Apport lock file root privilege escalation

Bug #1862348 reported by Maximilien Bourgeteau on 2020-02-07

This bug affects 1 person 260

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

Bug Description

```
Vulnerable source code (from data/apport):

    35 # create lock file directory
    36 try:
    37 os.mkdir("/var/lock/apport", mode=0o744)
    38 except FileExistsError as e:
    39 pass
    40
    41 # create a lock file
    42 try:
    43 fd = os.open("/var/lock/apport/lock", os.O_WRONLY | os.O_CREAT |
os.O_NOFOLLOW)
    44 except OSError as e:
    45 error_log('cannot create lock file (uid %i): %s' % (os.getuid(),
str(e)))
    46 sys.exit(1)

When invoked, Apport tries to create the directory /var/lock/apport and
continues its execution if the directory already exists.

Since /var/lock is a world writable tmpfs, the probability that
/var/lock/apport directory doesn't exist is high, which allows a malicious
user to create a symbolic link to the directory of its choice to control
the lock file location.

In this case, os.O_NOFOLLOW and fs.protected_symlinks (sysctl) have no
effect during os.open execution because the symbolic link isn't located in
the last component of the given path.

In addition, os.open is called without specifying the "mode" optional
argument which by default is set to 0o777. Thus the lock file is created
as root and is world writable which opens the door to several root
privilege escalation scenarios like, for example, creating the lock file
in a cron scripts directory.

All releases containing the bug 1839415 fix (https://bugs.launchpad
.net/apport/+bug/1839415) are affected.

Fix suggestions:
- If the /var/lock/apport directory already exists and isn't owned by root
or owned by root but world writable, remove it and recreate it.
- Specify a mode of 0o600 in the os.open call for the lock file.
```

See original description

Related branches

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

CVE References

2020-8831

2020-8833

Maximilien Bourgeteau (mbourget) on 2020-02-09

description:updated

Marc Deslauriers (mdeslaur) wrote on 2020-02-09: #1

```
Thanks for reporting this issue, we will investigate it shortly.
```

Seth Arnold (seth-arnold) wrote on 2020-02-11: #2

```
Hello, please use CVE-2020-8831 for this issue.

Thanks
```

Alex Murray (alexmurray) wrote on 2020-02-11: #3

```
Thanks again for reporting this issue. Do you have a proposed coordinated
release date (CRD) for this issue? We would prefer this to be in at least
a few weeks time so that patches and updated packages can be prepared in
advance so that when this issue is made public the fixed packages can be
available immediately. If you have a preferred date or timeline please let
us know, otherwise we can propose something.
```

Maximilien Bourgeteau (mbourget) wrote on 2020-02-11: #4

```
Hello, I don't have a CRD to propose, feel free to propose one.
```

Alex Murray (alexmurray) wrote on 2020-02-11: #5

apport_2.20.11-0ubuntu17.debdiff (2.6 KiB, text/plain)

Patches

apport_2.20.11-0ubuntu17.debdiff

Reworked patch to just move the lock file to /var/run directly

apport_2.20.11-0ubuntu19.debdiff

apport_2.20.11-0ubuntu19.debdiff

Add patch

Apport is maintained by the foundations team so I will wait to confer with them before determining a CRD. In the meantime I have come up with a patch which should resolve this - @mbourget as you reported this and also suggested an appropriate fix I would appreciate it if you could please review it. @bdmurray can you or someone appropriate from the foundations team please also review? Finally, @seth-arnold I would appreciate your review too.

Marc Deslauriers (mdeslaur) wrote on 2020-02-11: #6

I think shutil.rmtree can be tricked into deleting arbitrary files via a symlink attack. Perhaps the best approach is to simply set the apport lock file as /run/apport.lock?

Maximilien Bourgeteau (mbourget) wrote on 2020-02-11: #7

Yes, shutil.rmtree can be vulnerable to symlink attacks, Python 3 documentation has a special note about it (https://docs.python.org/3/library/shutil.html#shutil.rmtree):

"On platforms that support the necessary fd-based functions a symlink attack resistant version of rmtree() is used by default. On other platforms, the rmtree() implementation is susceptible to a symlink attack: given proper timing and circumstances, attackers can manipulate symlinks on the filesystem to delete files they wouldn't be able to access otherwise. Applications can use the rmtree.avoids_symlink_attacks function attribute to determine which case applies."

@alexmurray I think Marc approach is a better choice than my suggestions because only root can write into /run which avoids any symlink trickery (as far as I know).

Seth Arnold (seth-arnold) wrote on 2020-02-12: #8

I also endorse the /run/apport.lock file approach -- the os.mkdir() calls in the new code can be raced to create exceptions same as the top-level os.mkdir(). Probably the os.unlink() and shutil.rmtree() calls could also be raced to cause arbitrary files to be deleted.

Thanks

Alex Murray (alexmurray) wrote on 2020-02-12: #9

Thanks for the excellent reviews - I had already wondered about a symlink attack hence the code first checks if /var/run/apport a dir and if not unlinks it - but yes this could be raced still so that is a good point. It does indeed seem best to then just put it somewhere where only root has write access to start with (although I have a feeling that apport can sometimes run as a user which in this case this is not going to work but perhaps I am mistaken - although my patch still assumes this file must be root owned anyway which is no different) - so then /var/run/apport.lock does seem like a simpler and better solution. I'll cook up a patch based on that instead.

Alex Murray (alexmurray) wrote on 2020-02-12: #10

Reworked patch to just move the lock file to /var/run directly (1.6 KiB, text/plain)

Alex Murray (alexmurray) wrote on 2020-02-12: #11

Ok please see the reworked patch in comment:10 - I have verified this works as intended and the existing autopkgtest's for focal do not regress (there are still the same failures with this patch as without). If we can agree on this as a solution then we can start to look at backporting it and setting a CRD.

Marc Deslauriers (mdeslaur) wrote on 2020-02-12: #12

Patch in comment 10 LGTM, thanks!

Maximilien Bourgeteau (mbourget) wrote on 2020-02-13: #13

Looks good to me too, thanks!

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

Alex, often O_EXCL is used alongside O_CREAT to ensure the file doesn't already exist -- this seems likely to be something we'd want for a lock file.

Was this rejected or overlooked?

Thanks

Alex Murray (alexmurray) wrote on 2020-02-19: #15

This was overlooked - thanks Seth - however I just tried adding this and it breaks a bunch of the autopkgtests so I don't think it is appropriate to add that at this time - so I think we should proceed with the above patch in comment:10

Maximilien Bourgeteau (mbourget) wrote on 2020-02-21: #16

I think O_EXCL flag breaking tests is a normal behavior since Apport doesn't and shouldn't (AFAIK) remove the lock file once its execution is terminated because there could be another instance waiting for the lock to be released. Using O_EXCL would allow Apport to only run 1 time per boot.

Alex Murray (alexmurray) wrote on 2020-02-27: #17

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 #1862933 in a single update.

Maximilien Bourgeteau (mbourget) wrote on 2020-02-27: #18

Thanks for the patch Alex, here is my review:

The title ("World writable lock file created in word writable location") for the lock file issue seems a bit inaccurate to me, I would rather use "World writable root owned lock file created in user controllable location", also there is typo in the second issue title, the "n" of between has gone away.

About the lock file fix, shouldn't we use a mode of 0o600 instead of 0o400 since os.open is called with the O_WRONLY flag? The lock file is not removed until reboot, it's fine for the first call when the file is created but it seems a bit confusing for the future calls don't you think?

Alex Murray (alexmurray) wrote on 2020-02-28: #19

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

Thanks again for the review Maximilien - I've updated the changelog description and changed the mode to 0o600 as suggested - this does make more sense.

Maximilien Bourgeteau (mbourget) wrote on 2020-03-02: #20

Thanks for the update Alex, the patch in comment 19 looks good to me, also, are there any news about the CRD?

Maximilien Bourgeteau (mbourget) wrote on 2020-03-21: #21

Hello, how it's going?

Alex Murray (alexmurray) wrote on 2020-03-23: #22

@Maximilien - thanks for the reminder re CRD - I want to plan this around some existing updates for apport so will wait until I know when they are likely to be done - then we can look at scheduling this if that's ok?

@Brian - I notice you have apport versions in -proposed for eoan/bionic - do you know when they are likely to migrate?

Brian Murray (brian-murray) wrote on 2020-03-23: #23

I'll make sure the versions in -proposed get migrated on Thursday, March 26th.

Alex Murray (alexmurray) wrote on 2020-03-23: #24

Thanks Brian - Maximilien - in that case, I think 1 week after the 26th (2020-04-02 00:00:00 UTC) would be suitable for a CRD since this should provide enough time for me to backport this fix to our other releases and I don't want to drag this out any longer than necessary. Also to clarify, this would be the CRD for both this issue (CVE-2020-8831) and CVE-2020-8833 from https://bugs.launchpad.net/ubuntu/+source/apport/+bug/1862933 if that works for you?

Maximilien Bourgeteau (mbourget) wrote on 2020-03-27: #25

Hi Alex, yes that works for me, thanks.

Launchpad Janitor (janitor) wrote on 2020-04-02: #26

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

Launchpad Janitor (janitor) wrote on 2020-04-02: #27

This bug was fixed in the package apport - 2.20.11-0ubuntu8.8

---

apport (2.20.11-0ubuntu8.8) eoan-security; 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 11:40:00 +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
importance:Undecided → Critical
    status:New → Fix Released
```

See full activity log

To post a comment you must log in.