# heap-buffer-overflow in ff_hevc_put_unweighted_pred_8_sse when decoding file #237

Open leonzhao7 opened this issue on Dec 24, 2019 · 1 comment

leonzhao7 commented on Dec 24, 2019

## heap-buffer-overflow in ff_hevc_put_unweighted_pred_8_sse 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-ff_hevc_put_unweighted_pred_8_sse-heap_overflow.crash
WARNING: CTB outside of image area (concealing stream error...)
WARNING: CTB outside of image area (concealing stream error...)
WARNING: CTB outside of image area (concealing stream error...)
=================================================================
==69912==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x61e00000fc30 at pc 0x0000004cc8bf bp 0x7ffc1997ee70 sp 0x7ffc1997ee60
WRITE of size 4 at 0x61e00000fc30 thread T0
    #0 0x4cc8be in ff_hevc_put_unweighted_pred_8_sse(unsigned char*, long, short const*, long, int, int) /root/src/libde265/libde265/x86/sse-motion.cc:149
    #1 0x52bc86 in acceleration_functions::put_unweighted_pred(void*, long, short const*, long, int, int, int) const ../libde265/acceleration.h:260
    #2 0x521301 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:578
    #3 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
    #4 0x478f4a in read_prediction_unit(thread_context*, int, int, int, int, int, int, int, int, int, int) /root/src/libde265/libde265/slice.cc:4137
    #5 0x47a704 in read_coding_unit(thread_context*, int, int, int, int) /root/src/libde265/libde265/slice.cc:4492
    #6 0x47b6fe in read_coding_quadtree(thread_context*, int, int, int, int) /root/src/libde265/libde265/slice.cc:4647
    #7 0x47b611 in read_coding_quadtree(thread_context*, int, int, int, int) /root/src/libde265/libde265/slice.cc:4636
    #8 0x47b53f in read_coding_quadtree(thread_context*, int, int, int, int) /root/src/libde265/libde265/slice.cc:4630
    #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 0x7f931534a82f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)
    #21 0x402b28 in _start (/root/dec265+0x402b28)

AddressSanitizer can not describe address in more detail (wild memory access suspected).
SUMMARY: AddressSanitizer: heap-buffer-overflow /root/src/libde265/libde265/x86/sse-motion.cc:149 ff_hevc_put_unweighted_pred_8_sse(unsigned char*, long, short const*, long, int,
int)
Shadow bytes around the buggy address:
  0x0c3c7fff9f30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c3c7fff9f40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c3c7fff9f50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c3c7fff9f60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c3c7fff9f70: 00 00 fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c3c7fff9f80: fa fa fa fa fa fa[fa]fa fa fa fa fa fa fa fa fa
  0x0c3c7fff9f90: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c3c7fff9fa0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c3c7fff9fb0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c3c7fff9fc0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c3c7fff9fd0: 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
==69912==ABORTING
```

## POC file

libde265-ff_hevc_put_unweighted_pred_8_sse-heap_overflow.zip
password: leon.zhao.7

## CREDIT

Zhao Liang, Huawei Weiran Labs

**ist199099** commented on Oct 20

This was assigned CVE-2020-21598.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants