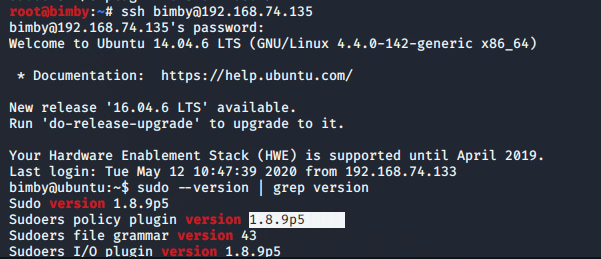
--user-- ALL=(ALL , !root) /usr/bin/id

The inserted entry shows that there is limited access due to the security policy that is created as the entry means that the id function is authorized to be run only by root user and the –user—is restricted from using

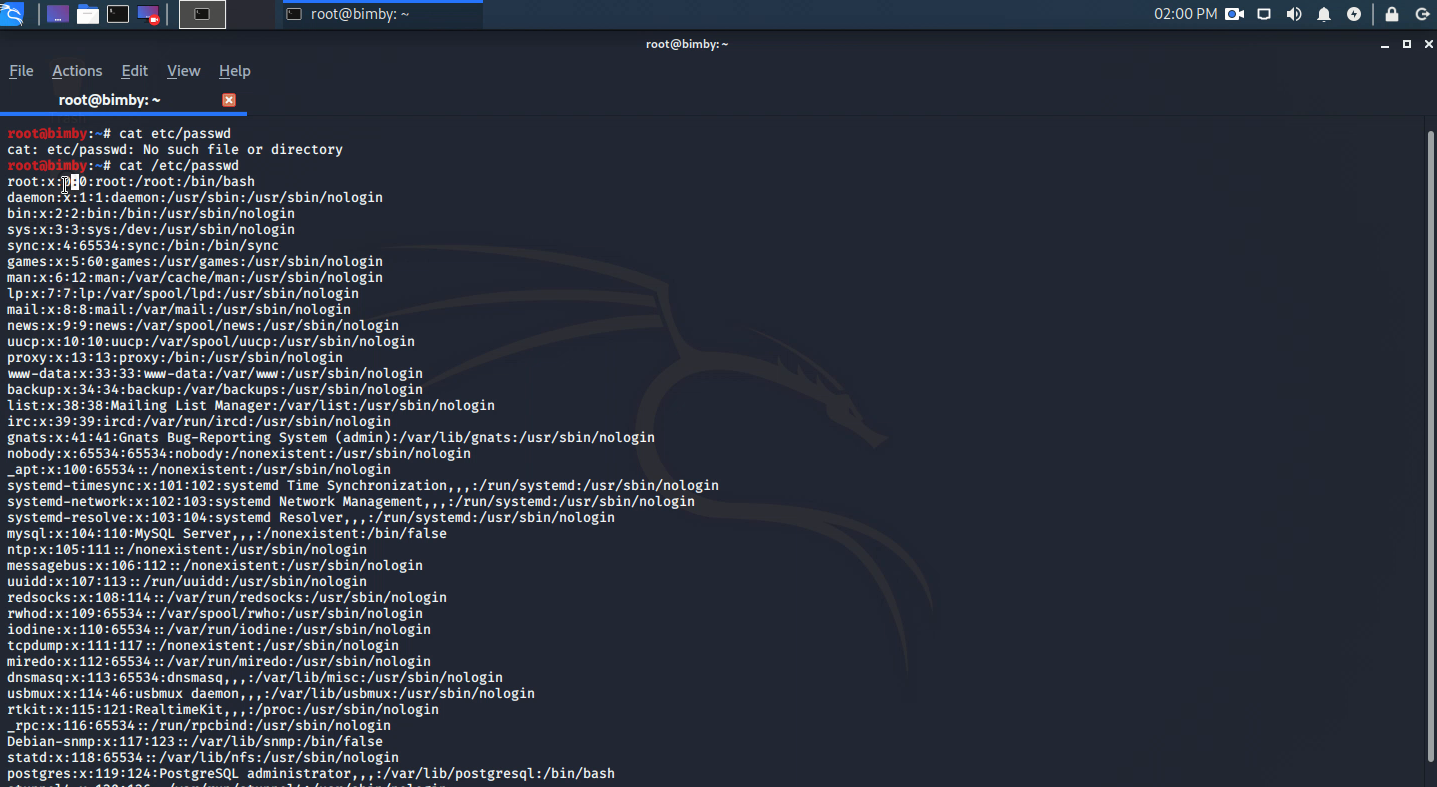
However, Joe Vennix from Apple Information Security found that there is a security issue in this function as it does not parse all the values correctly. This is because the command would run as root, bypassing the security policy if the parameter used id -1 or 4294967295, its undersigned number is given.

However, on a brighter note, the team of Sudo has introduced a secure version which is version 1.8.28 and over. It should also be noted that the security issue in Sudo takes place in a very specific configuration. As the issue is applicable only in a certain configuration and as the bug is fixed in the updated versions, the users of Sudo do not have to worry about the vulnerability found in this program

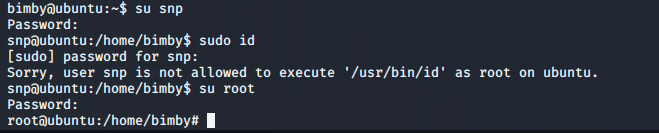
**Screen Shots**



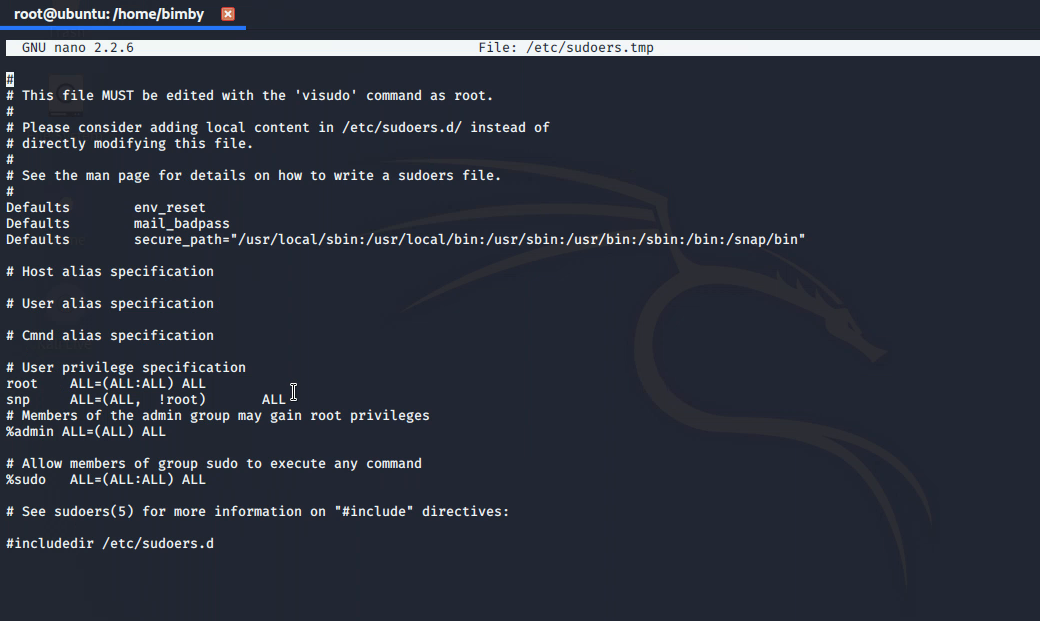
This screen shot shows the sudo id of my ubuntu system’s SUDO which is 1.8.9 so because it is less than 1.8.28 the exploit can be done from this system



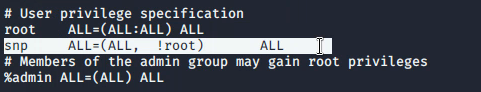
Here shows the ids and some further information about the users the important thing is that the id of root is 0 and the id of created users is more than 1000.



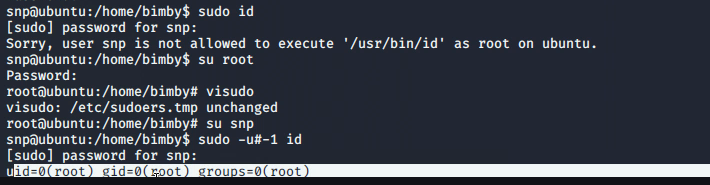
In this screen shot in line number 5 it shows that the user is not allowed to access function ID.



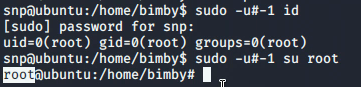
This is the sudoers file.



