LibreDWG / **libredwg** Public

<> Code  ⊙ **Issues** 77  Pull requests 2  Discussions  Actions  Security  ···

New issue

Jump to bottom

# heap-use-after-free exists in the function decode_preR13_section in decode_r11.c #487

⊙ Open  **cxlzff** opened this issue on Jun 6 · 2 comments

Assignees

Labels  bug  **fuzzing**  **invalid CVE**

**cxlzff** commented on Jun 6

## system info

Ubuntu x86_64, clang 6.0, dwg2dxf(0.12.4.4608)

## Command line

./programs/dwg2dxf -b -m @@ -o /dev/null

## AddressSanitizer output

```
==8989==ERROR: AddressSanitizer: heap-use-after-free on address 0x7ffff7e35838 at pc 0x0000007106ca
bp 0x7fffffffc8b0 sp 0x7fffffffc8a8
READ of size 8 at 0x7ffff7e35838 thread T0
#0 0x7106c9 in decode_preR13_section /testcase/libredwg/src/decode_r11.c:339:35
#1 0x705d0a in decode_preR13 /testcase/libredwg/src/decode_r11.c:830:12
#2 0x53245a in dwg_decode /testcase/libredwg/src/decode.c:209:23
#3 0x50d759 in dwg_read_file /testcase/libredwg/src/dwg.c:254:11
#4 0x50c454 in main /testcase/libredwg/programs/dwg2dxf.c:258:15
#5 0x7ffff6e22c86 in __libc_start_main /build/glibc-CVJwZb/glibc-2.27/csu/../csu/libc-start.c:310
#6 0x419ee9 in _start (/testcase/libredwg/programs/dwg2dxf+0x419ee9)
```

```
0x7ffff7e35838 is located 56 bytes inside of 172032-byte region [0x7ffff7e35800,0x7ffff7e5f800)
freed by thread T0 here:
#0 0x4d2968 in realloc /fuzzer/build/llvm_tools/llvm-4.0.0.src/projects/compiler-rt/lib/asan/asan_malloc_linux.cc:79
#1 0x5a05b5 in dwg_add_object /testcase/libredwg/src/decode.c:4730:35

previously allocated by thread T0 here:
#0 0x4d2750 in calloc /fuzzer/build/llvm_tools/llvm-4.0.0.src/projects/compiler-rt/lib/asan/asan_malloc_linux.cc:74
#1 0x5a0465 in dwg_add_object /testcase/libredwg/src/decode.c:4719:35

SUMMARY: AddressSanitizer: heap-use-after-free /testcase/libredwg/src/decode_r11.c:339:35 in decode_preR13_section
Shadow bytes around the buggy address:
0x10007efbeab0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x10007efbeac0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x10007efbead0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x10007efbeae0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x10007efbeaf0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x10007efbeb00: fd fd fd fd fd fd fd[fd]fd fd fd fd fd fd fd fd
0x10007efbeb10: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x10007efbeb20: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x10007efbeb30: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x10007efbeb40: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x10007efbeb50: 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
==8989==ABORTING
```

## poc

https://gitee.com/cxlzff/fuzz-poc/raw/master/libredwg/decode_preR13_uaf

**rurban** added bug fuzzing labels on Jun 7

**rurban** self-assigned this on Jun 7

**abergmann** commented on Jun 24

CVE-2022-33025 was assigned to this issue.

**rurban** commented on Jun 24 • edited ▾ Contributor

Hello? CVE's to unreleased experimental non-working code?

Invalid CVE, not repro in the latest release 0.12.5. decoding preR13 DWG's was never released.

**rurban** added the invalid CVE label on Jun 24

Assignees

rurban

Labels

bug fuzzing invalid CVE

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants