

file-roller symlink attack

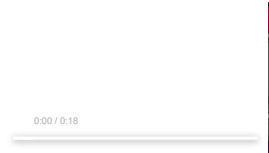
Summary: A malicious archive may be able to overwrite arbitrary files with file-roller

Steps to reproduce: 1- Download symlink.tar 2- Extract it with file-roller

Proof of concept: 2020-11-17_12-27-39.mp4

Version: Ubuntu 20.10

Thank you,



[file-roller symlink attack](#)

[symlink.tar](#)

📁 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 0	
Related merge requests 3	
Resolve "file-roller symlink attack"	2.67.6
Backport 12325 "file-roller symlink attack" to glib-2-66	2.66.8
Backport CVE-2021-28153 symlink fix to GLib 2.58	

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

Activity

- Yiğit Can Yılmaz** @yigitcanvilmaz046 · 2 years ago Author
[@mcatanzaro](#) Is there any progress on this issue? It's been six days.
- Michael Catanzaro** @mcatanzaro · 2 years ago Maintainer
Well you can see everything here. There are no comments from anybody else, so the file-roller maintainers have not yet acknowledged this issue.
- Yiğit Can Yılmaz** @yigitcanvilmaz046 · 2 years ago Author
Hello [@mcatanzaro](#), I have feel concern about this report. Please look at the <https://security.gnome.org/>. You will see "Your submission will generally be acknowledged within one business day, and you'll receive a more detailed response to your email within five business days indicating the next steps in handling your report." sentence
- Michael Catanzaro** @mcatanzaro · 2 years ago Maintainer
GNOME Security has acknowledged your submission.

The file-roller package maintainers have not. We can't help with that. They are volunteers and it's not unusual for GNOME maintainers to ignore bug reports, including security reports.

Keeping the issue confidential only makes sense if it's going to be fixed in the foreseeable future. What we can do, if you want, is make this issue public, so you can request a CVE and publicize the flaw. Then at least people could have some heads-up to be careful about using file-roller to open archives.
- Michael Catanzaro** changed due date to February 15, 2021 1 year ago
- Michael Catanzaro** @mcatanzaro · 1 year ago Maintainer
Reminder: this vulnerability will be made public on Feb 15.
- Bastien Nocera** @hadess · 1 year ago Developer
Tagging [@oholy](#), this looks like the same problem fixed in [gnome-autoar@ad0b67e6](#)
- Ondrej Holy** @oholy · 1 year ago Maintainer
Hmm, if this is not fixed by the commit [file-roller@21dfcd5](#) then we may have a problem in gnome-autoar as well although it uses different code. I will have a look at the archive and try to reproduce it...
- Ondrej Holy** added [1 Bug](#), [1 Security](#), [2 Needs Diagnosis](#) labels 1 year ago
- Ondrej Holy** @oholy · 1 year ago Maintainer
Ok, I can reproduce it with file-roller, but not with gnome-autoar. I am going to analyze what is wrong...
- Ondrej Holy** assigned to [@oholy](#) 1 year ago
- Ondrej Holy** @oholy · 1 year ago Maintainer
The archive contains symlink `foo` which points to `/tmp/foo` and file `foo`. gnome-autoar by default skips conflicting files, so it ignores the file `foo` (although it was not probably intended as per the code but that's another story). file-roller does not care about conflicts and simply calls `g_file_replace`, which should not be a problem as well. However, something seems to be wrong with `g_file_replace`, because it creates an empty file in the location of the symlink target (apart from overwriting the symlink!) So changing component to glib. [@youthful](#), don't you have the capacity to look at it?
- Ondrej Holy** unassigned [@oholy](#) 1 year ago
- Ondrej Holy** removed [2 Needs Diagnosis](#) label 1 year ago
- Ondrej Holy** added [Glib](#) label 1 year ago
- Ondrej Holy** @oholy · 1 year ago Maintainer
The following snippet illustrates what is wrong:

```
#include <glib.h>
#include <glib/gstdio.h>
#include <gio/gio.h>

#define PATH "/tmp/foo"
#define TARGET "/tmp/bar"

int
main (int argc, const char* argv[])
{
    GFile *file;
    GError *error = NULL;
    GFileOutputStream *stream;
```

```
// Be sure that none of the testing files exists yet
g_unlink (PATH);
g_unlink (TARGET);

// Create symlink PATH which points to TARGET
file = g_file_new_for_path (PATH);
g_file_make_symbolic_link (file, TARGET, NULL, &error);
g_assert_no_error (error);
g_assert_true (g_file_test (PATH, G_FILE_TEST_IS_SYMLINK));

// Replace symlink with regular file
stream = g_file_replace (file, NULL, FALSE, G_FILE_CREATE_REPLACE_DESTINATION, NULL, &error);
g_assert_no_error (error);
g_output_stream_write_all (G_OUTPUT_STREAM (stream), "foo", 4, NULL, NULL, &error);
g_output_stream_close (G_OUTPUT_STREAM (stream), NULL, &error);
g_assert_no_error (error);

// Be sure that symlink was replaced, not its target
g_assert_true (g_file_test (PATH, G_FILE_TEST_IS_REGULAR));
g_assert_false (g_file_test (TARGET, G_FILE_TEST_EXISTS)); // The file TARGET exists but it shouldn't!
}
```

Edited by [Ondrej Holy](#) 1 year ago

[Philip Withnall](#) @[pwithnall](#) · 1 year ago

Maintainer

[@pwithnall](#), don't you have the capacity to look at it?

Selfishly, I'd really rather not right now, as I've just lost two days to other security issues in Glib which pre-empted other things. If I have to work on this, it will be next week. Please don't disclose it before then, [@cncarlomagno046](#) / [@mcatanzaro](#).

From a quick look, I'm not sure this is actually exploitable to overwrite arbitrary files — only to create arbitrary (empty) files which didn't previously exist. Essentially, the issue is Glib using `O_CREAT` when opening the symlink to check whether it's a symlink, around line 861 in `handle_overwrite_open()` in `glocalfileoutputstream.c`.

You can see this with your reproducer above by commenting out the `g_unlink()` calls and creating `/tmp/bar` with some content before running the test. The content is unchanged after running the test.

I haven't opened the original `symlink.tar` from the report to verify whether it can overwrite arbitrary content.

To be sure, being able to create empty arbitrary files by opening an attacker-controlled tar file is a security issue which needs fixing quickly, but it's nowhere near as bad as writing arbitrary content to arbitrary files.

[Ondrej Holy](#) @[oholy](#) · 1 year ago

Maintainer

it's nowhere near as bad as writing arbitrary content to arbitrary files.

Just for your info, this is unfortunately also reality, see [file-roller#108 \(closed\)](#), which is also why I tried to pass this one to you..

[Michael Catanzaro](#) changed due date to April 15, 2021 1 year ago

[Ondrej Holy](#) mentioned in issue [gnome-autoar#12 \(closed\)](#) 1 year ago

[Philip Withnall](#) assigned to [@pwithnall](#) 1 year ago

[Philip Withnall](#) @[pwithnall](#) · 1 year ago

Maintainer

OK, I have a fix for the issue reproduced by [your test program](#), but need to add a whole load more unit tests to avoid regressions.

[Philip Withnall](#) @[pwithnall](#) · 1 year ago

Maintainer

Here's a fix for the issue with `g_file_replace()` creating the target of a dangling symlink, which I understand is the extent of the problem in Glib.

Please review the patches using comments in this issue, and keep them confidential. (Merge requests on GitHub are not confidential, sorry.) Once they're reviewed and ready to merge, we can agree on a date to lift embargo, at which point the patches will simultaneously land on Glib master and in distros. I expect this will get released in the [2.67.6 release](#) (due 2021-03-11). I don't expect this will be ready to land for 2.67.5 (due, er, yesterday).

[@smcv](#), you may be interested in this from the Debian side.

There are 6 patches in the series. Subject to review, I plan to backport all of them to 2.66. If distros want to backport anything to older Glib stable releases, 'glocalfileoutputstream: Fix CREATE_REPLACE_DESTINATION with symlinks' is the patch you really want.

The patches bump the code coverage of `glocalfileoutputstream.c` up from 51.4%, 85.3%, 44.2% to 63.1%, 88.2%, 57.0% of lines, functions and branches, respectively. Mostly what's missing coverage now (in the code relevant to this issue) are the error paths which are only reachable via race conditions.

[0001-glocalfileoutputstream-Fix-a-typo-in-a-comment-patch](#)

[0002-tests-Stop-using-o_test_base-in-file-tests-patch](#)

[0003-glocalfileoutputstream-Fix-CREATE_REPLACE_DESTINATION-patch](#)

[0004-glocalfileoutputstream-Add-a-missing-O_CLOEXEC-flag-patch](#)

[0005-tests-Add-comprehensive-tests-for-static-behaviour-o-patch](#)

[0006-oioerror-Add-conversion-from-ENXIO-to-G_IO_ERROR_NOT-patch](#)

Edited by [Philip Withnall](#) 1 year ago

[Philip Withnall](#) changed milestone to [%2.67.6](#) 1 year ago

[Michael Catanzaro](#) @[mcatanzaro](#) · 1 year ago

Maintainer

Regarding patch 4: we sometimes use a do/while loop and check for `EINTR` when using `fcntl`'s `F_SETFD`. That said, I'm not sure why, because it looks like `EINTR` can never be returned, at least not on Linux.

[Philip Withnall](#) @[pwithnall](#) · 1 year ago

Maintainer

I'm going to assume that `nan_fcntl1` is correct in saying that `EINTR` can only be returned by the locking `fcntl` modes, and hence isn't applicable here. Over half of the other `F_SETFD` calls in Glib don't check for `EINTR`.

[Philip Withnall](#) @[pwithnall](#) · 1 year ago

Maintainer

Did you review the rest of the patches closely, [@mcatanzaro](#), or was that a drive-by comment?

It would be good to get these in 2.68, and the 2.67.6 release (due 2021-03-11) is hard code freeze. The security disclosure fell around this is going to take a few days, so these need review from at least one other person fairly soon.

[Michael Catanzaro](#) @[mcatanzaro](#) · 1 year ago

Maintainer

That was very much a drive-by comment, sorry. I'm pretty sure your patches are not malware, which is nice.

Please [register](#) or [sign in](#) to reply

[Ondrej Holy](#) mentioned in commit [gnome-autoar@f397a6c](#) 1 year ago

[Emmanuel Basi](#) @[ebasi](#) · 1 year ago

Maintainer

The 0003 patch looks good to close the issue: a style nitpick would be to replace `(flags & G_FILE_CREATE_REPLACE_DESTINATION)` with `gboolean replace_destination_set = (flags & G_FILE_CREATE_REPLACE_DESTINATION) != 0`, but it's just a nitpick.

The large test in 0005 is pretty big, but I don't see anything wrong with its implementation.

The other patches are trivial changes, and I have no objections to them.

[Philip Withnall](#) @[pwithnall](#) · 1 year ago

Maintainer

Thanks, I'll fix that nitpick before merging, but will wait for [@sdroege](#)'s review before doing any changes.

[Sebastian Dröge](#) @[sdroege](#) · 1 year ago

Maintainer

In patch 0003:

- "Fix that by not opening the destination file if it's a symlink, and adjusting the rest of the code to cope with" in the commit message. Maybe also explain here what it does instead and what the effect would be for applications calling it on dangling symlinks. AFAIU this doesn't actually change anything in behaviour but just replaces the symlink (dangling or not), and in case of a

dangling symlink it wouldn't create an empty file at the destination (the actual problem here, also completely useless to do) but just replace the symlink with the file. From just reading the commit message it's not clear to me what will happen (you could think for example that it is an error now to call it on dangling symlinks).

In patch 0004:

- Maybe that `#ifdef` part should move to some private header, it's copied in many places. But can also happen in another MR later, not really important for this problem here.

In patch 0005:

- Almost 1000 lines of new tests 🍷
- `mknod()` is not available on Windows probably? Also does the symlink API work like this on Windows? On NTFS you can have symlinks IIRC but do they work the same way API-wise? How about `G_FILE_ATTRIBUTE_UNIX_MODE` ?

Looks all good to me otherwise. Gitlab MRs make reviews so much easier, you really only notice that when having to review diff files again... too bad confidential MRs are a Gitlab EE feature.



Michael Catanzaro @mcatanzaro · 1 year ago

Maintainer

Looks all good to me otherwise. Gitlab MRs make reviews so much easier, you really only notice that when having to review diff files again... too bad confidential MRs are a Gitlab EE feature.

There's a button to create confidential MRs at the top of this page: it's not an enterprise feature. But it requires creating a private fork, and GNOME Gitlab has some configuration to prevent that, so it won't work here.



Philip Withnall @pwithnall · 1 year ago

Maintainer

In patch 0003:

- "Fix that by not opening the destination file if it's a symlink, and adjusting the rest of the code to cope with" in the commit message. ...

Expanded a bit.

In patch 0004:

- Maybe that `#ifdef` part should move to some private header, it's copied in many places. But can also happen in another MR later, not really important for this problem here.

Adding a blanket definition of missing `open()` flags might mask situations where they are undefined incorrectly, or where the behaviour when they're not supported should not just be to ignore things and carry on.

In patch 0005:

- `mknod()` is not available on Windows probably? Also does the symlink API work like this on Windows? On NTFS you can have symlinks IIRC but do they work the same way API-wise? How about `G_FILE_ATTRIBUTE_UNIX_MODE` ?

Hide the whole lot behind `G_OS_UNIX` for now. A follow-up MR could expose the tests to Windows, but that can be done later and with the benefit of CI.



Sebastian Dröge @sdröge · 1 year ago

Maintainer

Ok, perfect ☺

Please [register](#) or [sign in](#) to reply



Philip Withnall @pwithnall · 1 year ago

Maintainer

The updated patches with review comments applied. Thanks for the reviews! I'll start coordinating disclosure on this now.

[0. 0001-giofileoutputstream-Fix-a-typo-in-a-comment.patch](#)

[0. 0002-tests-Stop-using-o_test_base-in-file-tests.patch](#)

[0. 0003-giofileoutputstream-Factor-out-a-flag-check.patch](#)

[0. 0004-giofileoutputstream-Fix-CREATE_REPLACE_DESTINATIO.patch](#)

[0. 0005-giofileoutputstream-Add-a-missing-O_CLOEXEC-flag.patch](#)

[0. 0006-tests-Add-comprehensive-tests-for-static-behaviour-o.patch](#)

[0. 0007-gioerror-Add-conversion-from-ENXIO-to-G_IO_ERROR_NOT.patch](#)



Philip Withnall mentioned in commit [a90fccc](#) · 1 year ago



Philip Withnall mentioned in commit [a2b64c8](#) · 1 year ago



Philip Withnall mentioned in merge request [11981 \(merged\)](#) · 1 year ago



Philip Withnall mentioned in merge request [11982 \(merged\)](#) · 1 year ago



Philip Withnall mentioned in commit [87e19535](#) · 1 year ago



Philip Withnall mentioned in commit [317b3b58](#) · 1 year ago



Philip Withnall @pwithnall · 1 year ago

Maintainer

Making the issue public as the MRs to fix it are public: [11981 \(merged\)](#) and [11982 \(merged\)](#).



Philip Withnall made the issue visible to everyone · 1 year ago



Philip Withnall closed via commit [c8b528f1](#) · 1 year ago



Philip Withnall closed via commit [87e19535](#) · 1 year ago



Yigit Can Yilmaz @yigitcanvilmaz046 · 1 year ago

Author

[@mcatanzaro](#) Can you give CVE to this issue?

Thanks,



Philip Withnall @pwithnall · 1 year ago

Maintainer

You're welcome to request a CVE for it yourself. I think it's a waste of my time to do so, though.



Michael Catanzaro @mcatanzaro · 1 year ago

Maintainer

Hi Yigit, you can request a CVE using <https://cveform.mitre.org/>.

Distributions will not fix the issue unless somebody requests a CVE.



Simon McVittie @smcv · 1 year ago

Maintainer

I'll do the CVE request, I've had a lot of practice at that :-)



Simon McVittie @smcv · 1 year ago

Maintainer

CVE ID requested from MITRE.



Simon McVittie @smcv · 1 year ago

Maintainer

This is CVE-2021-28153.

Please [register](#) or [sign in](#) to reply




Ondrej Holy mentioned in commit [gnome-autoar#26d32e8](#) · 1 year ago




Philip Withnall mentioned in commit [c8b528f1](#) · 1 year ago



Philip Withnall mentioned in commit [4e6da27f](#) · 1 year ago

 [Simon McVittie](#) mentioned in merge request [12002 \(merged\)](#) 1 year ago

 [Philip Withnall](#) mentioned in commit [153fa888](#) 1 year ago

 [baniusingshan](#) mentioned in issue [#2649](#) 5 months ago

Please [register](#) or [sign in](#) to reply