New issue

Jump to bottom

heap overflow in de265_image::available_zscan when decoding file #235

Open leonzhao7 opened this issue on Dec 24, 2019 · 2 comments

leonzhao7 commented on Dec 24, 2019

heap overflow in de265_image::available_zscan when decoding 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

root@ubuntu:~# uname -a
Linux ubuntu 4.15.0-45-generic #4816.04.1-Ubuntu SMP Tue Jan 29 18:03:48 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux

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:~# ./dec265 libde265-de265_image__available_zscan-heap_overflow.crash
WARNING: pps header invalid
WARNING: non-existing PPS referenced
WARNING: pps header invalid
WARNING: non-existing PPS referenced
=================================================================
==50404==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62a0000178bc at pc 0x000000443563 bp 0x7fff14a846d0 sp 0x7fff14a846c0
READ of size 4 at 0x62a0000178bc thread T0
    #0 0x443562 in de265_image::available_zscan(int, int, int, int) const /root/src/libde265/libde265/image.cc:760
    #1 0x443acf in de265_image::available_pred_blk(int, int, int, int, int, int, int, int, int, int) const /root/src/libde265/libde265/image.cc:796
    #2 0x521fd7 in derive_spatial_merging_candidates(MotionVectorAccess const&, de265_image const*, int, int, int, int, int, unsigned char, int, int, int, PBMotion*, int)
/root/src/libde265/libde265/motion.cc:808
    #3 0x525e21 in get_merge_candidate_list_without_step_9(base_context*, slice_segment_header const*, MotionVectorAccess const&, de265_image*, int, int, int, int, int, int,
int, int, PBMotion*) /root/src/libde265/libde265/motion.cc:1467
    #4 0x526732 in derive_luma_motion_merge_mode(base_context*, slice_segment_header const*, de265_image*, int, int, int, int, int, int, int, int, int, PBMotion*)
/root/src/libde265/libde265/motion.cc:1570
    #5 0x52afb3 in motion_vectors_and_ref_indices(base_context*, slice_segment_header const*, de265_image*, PBMotionCoding const&, int, int, int, int, int, int, int, int,
PBMotion*) /root/src/libde265/libde265/motion.cc:2029
    #6 0x52b8ae 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:2103
    #7 0x47995d in read_coding_unit(thread_context*, int, int, int, int) /root/src/libde265/libde265/slice.cc:4310
    #8 0x47b6fe in read_coding_quadtree(thread_context*, int, int, int, int) /root/src/libde265/libde265/slice.cc:4647
    #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 0x7f4581a5882f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)
    #21 0x402b28 in _start (/root/dec265+0x402b28)

0x62a0000178bc is located 1468 bytes to the right of 20736-byte region [0x62a000012200,0x62a000017300)
allocated by thread T0 here:
    #0 0x7f4582959532 in operator new(unsigned long) (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x99532)
    #1 0x42447e in __gnu_cxx::new_allocator<int>::allocate(unsigned long, void const*) /usr/include/c++/5/ext/new_allocator.h:104
    #2 0x422d9c in std::allocator_traits<std::allocator<int> >::allocate(std::allocator<int>&, unsigned long) /usr/include/c++/5/bits/alloc_traits.h:491
    #3 0x420d4f in std::_Vector_base<int, std::allocator<int> >::_M_allocate(unsigned long) /usr/include/c++/5/bits/stl_vector.h:170
    #4 0x455ef8 in std::vector<int, std::allocator<int> >::_M_default_append(unsigned long) /usr/include/c++/5/bits/vector.tcc:557
    #5 0x455c0c in std::vector<int, std::allocator<int> >::resize(unsigned long) /usr/include/c++/5/bits/stl_vector.h:676
    #6 0x451598 in pic_parameter_set::set_derived_values(seq_parameter_set const*) /root/src/libde265/libde265/pps.cc:589
    #7 0x450649 in pic_parameter_set::read(bitreader*, decoder_context*) /root/src/libde265/libde265/pps.cc:528
    #8 0x40a562 in decoder_context::read_pps_NAL(bitreader&) /root/src/libde265/libde265/decctx.cc:574
```

```
    #9 0x40dc78 in decoder_context::decode_NAL(NAL_unit*) /root/src/libde265/libde265/decctx.cc:1244
    #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 0x7f4581a5882f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)

SUMMARY: AddressSanitizer: heap-buffer-overflow /root/src/libde265/libde265/image.cc:760 de265_image::available_zscan(int, int, int, int) const
Shadow bytes around the buggy address:
  0x0c547fffaec0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c547fffaed0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c547fffaee0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c547fffaef0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c547fffaf00: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c547fffaf10: fa fa fa fa fa fa fa[fa]fa fa fa fa fa fa fa fa
  0x0c547fffaf20: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c547fffaf30: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c547fffaf40: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c547fffaf50: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c547fffaf60: 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
==50404==ABORTING
```

POC file

libde265-de265_image__available_zscan-heap_overflow.zip
password: leon.zhao.7

CREDIT

Zhao Liang, Huawei Weiran Labs

ist199099 commented on Oct 20

This was assigned CVE-2020-21599.

farindk commented on Oct 20 Contributor

Apparently fixed in `a3f1c6a` .
I could only reproduce it with Ubuntu 14.04

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants