

New issue

Jump to bottom

Crash during multiple concurrent/parallel decoding #708

Closed

novomesk opened this issue on Oct 8, 2021 · 9 comments · Fixed by #775

Assignees



Labels

bug decoding

novomesk commented on Oct 8, 2021

Contributor

Hello,

This crash occurs when a Qt application decode more JXL files at the same time via my [qt-jpegxl-image-plugin](#)

During the crash I see following message:

```
/var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/jxl/image_ops.h:48: JXL_DASSERT: rect_from.IsInside(from)

#0 jxl::Abort () at /var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/jxl/base/status.cc:42
#1 0x0000ffffe872f964 in jxl::CopyImageTo<float> () at /var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/jxl/image_ops.h:48
#2 0x0000ffffe8844075 in LoadBorders () at /var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/jxl/dec_cache.cc:110
#3 jxl::PassesDecoderState::FinalizeGroup () at /var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/jxl/dec_cache.cc:156
#4 0x0000ffffe8863925 in jxl::N_AVX2::DecodeGroupImpl () at /var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/jxl/dec_group.cc:441
#5 0x0000ffffe886458f in jxl::DecodeGroup () at /var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/jxl/dec_group.cc:754
#6 0x0000ffffe8848f9d in jxl::FrameDecoder::ProcessACGroup () at /var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/jxl/dec_frame.cc:579
#7 0x0000ffffe88490d5 in operator() () at /var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/jxl/dec_frame.cc:746
#8 CallDataFunc () at /var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/jxl/base/data_parallel.h:88
#9 0x0000ffffe8a011af in jpegxl::ThreadParallelRunner::RunRange () at /var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/threads/thread_parallel_runner_internal.cc:139
#10 0x0000ffffe8a012ab in jpegxl::ThreadParallelRunner::ThreadFunc () at /var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/threads/thread_parallel_runner_internal.cc:169
#11 0x0000fffff72b7e30 in std::execute_native_thread_routine (__p=0x7ffffe0005390) at /var/tmp/portage/sys-devel/gcc-10.3.0-r2/work/gcc-10.3.0/libstdc++-v3/src/c++11/thread.cc:80
#12 0x0000fffff7f80e3e in start_thread () from /lib64/libpthread.so.0
#13 0x0000fffff6fd32cf in clone () from /lib64/libc.so.6
```

Here is a simple console application I am able to reproduce crash easily:

[sources.zip](#)

How to compile and run:

```
qmake test_crash.pro
make
```

The application decodes bucresti2.jxl file in two threads (0 - main thread, 1 - worker thread). Each thread have different instance of the plug-in and each plug-in create own ParallelRunner. It may not crash during first iteration, but sooner or later it crashes. The output may run like this:

```
./test_crash
Iteration: 0
[0] 10916x9985 count:1
[1] 10916x9985 count:1
Iteration: 1
[0] 10916x9985 count:1
[1] 10916x9985 count:1
Iteration: 2
[1] 10916x9985 count:1
/var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/jxl/image_ops.h:48: JXL_DASSERT: rect_from.IsInside(from)
Nepriputná inštrukcia
```

Sometime it crashes immediately:

```
Iteration: 0
/var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/jxl/image_ops.h:48: JXL_DASSERT: rect_from.IsInside(from)
/var/tmp/portage/media-libs/libjxl-9999/work/libjxl-9999/lib/jxl/image_ops.h:48: JXL_DASSERT: rect_from.IsInside(from)
Nepriputná inštrukcia
```

When just one thread with plug-in is running at a time, there is no crash.

novomesk mentioned this issue on Oct 8, 2021

error printed by jxlinfo #699

Closed

novomesk commented on Oct 8, 2021

Contributor Author

If I put following call for JXL_DEC_FULL_IMAGE event into critical section, I can avoid crash:

```
status = JxlDecoderProcessInput(m_decoder);
```

novomesk commented on Oct 9, 2021

Contributor Author

I am able to reproduce the crash also via [gdk-pixbuf](#) loader which use ResizableParallelRunner.

It is enough to call following procedure in concurrently in multiple threads few times:

```
#include <gdk-pixbuf/gdk-pixbuf.h>

bool test_plugin_gdk(int id)
{
    GdkPixbuf *pixbuf = gdk_pixbuf_new_from_file("bucuresti2.jpg", NULL);
    if (!pixbuf) {
        printf("Error reading picture via GDK plugin\n");
        return false;
    }
    int x = gdk_pixbuf_get_width(pixbuf);
    int y = gdk_pixbuf_get_height(pixbuf);
    printf("GDK [%d] %dx%d\n", id, x, y);
    gdk_pixbuf_unref(pixbuf);
    return true;
}
```

saschanaz commented on Oct 11, 2021 • edited

djpg1 bucuresti2.jpg also fails, btw.

jonsneyers added bug decoder labels on Oct 13, 2021

deymo self-assigned this on Oct 13, 2021

deymo commented on Oct 18, 2021

Contributor

@novomesk what's the source image for that .jpg file? Does it have a vertical green rectangle on the top right corner of the image by any chance? I'm not sure if that rectangle was on the source image or that's also a (maybe different) bug.

I wasn't able to reproduce it directly with djpg but it has to do with the order in which the groups are processed so the more threads/CPU's you have the more likely it is. I was able to reliably reproduce it with a random-order single threaded parallel runner.

novomesk commented on Oct 19, 2021

Contributor Author

I think I found the source image here:

<https://www.skyscrapercity.com/threads/large-bucharest-photomap-10-000-x-10-000-pixels-38-mb-assembled-from-google-map-pieces.1673975/>

deymo added a commit to deymo/libjxl that referenced this issue on Oct 26, 2021

Fix out of bounds copy in LoadBorders() ...

279a2e6

deymo mentioned this issue on Oct 26, 2021

Fix out of bounds copy in LoadBorders() #775

→ Merged

deymo added a commit to deymo/libjxl that referenced this issue on Oct 26, 2021

Fix out of bounds copy in LoadBorders() ...

732cf50

deymo added a commit to deymo/libjxl that referenced this issue on Oct 26, 2021

Fix out of bounds copy in LoadBorders() ...

bb2b638

deymo added a commit to deymo/libjxl that referenced this issue on Oct 26, 2021

Fix out of bounds copy in LoadBorders() ...

e05e801

deymo closed this as completed in #775 on Oct 26, 2021

deymo added a commit that referenced this issue on Oct 26, 2021

Fix out of bounds copy in LoadBorders() (#775) ...

✓ e649705

deymo added a commit to deymo/libjxl that referenced this issue on Oct 27, 2021

Fix out of bounds copy in LoadBorders() (libjxl#775) ...

✗ 435f2ac

deymo commented on Oct 27, 2021

Contributor

Thanks @novomesk for the link. The green rectangle is in the source image too, so that's good.

deymo commented on Oct 29, 2021

Contributor

Note: this bug got assigned CVE-2021-22564

1

deymo commented on Nov 1, 2021

Contributor

@novomesk Please let me know if you would like to be credited in the CVE description and how (name, company affiliation, etc).

novomesk commented on Nov 1, 2021

Contributor

Author

Thanks, no need to give credit.

Assignees



Labels

bug decoder

Projects


None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

 [Fix out of bounds copy in LoadBorders\(\)](#)
deymo/libjxl

4 participants

