New issue

Several bugs found by fuzzing #179

⊙ Closed   **linhlhq** opened this issue on Jan 2, 2020 · 7 comments

Assignees	
Labels	bug **fuzzing**
Milestone	⊟ 0.10

---

linhlhq commented on Jan 2, 2020

Hi,

After fuzzing libredwg, I found the following bugs on the latest commit on master.

Command: ./dwg2svg2 $PoC

1.heap-buffer-overflow in read_pages_map ../../src/decode_r2007.c:1007

POC: https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_0.9.3.2564/id:000239%2Csig:06%2Csrc:007083%2Cop:havoc%2Crep:4

ASAN says:

```
==4335==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x611000000258 at pc 0x55f9e1e04e05 bp 0x7ffc92f94a40 sp 0x7ffc92f94a30
READ of size 8 at 0x611000000258 thread T0
    #0 0x55f9e1e04e04 in read_pages_map ../../src/decode_r2007.c:1007
    #1 0x55f9e1e04e04 in read_r2007_meta_data ../../src/decode_r2007.c:1774
    #2 0x55f9e1dd66d7 in decode_R2007 ../../src/decode.c:2973
    #3 0x55f9e1dd66d7 in dwg_decode ../../src/decode.c:241
    #4 0x55f9e177b466 in dwg_read_file ../../src/dwg.c:210
    #5 0x55f9e1776d4b in test_SVG ../../examples/dwg2svg2.c:116
    #6 0x55f9e1776d4b in main ../../examples/dwg2svg2.c:501
    #7 0x7f595f806b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
    #8 0x55f9e17779a9 in _start (/home/user/linhlhq/libredwg/asan_build/examples/dwg2svg2+0x2d59a9)

0x61100000025c is located 0 bytes to the right of 220-byte region [0x611000000180,0x61100000025c)
allocated by thread T0 here:
    #0 0x7f5960052d38 in __interceptor_calloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xded38)
    #1 0x55f9e1de73be in read_system_page ../../src/decode_r2007.c:635

SUMMARY: AddressSanitizer: heap-buffer-overflow ../../src/decode_r2007.c:1007 in read_pages_map
Shadow bytes around the buggy address:
  0x0c227fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c227fff8000: fa fa fa fa fa fa fa fa 00 00 00 00 00 00 00 00
  0x0c227fff8010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c227fff8020: 00 00 00 00 00 00 fa fa fa fa fa fa fa fa fa fa
  0x0c227fff8030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c227fff8040: 00 00 00 00 00 00 00 00 00 00 00[04]fa fa fa fa
  0x0c227fff8050: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c227fff8060: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c227fff8070: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c227fff8080: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c227fff8090: 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
==4335==ABORTING
```

---

linhlhq commented on Jan 2, 2020   Author

2.heap-buffer-overflow in bit_search_sentinel ../../src/bits.c:1844

POC: https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_0.9.3.2564/id:000361%2Csig:06%2Csrc:001213%2B008342%2Cop:splice%2Crep:4

ASAN says:

```
READ of size 1 at 0x7f8f1c662900 thread T0
    #0 0x556621519ff4 in bit_search_sentinel ../../src/bits.c:1844
    #1 0x556621ae0a78 in decode_R13_R2000 ../../src/decode.c:1437
    #2 0x556621b0cd42 in dwg_decode ../../src/decode.c:239
    #3 0x556621b2466 in dwg_read_file ../../src/dwg.c:210
    #4 0x5566214add4b in test_SVG ../../examples/dwg2svg2.c:116
    #5 0x5566214add4b in main ../../examples/dwg2svg2.c:501
    #6 0x7f8f1ae29b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
    #7 0x5566214ae9a9 in _start (/home/user/linhlhq/libredwg/asan_build/examples/dwg2svg2+0x2d59a9)

0x7f8f1c662900 is located 0 bytes to the right of 401664-byte region [0x7f8f1c600800,0x7f8f1c662900)
allocated by thread T0 here:
    #0 0x7f8f1b675d38 in __interceptor_calloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xded38)
    #1 0x5566214b234c in dat_read_file ../../src/dwg.c:73
    #2 0x5566214b234c in dwg_read_file ../../src/dwg.c:203
```

```
0x0ff2638c44d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ff2638c44e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ff2638c44f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ff2638c4500: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ff2638c4510: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0ff2638c4520:[fa]fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0ff2638c4530: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0ff2638c4540: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0ff2638c4550: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0ff2638c4560: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0ff2638c4570: 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
==4471==ABORTING
```

---

linhlhq commented on Jan 2, 2020   `Author`

3.heap-buffer-overflow in bfr_read ../../src/decode.c:1548
POC: https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_0.9.3.2564/id:000328%2Csig:06%2Csrc:007279%2Cop:havoc%2Crep:8
ASAN says:

```
==4589==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x61b0000014f4 at pc 0x7fe367220733 bp 0x7fff95bb63e0 sp 0x7fff95bb5b88
READ of size 96 at 0x61b0000014f4 thread T0
    #0 0x7fe367220732  (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79732)
    #1 0x559f5ee1a6a9 in memcpy /usr/include/x86_64-linux-gnu/bits/string_fortified.h:34
    #2 0x559f5ee1a6a9 in bfr_read ../../src/decode.c:1548
    #3 0x559f5ee1a6a9 in read_R2004_section_info ../../src/decode.c:2062
    #4 0x559f5f3e2392 in decode_R2004 ../../src/decode.c:2910
    #5 0x559f5f3eac7d in dwg_decode ../../src/decode.c:245
    #6 0x559f5ed90466 in dwg_read_file ../../src/dwg.c:210
    #7 0x559f5ed8bd4b in test_SVG ../../examples/dwg2svg2.c:116
    #8 0x559f5ed8bd4b in main ../../examples/dwg2svg2.c:501
    #9 0x7fe366a39b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
    #10 0x559f5ed8c9a9 in _start (/home/user/linhlhq/libredwg/asan_build/examples/dwg2svg2+0x2d59a9)

0x61b0000014f4 is located 0 bytes to the right of 1652-byte region [0x61b000000e80,0x61b0000014f4)
allocated by thread T0 here:
    #0 0x7fe367285d38 in __interceptor_calloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xded38)
    #1 0x559f5ee1a1e7 in read_R2004_section_info ../../src/decode.c:2007
    #2 0x2ddf1  (<unknown module>)

SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79732)
Shadow bytes around the buggy address:
  0x0c367fff8240: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c367fff8250: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c367fff8260: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c367fff8270: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c367fff8280: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c367fff8290: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00[04]fa
  0x0c367fff82a0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c367fff82b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c367fff82c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c367fff82d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c367fff82e0: 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
==4589==ABORTING
```

---

linhlhq commented on Jan 2, 2020   `Author`

4.Crafted input will lead to Memory allocation failed in read_sections_map ../../src/decode_r2007.c:917
POC: https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_0.9.3.2564/id:000271%2Csig:06%2Csrc:007728%2Cop:havoc%2Crep:2
ASAN says:

```
0x00008fff7000-0x02008fff7000
0x02008fff7000-0x10007fff8000
0x561598094000-0x561598fd9000 /home/user/linhlhq/libredwg/asan_build/examples/dwg2svg2
0x5615991d9000-0x5615991f1000 /home/user/linhlhq/libredwg/asan_build/examples/dwg2svg2
0x5615991f1000-0x5615992b7000 /home/user/linhlhq/libredwg/asan_build/examples/dwg2svg2
......
0x7feefb0c4000-0x7feefb0c8000
0x7feefb0c8000-0x7feefb0cf000 /lib/x86_64-linux-gnu/librt-2.27.so
0x7feefb0cf000-0x7feefb2ce000 /lib/x86_64-linux-gnu/librt-2.27.so
0x7feefb2ce000-0x7feefb2cf000 /lib/x86_64-linux-gnu/librt-2.27.so
0x7feefb2cf000-0x7feefb2d0000 /lib/x86_64-linux-gnu/librt-2.27.so
0x7feefb2d0000-0x7feefb2d3000 /lib/x86_64-linux-gnu/libdl-2.27.so
0x7feefb2d3000-0x7feefb4d2000 /lib/x86_64-linux-gnu/libdl-2.27.so
0x7feefb4d2000-0x7feefb4d3000 /lib/x86_64-linux-gnu/libdl-2.27.so
0x7feefb4d3000-0x7feefb4d4000 /lib/x86_64-linux-gnu/libdl-2.27.so
0x7feefb4d4000-0x7feefb6bb000 /lib/x86_64-linux-gnu/libc-2.27.so
0x7feefb6bb000-0x7feefb8bb000 /lib/x86_64-linux-gnu/libc-2.27.so
0x7feefb8bb000-0x7feefb8bf000 /lib/x86_64-linux-gnu/libc-2.27.so
0x7feefb8bf000-0x7feefb8c1000 /lib/x86_64-linux-gnu/libc-2.27.so
0x7feefb8c1000-0x7feefb8c5000
0x7feefb8c5000-0x7feefba62000 /lib/x86_64-linux-gnu/libm-2.27.so
0x7feefba62000-0x7feefbc61000 /lib/x86_64-linux-gnu/libm-2.27.so
0x7feefbc61000-0x7feefbc62000 /lib/x86_64-linux-gnu/libm-2.27.so
0x7feefbc62000-0x7feefbc63000 /lib/x86_64-linux-gnu/libm-2.27.so
0x7feefbc63000-0x7feefbdb3000 /usr/lib/x86_64-linux-gnu/libasan.so.4.0.0
0x7feefbdb3000-0x7feefbfb3000 /usr/lib/x86_64-linux-gnu/libasan.so.4.0.0
0x7feefbfb3000-0x7feefbfb6000 /usr/lib/x86_64-linux-gnu/libasan.so.4.0.0
0x7feefbfb6000-0x7feefbfb9000 /usr/lib/x86_64-linux-gnu/libasan.so.4.0.0
0x7feefbfb9000-0x7feefcc1e000
0x7feefcc1e000-0x7feefcc45000 /lib/x86_64-linux-gnu/ld-2.27.so
0x7feefccaa000-0x7feefce38000
0x7feefce38000-0x7feefce45000
0x7feefce45000-0x7feefce46000 /lib/x86_64-linux-gnu/ld-2.27.so
0x7feefce46000-0x7feefce47000 /lib/x86_64-linux-gnu/ld-2.27.so
0x7feefce47000-0x7feefce48000
0x7ffe47737000-0x7ffe47758000 [stack]
0x7ffe477e4000-0x7ffe477e7000 [vvar]
0x7ffe477e7000-0x7ffe477e9000 [vdso]
0xffffffffff600000-0xffffffffff601000 [vsyscall]
==4839==End of process memory map.
==4839==AddressSanitizer CHECK failed: ../../../../src/libsanitizer/sanitizer_common/sanitizer_common.cc:118 "((0 && "unable to mmap")) != (0)" (0x0, 0x0)
    #0 0x7feefbd4cc02  (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xe9c02)
    #1 0x7feefbd6b595 in __sanitizer::CheckFailed(char const*, int, char const*, unsigned long long, unsigned long long) (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x108595)
    #2 0x7feefbd56492  (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xf3492)
    #3 0x7feefbd628a5  (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xff8a5)
    #4 0x7feefbc8f8f1  (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x2c8f1)
    #5 0x7feefbc8a04b  (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x2704b)
    #6 0x7feefbd41d00 in __interceptor_calloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xded00)
    #7 0x5615989dad4d in read_sections_map ../../src/decode_r2007.c:917
    #8 0x5615989ecaee in read_r2007_meta_data ../../src/decode_r2007.c:1800
    #9 0x5615989c86d7 in decode_R2007 ../../src/decode.c:2973
    #10 0x5615989c86d7 in dwg_decode ../../src/decode.c:241
    #11 0x56159836d466 in dwg_read_file ../../src/dwg.c:210
    #12 0x561598368d4b in test_SVG ../../examples/dwg2svg2.c:116
    #13 0x561598368d4b in main ../../examples/dwg2svg2.c:501
    #14 0x7feefb4f5b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
    #15 0x5615983699a9 in _start (/home/user/linhlhq/libredwg/asan_build/examples/dwg2svg2+0x2d59a9)
```

---

linhlhq commented on Jan 2, 2020 · `Author`

5.heap-buffer-overflow in copy_compressed_bytes ../../src/decode_r2007.c:233
POC: https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_0.9.3.2564/id:000426%2Csig:06%2Csrc:009599%2Cop:havoc%2Crep:8
ASAN says:

```
==4975==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x629000009e7c at pc 0x564f138ce00b bp 0x7ffcab4ef510 sp 0x7ffcab4ef500
READ of size 8 at 0x629000009e7c thread T0
    #0 0x564f138ce00a in memcpy /usr/include/x86_64-linux-gnu/bits/string_fortified.h:34
    #1 0x564f138ce00a in copy_compressed_bytes ../../src/decode_r2007.c:233
    #2 0x564f138ce00a in decompress_r2007 ../../src/decode_r2007.c:519
    #3 0x564f138d42c7 in read_data_page ../../src/decode_r2007.c:694
    #4 0x564f138d42c7 in read_data_section ../../src/decode_r2007.c:758
    #5 0x564f138e1c04 in read_2007_section_handles ../../src/decode_r2007.c:1544
    #6 0x564f138e1c04 in read_r2007_meta_data ../../src/decode_r2007.c:1811
    #7 0x564f138bd6d7 in decode_R2007 ../../src/decode.c:2973
    #8 0x564f138bd6d7 in dwg_decode ../../src/decode.c:241
    #9 0x564f13262466 in dwg_read_file ../../src/dwg.c:210
    #10 0x564f1325dd4b in test_SVG ../../examples/dwg2svg2.c:116
    #11 0x564f1325dd4b in main ../../examples/dwg2svg2.c:501
    #12 0x7f78102e7b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
    #13 0x564f1325e9a9 in _start (/home/user/linhlhq/libredwg/asan_build/examples/dwg2svg2+0x2d59a9)

0x629000009e7c is located 2 bytes to the right of 19578-byte region [0x629000005200,0x629000009e7a)
allocated by thread T0 here:
    #0 0x7f7810b33d38 in __interceptor_calloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xded38)
    #1 0x564f138d3d31 in decode_rs ../../src/decode_r2007.c:580
    #2 0x564f138d3d31 in read_data_page ../../src/decode_r2007.c:689
    #3 0x564f138d3d31 in read_data_section ../../src/decode_r2007.c:758

SUMMARY: AddressSanitizer: heap-buffer-overflow /usr/include/x86_64-linux-gnu/bits/string_fortified.h:34 in memcpy
Shadow bytes around the buggy address:
  0x0c527fff9370: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c527fff9380: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c527fff9390: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c527fff93a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c527fff93b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c527fff93c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00[02]
  0x0c527fff93d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c527fff93e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c527fff93f0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c527fff9400: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c527fff9410: 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
```

```
     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
  ==4975==ABORTING
```

---

linhlhq commented on Jan 2, 2020   `Author`

6.NULL pointer dereference in get_next_owned_entity ../../src/dwg.c:935
POC: https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_0.9.3.2564/id:000113%2Csig:06%2Csrc:000000%2Cop:flip2%2Cpos:398289
ASAN says:

```
    =================================================================
    ==5183==ERROR: AddressSanitizer: SEGV on unknown address 0x0000000000b0 (pc 0x55c862f1795c bp 0x0c680000057d sp 0x7ffc02b51490 T0)
    ==5183==The signal is caused by a READ memory access.
    ==5183==Hint: address points to the zero page.
        #0 0x55c862f1795b in get_next_owned_entity ../../src/dwg.c:935
        #1 0x55c862f0814b in output_BLOCK_HEADER ../../examples/dwg2svg2.c:347
        #2 0x55c862f07078 in output_SVG ../../examples/dwg2svg2.c:395
        #3 0x55c862f07078 in test_SVG ../../examples/dwg2svg2.c:118
        #4 0x55c862f07078 in main ../../examples/dwg2svg2.c:501
        #5 0x7fd3ed723b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
        #6 0x55c862f079a9 in _start (/home/user/linhlhq/libredwg/asan_build/examples/dwg2svg2+0x2d59a9)

    AddressSanitizer can not provide additional info.
    SUMMARY: AddressSanitizer: SEGV ../../src/dwg.c:935 in get_next_owned_entity
    ==5183==ABORTING
```

