heap-buffer-overflow in decode file #231

Open **leonzhao7** opened this issue on Dec 23, 2019 · 1 comment

leonzhao7 commented on Dec 23, 2019

heap-buffer-overflow in decode file

I found some problems during fuzzing

Test Version

dev version, git clone https://github.com/strukturag/libde265

Test Environment

root@ubuntu:~# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 16.04.6 LTS
Release: 16.04
Codename: xenial

Test Configure

./configure
configure: ---------------------------------------
configure: Building dec265 example: yes
configure: Building sherlock265 example: no
configure: Building encoder: yes
configure: ---------------------------------------

Test Program

```
dec265 [infile]
```

Asan Output

```
root@ubuntu:~# /opt/asan/bin/dec265 libde265-mm_loadl_epi64-heap_overflow.crash
WARNING: maximum number of reference pictures exceeded
WARNING: end_of_sub_stream_one_bit not set to 1 when it should be
WARNING: faulty reference picture list
WARNING: coded parameter out of range
WARNING: end_of_sub_stream_one_bit not set to 1 when it should be
WARNING: faulty reference picture list
WARNING: faulty reference picture list
WARNING: maximum number of reference pictures exceeded
WARNING: maximum number of reference pictures exceeded
WARNING: end_of_sub_stream_one_bit not set to 1 when it should be
WARNING: faulty reference picture list
WARNING: faulty reference picture list
=================================================================
==129719==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62b000068560 at pc 0x0000004d0359 bp 0x7ffe48aefc20 sp 0x7ffe48aefc10
READ of size 8 at 0x62b000068560 thread T0
    #0 0x4d0358 in _mm_loadl_epi64(long long __vector(2) const*) /usr/lib/gcc/x86_64-linux-gnu/5/include/emmintrin.h:704
    #1 0x4d0358 in ff_hevc_put_hevc_epel_pixels_8_sse(short*, long, unsigned char const*, long, int, int, int, int, short*) /root/src/libde265/libde265/x86/sse-motion.cc:987
    #2 0x52bf76 in acceleration_functions::put_hevc_epel(short*, long, void const*, long, int, int, int, int, short*, int) const ../libde265/acceleration.h:296
    #3 0x52dc7a in void mc_chroma<unsigned short>(base_context const*, seq_parameter_set const*, int, int, int, int, short*, int, unsigned short const*, int, int, int, int)
/root/src/libde265/libde265/motion.cc:205
    #4 0x51f88a in generate_inter_prediction_samples(base_context*, slice_segment_header const*, de265_image*, int, int, int, int, int, int, int, PBMotion const*)
/root/src/libde265/libde265/motion.cc:382
    #5 0x52b8f9 in decode_prediction_unit(base_context*, slice_segment_header const*, de265_image*, PBMotionCoding const&, int, int, int, int, int, int, int, int)
/root/src/libde265/libde265/motion.cc:2107
    #6 0x47995d in read_coding_unit(thread_context*, int, int, int, int) /root/src/libde265/libde265/slice.cc:4310
    #7 0x47b6fe in read_coding_quadtree(thread_context*, int, int, int, int) /root/src/libde265/libde265/slice.cc:4647
    #8 0x47b611 in read_coding_quadtree(thread_context*, int, int, int, int) /root/src/libde265/libde265/slice.cc:4636
    #9 0x47338a in read_coding_tree_unit(thread_context*) /root/src/libde265/libde265/slice.cc:2861
    #10 0x47beb1 in decode_substream(thread_context*, bool, bool) /root/src/libde265/libde265/slice.cc:4736
    #11 0x47db9f in read_slice_segment_data(thread_context*) /root/src/libde265/libde265/slice.cc:5049
    #12 0x40bf17 in decoder_context::decode_slice_unit_sequential(image_unit*, slice_unit*) /root/src/libde265/libde265/decctx.cc:843
    #13 0x40c6d7 in decoder_context::decode_slice_unit_parallel(image_unit*, slice_unit*) /root/src/libde265/libde265/decctx.cc:945
    #14 0x40b589 in decoder_context::decode_some(bool*) /root/src/libde265/libde265/decctx.cc:730
