Heap-buffer-overflow in fallback-motion.cc: void put_epel_hv_fallback<unsigned short>(#345

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+0x148fda) in void put_epel_hv_fallback(short*, long, unsigned short const*, long, int, int, int, int, short*, int)

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 poc11-1
./dec265/dec265 poc11-2
```

ASAN

```
WARNING: pps header invalid
WARNING: CTB outside of image area (concealing stream error...)
WARNING: CTB outside of image area (concealing stream error...)
=================================================================
==61372==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62b00002951c at pc 0x7f3e99904fdb
READ of size 2 at 0x62b00002951c thread T0
    #0 0x7f3e99904fda in void put_epel_hv_fallback<unsigned short>(short*, long, unsigned short const
    #1 0x7f3e999332ca in acceleration_functions::put_hevc_epel_hv(short*, long, void const*, long, in
    #2 0x7f3e99935213 in void mc_chroma<unsigned short>(base_context const*, seq_parameter_set const*
    #3 0x7f3e99925b2d in generate_inter_prediction_samples(base_context*, slice_segment_header const*
    #4 0x7f3e9993290f in decode_prediction_unit(base_context*, slice_segment_header const*, de265_ima
    #5 0x7f3e9996d7e3 in read_prediction_unit(thread_context*, int, int, int, int, int, int, int, int
    #6 0x7f3e9996f39a in read_coding_unit(thread_context*, int, int, int, int) (/libde265/build/libde
    #7 0x7f3e99970250 in read_coding_quadtree(thread_context*, int, int, int, int) (/libde265/build/l
    #8 0x7f3e99970091 in read_coding_quadtree(thread_context*, int, int, int, int) (/libde265/build/l
    #9 0x7f3e99967726 in read_coding_tree_unit(thread_context*) (/libde265/build/libde265/liblibde265
    #10 0x7f3e999709ea in decode_substream(thread_context*, bool, bool) (/libde265/build/libde265/lib
    #11 0x7f3e9997270f in read_slice_segment_data(thread_context*) (/libde265/build/libde265/liblibde
    #12 0x7f3e998d16d2 in decoder_context::decode_slice_unit_sequential(image_unit*, slice_unit*) (/l
    #13 0x7f3e998d1ec1 in decoder_context::decode_slice_unit_parallel(image_unit*, slice_unit*) (/lib
    #14 0x7f3e998d0c0f in decoder_context::decode_some(bool*) (/libde265/build/libde265/liblibde265.s
    #15 0x7f3e998d093d in decoder_context::read_slice_NAL(bitreader&, NAL_unit*, nal_header&) (/libde
    #16 0x7f3e998d343e in decoder_context::decode_NAL(NAL_unit*) (/libde265/build/libde265/liblibde26
    #17 0x7f3e998d3ab3 in decoder_context::decode(int*) (/libde265/build/libde265/liblibde265.so+0x11
    #18 0x7f3e998bae95 in de265_decode (/libde265/build/libde265/liblibde265.so+0xfee95)
    #19 0x55a40ac18bc9 in main (/libde265/build/dec265/dec265+0x6bc9)
    #20 0x7f3e993ecc86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)
    #21 0x55a40ac169b9 in _start (/libde265/build/dec265/dec265+0x49b9)

0x62b00002951c is located 12 bytes to the right of 25360-byte region [0x62b000023200,0x62b000029510)
allocated by thread T0 here:
    #0 0x7f3e99de3790 in posix_memalign (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xdf790)
    #1 0x7f3e9990c1cb in ALLOC_ALIGNED(unsigned long, unsigned long) (/libde265/build/libde265/liblib
    #2 0x7f3e9990c99d in de265_image_get_buffer(void*, de265_image_spec*, de265_image*, void*) (/libd
    #3 0x7f3e9990ed1a in de265_image::alloc_image(int, int, de265_chroma, std::shared_ptr<seq_paramet
    #4 0x7f3e998f30cc in decoded_picture_buffer::new_image(std::shared_ptr<seq_parameter_set const>,
    #5 0x7f3e998d4824 in decoder_context::generate_unavailable_reference_picture(seq_parameter_set co
    #6 0x7f3e998d7332 in decoder_context::process_reference_picture_set(slice_segment_header*) (/libd
    #7 0x7f3e998dad70 in decoder_context::process_slice_segment_header(slice_segment_header*, de265_e
    #8 0x7f3e998d0246 in decoder_context::read_slice_NAL(bitreader&, NAL_unit*, nal_header&) (/libde2
    #9 0x7f3e998d343e in decoder_context::decode_NAL(NAL_unit*) (/libde265/build/libde265/liblibde265
    #10 0x7f3e998d3ab3 in decoder_context::decode(int*) (/libde265/build/libde265/liblibde265.so+0x11
    #11 0x7f3e998bae95 in de265_decode (/libde265/build/libde265/liblibde265.so+0xfee95)
    #12 0x55a40ac18bc9 in main (/libde265/build/dec265/dec265+0x6bc9)
    #13 0x7f3e993ecc86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)

SUMMARY: AddressSanitizer: heap-buffer-overflow (/libde265/build/libde265/liblibde265.so+0x148fda) in
Shadow bytes around the buggy address:
  0x0c567fffd250: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c567fffd260: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c567fffd270: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c567fffd280: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c567fffd290: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c567fffd2a0: 00 00 fa[fa]fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c567fffd2b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c567fffd2c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c567fffd2d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c567fffd2e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c567fffd2f0: 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
==61372==ABORTING
```

POC

https://github.com/FDU-Sec/poc/blob/main/libde265/poc11-1
https://github.com/FDU-Sec/poc/blob/main/libde265/poc11-2

Environment

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

Credit

Peng Deng (Fudan University)

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant