

New issue

Jump to bottom

A heap buffer overflow in bit_wcs2len at bits.c:1634 #255

Closed

seviezhou opened this issue on Jul 31, 2020 · 1 comment

Assignees



Labels

bug

fuzzing

Milestone

0.11

seviezhou commented on Jul 31, 2020

System info

Ubuntu X64, gcc (Ubuntu 5.5.0-12ubuntu1), dwg2dxf (latest master [aee0ea](#))

Configure

CONFIG="g -fsanitize=address" LD_FLAGS="-fsanitize=address" ./configure

Command line

./programs/dwg2dxf -b -m ./heap-buffer-overflow-bit_wcs2len-bits-1634 -o /dev/null

AddressSanitizer output

```
=====
==15425==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x61600000ec80 at pc 0x561b14ea8544 bp 0x7ffcc39e7810 sp 0x7ffcc39e7800
READ of size 2 at 0x61600000ec80 thread T0
#0 0x561b14ea8543 in bit_wcs2len /home/seviezhou/libredwg/src/bits.c:1634
#1 0x561b145837d0 in dwg_decode_LTYPE_private /home/seviezhou/libredwg/src/dwg.spec:3018
#2 0x561b14f5906b in dwg_decode_LTYPE /home/seviezhou/libredwg/src/dwg.spec:2936
#3 0x561b14f5906b in dwg_decode_add_object /home/seviezhou/libredwg/src/decode.c:5660
#4 0x561b14f60d90 in read_2004_section_handles /home/seviezhou/libredwg/src/decode.c:2835
#5 0x561b14f60d90 in decode_R2004 /home/seviezhou/libredwg/src/decode.c:3671
#6 0x561b14f6f3db in dwg_decode /home/seviezhou/libredwg/src/decode.c:242
#7 0x561b14e6a1fc in dwg_read_file /home/seviezhou/libredwg/src/dwg.c:251
#8 0x561b14e67594 in main /home/seviezhou/libredwg/programs/dwg2dxf.c:258
#9 0x7fdefebfb96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#10 0x561b14e68689 in _start (/home/seviezhou/libredwg/programs/dwg2dxf+0xa4b689)

0x61600000ec81 is located 0 bytes to the right of 513-byte region [0x61600000ea80,0x61600000ec81)
allocated by thread T0 here:
#0 0x7defeff4857aa in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x987aa)
#1 0x561b14ea4348 in bit_read_TF /home/seviezhou/libredwg/src/bits.c:1444



SUMMARY: AddressSanitizer: heap-buffer-overflow /home/seviezhou/libredwg/src/bits.c:1634 bit_wcs2len
Shadow bytes around the buggy address:
 0x0c2c7fff9d40: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c2c7fff9d50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c2c7fff9d60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c2c7fff9d70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c2c7fff9d80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c2c7fff9d90:[01]fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c2c7fff9da0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c2c7fff9db0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c2c7fff9dc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c2c7fff9dd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c2c7fff9de0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
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
==15425==ABORTING
```

POC

[heap-buffer-overflow-bit_wcs2len-bits-1634.zip](#)

rurban added this to the 0.11 milestone on Jul 31, 2020

  **rurban** self-assigned this on Jul 31, 2020

  **rurban** added `bug` `fuzzing` labels on Jul 31, 2020

rurban commented on Jul 31, 2020

Contributor

I had to add a bounded `bit_wcs2n1en` to check against 512 overflows here. Thanks.


 **rurban** added a commit that referenced this issue on Jul 31, 2020

 decode: protect `LTYPE.dash_i` from overflowing 512 ...

✗ 4b99edb

 **rurban** closed this as completed on Aug 1, 2020

 **rurban** added a commit that referenced this issue on Aug 2, 2020

 fixup `LTYPE.dashes` overflows ...

✗ dac8fcc

Assignees

 **rurban**

Labels

`bug` `fuzzing`

Projects

None yet

Milestone

0.11

Development

No branches or pull requests

2 participants

