Ubuntu
apport package

Log in / Register

Overview Code Bugs Blueprints Translations Answers

Apport crash report & cron script TOCTTOU

Bug #1862933 reported by Maximilien Bourgeteau on 2020-02-12

This bug affects 1 person

258

Affects	Status	Importance	Assigned to	Milestone
Apport	Fix Released	Critical	Unassigned	Apport 2.21.0
apport (Ubuntu)	Fix Released	Undecided	Unassigned	

Bug Description

Vulnerable code (from data/apport):

```
# we prefer having a file mode of 0 while writing; this doesn't work
# for suid binaries as we completely drop privs and thus can't chmod
# them later on
if pidstat.st_uid != os.getuid():
mode = 0o640
else:
mode = 0
reportfile = os.fdopen(os.open(report, os.O_RDWR | os.O_CREAT | os.O_EXCL, mode), 'w+b')
assert reportfile.fileno() > sys.stderr.fileno()

# Make sure the crash reporting daemon can read this report
try:
gid = pwd.getpwnam('whoopsie').pw_gid
os.chown(report, pidstat.st_uid, gid)
except (OSError, KeyError):
os.chown(report, pidstat.st_uid, pidstat.st_gid)
except (OSError, IOError) as e:
error_log('Could not create report file: %s' % str(e))
sys.exit(1)
```

The TOCTTOU takes place between the os.open and the os.chown call and can be fully achieved thanks to the Apport cron script (etc/cron.daily/apport) :

```
#!/bin/sh -e
# clean all crash reports which are older than a week.
[ -d /var/crash ] || exit 0
find /var/crash/. ! -name . -prune -type f \( \( -size 0 -a \! -name '*.upload*' -a \! -name '*.drkonqi*' \) -o -mtime +7 \) -exec rm -f -- '{}' \;
find /var/crash/. ! -name . -prune -type d -regextype posix-extended -regex '.*/[0-9]{12}$' \( -mtime +7 \) -exec rm -Rf -- '{}' \;
```

The interesting part in this daily script is that the crash reports gets removed if their size is 0.

Since Apport drops real uid and gid, the crashed process owner can send signals during the report file creation. At this time, effective uid and gid are still root.

We can also block Apport by replacing user settings file with a FIFO (~/.config/apport/settings). I'm using the FIFO way in my PoC but it can be done without it.

To make Apport read user settings, the crashing program must not be located in one of those directories (taken from apport/fileutils.py):

```
pkg_whitelist = ['/bin/', '/boot', '/etc/', '/initrd', '/lib', '/sbin/',
'/opt', '/usr/', '/var'] # packages only ship executables in these directories
```

Once Apport is blocked into FIFO reading, we send the SIGSTOP signal then we write "[main]\nunpackaged=1" into the FIFO so Apport won't exit after resuming (if unpackaged is 0 Apport directly exits because we're not in a "packaged" directory).

After that we "single step" through Apport by sending SIGCONT and SIGSTOP consecutively in a loop until the report file is created with os.open. We must make sure os.chown hasn't been called and then we wait for the cron script to remove the report (it's created as root with mode 0 so only root can remove it).

Once removed, we can replace it with a symbolic link/file of the same name, resume Apport with SIGCONT then the file will now be owned by the crashed process user and group.

I think the impact of this vulnerability alone is low because fs.protected_symlinks prevents symlink resolution since we're in a sticky world writable directory (/var/crash), but if it's disabled, you can escalate privileges very easily. It still can be used in some kind of exploit chain though.

My PoC does everything for you except symbolic link/file creation once the report gets removed by the cron script, you have to create it manually then press enter to resume Apport and let the chown happen. You could also create a new crontab entry and directory then copy Apport cron script into it so you don't have to wait the entire day.

Fix suggestions:
- Use reportfile.fileno() instead of the report string for the os.chown calls, and also add follow_symlinks=False argument just in case.
- Remove the size 0 condition in the cron script (not sure about this one, I suppose the condition was there for a reason).

