Closed · Issue created 1 year ago by Kevin Backhouse

CVE-2021-27219 (GHSL-2021-045): integer overflow in g_bytes_new/g_memdup

GitHub Security Lab (GHSL) Vulnerability Report: `GHSL-2021-045`

The GitHub Security Lab team has identified a potential security vulnerability in GLib.

We are committed to working with you to help resolve these issues. In this report you will find everything you need to effectively coordinate a resolution of these issues with the GHSL team.

If at any point you have concerns or questions about this process, please do not hesitate to reach out to us at `securitylab@github.com` (please include `GHSL-2021-045` as a reference).

If you are *NOT* the correct point of contact for this report, please let us know!

Summary

The function `g_bytes_new` has an integer overflow due to an implicit cast from 64 bits to 32 bits. The overflow could potentially lead to a memory corruption vulnerability.

Product

GLib

Tested Versions

- Ubuntu 20.04 (x86_64): version 2.64.6-1
- CentOS Stream (x86_64): version 2.56.4-9
- archlinux (x86_64): 2.66.4-2

Details

Issue 1: Integer overflow in g_bytes_new (GHSL-2021-045)

On 64-bit platforms, an integer overflow can occur in `g_bytes_new`, due to an implicit cast from `gsize` to `guint`. The overflow happens in the call to `g_memdup` (gbytes.c, line 98). The reason is that `size` is a 64-bit `gsize`, but `g_memdup` takes a 32-bit `guint`.

```
GBytes *
g_bytes_new (gconstpointer data,
             gsize         size)
{
  g_return_val_if_fail (data != NULL || size == 0, NULL);

  return g_bytes_new_take (g_memdup (data, size), size);  <===  integer overflow
}
```

When the overflow occurs, it does not cause the code to crash immediately. Instead, `g_memdup` creates a much smaller buffer than it should. This causes `g_bytes_new` to return a `GBytes` object containing a much smaller data buffer than it's size would suggest. For example, if `size` is `0x100000008` then `g_bytes_new` will return a `GBytes` object that claims to contain a 4GB buffer, but actually contains an 8 byte buffer.

We have attached a proof-of-concept which demonstrates that it is possible to trigger the overflow. The proof-of-concept triggers the overflow via `polkit-agent-helper-1`, which is a SUID binary. Luckily the poc only causes `polkit-agent-helper-1` to crash with a `SIGABRT`, due to an assertion failure. However, GLib is a very widely used library, so it is possible that other attack vectors exist.

To run the poc:

```
gcc polkithelperabort.c -o polkithelperabort
./polkithelperabort <username>
```

The poc will ask for the user's password, which is sent to `polkit-agent-helper-1` in the normal way, along with a 4GB "cookie", which triggers the overflow. Although the poc will only work with a valid password, it will work for any user account. So if you are worried about plugging a genuine password into the poc, just create a temporary user account for running the poc, and delete it when you are done.

The poc should trigger an assertion failure, with an error message like this:

```
GLib-GIO:ERROR:../glib/gio/gdbusmessage.c:2399:append_value_to_blob: assertion failed: (g_utf8_validate (v, -1, &end) && (end == v +
Bail out! GLib-GIO:ERROR:../glib/gio/gdbusmessage.c:2399:append_value_to_blob: assertion failed: (g_utf8_validate (v, -1, &end) && (
```

Impact

The poc which we have provided only causes a harmless crash. We are not currently aware of an exploitable attack vector for this issue. However, there is a risk that it could lead to memory corruption vulnerability, which could potentially be used to gain code execution in an application that uses GLib.

Remediation

The issue looks easy to fix by changing the type of the `byte_size` parameter of `g_memdup`, so we have a posted a merge request with that change.

Resources

Source code for poc: polkithelperabort.c

Credit

This issue was discovered and reported by GHSL team member @kevinbackhouse (Kevin Backhouse).

Contact

You can contact the GHSL team at `securitylab@github.com`, please include a reference to `GHSL-2021-045` in any communication regarding this issue.

Disclosure Policy

This report is subject to our coordinated disclosure policy.

Edited 1 year ago by Simon McVittie

Drag your designs here or click to upload.

Tasks 0

No tasks are currently assigned. Use tasks to break down this issue into smaller parts.

Linked items 1

g_memdup has an inconsistent interface, should be replaced or deprecated
#169

Related merge requests 16

Change type of byte_size to fix integer overflow. !1924	
Add g_memdup2() !1926	2.67.3
Backport !1926 "Add g_memdup2()" to glib-2-66 !1927	2.66.6
gstrfuncs: Deprecate g_memdup() in favour of g_memdup2() !1928	2.67.4
giochannel: Fix length_size bounds check !1931	2.66.7
gtlspassword: Fix inverted assertion !1932	2.66.7

[2.66] Fix regressions in 2.66.6 where negative gssize indicates strlen()
!1933 · 2.66.7

gkeyfilesettingsbackend: Fix basename handling when group is unset
!1941 · 2.67.4

CVE-2021-27218: gbytearray: Do not accept too large byte arrays
!1942

Backport !1941 "gkeyfilesettingsbackend: Fix basename handling when group is unset" to glib-2-66
!1943 · 2.66.7

Backport !1942 "CVE-2021-27218: gbytearray: Do not accept too large byte arrays" to glib-2-66
!1944 · 2.66.7

Backport CVE-2021-27219 integer overflow fix to GLib 2.58
!2000

toolbar: Use HdyHeaderBar
geary!658

Change all g_memdup() to g_memdup2()
gnome-remote-desktop!32

Change all g_memdup() to g_memdup2()
gnome-shell!1637

backports: Add a backport of g_memdup2()
libgline!46

When these merge requests are accepted, this issue will be closed automatically.

Activity

Kevin Backhouse changed the description 1 year ago

Emmanuele Bassi @ebassi · 1 year ago (Maintainer)

Thanks for your bug report.

Since we can't change `g_memdup()`'s arguments, we should at least validate that we aren't getting a size that would overflow `g_memdup()` before calling it.

Alternatively, we could have an internal copy of `g_memdup()` inside `gbytes.c` using the appropriate argument type.

Just as a side note: merge requests in GitLab cannot be made confidential. If you have a fix for a confidential issue, and you wish to keep the fix under embargo as well, we recommend attaching a patch to the confidential issue.

Philip Withnall @pwithnall · 1 year ago (Maintainer)

> Just as a side note: merge requests in GitLab cannot be made confidential. If you have a fix for a confidential issue, and you wish to keep the fix under embargo as well, we recommend attaching a patch to the confidential issue.

Which means the embargo has effectively been broken already, by creating an MR. Note also that git branches are public (so pushing a git branch and not creating an MR won't work either).

Philip Withnall @pwithnall · 1 year ago (Maintainer)

> Alternatively, we could have an internal copy of `g_memdup()` inside `gbytes.c` using the appropriate argument type.

That would work to fix this particular case, but there are ~60 uses of `g_memdup()` in GLib's codebase (and more outside), and I suspect a lot of them will (reasonably) assume it accepts `size_t`.

I suggest we:

1. Add a `g_memdup2()` (naming suggestions welcome) which has a `gsize` argument
2. Deprecate `g_memdup()` in favour of it
3. Do this all before the API freeze on 2021-02-11

Philip Withnall @pwithnall · 1 year ago (Maintainer)

I'm working on this now

Emmanuele Bassi @ebassi · 1 year ago (Maintainer)

I'm fine with `g_memdup2()`.

The main problem with adding a new function this late in the game, and deprecating `g_memdup()`, is the amount of churn that will likely break CI in every other GNOME project.

I count >600 occurrences of `g_memdup()` in the core jhbuild moduleset alone. If we do this, we'll need to notify all maintainers beforehand.

Philip Withnall @pwithnall · 1 year ago (Maintainer)

One option would be to skip the deprecation for now, but I fear that would just lead to a load of zero-day vulnerabilities in various projects that use GLib. I don't see a nice solution 🙁

I'll put the deprecation in a separate commit so we can ponder it separately. It shouldn't block the addition of `g_memdup2()` or fixing all the instances of it in GLib.

Kevin Backhouse @kevin.backhouse · 1 year ago (Author)

I am sorry about the merge request. I was trying to be helpful by offering a fix, but it was a mistake to make it public and I should have put the diff in this issue instead. I am also sorry that this is much more complicated to fix than I realized.

Philip Withnall @pwithnall · 1 year ago (Maintainer)

No worries, it's not particularly obvious that issues can be private but MRs can't (I would love it if they could be, but it requires some faffing about with private forks). Thanks for the finding the issue and reporting it. :) I look forward to not having to deal with a zero-day disclosure next time though!

Michael Catanzaro @mcatanzaro · 1 year ago (Maintainer)

Is it possible to define a g_memdup() macro that calls the new g_memdup2()? Then software recompiled against newer glib would get switched over automatically without breaking ABI. Right?

I wonder where else we are using guint/gint where gsize/gssize is required. :/

Michael Catanzaro @mcatanzaro · 1 year ago (Maintainer)

> Is it possible to define a g_memdup() macro that calls the new g_memdup2()? Then software recompiled against newer glib would get switched over automatically without breaking ABI. Right?

We do this to g_steal_pointer() to add typesafety, for example. (Although g_steal_pointer() evaluates its argument twice, which we shouldn't do here.)

Philip Withnall @pwithnall · 1 year ago (Maintainer)

> Then software recompiled against newer glib would get switched over automatically without breaking ABI. Right?

Yes, but that wouldn't address the likely other arithmetic errors in the calling code (if you've got one arithmetic error, you've probably got more). In about half of the `g_memdup()` calls I've fixed in GLib, changes to the surrounding code have been needed to actually fix the issue.

Philip Withnall @pwithnall · 1 year ago (Maintainer)

Adding macro wrappers around functions always seems to cause a long tail of problems. The `g_memdup()` call sites in other projects are going to need to be fixed at some point, why not now? The fixes in other projects should not need to be done by the API freeze, so there's a bit of time.

Edited by Philip Withnall 1 year ago

Please register or sign in to reply

Philip Withnall added 1 Security, @ibes, gstreamer labels 1 year ago

Philip Withnall mentioned in merge request !1926 (merged) 1 year ago

Philip Withnall @pwithnall · 1 year ago

Maintainer

Fix available in !1926 (merged). Given that this is essentially disclosed already, I thought that the benefit of putting it up for a review publicly as an MR outweighed the downside of giving it more visibility. This way, it can be reviewed in GitLab rather than as attached patches.

Please can people look at it as a matter of urgency, since this is a zero-day (and the new API is also subject to the API freeze on 2021-02-11). Sorry about this.

A suggestion for handling the fallout in other modules:

1. Review this
2. Merge everything except the commit which deprecates `g_memdup()`
3. Release GLib 2.67.3 and 2.66.6 with the fix
4. Tell module maintainers to fix their modules quickly
5. Merge the deprecation in time for the API freeze (and then release 2.67.4)

This will at least give module maintainers a new GLib release (2.67.3) to build and test against which has the new `g_memdup2()` API but not the deprecation.

Thoughts?

Philip Withnall mentioned in merge request !1927 (merged) 1 year ago

Philip Withnall closed via commit 20cfc75d 1 year ago

Philip Withnall mentioned in merge request !1928 (merged) 1 year ago

Philip Withnall @pwithnall · 1 year ago

Maintainer

Fix has landed on `master` as !1926 (merged), and a backport is available for `glib-2-66` as !1927 (merged). I don't plan to backport to any older releases, as we don't support them. The deprecation MR is !1928 (merged) and I'll merge it nearer the time of the API freeze.

I've mailed desktop-devel to announce the need to port to `g_memdup2()`, and cross-posted to Discourse.

I'll work on releasing 2.67.3 right now, and then 2.66.6 once !1927 (merged) is merged.

I'm going to make this issue un-confidential as everything's public already.

Edited by Philip Withnall 1 year ago

Philip Withnall made the issue visible to everyone 1 year ago

Jonas Ådahl mentioned in commit jadahl/mutter@65da83b7 1 year ago

Jonas Ådahl mentioned in commit jadahl/gnome-shell@f2070380 1 year ago

Jonas Ådahl mentioned in merge request gnome-shell!1637 (merged) 1 year ago

Jonas Ådahl mentioned in commit jadahl/mutter@ad1af0bb 1 year ago

Jonas Ådahl mentioned in commit jadahl/mutter@5b092396 1 year ago

Jonas Ådahl mentioned in commit jadahl/gnome-remote-desktop@1cba15cd 1 year ago

Jonas Ådahl mentioned in merge request gnome-remote-desktop!32 (merged) 1 year ago

Jonas Ådahl mentioned in commit jadahl/mutter@30e1c51b 1 year ago

Philip Withnall @pwithnall · 1 year ago

Maintainer

2.67.3 and 2.66.6 released and everything's wrapped up here.

Rico Tzschichholz mentioned in commit vala@4209a8bf 1 year ago

Jonas Ådahl mentioned in commit jadahl/gnome-shell@3d737b95 1 year ago

Jonas Ådahl mentioned in commit jadahl/gnome-shell@f14af9ab 1 year ago

Jonas Ådahl mentioned in commit jadahl/gnome-remote-desktop@3f05ea2b 1 year ago

Jonas Ådahl mentioned in commit jadahl/gnome-remote-desktop@43c96612 1 year ago

Jonas Ådahl mentioned in commit jadahl/gnome-remote-desktop@4932bb1b 1 year ago

Jonas Ådahl mentioned in commit jadahl/gnome-shell@463000d0 1 year ago

Jonas Ådahl mentioned in commit jadahl/gnome-remote-desktop@19122267 1 year ago

Jonas Ådahl mentioned in commit jadahl/gnome-remote-desktop@1cdeeba6 1 year ago

Jonas Ådahl mentioned in commit jadahl/gnome-remote-desktop@c9655445 1 year ago

Jonas Ådahl mentioned in commit jadahl/gnome-remote-desktop@4dc66a68 1 year ago

Jonas Ådahl mentioned in commit jadahl/gnome-remote-desktop@c23cd885 1 year ago

Michael Gratton mentioned in merge request geary!658 (merged) 1 year ago

Philip Withnall mentioned in commit 5e5f75a7 1 year ago

Philip Withnall mentioned in commit be883434 1 year ago

Philip Withnall mentioned in commit 6110caea 1 year ago

Philip Withnall mentioned in commit 8736b7c1 1 year ago

Philip Withnall mentioned in commit 0cbad673 1 year ago

Philip Withnall mentioned in commit f9ee2275 1 year ago

Philip Withnall mentioned in commit 2aaf593a 1 year ago

Philip Withnall mentioned in commit ba8ca443 1 year ago

Philip Withnall mentioned in commit 65ec7f4d 1 year ago

Philip Withnall mentioned in commit 777b95a8 1 year ago

Philip Withnall mentioned in commit ecdf9140 1 year ago

Simon McVittie @smcv · 1 year ago

Maintainer

Distributions backporting fixes for this integer overflow to older GLib releases should note that there were some regressions caused by !1926 (merged) and !1927 (merged). The ones I know about so far were fixed in !1931 (merged) and !1932 (merged) for 2.67.x, and in !1933 (merged) for 2.66.x.

Philip Withnall @pwithnall · 1 year ago

Maintainer

There is an additional regression fixed by !1941 (merged), and a backport to 2.66 is available as !1943 (merged).

Edited by Philip Withnall 1 year ago

Philip Withnall @pwithnall · 1 year ago Maintainer

And a related security fix (not a regression!) in !1942 (merged). Backport in !1944 (merged). I'll do a 2.66.7 release once the backport has landed, to tie things together.

Edited by Philip Withnall 1 year ago

Philip Withnall @pwithnall · 1 year ago Maintainer

2.66.7 released.

Please register or sign in to reply

Philip Withnall mentioned in commit `f8cf0b86` 1 year ago

Philip Withnall mentioned in commit `73b293fd` 1 year ago

Philip Withnall mentioned in commit `f10101b9` 1 year ago

Philip Withnall mentioned in commit `19470722` 1 year ago

Philip Withnall mentioned in commit `81a45423` 1 year ago

Philip Withnall mentioned in commit `41d5eeda` 1 year ago

Philip Withnall mentioned in commit `9acebef7` 1 year ago

Philip Withnall mentioned in commit `7781a9cb` 1 year ago

Philip Withnall mentioned in commit `a2e38fd2` 1 year ago

Philip Withnall mentioned in commit `a8b204ff` 1 year ago

Philip Withnall mentioned in commit `0cc11f74` 1 year ago

Philip Withnall mentioned in commit `20cfc75d` 1 year ago

Simon McVittie @smcv · 1 year ago Maintainer

I've requested a CVE ID from MITRE for this issue.

Simon McVittie @smcv · 1 year ago Maintainer

CVE-2021-27219 has been allocated for this.

Simon McVittie changed title from **GHSL-2021-045: integer overflow in g_bytes_new/g_memdup** to **CVE-2021-27219 (GHSL-2021-045): integer overflow in g_bytes_new/g_memdup** 1 year ago

Philip Withnall mentioned in commit `24b94469` 1 year ago

Simon McVittie mentioned in commit `3e0bb3bf` 1 year ago

Philip Withnall mentioned in commit `0ace82d7` 1 year ago

Philip Withnall mentioned in commit `c921c826` 1 year ago

Philip Withnall mentioned in commit `2424eeaf` 1 year ago

Simon McVittie mentioned in commit `d9c23ef8` 1 year ago

Philip Withnall mentioned in commit `2fec1db0` 1 year ago

Philip Withnall mentioned in commit `52458318` 1 year ago

Simon McVittie mentioned in commit `65be56a1` 1 year ago

Philip Withnall mentioned in commit `8367c73d` 1 year ago

Philip Withnall mentioned in commit `1ad50f84` 1 year ago

Simon McVittie mentioned in merge request !2000 (merged) 1 year ago

Simon McVittie mentioned in commit `1436fedb` 1 year ago

Philip Withnall mentioned in commit `7f308de3` 1 year ago

Philip Withnall mentioned in commit `5c26b6a7` 1 year ago

Philip Withnall marked #169 (closed) as a duplicate of this issue 1 year ago

Philip Withnall marked this issue as related to #169 (closed) 1 year ago

Philip Withnall mentioned in commit `4669c799` 1 year ago

Simon McVittie mentioned in commit `324a29c6` 1 year ago

Philip Withnall mentioned in commit `b1a857f8` 1 year ago

Philip Withnall mentioned in commit `64bcc117` 1 year ago

Philip Withnall mentioned in commit `5d1d065a` 1 year ago

Simon McVittie mentioned in commit `41baea68` 1 year ago

Philip Withnall mentioned in commit `8b55fa39` 1 year ago

Philip Withnall mentioned in commit `54d088f4` 1 year ago

Philip Withnall mentioned in commit `66c44d68` 1 year ago

Philip Withnall mentioned in commit `c1ded863` 1 year ago

Simon McVittie mentioned in commit `libglnx@73210dce` 1 month ago

Simon McVittie mentioned in merge request libglnx!46 (merged) 1 month ago

Simon McVittie mentioned in commit `libglnx@8a293253` 1 month ago

Please register or sign in to reply