

New issue

Jump to bottom

# Several bugs found by fuzzing #188



linhlhq opened this issue on Jan 14, 2020 · 15 comments

Assignees



Labels

bug

fuzzing

Milestone



0.11

linhlhq commented on Jan 14, 2020

Hi,

After fuzzing libredwg, I found the following bugs on the latest commit on master.  
Command: ./dwgbmp \$PoC

## 1.NULL pointer dereference in read\_2004\_compressed\_section ../../src/decode.c:2417

POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_4c26d37/id:000012%2Csig:06%2Csrc:002489%2Ccop:havoc%2Crep:16](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_4c26d37/id:000012%2Csig:06%2Csrc:002489%2Ccop:havoc%2Crep:16)

```
=====
==20486==ERROR: AddressSanitizer: SEGV on unknown address 0x62904165558b (pc 0x7fa1bce115c5 bp 0x7fff8e06b350 sp 0x7fff8e06aab8 T0)
==20486==The signal is caused by a READ memory access.
#0 0x7fa1bce115c4 (/lib/x86_64-linux-gnu/libc.so.6+0xbb5c4)
#1 0x7fa1bd55e6ce (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x796ce)
#2 0x5634364fc7b6 in memcpy /usr/include/x86_64-linux-gnu/bits/string_fortified.h:34
#3 0x5634364fc7b6 in read_2004_compressed_section ../../src/decode.c:2417
#4 0x563436ac5dd2 in read_2004_section_summary ../../src/decode.c:2785
#5 0x563436ac5dd2 in decode_R2004 ../../src/decode.c:3352
#6 0x563436ad208d in dwg_decode ../../src/decode.c:246
#7 0x563436464fae in dwg_read_file ../../src/dwg.c:211
#8 0x5634364639d0 in get_bmp ../../programs/dwgbmp.c:111
#9 0x563436463066 in main ../../programs/dwgbmp.c:280
#10 0x7fa1bcd77b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#11 0x5634364636c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV (/lib/x86_64-linux-gnu/libc.so.6+0xbb5c4)
```

linhlhq commented on Jan 14, 2020

Author

## 2.NULL pointer dereference n get\_bmp ../../programs/dwgbmp.c:164

POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_4c26d37/id:000203%2Csig:06%2Csrc:007151%2B000917%2Ccop:splice%2Crep:64](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_4c26d37/id:000203%2Csig:06%2Csrc:007151%2B000917%2Ccop:splice%2Crep:64)

```
=====
==31948==ERROR: AddressSanitizer: SEGV on unknown address 0x62c097cf2538 (pc 0x7f0721de463e bp 0x0000e1e1e1e1 sp 0x7ffe6d2c5ea8 T0)
==31948==The signal is caused by a READ memory access.
#0 0x7f0721de463d (/lib/x86_64-linux-gnu/libc.so.6+0xbb63d)
#1 0x7f0721db4993 in _IO_file_xsputn (/lib/x86_64-linux-gnu/libc.so.6+0x8b993)
#2 0x7f0721da8976 in _IO_fwrite (/lib/x86_64-linux-gnu/libc.so.6+0x7f976)
#3 0x555a28f91ee4 in get_bmp ../../programs/dwgbmp.c:164
#4 0x555a28f91066 in main ../../programs/dwgbmp.c:280
#5 0x7f0721d4ab96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#6 0x555a28f916c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV (/lib/x86_64-linux-gnu/libc.so.6+0xbb63d)
```

linhlhq commented on Jan 14, 2020

Author

## 3.heap-buffer-overflow in read\_2004\_compressed\_section ../../src/decode.c:2417

POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_4c26d37/id:000046%2Csig:06%2Csrc:005438%2B0002843%2Ccop:splice%2Crep:2](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_4c26d37/id:000046%2Csig:06%2Csrc:005438%2B0002843%2Ccop:splice%2Crep:2)

```
=====
==2469==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62c0000070c0 at pc 0x7f300b076733 bp 0x7ffe0bc1be00 sp 0x7ffe0bc1b5a8
READ of size 33554560 at 0x62c0000070c0 thread T0
#0 0x7f300b076732 (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79732)
#1 0x55cdec4b77b6 in memcpy /usr/include/x86_64-linux-gnu/bits/string_fortified.h:34
#2 0x55cdec4b77b6 in read_2004_compressed_section ../../src/decode.c:2417
#3 0x55cdec47ea88 in read_2004_section_filedeplist ../../src/decode.c:2888
#4 0x55cdec47ea88 in decode_R2004 ../../src/decode.c:3361
#5 0x55cdec48d08d in dwg_decode ../../src/decode.c:246
#6 0x55cdec41ffae in dwg_read_file ../../src/dwg.c:211
#7 0x55cdec41e9d0 in get_bmp ../../programs/dwgbmp.c:111
#8 0x55cdec41e066 in main ../../programs/dwgbmp.c:280
#9 0x7f300a88fb96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#10 0x55cdec41e6c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)

0x62c0000070c0 is located 0 bytes to the right of 28352-byte region [0x62c000000200,0x62c0000070c0)
allocated by thread T0 here:
#0 0x7f300b00db38 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xded38)
#1 0x55cdec41ff41 in dat_read_file ../../src/dwg.c:74
#2 0x55cdec41ff41 in dwg_read_file ../../src/dwg.c:204

SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79732)
Shadow bytes around the buggy address:
 0x0c587fff8dc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

```
0x0c36711f8e00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c587fff8e10: 00 00 00 00 00 00 00 00[fa]fa fa fa fa fa fa
0x0c587fff8e20: fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e30: fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e40: fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e50: fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e60: 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
==2469==ABORTING
```

linhlhq commented on Jan 14, 2020

Author

#### 4.Memory leaks in dwg\_decode\_eed ../../src/decode.c:3638

POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_4c26d37/id:000171%2Csig:06%2Csrc:004378%2B003724%2Ccop:splice%2Crep:2](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_4c26d37/id:000171%2Csig:06%2Csrc:004378%2B003724%2Ccop:splice%2Crep:2)

```
=====
==6139==ERROR: LeakSanitizer: detected memory leaks

Direct leak of 80 byte(s) in 2 object(s) allocated from:
#0 0x7f50ca2edd38 in __interceptor_calloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xdd38)
#1 0x561a721a5b36 in dwg_decode_eed ../../src/decode.c:3638

SUMMARY: AddressSanitizer: 80 byte(s) leaked in 2 allocation(s).
```

linhlhq commented on Jan 14, 2020

Author

#### 5.heap-buffer-overflow in read\_2004\_section\_appinfo ../../src/decode.c:2842

POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_4c26d37/id:000091%2Csig:06%2Csrc:005906%2B004569%2Ccop:splice%2Crep:64](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_4c26d37/id:000091%2Csig:06%2Csrc:005906%2B004569%2Ccop:splice%2Crep:64)

```
=====
==8956==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62c0000070c0 at pc 0x7ff0838d6733 bp 0x7ffd5c5ceb70 sp 0x7ffd5c5ce318
READ of size 32768 at 0x62c0000070c0 thread T0
#0 0x7ff0838d6732 (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79732)
#1 0x5599a2abd7b6 in memcpy /usr/include/x86_64-linux-gnu/bits/string_fortified.h:34
#2 0x5599a2abd7b6 in read_2004_compressed_section ../../src/decode.c:2417
#3 0x5599a3083a9b in read_2004_section_appinfo ../../src/decode.c:2842
#4 0x5599a3083a9b in decode_R2004 ../../src/decode.c:3359
#5 0x5599a309308d in dwg_decode ../../src/decode.c:246
#6 0x5599a2a25fae in dwg_read_file ../../src/dwg.c:211
#7 0x5599a2a249d0 in get_bmp ../../programs/dwgbmp.c:111
#8 0x5599a2a24066 in main ../../programs/dwgbmp.c:280
#9 0x7ff0830efb96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#10 0x5599a2a246c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)

0x62c0000070c0 is located 0 bytes to the right of 28352-byte region [0x62c000000200,0x62c0000070c0)
allocated by thread T0 here:
#0 0x7ff08393bd38 in __interceptor_calloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xdd38)
#1 0x5599a2a25f41 in dat_read_file ../../src/dwg.c:74
#2 0x5599a2a25f41 in dwg_read_file ../../src/dwg.c:204

SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79732)
Shadow bytes around the buggy address:
0x0c587fff8dc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8dd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8de0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8df0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8e00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c587fff8e10: 00 00 00 00 00 00 00 00[fa]fa fa fa fa fa fa
0x0c587fff8e20: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e30: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e40: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e50: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e60: 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
```

linhlhq commented on Jan 14, 2020

Author

#### 6.heap-buffer-overflow in read\_2004\_compressed\_section ../../src/decode.c:2417

POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_4c26d37/id:000004%2Csig:06%2Csrc:000240%2Cop:havoc%2Crep:4](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_4c26d37/id:000004%2Csig:06%2Csrc:000240%2Cop:havoc%2Crep:4)

```
=====
==11971==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62d0000085c0 at pc 0x7f21ca533733 bp 0x7ffc413ece40 sp 0x7ffc413ce5e8
READ of size 1179776 at 0x62d0000085c0 thread T0
#0 0x7f21ca533732 in /usr/lib/x86_64-linux-gnu/libasan.so.4+0x79732
#1 0x5622b14827b6 in memcpy /usr/include/x86_64-linux-gnu/bits/string_fortified.h:34
#2 0x5622b14827b6 in read_2004_compressed_section ../../src/decode.c:2417
#3 0x5622b148b0d2 in read_2004_section_summary ../../src/decode.c:2785
#4 0x5622b148b0d2 in decode_R2004 ../../src/decode.c:3352
#5 0x5622b145808d in dwg_decode ../../src/decode.c:246
#6 0x5622b13eafae in dwg_read_file ../../src/dwg.c:211
#7 0x5622b13e99d0 in get_bmp ../../programs/dwgbmp.c:111
#8 0x5622b13e9066 in main ../../programs/dwgbmp.c:280
#9 0x7f21c9d4cb96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#10 0x5622b13e96c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)

0x62d0000085c0 is located 0 bytes to the right of 33216-byte region [0x62d000000400,0x62d0000085c0)
allocated by thread T0 here:
#0 0x7f21ca598d38 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xd38)
#1 0x5622b13eaf41 in dat_read_file ../../src/dwg.c:74
#2 0x5622b13eaf41 in dwg_read_file ../../src/dwg.c:204

SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79732)
Shadow bytes around the buggy address:
 0x0c5a7fff90e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c5a7fff9070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c5a7fff9080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c5a7fff9090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c5a7fff90a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c5a7fff90b0: 00 00 00 00 00 00 00 00 00[fa]fa fa fa fa fa fa
0x0c5a7fff90c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c5a7fff90d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c5a7fff90e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c5a7fff90f0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c5a7fff9100: 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
==11971==ABORTING
```

linhlhq commented on Jan 14, 2020

Author

#### 7.NULL pointer dereference in read\_2004\_compressed\_section ../../src/decode.c:2337

POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_4c26d37/id:000036%2Csig:06%2Csrc:004273%2Cop:havoc%2Crep:8](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_4c26d37/id:000036%2Csig:06%2Csrc:004273%2Cop:havoc%2Crep:8)

```
=====
==15970==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000000 (pc 0x55e389e038c1 bp 0x000000000000 sp 0x7ffe410ebab0 T0)
==15970==The signal is caused by a READ memory access.
==15970==Hint: address points to the zero page.
#0 0x55e389e038c0 in read_2004_compressed_section ../../src/decode.c:2337
#1 0x55e38a3cb81f in read_2004_section_template ../../src/decode.c:3136
#2 0x55e38a3cb81f in decode_R2004 ../../src/decode.c:3365
#3 0x55e38a3d908d in dwg_decode ../../src/decode.c:246
#4 0x55e389d0bfae in dwg_read_file ../../src/dwg.c:211
#5 0x55e389d0a9d0 in get_bmp ../../programs/dwgbmp.c:111
#6 0x55e389d0a066 in main ../../programs/dwgbmp.c:280
#7 0x7f67aa354b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#8 0x55e389d06a6c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV ../../src/decode.c:2337 in read_2004_compressed_section
==15970==ABORTING
```

linhlhq commented on Jan 14, 2020

Author

#### 8.heap-buffer-overflow in bit\_calc\_CRC ../../src/bits.c:2213

POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_4c26d37/id:000034%2Csig:06%2Csrc:003269%2B004108%2Cop:splice%2Crep:4](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_4c26d37/id:000034%2Csig:06%2Csrc:003269%2B004108%2Cop:splice%2Crep:4)

```
=====
==18234==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62b000006a00 at pc 0x55d63177a35 bp 0x7ffdcca65160 sp 0x7ffdcca65150
READ of size 1 at 0x62b000006a00 thread T0
#0 0x55d63177a34 in bit_calc_CRC ../../src/bits.c:2213
#1 0x55d63177a34 in bit_check_CRC ../../src/bits.c:1279
#2 0x55d63174e8b7 in dwg_decode_add_object ../../src/decode.c:5555
```

```
#0 0x55063312406c in dwg_read_file ../../src/dwg.c:211
#7 0x55d6331249d0 in get_bmp ../../programs/dwgbmp.c:111
#8 0x55d633124066 in main ../../programs/dwgbmp.c:280
#9 0x7f3fdd3f6b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#10 0x55d6331246c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)
```

0x62b000006a00 is located 0 bytes to the right of 26624-byte region [0x62b000000200,0x62b000006a00) allocated by thread T0 here:

```
#0 0x7f3fddc42d38 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xdd38)
#1 0x55d6331bcf25 in read_2004_compressed_section ../../src/decode.c:2321
#2 0x55d633c674b2 (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0xdd24b2)
```

SUMMARY: AddressSanitizer: heap-buffer-overflow ../../src/bits.c:2213 in bit\_calc\_CRC

Shadow bytes around the buggy address:

```
0x0c567fff8cf0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c567fff8d00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c567fff8d10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c567fff8d20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c567fff8d30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c567fff8d40:[fa]fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c567fff8d50: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c567fff8d60: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c567fff8d70: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c567fff8d80: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c567fff8d90: fa fa fa fa fa 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
==18234==ABORTING
```

linhlhq commented on Jan 14, 2020

Author

### 9.heap-buffer-overflow in read\_2004\_section\_handles ../../src/decode.c:2637

POC:[https://github.com/linhlhq/research/blob/master/PoCs/libredWG\\_4c26d37/id:000168%2Csig:06%2Csrc:001038%2Cophavoc%2Crep:4](https://github.com/linhlhq/research/blob/master/PoCs/libredWG_4c26d37/id:000168%2Csig:06%2Csrc:001038%2Cophavoc%2Crep:4)

```
=====
==21207==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x629000004f20 at pc 0x7fbed213733 bp 0x7ffdd9fa0a80 sp 0x7ffdd9fa0228
READ of size 29696 at 0x629000004f20 thread T0
#0 0x7fbed213732 (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79732)
#1 0x558f0e7517b6 in memcpy /usr/include/x86_64-linux-gnu/bits/string_fortified.h:34
#2 0x558f0e7517b6 in read_2004_compressed_section ../../src/decode.c:2417
#3 0x558f0ed15353 in read_2004_section_handles ../../src/decode.c:2637
#4 0x558f0ed15353 in decode_R2004 ../../src/decode.c:3354
#5 0x558f0ed2708d in dwg_decode ../../src/decode.c:246
#6 0x558f0e6b9fae in dwg_read_file ../../src/dwg.c:211
#7 0x558f0e6b89d0 in get_bmp ../../programs/dwgbmp.c:111
#8 0x558f0e6b8066 in main ../../programs/dwgbmp.c:280
#9 0x7fbed8ca2cb96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#10 0x558f0e6b86c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)
```

0x629000004f20 is located 0 bytes to the right of 19744-byte region [0x629000000200,0x629000004f20) allocated by thread T0 here:

```
#0 0x7fbed278d38 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xdd38)
#1 0x558f0e6b9f41 in dat_read_file ../../src/dwg.c:74
#2 0x558f0e6b9f41 in dwg_read_file ../../src/dwg.c:204
```

SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86\_64-linux-gnu/libasan.so.4+0x79732)

Shadow bytes around the buggy address:

```
0x0c527fff8990: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c527fff89a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c527fff89b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c527fff89c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c527fff89d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c527fff89e0: 00 00 00 00[fa]fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c527fff89f0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c527fff8a00: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c527fff8a10: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c527fff8a20: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c527fff8a30: fa fa fa fa fa fa 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
```

linhlhq commented on Jan 14, 2020

Author

#### 10.heap-buffer-overflow in read\_2004\_section\_classes ../../src/decode.c:2440

POC:[https://github.com/linhlhq/research/blob/master/PoCs/libredWG\\_4c26d37/id:000031%2Csig:06%2Csrc:002636%2Copenhavoc%2Crep:8](https://github.com/linhlhq/research/blob/master/PoCs/libredWG_4c26d37/id:000031%2Csig:06%2Csrc:002636%2Copenhavoc%2Crep:8)

```
=====
==23808==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62c0000070c0 at pc 0x7f52bd72c733 bp 0x7ffc9c1738a0 sp 0x7ffc9c173848
READ of size 29696 at 0x62c0000070c0 thread T0
#0 0x7f52bd72c732 (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79732)
#1 0x55e8f86a57b6 in memcpy /usr/include/x86_64-linux-gnu/bits/string_fortified.h:34
#2 0x55e8f86a57b6 in read_2004_compressed_section ../../src/decode.c:2417
#3 0x55e8f86aa24f in read_2004_section_classes ../../src/decode.c:2440
#4 0x55e8f8c690d0 in decode_R2004 ../../src/decode.c:3353
#5 0x55e8f8c7b08d in dwg_decode ../../src/decode.c:246
#6 0x55e8f860d0fae in dwg_read_file ../../src/dwg.c:211
#7 0x55e8f860c9d0 in get_bmp ../../programs/dwgbmp.c:111
#8 0x55e8f860c066 in main ../../programs/dwgbmp.c:280
#9 0x7f52bcf45b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#10 0x55e8f860c6c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)

0x62c0000070c0 is located 0 bytes to the right of 28352-byte region [0x62c000000200,0x62c0000070c0)
allocated by thread T0 here:
#0 0x7f52bd791d38 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x0d38)
#1 0x55e8f860d0f41 in dat_read_file ../../src/dwg.c:74
#2 0x55e8f860d0f41 in dwg_read_file ../../src/dwg.c:204
```

SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86\_64-linux-gnu/libasan.so.4+0x79732)

Shadow bytes around the buggy address:

```
0x0c587fff8dc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8dd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8de0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8df0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8e00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c587fff8e10: 00 00 00 00 00 00 00 00 00[fa]fa fa fa fa fa fa fa
0x0c587fff8e20: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e30: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e40: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e50: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e60: 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
==23808==ABORTING
```

linhlhq commented on Jan 14, 2020

Author

#### 11.heap-buffer-overflow in read\_2004\_section\_preview ../../src/decode.c:3175

POC:[https://github.com/linhlhq/research/blob/master/PoCs/libredWG\\_4c26d37/id:000016%2Csig:06%2Csrc:000009%2Copenhavoc%2Cpos:27771](https://github.com/linhlhq/research/blob/master/PoCs/libredWG_4c26d37/id:000016%2Csig:06%2Csrc:000009%2Copenhavoc%2Cpos:27771)

```
=====
==28918==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62c0000070c0 at pc 0x7fee7c87d733 bp 0x7ffdb7a72050 sp 0x7ffdb7a717f8
READ of size 29856 at 0x62c0000070c0 thread T0
#0 0x7fee7c87d732 (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79732)
#1 0x55ba7a6407b6 in memcpy /usr/include/x86_64-linux-gnu/bits/string_fortified.h:34
#2 0x55ba7a6407b6 in read_2004_compressed_section ../../src/decode.c:2417
#3 0x55ba7a640a02 in read_2004_section_preview ../../src/decode.c:3175
#4 0x55ba7a640a02 in decode_R2004 ../../src/decode.c:3356
#5 0x55ba7a61608d in dwg_decode ../../src/decode.c:246
#6 0x55ba7a5a8fae in dwg_read_file ../../src/dwg.c:211
#7 0x55ba7a5a79d0 in get_bmp ../../programs/dwgbmp.c:111
#8 0x55ba7a5a7066 in main ../../programs/dwgbmp.c:280
#9 0x7fee7c096b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#10 0x55ba7a5a76c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)
```

0x62c0000070c0 is located 0 bytes to the right of 28352-byte region [0x62c000000200,0x62c0000070c0)

allocated by thread T0 here:

```
#0 0x7fee7c8e2d38 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x0d38)
#1 0x55ba7a5a8f41 in dat_read_file ../../src/dwg.c:74
#2 0x55ba7a5a8f41 in dwg_read_file ../../src/dwg.c:204
```

SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86\_64-linux-gnu/libasan.so.4+0x79732)

Shadow bytes around the buggy address:

```
0x0c587fff8dc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8dd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8de0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8df0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8e00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c587fff8e10: 00 00 00 00 00 00 00 00 00[fa]fa fa fa fa fa fa fa
0x0c587fff8e20: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e30: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e40: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e50: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e60: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
```

```
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

==28918==ABORTING
```

linhlhq commented on Jan 14, 2020

Author

## 12. heap-buffer-overflow in bit\_search\_sentinel ../../src/bits.c:1985

POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_4c26d37/id:000009%2Csig:06%2Csrc:000009%2Ccop:flip1%2Cpos:27725](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_4c26d37/id:000009%2Csig:06%2Csrc:000009%2Ccop:flip1%2Cpos:27725)

```
=====
==30593==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6020000003f1 at pc 0x562e9b28f678 bp 0x7ffe7e543550 sp 0x7ffe7e543540
READ of size 1 at 0x6020000003f1 thread T0
#0 0x562e9b28f677 in bit_search_sentinel ../../src/bits.c:1985
#1 0x562e9b2bf2c4 in read_2004_section_classes ../../src/decode.c:2449
#2 0x562e9b87e0d0 in decode_R2004 ../../src/decode.c:3353
#3 0x562e9b89008d in dwg_decode ../../src/decode.c:246
#4 0x562e9b222fae in dwg_read_file ../../src/dwg.c:211
#5 0x562e9b2219d0 in get_bmp ../../programs/dwgbmp.c:111
#6 0x562e9b221066 in main ../../programs/dwgbmp.c:280
#7 0x7f561af17b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#8 0x562e9b2216c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)
```

0x6020000003f1 is located 0 bytes to the right of 1-byte region [0x6020000003f0,0x6020000003f1) allocated by thread T0 here:

```
#0 0x7f561b763d38 in __interceptor_calloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xded38)
#1 0x562e9b2b9f25 in read_2004_compressed_section ../../src/decode.c:2321
#2 0x562e9bd644b2 (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0xdd24b2)
```

SUMMARY: AddressSanitizer: heap-buffer-overflow ../../src/bits.c:1985 in bit\_search\_sentinel

Shadow bytes around the buggy address:

```
0x0c047fff8020: fa fa 00 00 fa fa 00 fa fa 00 fa fa 00 fa fa 00 fa
0x0c047fff8030: fa fa 00 fa fa 00 fa fa 00 fa fa 00 fa fa 04 fa
0x0c047fff8040: fa fa 02 fa fa 02 fa fa 02 fa fa 02 fa fa 04 fa
0x0c047fff8050: fa fa 02 fa fa 02 fa fa 02 fa fa 02 fa fa 02 fa
0x0c047fff8060: fa fa 02 fa fa 02 fa fa 02 fa fa 02 fa fa 00 02
=>0x0c047fff8070: fa fa 02 fa fa 02 fa fa 02 fa fa 02 fa fa [01]fa
0x0c047fff8080: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c047fff8090: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c047fff80a0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c047fff80b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c047fff80c0: 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

==30593==ABORTING
```

linhlhq commented on Jan 14, 2020

Author

## 13.memcpy-param-overlap in read\_2004\_section\_header ../../src/decode.c:2580

POC:[https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_4c26d37/id:000000%2Csig:06%2Csrc:000009%2Ccop:flip1%2Cpos:27641](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_4c26d37/id:000000%2Csig:06%2Csrc:000009%2Ccop:flip1%2Cpos:27641)

```
=====
==32738==ERROR: AddressSanitizer: memcpy-param-overlap: memory ranges [0x62c000008200,0x62c00000f600) and [0x62c0000067a0, 0x62c00000dba0) overlap
#0 0x7f52a17f7425 (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79425)
#1 0x55cd4d5ce7b6 in memcpy /usr/include/x86_64-linux-gnu/bits/string_fortified.h:34
#2 0x55cd4d5ce7b6 in read_2004_compressed_section ../../src/decode.c:2417
#3 0x55cd4db91900 in read_2004_section_header ../../src/decode.c:2580
#4 0x55cd4db91900 in decode_R2004 ../../src/decode.c:3350
#5 0x55cd4dba408d in dwg_decode ../../src/decode.c:246
#6 0x55cd4d536fae in dwg_read_file ../../src/dwg.c:211
#7 0x55cd4d5359d0 in get_bmp ../../programs/dwgbmp.c:111
#8 0x55cd4d535066 in main ../../programs/dwgbmp.c:280
#9 0x7f52a1010b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#10 0x55cd4d5356c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)
```

0x62c000008200 is located 0 bytes inside of 29696-byte region [0x62c000008200,0x62c00000f600) allocated by thread T0 here:

```
#0 0x7f52a185cd38 in __interceptor_calloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xded38)
#1 0x55cd4d5cdf25 in read_2004_compressed_section ../../src/decode.c:2321
```

```
allocated by chrtread to here.
#0 0x7f52a185cd38 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xded38)
#1 0x55cd4d536f41 in dat_read_file ../../src/dwg.c:74
#2 0x55cd4d536f41 in dwg_read_file ../../src/dwg.c:204

SUMMARY: AddressSanitizer: memcpy-param-overlap (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79425)
==32738==ABORTING
```

linhlhq commented on Jan 14, 2020

Author

#### 14.heap-buffer-overflow in read\_2004\_section\_revhistory ../../src/decode.c:3051

POC:[https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_4c26d37/id:000006%2Csig:06%2Csrc:000009%2Ccop:flip1%2Cpos:27665](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_4c26d37/id:000006%2Csig:06%2Csrc:000009%2Ccop:flip1%2Cpos:27665)

```
=====
==3444==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62c0000070c0 at pc 0x7f7b0a9aa733 bp 0x7ffde99cc390 sp 0x7ffde99cbb38
READ of size 29696 at 0x62c0000070c0 thread T0
#0 0x7f7b0a9aa732 (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79732)
#1 0x5651577e87b6 in memcpy /usr/include/x86_64-linux-gnu/bits/string_fortified.h:34
#2 0x5651577e87b6 in read_2004_compressed_section ../../src/decode.c:2417
#3 0x5651577ea80f in read_2004_section_revhistory ../../src/decode.c:3051
#4 0x565157db075a in decode_R2004 ../../src/decode.c:3363
#5 0x565157db08d in dwg_decode ../../src/decode.c:246
#6 0x565157750fae in dwg_read_file ../../src/dwg.c:211
#7 0x56515774f9d0 in get_bmp ../../programs/dwgbmp.c:111
#8 0x56515774f066 in main ../../programs/dwgbmp.c:280
#9 0x7f7b0a1c3b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#10 0x56515774f6c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)
```

0x62c0000070c0 is located 0 bytes to the right of 28352-byte region [0x62c000000200,0x62c0000070c0)  
allocated by thread T0 here:

```
#0 0x7f7b0a0fd38 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xded38)
#1 0x565157750f41 in dat_read_file ../../src/dwg.c:74
#2 0x565157750f41 in dwg_read_file ../../src/dwg.c:204
```

SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86\_64-linux-gnu/libasan.so.4+0x79732)

Shadow bytes around the buggy address:

```
0x0c587fff8dc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8dd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8de0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8df0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c587fff8e00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
=>0x0c587fff8e10: 00 00 00 00 00 00 00 00 00[fa]fa fa fa fa fa fa fa
0x0c587fff8e20: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e30: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e40: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e50: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587fff8e60: 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
```

==3444==ABORTING

linhlhq commented on Jan 14, 2020

Author

#### 15. heap-buffer-overflow in bit\_read\_B ../../src/bits.c:135

POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_4c26d37/id:000008%2Csig:06%2Csrc:000009%2Ccop:flip1%2Cpos:27724](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_4c26d37/id:000008%2Csig:06%2Csrc:000009%2Ccop:flip1%2Cpos:27724)

```
=====
==6101==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60c000000998 at pc 0x55fc249a4769 bp 0x7ffe70852d70 sp 0x7ffe70852d60
READ of size 1 at 0x60c000000998 thread T0
#0 0x55fc249a4768 in bit_read_B ../../src/bits.c:135
#1 0x55fc2502c10f in section_string_stream ../../src/decode_r2007.c:1372
#2 0x55fc24a1f8b1 in read_2004_section_classes ../../src/decode.c:2488
#3 0x55fc24fdcd0d in decode_R2004 ../../src/decode.c:3353
#4 0x55fc24fee08d in dwg_decode ../../src/decode.c:246
#5 0x55fc24980fae in dwg_read_file ../../src/dwg.c:211
#6 0x55fc2497f9d0 in get_bmp ../../programs/dwgbmp.c:111
#7 0x55fc2497f066 in main ../../programs/dwgbmp.c:280
#8 0x7fa7b531fb96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#9 0x55fc2497f6c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)
```

Address 0x60c000000998 is a wild pointer.

SUMMARY: AddressSanitizer: heap-buffer-overflow ../../src/bits.c:135 in bit\_read\_B

Shadow bytes around the buggy address:

```
0x0c187fff80e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c187fff80f0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c187fff8100: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c187fff8110: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c187fff8120: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c187fff8130: fa fa[fa]fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c187fff8140: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c187fff8150: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c187fff8160: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
```

```
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
==6101==ABORTING
```

linhlhq commented on Jan 14, 2020

Author

### 16.heap-buffer-overflow in bit\_read\_RC ../../src/bits.c:318

POC:[https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_4c26d37/id:000015%2Csig:06%2Csrc:000009%2Cop:flip2%2Cpos:27803](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_4c26d37/id:000015%2Csig:06%2Csrc:000009%2Cop:flip2%2Cpos:27803)


```
=====
==9805==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60e00000700 at pc 0x55924d56766b bp 0x7ffdb8bc7a50 sp 0x7ffdb8bc7a40
READ of size 1 at 0x60e00000700 thread T0
#0 0x55924d56766a in bit_read_RC ../../src/bits.c:318
#1 0x55924d543967 in dwg_bmp ../../src/dwg.c:468
#2 0x55924d53da29 in get_bmp ../../programs/dwgbmp.c:120
#3 0x55924d53d066 in main ../../programs/dwgbmp.c:280
#4 0x7f6cb0f7bb96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#5 0x55924d53d6c9 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0x28f6c9)

0x60e00000700 is located 0 bytes to the right of 160-byte region [0x60e00000660,0x60e00000700)
allocated by thread T0 here:
#0 0x7f6cb17c7d38 in __interceptor_calloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xded38)
#1 0x55924d5d5f25 in read_2004_compressed_section ../../src/decode.c:2321
#2 0x55924e0804b2 (/home/user/linhlhq/libredwg/asan_build/programs/dwgbmp+0xdd24b2)


SUMMARY: AddressSanitizer: heap-buffer-overflow ../../src/bits.c:318 in bit_read_RC
Shadow bytes around the buggy address:
 0x0c1c7fff8090: fa fa fa 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c1c7fff80a0: 00 00 00 00 00 00 00 00 fa fa fa fa fa fa fa
 0x0c1c7fff80b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c1c7fff80c0: 00 00 00 00 fa fa fa fa fa fa fa fa fa fa
 0x0c1c7fff80d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00
->0x0c1c7fff80e0: [fa]fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c1c7fff80f0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c1c7fff8100: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c1c7fff8110: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c1c7fff8120: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c1c7fff8130: 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
==9805==ABORTING
```

 rurban self-assigned this on Jan 16, 2020

 rurban added the bug label on Jan 16, 2020


 rurban added this to the 0.11 milestone on Jan 16, 2020

 rurban added a commit that referenced this issue on Jan 16, 2020

 dwgbmp: protect against BMP size overflow ...

6757b07


 rurban added a commit that referenced this issue on Jan 16, 2020

 decode: check section sizes ...

d447393



---

 decode: check section sizes ...

6ea3bdb

 **rurban** added a commit that referenced this issue on Jan 16, 2020


 decode: fix uncompressed section overflow ...

7fc2102

 **rurban** added a commit that referenced this issue on Jan 16, 2020

 decode: re-add one more sections check ...

f955c0b

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

 **rurban** added a commit that referenced this issue on Jan 16, 2020

 decode: protect overlarge section sizes ...

185889b

 **rurban** closed this as completed on Jan 16, 2020

---

Assignees

 **rurban**

Labels

bug **fuzzing**

Projects

None yet

Milestone

0.11

Development

No branches or pull requests

2 participants