---

linhlhq commented on Jan 2, 2020   `Author`

7.NULL pointer dereference in dwg_dynapi_entity_value /home/user/linhlhq/libredwg/asan_build/src/gen-dynapi.pl:1395
POC: https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_0.9.3.2564/id:000026%2Csig:06%2Csrc:000000%2Cop:flip1%2Cpos:132007
ASAN says:

```
    =================================================================
    ==5356==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000010 (pc 0x7f0447efc57e bp 0x7ffe6ac888f0 sp 0x7ffe6ac88058 T0)
    ==5356==The signal is caused by a READ memory access.
    ==5356==Hint: address points to the zero page.
        #0 0x7f0447efc57d  (/lib/x86_64-linux-gnu/libc.so.6+0xbb57d)
        #1 0x7f04486496ce  (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x796ce)
        #2 0x56329a7c1785 in memcpy /usr/include/x86_64-linux-gnu/bits/string_fortified.h:34
        #3 0x56329a7c1785 in dwg_dynapi_entity_value /home/user/linhlhq/libredwg/asan_build/src/gen-dynapi.pl:1395
        #4 0x563299fbaf1f in output_BLOCK_HEADER ../../examples/dwg2svg2.c:335
        #5 0x563299fba078 in output_SVG ../../examples/dwg2svg2.c:395
        #6 0x563299fba078 in test_SVG ../../examples/dwg2svg2.c:118
        #7 0x563299fba078 in main ../../examples/dwg2svg2.c:501
        #8 0x7f0447e62b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
        #9 0x563299fba9a9 in _start (/home/user/linhlhq/libredwg/asan_build/examples/dwg2svg2+0x2d59a9)

    AddressSanitizer can not provide additional info.
    SUMMARY: AddressSanitizer: SEGV (/lib/x86_64-linux-gnu/libc.so.6+0xbb57d)
    ==5356==ABORTING
```

✎ **linhlhq** changed the title ~~Sever~~ Several bugs found by fuzzing on Jan 2, 2020

---

rurban commented on Jan 3, 2020 • edited ▾   `Contributor`

Thanks, but dwg2svg2 is not even installed. cannot you just fuzz the official dwg2SVG instead?
Which master exactly? I'm assuming tag 0.9.3.2564. I'm constantly fixing fuzzing bugs right now, just >300 in the last 2 days.

👤 **rurban** self-assigned this on Jan 3, 2020

🏷 **rurban** added the `bug` label on Jan 3, 2020

↗ **rurban** added a commit that referenced this issue on Jan 3, 2020

    🖼 dwg2svg2: fail on NULL _hdr ⋯   757b87f

↗ **rurban** added a commit that referenced this issue on Jan 3, 2020

    🖼 Fix NULL ptr deref in get_next_owned_entity ⋯   540e1e4

protect r2007 decode compression length   c893d84

rurban added a commit that referenced this issue on Jan 3, 2020

protect r2007 section.num_pages overflow   ...   d5fe684

rurban added a commit that referenced this issue on Jan 3, 2020

fix read_R2004_section_info out of range check   ...   25228ca

rurban added a commit that referenced this issue on Jan 3, 2020

fix bit_search_sentinel out of range check   ...   5c19f59

rurban added a commit that referenced this issue on Jan 3, 2020

add r2007 Page out of bounds check   ...   fc8c16c

rurban closed this as completed on Jan 3, 2020

rurban added this to the **0.10** milestone on Jan 6, 2020

rurban added the fuzzing label on Jan 16, 2020

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

in_json: fix heap-overflow on parser errmsg   ...   932fdb2

DavidKorczynski mentioned this issue on Feb 23, 2021

libredwg: initial integration. google/oss-fuzz#5226

⤷ Merged

Assignees

rurban

Labels

bug   fuzzing

Projects

None yet

Milestone

0.10

Development

No branches or pull requests

2 participants