Integer Overflow in num_images #1338

Closed NigelX opened this issue on Mar 24, 2021 · 10 comments · Fixed by #1395

NigelX commented on Mar 24, 2021 • edited

Hello openjpeg2 team,
I found an integer overflow vulnerability in the command line options.

```
-ImgDir
```

If there are many files in the imgdir directory The number of files read by opj_compress will overflow.

openjpeg2(tested with revision * master `0bda718`).

run commd

```
./opj_compress -ImgDir testcase/ -OutFor outcase/t.jp2
```

asan info

```
Folder opened successfully
UndefinedBehaviorSanitizer:DEADLYSIGNAL
==1852564==ERROR: UndefinedBehaviorSanitizer: SEGV on unknown address 0x000001183310 (pc 0x7ffff764cefa bp 0x0000000fffff sp 0x7fffffff3988 T1852564)
==1852564==The signal is caused by a WRITE memory access.
    #0 0x7ffff764cefa  /build/glibc-eX1tMB/glibc-2.31/string/../sysdeps/x86_64/multiarch/strcpy-avx2.S:630
    #1 0x42d9a5 in load_images /home/test/Downloads/openjpeg/src/bin/jp2/opj_compress.c:508:9
    #2 0x429366 in main /home/test/Downloads/openjpeg/src/bin/jp2/opj_compress.c:1924:13
    #3 0x7ffff74e70b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/csu/../csu/libc-start.c:308:16
    #4 0x408c7d in _start (/home/test/Downloads/openjpeg/fast_build64/bin/opj_compress+0x408c7d)

UndefinedBehaviorSanitizer can not provide additional info.
SUMMARY: UndefinedBehaviorSanitizer: SEGV /build/glibc-eX1tMB/glibc-2.31/string/../sysdeps/x86_64/multiarch/strcpy-avx2.S:630
==1852564==ABORTING
```

```
/* Read directory if necessary */
if (img_fol.set_imgdir == 1) {
    num_images = get_num_images(img_fol.imgdirpath);
    dirptr = (dircnt_t*)malloc(sizeof(dircnt_t));
    if (dirptr) {
        dirptr->filename_buf = (char*)malloc(num_images * OPJ_PATH_LEN * sizeof(
                char));
        dirptr->filename = (char**) malloc(num_images * sizeof(char*));
        if (!dirptr->filename_buf) {
            ret = 0;
            goto fin;
        }
        for (i = 0; i < num_images; i++) {
            dirptr->filename[i] = dirptr->filename_buf + i * OPJ_PATH_LEN;
        }
    }
    if (load_images(dirptr, img_fol.imgdirpath) == 1) {
        ret = 0;
        goto fin;
    }
    if (num_images == 0) {
        fprintf(stdout, "Folder is empty\n");
        ret = 0;
        goto fin;
    }
```

When num_images is equal to 1048576, multiplying with OPJ_PATH_LEN will produce an overflow result of 0
poc.zip

HX from **Topsec alpha Security Team**

1

carnil commented on Apr 14, 2021

This appears to have been assigned CVE-2021-29338

1

Abhishek-sin commented on Apr 26, 2021

Is there any manual fix we can use/apply here till we get an patch update ??

stevebeattie commented on May 19, 2021

It looks like the pull request #1346 is intended to cover this issue as well; I believe Alpine Linux has already released an update for the issue with an earlier iteration of the proposed pull request.

tony-- commented on Jun 29, 2021

#1346 was replaced with `f0629cb` . Does that mean CVE-2021-29338 is fixed in master, **@rouault**?

rouault commented on Jun 29, 2021 Collaborator

> #1346 was replaced with f0629cb. Does that mean CVE-2021-29338 is fixed in master, **@rouault**?

I don't think so. I don't see `f0629cb` changing the code path pointed above

baparham mentioned this issue on Dec 21, 2021

opj_{compress,decompress,dump}: fix possible buffer overflows in path manipulation functions #1346

Closed

baparham commented on Jan 12 Contributor

now that we might be back after some time away, I'll ping again but over in this issue. **@rouault** or **@kaniini** are there any plans or PRs in the works to fix this cve since it was not included in `f0629cb` and #1346 wasn't merged fully?

rouault commented on Jan 12 Collaborator

> are there any plans or PRs in the works to fix this cve since it was not included in f0629cb and #1346 wasn't merged fully?

if you believe there's something left to fix, please issue a pull request to fix it. That's the effective way to make changes happen

baparham commented on Jan 12 Contributor

I understand and agree, but I am not really good at c these days. Since you and **@kaniini** have previously made changes in this area, I figured either of you two would be the quickest at making such a PR, and ensuring that it actually is correct. A PR from me would basically be trying to copy and paste code from **@kaniini** where it fits the latest master code, which seems inappropriate and bug prone.

Either way, it sounds like the answer is no, so I'll try and cobble something together (just what you want to hear when fixing a CVE :-)) and see if the CI and reviewers like it.

baparham added a commit to baparham/openjpeg that referenced this issue on Jan 12

`Fix integer overflow in num_images` ··· `f0727df`

baparham mentioned this issue on Jan 12

Fix integer overflow in num_images #1395

Merged

baparham commented on Jan 12 Contributor

Is it possible to confirm that this issue doesn't affect the `lib` code? I'm not really sure how they are intertwined, but for example, `pdfium` in the chromium project seems to just makes use of the code under `lib` (reference) which to me seems to indicate that it is not vulnerable to this CVE.

thoughts?

rouault commented on Jan 12 Collaborator

> Is it possible to confirm that this issue doesn't affect the `lib` code?

if the code source changes are in src/bin/ only, it means that it affects only the utilities

rouault closed this as completed in #1395 on Jan 12

---

rouault pushed a commit that referenced this issue on Jan 12

`opj_compress/opj_uncompress: fix integer overflow in num_images (#1395)` ··· ✓ `79c7d7a`

kraj pushed a commit to YoeDistro/meta-openembedded that referenced this issue on Feb 15

`openjpeg: fix CVE-2021-29338` ··· `d48a814`

kraj pushed a commit to YoeDistro/meta-openembedded that referenced this issue on Feb 15

`openjpeg: fix CVE-2021-29338` ··· `a847d84`

kraj pushed a commit to YoeDistro/meta-openembedded that referenced this issue on Feb 16

`openjpeg: fix CVE-2021-29338` ··· `33efb90`

halstead pushed a commit to openembedded/meta-openembedded that referenced this issue on Feb 23

openjpeg: fix CVE-2021-29338 ··· b4f4177

radarhere mentioned this issue on Apr 29

Vulnerability in a library that Pillow depends on python-pillow/Pillow#6251

Closed

amstewart pushed a commit to ni/meta-openembedded that referenced this issue on May 2

openjpeg: fix CVE-2021-29338 ··· fb00512

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

Fix integer overflow in num_images
baparham/openjpeg

7 participants