New issue

Jump to bottom

Heap over flow #183

Closed **linhlhq** opened this issue on Jan 12, 2020 · 2 comments

Assignees	
Labels	bug fuzzing
Milestone	0.11

linhlhq commented on Jan 12, 2020

I found a bug in dwg2dxf.
POC: https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_69b5609/id:000000%2Csig:06%2Csrc:000001%2Cop:flip4%2Cpos:27167

```
==29243==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6140000005b2 at pc 0x7ffafa9dc77a bp 0x7ffcc19b07e0 sp 0x7ffcc19aff88
WRITE of size 126 at 0x6140000005b2 thread T0
    #0 0x7ffafa9dc779  (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79779)
    #1 0x561777b62e9c in memcpy /usr/include/x86_64-linux-gnu/bits/string_fortified.h:34
    #2 0x561777b62e9c in read_2004_compressed_section ../../src/decode.c:2379
    #3 0x56177811f8af in read_2004_section_preview ../../src/decode.c:2778
    #4 0x56177811f8af in decode_R2004 ../../src/decode.c:2965
    #5 0x56177812c264 in dwg_decode ../../src/decode.c:245
    #6 0x561777adb7c2 in dwg_read_file ../../src/dwg.c:211
    #7 0x561777ad9550 in main ../../programs/dwg2dxf.c:255
    #8 0x7ffafa1f5b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
    #9 0x561777adaa69 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwg2dxf+0x363a69)

Address 0x6140000005b2 is a wild pointer.
SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79779)
Shadow bytes around the buggy address:
  0x0c287fff8060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c287fff8070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c287fff8080: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c287fff8090: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c287fff80a0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c287fff80b0: fa fa fa fa fa fa[fa]fa fa fa fa fa fa fa fa fa
  0x0c287fff80c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c287fff80d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c287fff80e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c287fff80f0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c287fff8100: 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
==29243==ABORTING
```

rurban commented on Jan 13, 2020 Contributor

Ha, and I just released 0.10.1

rurban self-assigned this on Jan 13, 2020

rurban added the bug label on Jan 13, 2020

rurban added this to the **0.11** milestone on Jan 13, 2020

rurban added a commit that referenced this issue on Jan 13, 2020

`read_2004_compressed_section: adjust for empty sections` ... bcec483

rurban commented on Jan 13, 2020 Contributor

Fixed in master, with `bcec483`

rurban closed this as completed on Jan 13, 2020

rurban added the **fuzzing** label on Jan 16, 2020

attritionorg mentioned this issue on Jul 18, 2020

LibreDWG "read_2004_compressed_section" function heap overflow vulnerability #249

Closed

Assignees

rurban

Labels

bug fuzzing

Projects

None yet

Milestone

0.11

Development

No branches or pull requests

2 participants