☆ Starred by 1 user

Owner:	wtc@google.com
CC:	----
Status:	Fixed *(Closed)*
Components:	----
Modified:	Dec 23, 2020

Type-Defect
Priority-Medium

Issue 2905: Null pointer dereference in av1/av1_dx_iface.c:970

Reported by zodf0...@gmail.com on Sat, Dec 19, 2020, 6:47 AM EST

Code ‹ Prev 396 of 2974 Next ›
Back to list

What version / commit were you testing with?
commit 7ddc21b

What steps will reproduce the problem?
1.cd aom/build
2.cmake ..
3.make
4.aomdec -o /dev/null --framestats=/dev/null ./poc

What is the expected output?

This is ASAN report:

```
➜ aomdec -o /dev/null --framestats=/dev/null ./poc
Warning: Read invalid frame size (2147483978)
ASAN:DEADLYSIGNAL
=================================================================
==9853==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000020 (pc 0x5633420d7f63 bp 0x7ffe9ba662e0 sp 0x7ffe9ba65b60 T0)
==9853==The signal is caused by a READ memory access.
==9853==Hint: address points to the zero page.
    #0 0x5633420d7f62 in ctrl_get_last_quantizer /home/yuan/afl-target/aom/av1/av1_dx_iface.c:970
    #1 0x5633420c2280 in aom_codec_control /home/yuan/afl-target/aom/aom/src/aom_codec.c:108
    #2 0x563341eeff35 in aom_codec_control_typechecked_AOMD_GET_LAST_QUANTIZER /home/yuan/afl-target/aom/aom/aomdx.h:452
    #3 0x563341eeff35 in main_loop /home/yuan/afl-target/aom/apps/aomdec.c:772
    #4 0x563341ee02c3 in main /home/yuan/afl-target/aom/apps/aomdec.c:1035
    #5 0x7f6e12f94bf6 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21bf6)
    #6 0x563341ee7329 in _start (/home/yuan/afl-target/aom/cbuild/aomdec+0x6d329)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV /home/yuan/afl-target/aom/av1/av1_dx_iface.c:970 in ctrl_get_last_quantizer
==9853==ABORTING

```

poc
441 bytes View Download

Comment 1 by wtc@google.com on Mon, Dec 21, 2020, 4:54 PM EST Project Member

Status: Started (was: New)
Owner: wtc@google.com
Cc: a_deleted_user a_deleted_user

zodf0055980: Thank you very much for the bug report and the poc file to reproduce the bug.

We can defend against this crash at several levels. I will start with adding a null pointer check for ctx->frame_worker to ctrl_get_last_quantizer().

Aasaipriya, Mufaddal: I am cc'ing you because I will add a similar null pointer check to the ctrl_get_xxx() functions that you added to aom/av1/av1_dx_iface.c in June or July.

Comment 2 by wtc@google.com on Mon, Dec 21, 2020, 4:57 PM EST Project Member

The code related to AOMD_GET_LAST_QUANTIZER and ctrl_get_last_quantizer() was added in commit a1f6432dfac92c17672071d9da5c6114fa4faa75:

https://aomedia-review.googlesource.com/c/aom/+/7140

Comment 3 by bugdroid on Mon, Dec 21, 2020, 7:02 PM EST Project Member

The following revision refers to this bug:
https://aomedia.googlesource.com/aom/+/be4ee75fd762d361d0679cc892e4c74af8140093

commit be4ee75fd762d361d0679cc892e4c74af8140093
Author: Wan-Teh Chang <wtc@google.com>
Date: Mon Dec 21 23:59:42 2020

Improve error checking in several ctrl_get_* funcs

Improve error checking in several ctrl_get_*() functions. They all have
the following error-checking logic:
1. If the output parameter is a null pointer, return AOM_CODEC_INVALID_PARAM.
2. If ctx->frame_worker is a null pointer (i.e., the decoder is not
initialized), return AOM_CODEC_ERROR.
3. Otherwise, return AOM_CODEC_OK.

The error-checking logic is realized in two ways to preserve the
original control structures in these functions.

To fix the crash in ~~bug aomedia:2905~~, only the change to the
ctrl_get_last_quantizer() function is needed. I took the opportunity to
review the enture av1/av1_dx_iface.c file.

~~BUG=aomedia:2905~~
Change-Id: I66e48dd21fec1102567aad22608673945d5743c7

[modify] https://crrev.com/be4ee75fd762d361d0679cc892e4c74af8140093/av1/av1_dx_iface.c

Comment 4 by wtc@google.com on Wed, Dec 23, 2020, 10:02 PM EST Project Member

Status: Fixed (was: Started)

Marked the bug Fixed.

There is more that we can do about this bug, but commit be4ee75fd762d361d0679cc892e4c74af8140093 alone fixes the crash.

The root cause of this crash is that ivf_read_frame() may set *bytes_read (which is the bytes_in_buffer variable in aomdec.c) to 0 and return 0 (success) on certain errors, such as an IVF frame size > 256 * 1024 * 1024, which is the case in this poc.

Initially, the 'buf' variable is NULL. If bytes_in_buffer is set to 0, then we call aom_codec_decode() with buf=NULL and bytes_in_buffer=0. aom_codec_decode() interprets the NULL, 0 inputs as a decoder flush operation. So it returns successfully without initializing the decoder.

So, a possible fix is to make ivf_read_frame() return 1 (failure) on a huge IVF frame size. Another possible fix is to make ivf_read_frame() allocates its internal buffer with a minimum buffer size (say 1024 bytes), so that the 'buf' variable won't be NULL, and then aom_codec_decode() will treat the non-NULL, 0 inputs as an error.