    #15 0x40b2f2 in decoder_context::read_slice_NAL(bitreader&, NAL_unit*, nal_header&) /root/src/libde265/libde265/decctx.cc:688
    #16 0x40dbb3 in decoder_context::decode_NAL(NAL_unit*) /root/src/libde265/libde265/decctx.cc:1230
    #17 0x40e17b in decoder_context::decode(int*) /root/src/libde265/libde265/decctx.cc:1318
    #18 0x405a61 in de265_decode /root/src/libde265/libde265/de265.cc:346
    #19 0x404972 in main /root/src/libde265/dec265/dec265.cc:764
    #20 0x7f56cd48d82f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)
    #21 0x402b28 in _start (/opt/asan/bin/dec265+0x402b28)

0x62b000068560 is located 80 bytes to the right of 25360-byte region [0x62b000062200,0x62b000068510)
allocated by thread T0 here:
    #0 0x7f56ce38e076 in __interceptor_posix_memalign (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x99076)
    #1 0x43e00d in ALLOC_ALIGNED /root/src/libde265/libde265/image.cc:54
    #2 0x43e725 in de265_image_get_buffer /root/src/libde265/libde265/image.cc:132
    #3 0x440639 in de265_image::alloc_image(int, int, de265_chroma, std::shared_ptr<seq_parameter_set const>, bool, decoder_context*, long, void*, bool)
/root/src/libde265/libde265/image.cc:384
    #4 0x43afa4 in decoded_picture_buffer::new_image(std::shared_ptr<seq_parameter_set const>, decoder_context*, long, void*, bool) /root/src/libde265/libde265/dpb.cc:262
    #5 0x40ee8b in decoder_context::generate_unavailable_reference_picture(seq_parameter_set const*, int, bool) /root/src/libde265/libde265/decctx.cc:1418
    #6 0x411722 in decoder_context::process_reference_picture_set(slice_segment_header*) /root/src/libde265/libde265/decctx.cc:1648
    #7 0x414cc9 in decoder_context::process_slice_segment_header(slice_segment_header*, de265_error*, long, nal_header*, void*) /root/src/libde265/libde265/decctx.cc:2066
    #8 0x40acad in decoder_context::read_slice_NAL(bitreader&, NAL_unit*, nal_header&) /root/src/libde265/libde265/decctx.cc:639
    #9 0x40dbb3 in decoder_context::decode_NAL(NAL_unit*) /root/src/libde265/libde265/decctx.cc:1230
    #10 0x40e17b in decoder_context::decode(int*) /root/src/libde265/libde265/decctx.cc:1318
    #11 0x405a61 in de265_decode /root/src/libde265/libde265/de265.cc:346
    #12 0x404972 in main /root/src/libde265/dec265/dec265.cc:764
    #13 0x7f56cd48d82f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)

SUMMARY: AddressSanitizer: heap-buffer-overflow /usr/lib/gcc/x86_64-linux-gnu/5/include/emmintrin.h:704 _mm_loadl_epi64(long long __vector(2) const*)
Shadow bytes around the buggy address:
  0x0c5680005050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c5680005060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c5680005070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c5680005080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c5680005090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c56800050a0: 00 00 fa fa fa fa fa fa fa fa fa fa[fa]fa fa fa
  0x0c56800050b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c56800050c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c56800050d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c56800050e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c56800050f0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Heap right redzone:      fb
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack partial redzone:   f4
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
==129719==ABORTING
```

POC file

libde265-mm_loadl_epi64-heap_overflow.zip
password: leon.zhao.7

CREDIT

Zhao Liang, Huawei Weiran Labs

coldtobi commented last week • edited

According to Debian this is CVE-2020-21604

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
No branches or pull requests

2 participants