

New issue

[Jump to bottom](#)

## Heap-use-after-free in intrapred.h when decoding file #299

🔒 Closed dhbbb opened this issue on Jun 22, 2021 · 4 comments

dhbbb commented on Jun 22, 2021

Hello,

A Heap-use-after-free has occurred when running program dec265

System info:

Ubuntu 20.04.1 : clang 10.0.0 , gcc 9.3.0

Dec265 v1.0.8

[poc.zip](#)

Verification steps:

1.Get the source code of libde265

2.Compile

```
cd libde265
mkdir build && cd build
cmake ../ -DCMAKE_CXX_COMPILER=clang++ -DCMAKE_CXX_FLAGS="-fsanitize=address"
make -j 32
```

3.run dec265

```
./dec265 poc
```

asan info

```
=====
==1538158==ERROR: AddressSanitizer: heap-use-after-free on address 0x625000007e04 at pc 0x7efe5f2b9526 bp 0x7ffceaaa13c0 sp 0x7ffceaaa13b0
READ of size 4 at 0x625000007e04 thread T0
#0 0x7efe5f2b9525 in intra_border_computer<unsigned char>::fill_from_image() /home/dh/sda3/libde265-master/libde265-master/libde265/intrapred.h:552
#1 0x7efe5f2ba6e9 in void fill_border_samples<unsigned char>(de265_image*, int, int, int, int, unsigned char*) /home/dh/sda3/libde265-master/libde265-master/libde265/intrapred.cc:260
#2 0x7efe5f2ba6e9 in void decode_intra_prediction_internal<unsigned char>(de265_image*, int, int, IntraPredMode, unsigned char*, int, int, int) /home/dh/sda3/libde265-master/libde265-master/libde265/intrapred.cc:284
#3 0x7efe5f2a5383 in decode_intra_prediction(de265_image*, int, int, IntraPredMode, int, int) /home/dh/sda3/libde265-master/libde265-master/libde265/intrapred.cc:335
#4 0x7efe5f31dc52 in decode_TU /home/dh/sda3/libde265-master/libde265-master/libde265/slice.cc:3453
#5 0x7efe5f342e76 in read_transform_unit(thread_context*, int, int, int, int, int, int, int, int, int, int) /home/dh/sda3/libde265-master/libde265-master/libde265/slice.cc:3665
#6 0x7efe5f347191 in read_transform_tree(thread_context*, int, int, int, int, int, int, int, int, int, int, int, PredMode, unsigned char, unsigned char) /home/dh/sda3/libde265-master/libde265-master/libde265/slice.cc:3942
#7 0x7efe5f34e119 in read_coding_unit(thread_context*, int, int, int, int) /home/dh/sda3/libde265-master/libde265-master/libde265/slice.cc:4575
#8 0x7efe5f3548f2 in read_coding_quadtree(thread_context*, int, int, int, int) /home/dh/sda3/libde265-master/libde265-master/libde265/slice.cc:4652
#9 0x7efe5f354357 in read_coding_quadtree(thread_context*, int, int, int, int) /home/dh/sda3/libde265-master/libde265-master/libde265/slice.cc:4635
#10 0x7efe5f356564 in decode_substream(thread_context*, bool, bool) /home/dh/sda3/libde265-master/libde265-master/libde265/slice.cc:4741
#11 0x7efe5f358dbd in read_slice_segment_data(thread_context*) /home/dh/sda3/libde265-master/libde265-master/libde265/slice.cc:5054
#12 0x7efe5f23d75 in decoder_context::decode_slice_unit_sequential(image_unit*, slice_unit*) /home/dh/sda3/libde265-master/libde265-master/libde265/dectx.cc:843
#13 0x7efe5f240c0f in decoder_context::decode_slice_unit_parallel(image_unit*, slice_unit*) /home/dh/sda3/libde265-master/libde265-master/libde265/dectx.cc:945
#14 0x7efe5f241715 in decoder_context::decode_some(bool*) /home/dh/sda3/libde265-master/libde265-master/libde265/dectx.cc:730
#15 0x7efe5f24695e in decoder_context::decode(int*) /home/dh/sda3/libde265-master/libde265-master/libde265/dectx.cc:1329
#16 0x55990c1348fd in main /home/dh/sda3/libde265-master/libde265-master/dec265/dec265.cc:764
#17 0x7efe5ed950b2 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x270b2)
#18 0x55990c13776d in _start (/home/dh/sda3/libde265-master/libde265-master/dec265+0xa76d)

0x625000007e04 is located 1284 bytes inside of 8600-byte region [0x625000007900,0x625000009a98)
freed by thread T0 here:
#0 0x7efe5f6408df in operator delete(void*) (/lib/x86_64-linux-gnu/libasan.so.5+0x1108df)
#1 0x7efe5f24b576 in std::_Sp_counted_ptr_inplace<pic_parameter_set, std::allocator<pic_parameter_set>, (__gnu_cxx::::lock_policy)2>::~_M_destroy()
/usr/include/c++/9/ext/new_allocator.h:128
#2 0x7efe5f4d996f (/home/dh/sda3/libde265-master/libde265-master/build/libde265/liblibde265.so+0x37d96f)

previously allocated by thread T0 here:
#0 0x7efe5f63f947 in operator new(unsigned long) (/lib/x86_64-linux-gnu/libasan.so.5+0x10f947)
#1 0x7efe5f22cf3f in std::shared_ptr<pic_parameter_set> std::make_shared<pic_parameter_set>() /usr/include/c++/9/ext/new_allocator.h:114
#2 0x7efe5f22cf3f in decoder_context::read_pps_NAL(bitreader&) /home/dh/sda3/libde265-master/libde265-master/libde265/dectx.cc:572
#3 0x7efe5b1ff7ff (<unknown module>)
#4 0x614ffffffffff (<unknown module>)
```

SUMMARY: AddressSanitizer: heap-use-after-free /home/dh/sda3/libde265-master/libde265-master/libde265/intrapred.h:552 in intra\_border\_computer&lt;unsigned char&gt;::fill\_from\_image()

Shadow bytes around the buggy address:

```
0x0c4a7fff8f70: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x0c4a7fff8f80: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x0c4a7fff8f90: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x0c4a7fff8fa0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x0c4a7fff8fb0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
=>0x0c4a7fff8fc0:[fd]fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x0c4a7fff8fd0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x0c4a7fff8fe0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x0c4a7fff8ff0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x0c4a7fff9000: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x0c4a7fff9010: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
```


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
Shadow gap:          cc
==1538158==ABORTING
```

**stevebeattie** commented on Jan 12

This issue was assigned [CVE-2021-36408](#).

 **farindk** added a commit that referenced this issue on Apr 5

 **fix** streams where SPS image size changes without refreshing PPS ([#299](#))

 [f538254](#)

**farindk** commented on Apr 5

Contributor

Thank you.  
Please confirm that it is fixed with the above change.


**ist199099** commented on Oct 1

This is fixed in the tip of the master branch (commit [b371427](#) ) on Ubuntu 20.04 (with GCC 9.4.0 and Clang 10.0.0) on the x86\_64 and aarch64 architectures.

**farindk** commented on Oct 1

Contributor

[@ist199099](#) Thank you for cross checking this.

 **farindk** closed this as completed on Oct 1

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

4 participants

