**SUDO vulnerability**

SUDO also known as super user do is a very essential and quite powerful mechanisms inside the Linux environment. It is a utility for UNIX and LINUX based systems and provides an efficient way to give users most privileges to run specific commands that gives root access to a system.

Examples of what SUDO can do are:

* Give 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 passwords and been granted appropriate privileges.

Privilege escalation is the act of exploiting system vulnerabilities to increase one’s privileges to use a higher power than the administration or developers intended.

Many organizations nowadays are statistically expected to have more Windows machines present, Linux privilege escalation attacks are noteworthy threats to take into consideration for when considering a company’s information security. Let us assume that an organization’s most important infrastructure, such as web servers, firewalls ,databases, etc. are running a Linux operating system. Hijacking to these devices have the potential to severely disrupt a company’s operations, if not destroy them entirely. In the near future, Internet of Things (IoT) and embedded systems are becoming the next step in the advancement of the workplace, thereby increasing the number of possible targets for malicious hackers. The occurrence of Linux devices in the workplace are highly take to consideration, it is of utmost importance that organizations harden and secure these devices.

This SUDO vulnerability is very easy to hack and by using this flaw a hacker can either remotely access or locally access and then conduct a privilege escalation for any user.

This vulnerability which found on October 14th  has a CVSS of 7.8 (as said by REDHAT security) this attack poses threat to the systems CIA triad as confidentiality, integrity, and availability.

The exploitation method I chose is to exploit privilege escalation using SUDO.

**Person who found**

Joe Vennix from apple information security found and analyzed.

**When was it found?**

14th October 2019

This bug was fixed is SUDO 1.8.28 version.

**Impact caused by the bug**

* Confidentiality issues as root privileges can be changed thus can access data with higher privileges.
* Data integrity is lost.

**The bug explained**

This vulnerability affects sudo versions bellow 1.8.28 as mentioned above. Normally, when a root user checks his/her id the root user always has an id of 0. But now in the case of the non-user, the non-user runs the command sudo -u#-1 id, the -1 goes to a function known as the setresuid(). These functions are written in a way that when -1 is sent as arguments the functions return the value what ever the function had before but wont change to -1. Thus, normally before writing -1 the user anyway has 0 in the function thus the function assigns this new user 0 thus making the new no privileged user a root user even though the sudoers file specifically says the non-user is !root.

By using the same sudo -u#-1 command and visudo, the visudo file can be changed to any of the commands allocated in the bin file thus the attacker can use any command.

This vulnerability is most affected by the users who use the sudoers file a lot of times otherwise there is not much harm intended to do so.

**Affected SUDO versions**

|  |  |  |
| --- | --- | --- |
| **Vendor** | **Product** | **Versions** |
| [Netapp](https://vulmon.com/searchpage?q=netapp) | [Element Software Management Node](https://vulmon.com/searchpage?q=element_software_management_node) | - |
| [SUDO Project](https://vulmon.com/searchpage?q=sudo_project) | [SUDO](https://vulmon.com/searchpage?q=sudo) | 1.3.0, 1.3.1, 1.4.0, 1.5.0, 1.5.1, 1.5.2, 1.5.3,  1.5.4, 1.5.6, 1.5.7, 1.5.8, 1.5.9, 1.6.0, 1.6.1,  1.6.2, 1.6.3, 1.6.4, 1.6.5, 1.6.6, 1.6.7, 1.6.8,  1.7.0,1.7.1, 1.7.2, 1.7.3, 1.7.4, 1.7.5, 1.7.6,  1.7.7, 1.7.8, 1.7.9, 1.7.10, 1.8.0, 1.8.1, 1.8.2,  1.8.3, 1.8.4, 1.8.5, 1.8.6, 1.8.7, 1.8.8, 1.8.9,  1.8.10, 1.8.11, 1.8.12, 1.8.13, 1.8.14, 1.8.15,  1.8.16, 1.8.17, 1.8.18,1.8.19, 1.8.20, 1.8.21,  1.8.22, 1.8.23, 1.8.24, 1.8.25, 1.8.26, 1.8.27 |
| [Canonical](https://vulmon.com/searchpage?q=canonical) | [Ubuntu Linux](https://vulmon.com/searchpage?q=ubuntu_linux) | 12.04, 14.04, 16.04, 18.04, 19.04 |
| [Debian](https://vulmon.com/searchpage?q=debian) | [Debian Linux](https://vulmon.com/searchpage?q=debian_linux) | 8.0,9.0,10 |
| [Fedora project](https://vulmon.com/searchpage?q=fedoraproject) | [Fedora](https://vulmon.com/searchpage?q=fedora) | 30,31 |
| [Opensuse](https://vulmon.com/searchpage?q=opensuse) | [Leap](https://vulmon.com/searchpage?q=leap) | 15.0, 15.1 |

**Exploitation techniques**

The techniques explained include current kernel exploits, wildcard injections, The SUID executable, SUDO exploits, physical access attacks, and the LUKS vulnerability.

**Kernel Exploits**

Kernel exploits are exploited by the use of a syscall and with arguments designed specially to cause unintended behaviour. Kernel exploits are fragile. Successful kernel exploits typically

give attackers super user access to target systems in the form of a root command prompt.

One kernel exploit, Dirty COW, received a lot of attention because of its extreme and extensive impact on millions of Linux devices. Kernel exploits can be an effective means to escalate privileges, they can often result in system instability because they interfere with the very basement of the operating system. Moreover, kernel exploits are less likely to be successful in environments with thorough patching practices. For some reasons, penetration testers and attackers alike often focus on identifying and exploiting weak Linux services and configurations due to their relative stability and also because of their importance on modern Linux systems.

**Wildcard Injection**

A wildcard is a character or set of characters that can be used as a replacement for some range/class of characters. Implemented in shell before any action is taken.

Most Linux administrators are fairly familiar with Linux wild cards, mainly

the asterisk (\*),(?),[],(-),(~). Wildcards make it easy for users to perform operations on arbitrary ranges or classes of characters.

What many Linux users may not know is that wild cards can be manipulated to perform many various attacks. Using wild cards to exploit Linux systems was first published by Leon Juranic in his paper, “Back to the Future: Unix Wildcards Gone Wild”

during 2014. The premise of Juranic’s paper is that wild cards can be utilized to inject arbitrary commands into the Linux environment.

**The SUID Executable**

SUID, which stands for set user ID**,** itis a way in UNIX-like operating systems of running a command as another user without providing credentials.

-s-x-x 1 root root 147044 sep 30 2013 /usr/bin/sudo

The s denodes that the user permissions shows that the file had suid set, if s if there in x in group then guid is set.

Using suid a hacker can easily get access and run the exploit code logs in with a valid user id and the change the uid.

As a safety precaution these suid files are run with shell intepreters thus when run with interpreters the privileges are dropped and cannot run at higher privileges.

**Exploiting SUDO Users**

Perpetrators are mostly attentive in compromising SUDO users, that is, users who can execute SUDO commands. If an attacker can compromise a user who has SUDO rights, he can potentially execute arbitrary commands with root privileges. Administrators are normally aware that they need to appropriately manage their SUDO users just as they manage the root account to prevent SUDO misuse. However, serious vulnerabilities are regularly introduced by administrators because they may have installed a poorly configured third party application or issued SUDO rights without awareness of command execution vulnerabilities. Programming language compilers and interpreters are just as under red alert to SUDO abuse.

**Physical Access Attacks**

Physical access allows [hardware keyloggers](https://en.wikipedia.org/wiki/Hardware_keylogger) and various other spyware and malware to be installed. An intruder can boot from a DVD or other external media and then read unencrypted data on the hard drive. An attacker can use the advantage of pressing F8 in certain versions of Microsoft Windows are booting, specifying 'init=/bin/sh' as a boot parameter to Linux. Another attacker can use some device and then wander an unsecured network

**LUKS Vulnerability**

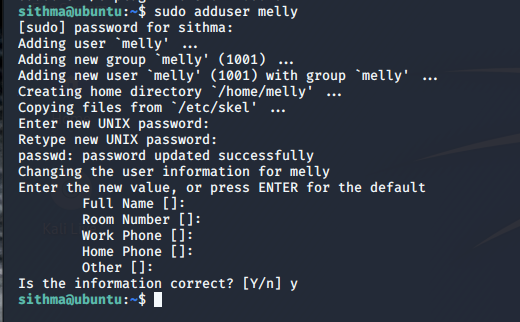
This vulnerability occurs because of a flaw in the password check function of the file “/scripts/local-top/cryptroot/” which is part of the Cryptsetup utility.

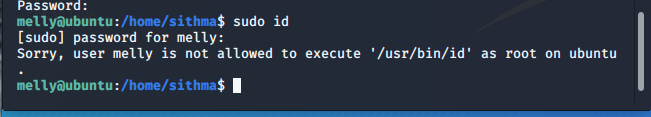
The Cryptsetup utility is the standard implementation of disk encryption on Linux-based systems. The subject flaw affects to how the Cryptsetup utility processes repeat password failures.

When a user exceeds the maximum number of passwords attempts the boot sequence continues. Though, the calling script, “/scripts/local” handles the authentication error as incase if it was caused by a slow device that requires more time. The booting scripts then try to mend the “not connecting” device. After this process occurs for a number of times a hardware fault is reached. At this point, the top-level script is not aware of the root cause of the fault and drops the user in a root shell leaving the attacker freely to do any chaos to the system and the plus point is that the hard drives and disks are still encrypted as the attacker does not have the LUKS password. But still the attackers can implement root owned SUID binaries into the non-encrypted parts of the hard drives and thus then logs into the system using low privileged credentials and then runs the SUID binaries thus bringing the system to the attackers needs.

A screenshot of a computer screen

Description automatically generated**Screenshots**



A close up of a screen

Description automatically generatedA close up of a screen

Description automatically generatedA screenshot of a computer screen

Description automatically generated



**A screen shot of a computer

Description automatically generated**

**A picture containing food

Description automatically generated**

**A close up of a sign

Description automatically generated**

**Conclusion**

This vulnerability is quite a very dangerous vulnerability as using this vulnerability and code can be used in the bin file thus posing a huge threat for the system. There are nearly 2000 codes that are only provided access by the bin and the non-user can get access to. But there is no need to worry though as now this vulnerability is patched. Simply updating the sudo version will fix this issue.

**References**

* <https://www.cynet.com/cyber-attacks/privilege-escalation/>
* https://medium.com/@isharaabeythissa/cve-2019-14287-sudo-will-hit-your-root-4df17e6a089b
* https://resources.whitesourcesoftware.com/blog-whitesource/new-vulnerability-in-sudo-cve-2019-14287
* https://access.redhat.com/security/cve/cve-2019-14287
* https://github.com/n0w4n/CVE-2019-14287/blob/master/sudo.sh
* https://www.suse.com/security/cve/CVE-2019-14287/
* https://medium.com/@jamesluo\_843/exploiting-cve-2019-14287-37eac7023a4
* https://access.redhat.com/solutions/4502891
* https://sysdig.com/blog/detecting-cve-2019-14287/