Closed Issue created 2 years ago by Yiğit Can Yılmaz

CVE-2020-36314: GNOME Archive Manager Traversal Attack

0:00 / 0:06

2020-11-11_18-28-38

dirsymlink2a.tar

Summary: A malicious archive may be able to overwrite arbitrary files with GNOME Archive Manager

Steps to reproduce: 1- Download dirsymlink2a.tar 2- Extract it with Archive Manager

Proof of concept: 2020-11-11_18-28-38.mp4

Version: Ubuntu 20.10

Thank you,

Edited 1 year ago by Ondrej Holy

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

Activity

GitLab Support Bot mentioned in issue gnome-autoar#7 (closed) 2 years ago

Michael Catanzaro made the issue confidential 2 years ago

Yiğit Can Yılmaz @yigitcanyilmaz046 · 2 years ago Author

@mcatanzaro Is there any progress on this issue? It's been a week

Michael Catanzaro @mcatanzaro · 2 years ago Developer

If there was progress, it would be reflected in this ticket. file-roller is maintained by volunteers, so it's not surprising that it might take a while before the issue is assessed.

Yiğit Can Yılmaz @yigitcanyilmaz046 · 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 Developer

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 14, 2021 1 year ago

Michael Catanzaro changed due date to February 15, 2021 1 year ago

Michael Catanzaro @mcatanzaro · 1 year ago Developer

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@adb067e6`

Ondrej Holy @oholy · 1 year ago Developer

Hmm, if this is not fixed by the commit `21dfcdbf` 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 Developer

Ok, I can reproduce it with file-roller and unfortunatelly also with the recent gnome-autoar fix. I am going to analyze what is wrong...

Ondrej Holy assigned to @oholy 1 year ago

Ondrej Holy @oholy · 1 year ago Developer

The attached archive contains symlink `cur` which points to `.` , symlink `par` which points to `cur/..` and file with `par/moo` path. In `gnome-autoar@adb067e6` fix, there is problem with `g_file_resolve_relative_path` which resolves `/path/destination/cur/..` to `/path/destination` as it doesn't care about links, however, it should be just `/path` instead. So I have to probably implement custom `g_file_resolve_relative_path` which will resolve symlinks. Althought, file-roller implements this itself by `21dfcdbf`, it does the same mistake. I will try to provide fix for it, but I am not sure I will find time before the due date...

Ondrej Holy removed 2. Needs Diagnosis label 1 year ago

Michael Catanzaro @mcatanzaro · 1 year ago Developer

That's fine. Help is much appreciated.

The due date is just the date the issue will be made public. World will continue spinning if we miss by a little.

Ondrej Holy mentioned in issue glib#2325 (closed) 1 year ago

Ondrej Holy @oholy · 1 year ago Developer

I was thinking about this issue a bit more to be sure that we will fix it correctly now. Writing a custom file path resolver is just a room for other bugs, so it would be better to base our solution on something like `realpath(3)` to implement something like `realpath --canonicalize-missing --relative-base=...` if at all. Because, although I found a way how to create an archive over `tar` where a parent is a symlink, it seems that it is not possible to extract it over `tar -xf` as it fails with `Not a directory` (even with options like `--overwrite`). Also `tar` by default archives only symlinks and doesn't follow them, or follows them, which means that it doesn't contain symlinks at all. So I suppose that it would be better to refuse files with symlinks in parents completely. What do you think?

/cc @antoniof @pwithnall

Ondrej Holy added 2. RFC label 1 year ago

Michael Catanzaro @mcatanzaro · 1 year ago Developer

I don't think g_file_resolve_relative_path() is intended to resolve symlinks, though. Its documentation just says it resolves a relative path to an absolute path. So this is not a glib issue, right?

> So I suppose that it would be better to refuse files with symlinks in parents completely. What do you think?

I think the problem is that not all the world is `tar` . We also have to handle `zip` and `rar` and `7z` and such, right? I think symlinks really are allowed in `zip` at least? Edit: internet says yes

Edited by Michael Catanzaro 1 year ago

Philip Withnall @pwithnall · 1 year ago Developer

> I don't think g_file_resolve_relative_path() is intended to resolve symlinks, though. Its documentation just says it resolves a relative path to an absolute path. So this is not a glib issue, right?

Indeed, `g_file_resolve_relative_path()` only looks at the path strings. It doesn't resolve symlinks.

GLib doesn't provide a function which behaves like `realpath(3)` .

Please register or sign in to reply

Ondrej Holy @oholy · 1 year ago Developer

> So this is not a glib issue, right?

Yes, this exactly is not a glib bug, this is my fault that I used that function. The function does exactly what it says.

However, glib uses the same logic internally as well, which I suppose may lead to other issues in other places, e.g.:

```
$ pwd
/home/user
$ ln -s ../ LINK
$ ls LINK/../
# The content of the / directory...
$ gio list LINK/../
# The content of the /home/user directory!!!
```

But this is unrelated to this issue.

> I think the problem is that not all the world is `tar` . We also have to handle `zip` and `rar` and `7z` and such, right? I think symlinks really are allowed in `zip` at least? Edit: internet says yes

You are right, it is not supported only by TAR, but surprisingly almost all other formats. TAR was just used as a demonstration, I don't really have the capacity to investigate ZIP, 7Z, RAR internals. My point is that symlinks in parents sound unlogical to me because such a file can be always included using the real path if it is in the same destination dir and if it is not, then it won't be extracted anyway. I can imagine that such archives could be created for special purposes, but then they won't be most probably extracted over Nautilus but using special tools. I have just quickly looked at how KDE/ark deals with symlinks. Their libarchive plugin is based on archive_write_disk APIs, so libarchive is responsible for extraction. They use `ARCHIVE_EXTRACT_SECURE_SYMLINKS` and `ARCHIVE_EXTRACT_SECURE_NODOTDOT` flags, which causes errors if some file contains symlinks, or `..` in parents. See: https://github.com/KDE/ark/blob/master/plugins/libarchive/libarchiveplugin.cpp.

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

Ondrej Holy mentioned in issue gnome-autoar#17 (closed) 1 year ago

Ondrej Holy @oholy · 1 year ago Developer

So something like libarchive-Skip-files-with-symlinks-in-parents.patch could fix this security issue. The patch simply skips the problematic file if there is any symlink in parents. Somebody willing to review it? I have proposed similar change for gnome-autoar.

Ondrej Holy mentioned in commit `gnome-autoar@8109c368` 1 year ago

Emmanuele Bassi @ebassi · 1 year ago Developer

```
-                               if ((load_data->error == NULL) && _symlink_is_external_to_destination (file, archive_entry
-                                       g_hash_table_insert (external_links, g_object_ref (file), GINT_TO_POINTER (1));
+                               if (load_data->error == NULL)
+                                       g_hash_table_insert (symlinks, g_object_ref (file), GINT_TO_POINTER (1));
```

I know it's a change, but I think it'd be best to replace `g_hash_table_insert()` with `g_hash_table_add (symlinks, g_object_ref (file));` and then:

```
-               if (_g_file_is_external_link (parent, destination, external_links)) {
-                       external = TRUE;
+               if (g_hash_table_lookup (symlinks, parent) != NULL) {
+                       contains_symlinks = TRUE;
```

replace `if (g_hash_table_lookup (symlinks, parent) != NULL)` with `if (g_hash_table_contains (symlinks, parent))` .

Other than that, I think the change is correct.

Ondrej Holy @oholy · 1 year ago Developer

Thanks @ebassi, that definitely makes sense. So there is updated version: libarchive-Skip-files-with-symlinks-in-parents.patch. @paobac Any objections to merging this patch?

Ondrej Holy closed via commit `e970f496` 1 year ago

Ondrej Holy mentioned in commit `0bd4a14f` 1 year ago

Ondrej Holy mentioned in commit `4aaa7c65` 1 year ago

Paolo Bacchilega @paobac · 1 year ago Maintainer

Patch merged, thank you.

Michael Catanzaro made the issue visible to everyone 1 year ago

Ondrej Holy @oholy · 1 year ago Developer

Just note that I've just requested CVE number for it.

Ondrej Holy @oholy · 1 year ago Developer

CVE-2020-36314

Michael Catanzaro @mcatanzaro · 1 year ago Developer

For a vulnerability disclosed in 2021? OK...

Ondrej Holy @oholy · 1 year ago Developer

That's what I get from mitre 🤷

Please register or sign in to reply

Ondrej Holy changed title from **GNOME Archive Manager Traversal Attack** to **CVE-2020-36314: GNOME Archive Manager Traversal Attack** 1 year ago

Ondrej Holy mentioned in commit `e970f496` 8 months ago

Please register or sign in to reply