libjxl / **libjxl** Public

Code | Issues 130 | Pull requests 28 | Actions | Security 2 | Insights

New issue

Jump to bottom

Assert failure in jxl::LowMemoryRenderPipeline::Init #1477

Closed **sleicasper** opened this issue on Jun 6 · 10 comments

sleicasper commented on Jun 6 • edited

desc

There is a assert failure in libjxl before version 0.6.1 that could cause deny of service attack.

asan output

```
./lib/jxl/render_pipeline/low_memory_render_pipeline.cc:312: JXL_ASSERT: first_image_dim_stage_ ==
stages_.size() || i >= first_image_dim_stage_
    #0 0x558c6d05047e in __sanitizer_print_stack_trace /fuzz/fuzzdeps/llvm-project-
11.0.0/compiler-rt/lib/asan/asan_stack.cpp:86:3
    #1 0x7fd128ed84b8 in jxl::Abort() /libjxl/SRC/lib/jxl/base/status.h:132:3
    #2 0x7fd12976cc2b in jxl::LowMemoryRenderPipeline::Init()
/libjxl/SRC/lib/jxl/render_pipeline/low_memory_render_pipeline.cc:311:9
    #3 0x7fd12978248d in jxl::RenderPipeline::Builder::Finalize(jxl::FrameDimensions) &&
/libjxl/SRC/lib/jxl/render_pipeline/render_pipeline.cc:91:8
    #4 0x7fd1293a62af in jxl::PassesDecoderState::PreparePipeline(jxl::ImageBundle*,
jxl::PassesDecoderState::PipelineOptions) /libjxl/SRC/lib/jxl/dec_cache.cc:198:40
    #5 0x7fd1293c5964 in jxl::FrameDecoder::ProcessSections(jxl::FrameDecoder::SectionInfo const*,
unsigned long, jxl::FrameDecoder::SectionStatus*) /libjxl/SRC/lib/jxl/dec_frame.cc:775:5
    #6 0x7fd1295aa44a in jxl::(anonymous
namespace)::JxlDecoderProcessCodestream(JxlDecoderStruct*, unsigned char const*, unsigned long)
/libjxl/SRC/lib/jxl/decode.cc:1555:27
    #7 0x7fd1295aa44a in HandleBoxes(JxlDecoderStruct*) /libjxl/SRC/lib/jxl/decode.cc:2079:11
    #8 0x7fd1295a25da in JxlDecoderProcessInput /libjxl/SRC/lib/jxl/decode.cc:2251:29
    #9 0x558c6d07ed4a in DecodeJpegXlOneShot(unsigned char const*, unsigned long,
std::vector<float, std::allocator<float> >*, unsigned long*, unsigned long*, std::vector<unsigned
char, std::allocator<unsigned char> >*) /libjxl/SRC/examples/decode_oneshot.cc:58:31
    #10 0x558c6d080317 in main /libjxl/SRC/examples/decode_oneshot.cc:233:8
    #11 0x7fd12892b082 in __libc_start_main /build/glibc-SzIz7B/glibc-2.31/csu/../csu/libc-
start.c:308:16
    #12 0x558c6cfa152d in _start (/libjxl/fuzzrun/decode_oneshot+0x1f52d)

[1]    888096 illegal hardware instruction  ./decode_oneshot /tmp/poc /dev/null /dev/null
```

reproduce

- compile libjxl with address sanitizer
- run `./decode_oneshot ./poc /dev/null /dev/null`

mo271 commented on Jun 9 Member

Could you point to the commit where you observe this issue?

You write "before version 0.6.1", but the file mentioned in the asan output in not present at or before version 0.6.1. It is present at current main, but there I cannot reproduce the issue.

sleicasper commented on Jun 12 Author

I can still reproduce this issue using commit `ec09355`

mo271 commented on Jun 15 Member

When I compile with asan and try to `decode_oneshot` the image `poc`, I can't trigger the assert failure at `ec09355`:

```
~/libjxl ((ec093557…))> ./ci.sh asan
[...]
~/libjxl ((ec093557…))> ./build/decode_oneshot poc /dev/null /dev/null
Decoder error
Error while decoding the jxl file
~/libjxl ((ec093557…))> md5sum poc
032c2ff7b122977ec747ed69c5e65207  poc
```

So in order to reproduce this, we need more specific info about the system that you are running this on.

sleicasper commented on Jun 15 Author

> When I compile with asan and try to `decode_oneshot` the image `poc`, I can't trigger the assert failure at ec09355:
>
> ```
> ~/libjxl ((ec093557…))> ./ci.sh asan
> [...]
> ~/libjxl ((ec093557…))> ./build/decode_oneshot poc /dev/null /dev/null
> Decoder error
> ```

> ```
> Error while decoding the jxl file
> ~/libjxl ((ec093557…))> md5sum poc
> 032c2ff7b122977ec747ed69c5e65207  poc
> ```
>
> So in order to reproduce this, we need more specific info about the system that you are running this on.

My running environment is ubuntu20.04.

sleicasper commented on Jun 15 Author

> > When I compile with asan and try to `decode_oneshot` the image `poc` , I can't trigger the assert failure at ec09355:
> >
> > ```
> > ~/libjxl ((ec093557…))> ./ci.sh asan
> > [...]
> > ~/libjxl ((ec093557…))> ./build/decode_oneshot poc /dev/null /dev/null
> > Decoder error
> > Error while decoding the jxl file
> > ~/libjxl ((ec093557…))> md5sum poc
> > 032c2ff7b122977ec747ed69c5e65207  poc
> > ```
> >
> > So in order to reproduce this, we need more specific info about the system that you are running this on.
>
> My running environment is ubuntu20.04.

I didn't use ci.sh to compile libjxl.
I used the following commands to compile libjxl.

```
export CFLAGS=-fsanitize=address
export CXXFLAGS=-fsanitize=address
export LDFLAGS=-fsanitize=address
cmake ../src -DBUILD_TESTING=OFF
make
```

mo271 commented on Jun 16 Member

I still cannot repro this, using the same way you compile libjxl. What compiler/version do you use?

sleicasper commented on Jun 16 Author

> I still cannot repro this, using the same way you compile libjxl. What compiler/version do you use?

sorry about the wrong uploaded poc.

I upload new poc here:
poc.zip

mo271 commented on Jun 17 Member

With the other poc.zip, I can now repro this. The failure occurs when the following assert is triggered:

libjxl/lib/jxl/render_pipeline/low_memory_render_pipeline.cc
Line 311 in `1354a06`

```
311      JXL_ASSERT(first_image_dim_stage_ == stages_.size() ||
```

with the following values:

```
stages_.size() == 4
first_image_dim_stage_ == 3
i == 1
```

Not sure if the `illegal hardware instruction` triggered in asan is the problem or if the assert is triggered erroneously..
Any thoughts, **@veluca93**?

szabadka commented on Jul 11 Contributor

The JXL_ASSERT that is triggered here was removed in #1551

Could you verify that it is fixed with the newest version?

wip-sync pushed a commit to NetBSD/pkgsrc-wip that referenced this issue on Jul 17

`libjxl: vulnerability was in git version, not in packaged 0.6.1` … `75e01d3`

szabadka closed this as completed on Aug 5

malaterre commented on Sep 26 Contributor

Issue was solved by commit `aff17c4`

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

4 participants