See original description

Report a bug

This report contains **Public Security** information

Everyone can see this security related information.

You are not directly subscribed to this bug's notifications.

Edit bug mail

Other bug subscribers

Subscribe someone else

Notified of all changes

Maximilien Bourge...
Ubuntu Security Team

May be notified

Alejandro J. Alva...
Ashani Holland
Benjamin Drung
Brian Murray
Bruno Garcia
CRC
Charlie_Smotherman
Christina A Reitb...
Debian PTS
Doraann2
Franko Fang
Hans Christian Holm
HaySayCheese
Hidagawa
Jesse Jones
José Alfonso
Kees Cook
Matt j
Micah Gersten
Michael Rowland H...
Mr. MInhaj
Name Changed
PCTeacher012
Paolo Topa
PechayClub Inc.
Peter Bullert
Philip Muškovac
Punnsa
Richard Seguin
Richard Williams
Tom Weiss
Ubuntu Foundation...
Ubuntu Touch seed...
Vasanth
Vic Parker
ahepas
basilisgabri
dsfkj dfjx
eoininmoran
ganesh
linuxgijs
miked
nikonikic42
projevie@hotmail.com
qadir
sankaran
van

Patches

apport_2.20.11-0ubuntu19.debdiff
apport_2.20.11-0ubuntu19.debdiff
Add patch

Bug attachments

poc.c
Add attachment

Related branches

lp:~ubuntu-core-dev/ubuntu/focal/apport/ubuntu

CVE References

2020-8831

2020-8833

Maximilien Bourgeteau (mbourget) wrote on 2020-02-12: #1

poc.c (3.7 KiB, text/x-csrc)

Maximilien Bourgeteau (mbourget) on 2020-02-12

description:updated

Maximilien Bourgeteau (mbourget) on 2020-02-12

description:updated

Marc Deslauriers (mdeslaur) wrote on 2020-02-14: #2

Thanks for reporting this issue, we'll investigate promptly.

Seth Arnold (seth-arnold) wrote on 2020-02-14: #3

Please use CVE-2020-8833 for this issue: chmod(2) instead of fchmod(2) of the report.

Thanks for the report.

Seth Arnold (seth-arnold) wrote on 2020-02-14: #4

Of course that's my mistake, thanks Marc, chown(2) instead of fchown(2).

Alex Murray (alexmurray) wrote on 2020-02-27: #5

apport_2.20.11-0ubuntu19.debdiff (2.8 KiB, text/plain)

See attached for a proposed patch against apport in focal to fix both this and LP #1862348 in a single update.

Alex Murray (alexmurray) wrote on 2020-02-28: #6

apport_2.20.11-0ubuntu19.debdiff (2.8 KiB, text/plain)

Updated version of patch based on feedback in #1862348

Launchpad Janitor (janitor) wrote on 2020-04-02: #7

This bug was fixed in the package apport - 2.20.11-0ubuntu22

```
---------------
apport (2.20.11-0ubuntu22) focal; urgency=medium

  * SECURITY UPDATE: World writable root owned lock file created in user
    controllable location (LP: #1862348)
    - data/apport: Change location of lock file to be directly under
      /var/run so that regular users can not directly access it or perform
      symlink attacks.
    - CVE-2020-8831
  * SECURITY UPDATE: Race condition between report creation and ownership
    (LP: #1862933)
    - data/apport: When setting owner of report file use a file-descriptor
      to the report file instead of its path name to ensure that users can
      not cause Apport to change the ownership of other files via a
      symlink attack.
    - CVE-2020-8833

 -- Alex Murray <email address hidden> Wed, 25 Mar 2020 11:28:58 +1030
```

Changed in apport (Ubuntu):
status: New → Fix Released

Alex Murray (alexmurray) on 2020-04-02

information type: Private Security → Public Security

Benjamin Drung (bdrung) on 2022-06-27

Changed in apport:
milestone:none → 2.21.0
status: New → Fix Released
importance: Undecided → Critical

See full activity log

To post a comment you must log in.