New issue

Jump to bottom

Heap-buffer-overflow in sse-motion.cc: ff_hevc_put_hevc_qpel_h_2_v_1_sse #335

Open FDU-Sec opened this issue on Oct 10 · 0 comments

FDU-Sec commented on Oct 10

Description

Heap-buffer-overflow (/libde265/build/libde265/liblibde265.so+0x2831a1) in ff_hevc_put_hevc_qpel_h_2_v_1_sse(short*, long, unsigned char const*, long, int, int, short*)

Version

```
$ ./dec265 -h
 dec265  v1.0.8
--------------
usage: dec265 [options] videofile.bin
The video file must be a raw bitstream, or a stream with NAL units (option -n).

options:
  -q, --quiet       do not show decoded image
  -t, --threads N   set number of worker threads (0 - no threading)
  -c, --check-hash  perform hash check
  -n, --nal         input is a stream with 4-byte length prefixed NAL units
  -f, --frames N    set number of frames to process
  -o, --output      write YUV reconstruction
  -d, --dump        dump headers
  -0, --noaccel     do not use any accelerated code (SSE)
  -v, --verbose     increase verbosity level (up to 3 times)
  -L, --no-logging  disable logging
  -B, --write-bytestream FILENAME  write raw bytestream (from NAL input)
  -m, --measure YUV compute PSNRs relative to reference YUV
  -T, --highest-TID select highest temporal sublayer to decode
      --disable-deblocking   disable deblocking filter
      --disable-sao          disable sample-adaptive offset filter
  -h, --help        show help
```

Replay

```
git clone https://github.com/strukturag/libde265.git
cd libde265
mkdir build
cd build
cmake ../ -DCMAKE_CXX_FLAGS="-fsanitize=address"
make -j$(nproc)
./dec265/dec265 poc1
```

ASAN

```
WARNING: non-existing PPS referenced
WARNING: non-existing PPS referenced
WARNING: CTB outside of image area (concealing stream error...)
WARNING: slice header invalid
=================================================================
==8080==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x7f80809038b9 at pc
0x7f807f61d1a2 bp 0x7fff6fd46c30 sp 0x7fff6fd46c20
READ of size 16 at 0x7f80809038b9 thread T0
    #0 0x7f807f61d1a1 in ff_hevc_put_hevc_qpel_h_2_v_1_sse(short*, long, unsigned char const*,
long, int, int, short*) (/libde265/build/libde265/liblibde265.so+0x2831a1)
    #1 0x7f807f51137d in acceleration_functions::put_hevc_qpel(short*, long, void const*, long,
int, int, short*, int, int, int) const (/libde265/build/libde265/liblibde265.so+0x17737d)
    #2 0x7f807f5128ab in void mc_luma<unsigned char>(base_context const*, seq_parameter_set
const*, int, int, int, int, short*, int, unsigned char const*, int, int, int, int)
(/libde265/build/libde265/liblibde265.so+0x1788ab)
    #3 0x7f807f503995 in generate_inter_prediction_samples(base_context*, slice_segment_header
const*, de265_image*, int, int, int, int, int, int, int, PBMotion const*)
(/libde265/build/libde265/liblibde265.so+0x169995)
    #4 0x7f807f51090f in decode_prediction_unit(base_context*, slice_segment_header const*,
de265_image*, PBMotionCoding const&, int, int, int, int, int, int, int, int)
(/libde265/build/libde265/liblibde265.so+0x17690f)
    #5 0x7f807f54b7e3 in read_prediction_unit(thread_context*, int, int, int, int, int, int, int,
int, int) (/libde265/build/libde265/liblibde265.so+0x1b17e3)
    #6 0x7f807f54d264 in read_coding_unit(thread_context*, int, int, int, int)
(/libde265/build/libde265/liblibde265.so+0x1b3264)
    #7 0x7f807f54e250 in read_coding_quadtree(thread_context*, int, int, int, int)
(/libde265/build/libde265/liblibde265.so+0x1b4250)
    #8 0x7f807f54e218 in read_coding_quadtree(thread_context*, int, int, int, int)
(/libde265/build/libde265/liblibde265.so+0x1b4218)
    #9 0x7f807f54e218 in read_coding_quadtree(thread_context*, int, int, int, int)
(/libde265/build/libde265/liblibde265.so+0x1b4218)
    #10 0x7f807f54e218 in read_coding_quadtree(thread_context*, int, int, int, int)
(/libde265/build/libde265/liblibde265.so+0x1b4218)
    #11 0x7f807f545726 in read_coding_tree_unit(thread_context*)
(/libde265/build/libde265/liblibde265.so+0x1ab726)
    #12 0x7f807f54e9ea in decode_substream(thread_context*, bool, bool)
(/libde265/build/libde265/liblibde265.so+0x1b49ea)
    #13 0x7f807f55070f in read_slice_segment_data(thread_context*)
(/libde265/build/libde265/liblibde265.so+0x1b670f)
    #14 0x7f807f4af6d2 in decoder_context::decode_slice_unit_sequential(image_unit*, slice_unit*)
(/libde265/build/libde265/liblibde265.so+0x1156d2)
    #15 0x7f807f4afec1 in decoder_context::decode_slice_unit_parallel(image_unit*, slice_unit*)
(/libde265/build/libde265/liblibde265.so+0x115ec1)
```

```
    #16 0x7f807f4aec0f in decoder_context::decode_some(bool*)
(/libde265/build/libde265/liblibde265.so+0x114c0f)
    #17 0x7f807f4ae93d in decoder_context::read_slice_NAL(bitreader&, NAL_unit*, nal_header&)
(/libde265/build/libde265/liblibde265.so+0x11493d)
    #18 0x7f807f4b143e in decoder_context::decode_NAL(NAL_unit*)
(/libde265/build/libde265/liblibde265.so+0x11743e)
    #19 0x7f807f4b1ab3 in decoder_context::decode(int*)
(/libde265/build/libde265/liblibde265.so+0x117ab3)
    #20 0x7f807f498e95 in de265_decode (/libde265/build/libde265/liblibde265.so+0xfee95)
    #21 0x55c0d6940bc9 in main (/libde265/build/dec265/dec265+0x6bc9)
    #22 0x7f807efcac86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)
    #23 0x55c0d693e9b9 in _start (/libde265/build/dec265/dec265+0x49b9)

0x7f80809038b9 is located 169 bytes to the right of 131088-byte region
[0x7f80808e3800,0x7f8080903810)
allocated by thread T0 here:
    #0 0x7f807f9c1790 in posix_memalign (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xdf790)
    #1 0x7f807f4ea1cb in ALLOC_ALIGNED(unsigned long, unsigned long)
(/libde265/build/libde265/liblibde265.so+0x1501cb)
    #2 0x7f807f4ea92a in de265_image_get_buffer(void*, de265_image_spec*, de265_image*, void*)
(/libde265/build/libde265/liblibde265.so+0x15092a)
    #3 0x7f807f4ecd1a in de265_image::alloc_image(int, int, de265_chroma,
std::shared_ptr<seq_parameter_set const>, bool, decoder_context*, long, void*, bool)
(/libde265/build/libde265/liblibde265.so+0x152d1a)
    #4 0x7f807f4d10cc in decoded_picture_buffer::new_image(std::shared_ptr<seq_parameter_set
const>, decoder_context*, long, void*, bool) (/libde265/build/libde265/liblibde265.so+0x1370cc)
    #5 0x7f807f4b2824 in decoder_context::generate_unavailable_reference_picture(seq_parameter_set
const*, int, bool) (/libde265/build/libde265/liblibde265.so+0x118824)
    #6 0x7f807f4b57f5 in decoder_context::process_reference_picture_set(slice_segment_header*)
(/libde265/build/libde265/liblibde265.so+0x11b7f5)
    #7 0x7f807f4b8d70 in decoder_context::process_slice_segment_header(slice_segment_header*,
de265_error*, long, nal_header*, void*) (/libde265/build/libde265/liblibde265.so+0x11ed70)
    #8 0x7f807f4ae246 in decoder_context::read_slice_NAL(bitreader&, NAL_unit*, nal_header&)
(/libde265/build/libde265/liblibde265.so+0x114246)
    #9 0x7f807f4b143e in decoder_context::decode_NAL(NAL_unit*)
(/libde265/build/libde265/liblibde265.so+0x11743e)
    #10 0x7f807f4b1ab3 in decoder_context::decode(int*)
(/libde265/build/libde265/liblibde265.so+0x117ab3)
    #11 0x7f807f498e95 in de265_decode (/libde265/build/libde265/liblibde265.so+0xfee95)
    #12 0x55c0d6940bc9 in main (/libde265/build/dec265/dec265+0x6bc9)
    #13 0x7f807efcac86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)

SUMMARY: AddressSanitizer: heap-buffer-overflow (/libde265/build/libde265/liblibde265.so+0x2831a1)
in ff_hevc_put_hevc_qpel_h_2_v_1_sse(short*, long, unsigned char const*, long, int, int, short*)
Shadow bytes around the buggy address:
  0x0ff0901186c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ff0901186d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ff0901186e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ff0901186f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ff090118700: 00 00 fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0ff090118710: fa fa fa fa fa fa fa[fa]fa fa fa fa fa fa fa fa
  0x0ff090118720: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0ff090118730: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0ff090118740: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0ff090118750: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0ff090118760: 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
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  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
  Left alloca redzone:     ca
  Right alloca redzone:    cb
==8080==ABORTING
```

POC

https://github.com/FDU-Sec/poc/blob/main/libde265/poc1

Environment

```
Ubuntu 16.04
Clang 10.0.1
gcc 5.5
```

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

---

1 participant