Here from the highlighted line it shows that the user snp cant access and root functions it is denoted by (ALL, !root) ALL



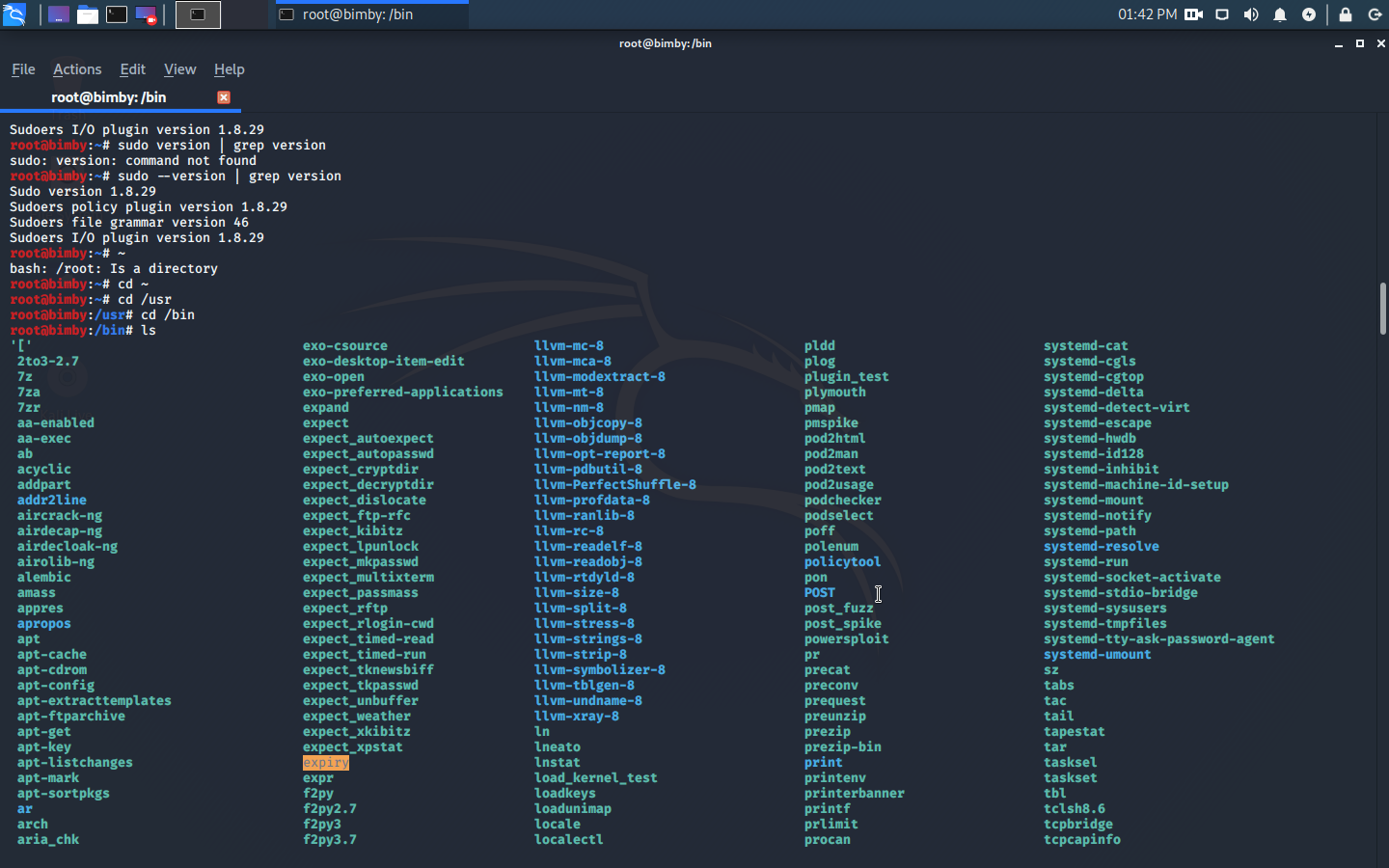
Although it is said that snp cannot access and sudo functions here using the exploit it accessed the sudo function id and got the desired output



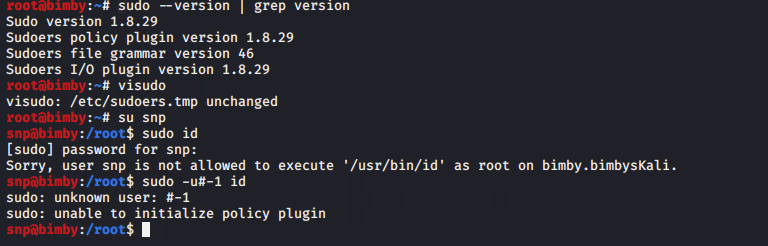
Here it shows the most dangerous part of the exploit where using this vulnerability I became the root user without specifying passwords



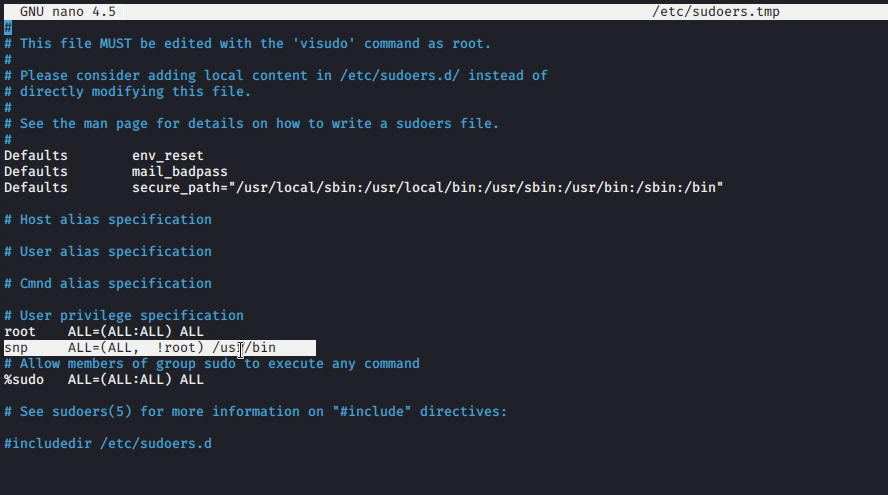
Here it is proven that the user actually is set as root when u use -u#-1



It shows some of the functions in the usr/bin directory so using this exploit any of the functions can be accessed and there are about 2000 functions.



This shows my kali box and it has SUDO higher than 1.8.29 and it is proven here that the problem did get fixed in 1.8.28



The sudoers file of my kali box

**Types Of Exploits**

One of the common ways that an attacker gains access to a system is through privilege escalation. Attackers find vulnerabilities in an organization’s defenses and gains access to a system. After wards the attacker escalates its privileges thus the attacker can do what the attacker intended to do to the system.

There are two types of privilege escalation.

* Horizontal privilege escalation
* Vertical privilege escalation

**Types of exploiting techniques.**

1. Unquoted service paths

This vulnerability occurs when the service path is not closed with quotation marks and thus contains spaces.

An unquoted service path

1. DLL hijacking

This attack can happen when a file or application dynamically loads without specifying a correct path name. The attacker will access the search paths and then include a malicious code of DLL and this newly loaded DLL will have admin privileges thus the perpetrator can escalate root privileges.

### **Kernel exploits**

Kernel exploits are programs that exploits the kernel vulnerabilities and then execute code with increased privileges. successful exploits grant perpetrators with super user permissions in the form of root.

Unprivileged user running exploit

1. Trick the kernel into running our payload in kernel mode  
2. Manipulate kernel data.   
3. Launch a shell with new privileges

Consider that for a kernel exploit attack to succeed, an opponent requires four conditions:

* 1. A vulnerable kernel  
     2. A matching exploit  
     3. The ability to transfer the exploit onto the target  
     4. The ability to execute the exploit on the target

**Exploiting services which are running as root**

With just one exploit code, an attacker can get remote code execution and Local Privilege Escalation. Heavily used to spread ransomware across of the globe because of this deadly combination.

**Exploiting SUID Executables**

SUID which stands for set user ID, is a Linux feature that allows users to execute a file with the permissions of a particular user. For instance, the Linux ping command requires root permissions in order to open network sockets, by marking the ping program as SUID with the owner as root, ping executes with root privileges anytime a low privilege user executes the program.

**Exploiting SUDO rights/user**

Once the attacker has access to any of the sudo users, the attacker can basically execute any commands with root privileges.

Administrators might just allow the users to run a few commands through SUDO and not all of them but even with this configuration, but users may abuse these privileges and they might introduce vulnerabilities unknowingly or knowingly which can lead to privilege escalation The exploit I used as sudo exploit cve 2019 - 14287

**Exploiting badly configured cron jobs**

Cron jobs, if not configured properly can be exploited to get root privilege.

Cron jobs generally run with root privileges. If an attacker finds a successful way to tamper any script or binary which are defined in the cron jobs then we can execute code with root privilege.

**Exploiting users with “.” in their PATH**

Introducing ‘.’ in the path means that the user is able to execute binaries/scripts from the current directory. This can be used as an excellent method for an attacker to escalate their privileges.

Eg:-

With “.” in path – text  
Without “.” in path – ./text

This error occurs because Linux first searches for the program in the current directory but when “.” is added in the path at the beginning the kernel searches anywhere else.

**Conclusion**

As it is show through out this document and my video this exploit is a really dangerous exploit of privilege escalation so it was a good thing that this error was fixed as early as possible and it is the responsibility of citizens to find these kinds of errors and notify the related party so that users can spend their time without worrying that some hacker will steal their classified data.