New issue

Jump to bottom

Heap-buffer-overflow in decode_preR13 line 470 #325

Closed zodf0055980 opened this issue on Mar 3, 2021 · 2 comments

Assignees

Labels fuzzing

zodf0055980 commented on Mar 3, 2021 • edited

I found a heap buffer overflow in the current master (`8072563`).

Configure

CFLAGS="-g -fsanitize=address" LDFLAGS="-fsanitize=address" ./configure

Command

```
./dwgread ./poc
```

ASAN report

```
➜  ./dwgread ./poc
ERROR: This version of LibreDWG is only capable of decoding version r13-r2018 (code: AC1012-AC1032) DWG files.
We don't decode many entities and no blocks yet.
ERROR: offset -39
ERROR: offset -85
=================================================================
==12538==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6120000001c8 at pc 0x7ffff29b78be bp 0x7fffffffa8e0 sp 0x7fffffffa8d0
WRITE of size 8 at 0x6120000001c8 thread T0
    #0 0x7ffff29b78bd in decode_preR13_section /home/yuan/afl-target/libredwg-asan/src/decode.c:470
    #1 0x7ffff2a26e90 in decode_preR13 /home/yuan/afl-target/libredwg-asan/src/decode.c:744
    #2 0x7ffff29a43e9 in dwg_decode /home/yuan/afl-target/libredwg-asan/src/decode.c:235
    #3 0x7ffff297afa4 in dwg_read_file /home/yuan/afl-target/libredwg-asan/src/dwg.c:253
    #4 0x5555555576f3 in main /home/yuan/afl-target/libredwg-asan/programs/dwgread.c:251
    #5 0x7ffff1f52bf6 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21bf6)
    #6 0x5555555562a9 in _start (/home/yuan/afl-target/libredwg-asan/programs/.libs/dwgread+0x22a9)

Address 0x6120000001c8 is a wild pointer.
SUMMARY: AddressSanitizer: heap-buffer-overflow /home/yuan/afl-target/libredwg-asan/src/decode.c:470 in decode_preR13_section
Shadow bytes around the buggy address:
  0x0c247fff7fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c247fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c247fff8000: fa fa fa fa fa fa fa fa 00 00 00 00 00 00 00 00
  0x0c247fff8010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c247fff8020: 00 00 00 00 00 00 00 00 00 00 00 00 00 fa fa fa
=>0x0c247fff8030: fa fa fa fa fa fa fa fa fa[fa]fa fa fa fa fa fa
  0x0c247fff8040: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c247fff8050: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c247fff8060: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c247fff8070: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c247fff8080: 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
==12538==ABORTING
```

POC

poc.zip

rurban added the fuzzing label on Mar 6, 2021

rurban self-assigned this on Mar 6, 2021

rurban commented on Mar 6, 2021 • edited Contributor

We don't release this code-path yet. fuzzers should configure with `--enable-release`. it is only run with git checkouts.

Fixed with `ea0b952`

rurban added a commit that referenced this issue on Mar 6, 2021

`decode_preR13: more PREP_TABLE protections` ··· ✓ `ea0b952`

rurban changed the title ~~Heap-buffer-overflow in decode.c:470~~ **Heap-buffer-overflow in decode_preR13 line 470** on Mar 6, 2021

zodf0055980 commented on Mar 6, 2021 Author

Sorry, I will add `--enable-release` to run fuzzing, thanks.

rurban closed this as completed on Mar 8, 2021

Assignees

rurban

Labels

fuzzing

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants