

[New issue](#)[Jump to bottom](#)

Heap-buffer-overflow in motion.cc: mc_luma<unsigned char> #340

[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+0x1787af) in void mc_luma(base_context const*, seq_parameter_set const*, int, int, int, int, short*, int, unsigned char const*, int, int, int, 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 poc6-1
./dec265/dec265 poc6-2

```

ASAN

```

WARNING: end_of_sub_stream_one_bit not set to 1 when it should be
WARNING: CTB outside of image area (concealing stream error...)
WARNING: non-existing PPS referenced
WARNING: non-existing PPS referenced
WARNING: CTB outside of image area (concealing stream error...)
=====
==45304==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x7f052d0ee810 at pc 0x7f052bc947b0
READ of size 1 at 0x7f052d0ee810 thread T0
#0 0x7f052bc947af in void mc_luma<unsigned char>(base_context const*, seq_parameter_set const*, i
#1 0x7f052bc85995 in generate_inter_prediction_samples(base_context*, slice_segment_header const*
#2 0x7f052bc9290f in decode_prediction_unit(base_context*, slice_segment_header const*, de265_ima
#3 0x7f052bcce2d9 in read_coding_unit(thread_context*, int, int, int, int) (/libde265/build/libde
#4 0x7f052bcd0250 in read_coding_quadtree(thread_context*, int, int, int, int) (/libde265/build/l
#5 0x7f052bcd0163 in read_coding_quadtree(thread_context*, int, int, int, int) (/libde265/build/l
#6 0x7f052bcc7726 in read_coding_tree_unit(thread_context*) (/libde265/build/libde265/liblibde265
#7 0x7f052bcd09ea in decode_substream(thread_context*, bool, bool) (/libde265/build/libde265/libl
#8 0x7f052bcd270f in read_slice_segment_data(thread_context*) (/libde265/build/libde265/liblibde2
#9 0x7f052bc316d2 in decoder_context::decode_slice_unit_sequential(image_unit*, slice_unit*) (/li
#10 0x7f052bc31ec1 in decoder_context::decode_slice_unit_parallel(image_unit*, slice_unit*) (/lib
#11 0x7f052bc30c0f in decoder_context::decode_some(bool*) (/libde265/build/libde265/liblibde265.s
#12 0x7f052bc3093d in decoder_context::read_slice_NAL(bitreader&, NAL_unit*, nal_header&) (/libde
#13 0x7f052bc3343e in decoder_context::decode_NAL(NAL_unit*) (/libde265/build/libde265/liblibde26
#14 0x7f052bc33ab3 in decoder_context::decode(int*) (/libde265/build/libde265/liblibde265.so+0x11
#15 0x7f052bc1ae95 in de265_decode (/libde265/build/libde265/liblibde265.so+0xfee95)
#16 0x562938164bc9 in main (/libde265/build/dec265/dec265+0x6bc9)
#17 0x7f052b74cc86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)
#18 0x5629381629b9 in _start (/libde265/build/dec265/dec265+0x49b9)

```

0x7f052d0ee810 is located 0 bytes to the right of 131088-byte region [0x7f052d0ce800,0x7f052d0ee810) allocated by thread T0 here:

```

#0 0x7f052c143790 in posix_memalign (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xdf790)
#1 0x7f052bc6c1cb in ALLOC_ALIGNED(unsigned long, unsigned long) (/libde265/build/libde265/liblibl
#2 0x7f052bc6c92a in de265_image_get_buffer(void*, de265_image_spec*, de265_image*, void*) (/libd
#3 0x7f052bc6ed1a in de265_image::alloc_image(int, int, de265_chroma, std::shared_ptr<seq_paramet
#4 0x7f052bc530cc in decoded_picture_buffer::new_image(std::shared_ptr<seq_parameter_set const>,
#5 0x7f052bc34824 in decoder_context::generate_unavailable_reference_picture(seq_parameter_set co
#6 0x7f052bc37332 in decoder_context::process_reference_picture_set(slice_segment_header*) (/libd
#7 0x7f052bc3ad70 in decoder_context::process_slice_segment_header(slice_segment_header*, de265_e
#8 0x7f052bc30246 in decoder_context::read_slice_NAL(bitreader&, NAL_unit*, nal_header&) (/libde2
#9 0x7f052bc3343e in decoder_context::decode_NAL(NAL_unit*) (/libde265/build/libde265/liblibde265
#10 0x7f052bc33ab3 in decoder_context::decode(int*) (/libde265/build/libde265/liblibde265.so+0x11
#11 0x7f052bc1ae95 in de265_decode (/libde265/build/libde265/liblibde265.so+0xfee95)

```

```
#12 0x562938164bc9 in main (/libde265/build/dec265/dec265+0x6bc9)
#13 0x7f052b74cc86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)
```

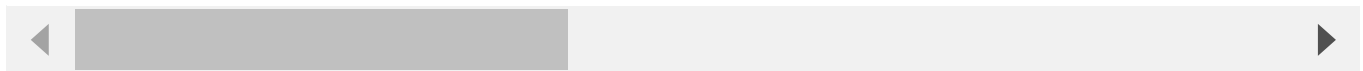
SUMMARY: AddressSanitizer: heap-buffer-overflow (/libde265/build/libde265/liblibde265.so+0x1787af) in
Shadow bytes around the buggy address:

```
0x0fe125a15cb0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0fe125a15cc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0fe125a15cd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0fe125a15ce0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0fe125a15cf0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0fe125a15d00: 00 00[fa]fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0fe125a15d10: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0fe125a15d20: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0fe125a15d30: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0fe125a15d40: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0fe125a15d50: fa 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
```

```
==45304==ABORTING
```



POC

<https://github.com/FDU-Sec/poc/blob/main/libde265/poc6-1>

<https://github.com/FDU-Sec/poc/blob/main/libde265/poc6-2>

Environment

Ubuntu 16.04

Clang 10.0.1

gcc 5.5

Credit

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant

