

New issue

Jump to bottom

# Several bugs found by fuzzing #182



linhlhq opened this issue on Jan 10, 2020 · 9 comments

Assignees



Labels

bug

fuzzing

Milestone

0.11

linhlhq commented on Jan 10, 2020

Hi,  
After fuzzing libredwg, I found the following bugs on the latest commit on master.  
Command: ./dwg2SVG \$PoC  
1.NULL pointer dereference in htmslescape ../programs/escape.c:29  
POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_7b9cb829/id:000013%2Csig:06%2Csrc:000000%2Cop:flip1%2Cpos:46417](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_7b9cb829/id:000013%2Csig:06%2Csrc:000000%2Cop:flip1%2Cpos:46417)  
ASAN says:

```
=====
==19607==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000000 (pc 0x7f433bb56646 bp 0x7ffed1176670 sp 0x7ffed1175de8 T0)
==19607==The signal is caused by a READ memory access.
==19607==Hint: address points to the zero page.
#0 0x7f433bb56645 (/lib/x86_64-linux-gnu/libc.so.6+0xb1645)
#1 0x7f433c28557b (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x5157b)
#2 0x55950b27cdd4 in htmslescape ../programs/escape.c:29
#3 0x55950b27abb5 in output_TEXT ../programs/dwg2SVG.c:113
#4 0x55950b27abb5 in output_object ../programs/dwg2SVG.c:312
#5 0x55950b27abb5 in output_BLOCK_HEADER ../programs/dwg2SVG.c:371
#6 0x55950b278477 in output_SVG ../programs/dwg2SVG.c:411
#7 0x55950b278477 in main ../programs/dwg2SVG.c:525
#8 0x7f433bac6b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#9 0x55950b278d19 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwg2SVG+0x27ad19)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV (/lib/x86_64-linux-gnu/libc.so.6+0xb1645)
==19607==ABORTING
```

linhlhq commented on Jan 10, 2020

Author

2.heap-buffer-overflow in htmslescape ../programs/escape.c:46  
POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_7b9cb829/id:000016%2Csig:06%2Csrc:000000%2Cop:flip1%2Cpos:125355](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_7b9cb829/id:000016%2Csig:06%2Csrc:000000%2Cop:flip1%2Cpos:125355)  
ASAN says:

```
==4460==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x603000044cc6 at pc 0x7f2ccb2af66e bp 0x7fffd5b9c04e0 sp 0x7fffd5b9bfc88
READ of size 22 at 0x603000044cc6 thread T0
#0 0x7f2ccb2af66d (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x5166d)
#1 0x561be92280e7 in strcat /usr/include/x86_64-linux-gnu/bits/string_fortified.h:128
#2 0x561be92280e7 in htmslescape ../programs/escape.c:46
#3 0x561be9225b38 in output_BLOCK_HEADER ../programs/dwg2SVG.c:361
#4 0x561be9223544 in output_SVG ../programs/dwg2SVG.c:418
#5 0x561be9223544 in main ../programs/dwg2SVG.c:525
#6 0x7f2ccb270b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#7 0x561be9223d19 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwg2SVG+0x27ad19)

0x603000044cc6 is located 0 bytes to the right of 22-byte region [0x603000044cb0,0x603000044cc6)
allocated by thread T0 here:
#0 0x7f2ccbabcb50 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xdeb50)
#1 0x561be9222de0 in htmslescape ../programs/escape.c:30

SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x5166d)
Shadow bytes around the buggy address:
0x0c0680000940: fa fa 00 00 00 00 fa fa 00 00 00 00 fa fa 00 00
0x0c0680000950: 00 00 fa fa 00 00 00 fa fa fa 00 00 00 00 fa fa
0x0c0680000960: 00 00 00 00 fa fa 00 00 00 00 fa fa 00 00 00 fa
0x0c0680000970: fa fa 00 00 00 00 fa fa 00 00 00 00 fa fa 00 00
0x0c0680000980: 00 00 fa fa fd fd fd fa fa fd fd fd fa fa
=>0x0c0680000990: fd fd fd fa fa 00 00[06]fa fa fa fa fa fa
0x0c06800009a0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c06800009b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c06800009c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c06800009d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c06800009e0: 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 10, 2020

Author

### 3.heap-buffer-overflow in htmlescape ../../programs/escape.c:51

POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_7b9cb829/id:000015%2Csig:06%2Csrc:000000%2Ccop:flip1%2Cpos:46436](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_7b9cb829/id:000015%2Csig:06%2Csrc:000000%2Ccop:flip1%2Cpos:46436)

ASAN says:

```
==1959==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x603000044c67 at pc 0x7f57a7a3f66e bp 0x7fff31617330 sp 0x7fff31616ad8
READ of size 16 at 0x603000044c67 thread T0
#0 0x7f57a7a3f66d (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x5166d)
#1 0x561c57375e53 in strcat /usr/include/x86_64-linux-gnu/bits/string_fortified.h:128
#2 0x561c57375e53 in htmlescape ../../programs/escape.c:51
#3 0x561c57373bb5 in output_TEXT ../../programs/dwg2SVG.c:113
#4 0x561c57373bb5 in output_object ../../programs/dwg2SVG.c:312
#5 0x561c57373bb5 in output_BLOCK_HEADER ../../programs/dwg2SVG.c:371
#6 0x561c57371477 in output_SVG ../../programs/dwg2SVG.c:411
#7 0x561c57371477 in main ../../programs/dwg2SVG.c:525
#8 0x7f57a7280b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#9 0x561c57371d19 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwg2SVG+0x27ad19)

0x603000044c67 is located 0 bytes to the right of 23-byte region [0x603000044c50,0x603000044c67)
allocated by thread T0 here:
#0 0x7f57a7accb50 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xdeb50)
#1 0x561c57375de0 in htmlescape ../../programs/escape.c:30

SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x5166d)
Shadow bytes around the buggy address:
 0x0c0680000930: 00 00 00 00 fa 00 00 00 00 fa 00 00 00 00 fa
 0x0c0680000940: fa fa 00 00 00 00 fa fa 00 00 00 00 fa fa 00 00
 0x0c0680000950: 00 00 fa fa 00 00 00 fa fa 00 00 00 00 fa fa
 0x0c0680000960: 00 00 00 00 fa fa 00 00 00 fa fa 00 00 00 fa
 0x0c0680000970: fa fa 00 00 00 00 fa fa 00 00 00 00 fa fa 00 00
=>0x0c0680000980: 00 00 fa fa fd fd fd fa fa 00 00[07]fa fa fa
 0x0c0680000990: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c06800009a0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c06800009b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c06800009c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c06800009d0: 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
==1959==ABORTING
```

linhlhq commented on Jan 10, 2020

Author

### 4.NULL pointer dereference in output\_TEXT ../../programs/dwg2SVG.c:114

POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_7b9cb829/id:000020%2Csig:06%2Csrc:000000%2Ccop:flip1%2Cpos:138350](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_7b9cb829/id:000020%2Csig:06%2Csrc:000000%2Ccop:flip1%2Cpos:138350)

ASAN says:

```
=====
==25992==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000000 (pc 0x55e38790e95d bp 0x7ffffbed38d0 sp 0x7ffffbed37b0 T0)
==25992==The signal is caused by a READ memory access.
==25992==Hint: address points to the zero page.
#0 0x55e38790e95c in output_TEXT ../../programs/dwg2SVG.c:114
#1 0x55e38790e95c in output_object ../../programs/dwg2SVG.c:312
#2 0x55e38790e95c in output_BLOCK_HEADER ../../programs/dwg2SVG.c:371
#3 0x55e38790d477 in output_SVG ../../programs/dwg2SVG.c:411
#4 0x55e38790d477 in main ../../programs/dwg2SVG.c:525
#5 0x7f08b41b1b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#6 0x55e38790dd19 in _start (/home/user/linhlhq/libredwg/asan_build/programs/dwg2SVG+0x27ad19)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV ../../programs/dwg2SVG.c:114 in output_TEXT
==25992==ABORTING
```

linhlhq commented on Jan 10, 2020

Author

### 5.heap-buffer-overflow in output\_TEXT ../../programs/dwg2SVG.c:114

POC: [https://github.com/linhlhq/research/blob/master/PoCs/libreDWG\\_7b9cb829/id:000109%2Csig:06%2Csrc:001432%2B002572%2Ccop:splice%2Crep:2](https://github.com/linhlhq/research/blob/master/PoCs/libreDWG_7b9cb829/id:000109%2Csig:06%2Csrc:001432%2B002572%2Ccop:splice%2Crep:2)

ASAN says:

```
=====
==17789==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x603000048c0 at pc 0x56426727812a bp 0x7ffe21f09bc0 sp 0x7ffe21f09bb0
READ of size 8 at 0x603000048c0 thread T0
#0 0x564267278129 in output_TEXT ../../programs/dwg2SVG.c:114
#1 0x564267278129 in output_object ../../programs/dwg2SVG.c:312
#2 0x564267278129 in output_BLOCK_HEADER ../../programs/dwg2SVG.c:371
#3 0x564267274544 in output_SVG ../../programs/dwg2SVG.c:418
```

```
0x603000048c0 is located 0 bytes to the right of 32-byte region [0x603000048a0,0x603000048c0)
allocated by thread T0 here:
#0 0x7f496174fd38 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xded38)
#1 0x56426731003d in dwg_decode_handlerref_with_code ../../src/decode.c:3890
```

SUMMARY: AddressSanitizer: heap-buffer-overflow ../../programs/dwg2SVG.c:114 in output\_TEXT

Shadow bytes around the buggy address:

```
0x0c067fff88c0: 00 00 00 00 fa 00 00 00 00 fa 00 00 00 00
0x0c067fff88d0: fa fa 00 00 00 00 fa 00 00 00 00 fa 00 00
0x0c067fff88e0: 00 00 fa fa 00 00 00 00 fa fa 00 00 00 fa fa
0x0c067fff88f0: 00 00 00 00 fa 00 00 00 fa fa 00 00 00 00
0x0c067fff8900: fa fa 00 00 00 00 fa 00 00 00 00 fa 00 00
=>0x0c067fff8910: 00 fa fa 00 00 00 00[fa]fa 00 00 00 fa fa
0x0c067fff8920: 00 00 00 00 fa 00 00 00 00 fa 00 00 00 00
0x0c067fff8930: fa fa 00 00 00 00 fa 00 00 00 00 fa 00 00
0x0c067fff8940: 00 00 fa fa 00 00 00 00 fa 00 00 00 00 fa fa
0x0c067fff8950: 00 00 00 00 fa 00 00 00 00 fa 00 00 00 00
0x0c067fff8960: fa fa 00 00 00 00 fa 00 00 00 00 fa 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
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
==17789==ABORTING
```

linlhq commented on Jan 10, 2020

Author

#### 6.heap-buffer-overflow in htmlwescapce ../../programs/escape.c:97

POC: [https://github.com/linlhq/research/blob/master/PoCs/libreDWG\\_7b9cb829/id:000025%2Csig:06%2Csrc:000804%2Ccop:havoc%2Crep:2](https://github.com/linlhq/research/blob/master/PoCs/libreDWG_7b9cb829/id:000025%2Csig:06%2Csrc:000804%2Ccop:havoc%2Crep:2)

ASAN says:

```
=====
==19465==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60300000725c at pc 0x7f769710266e bp 0x7ffef73ec550 sp 0x7ffef73ebcf8
READ of size 28 at 0x60300000725c thread T0
#0 0x7f769710266d (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x5166d)
#1 0x55d1b06549e7 in strcat /usr/include/x86_64-linux-gnu/bits/string_fortified.h:128
#2 0x55d1b06549e7 in htmlwescapce ../../programs/escape.c:97
#3 0x55d1b0650235 in output_BLOCK_HEADER ../../programs/dwg2SVG.c:359
#4 0x55d1b064f503 in output_SVG ../../programs/dwg2SVG.c:413
#5 0x55d1b064f503 in main ../../programs/dwg2SVG.c:525
#6 0x7f7696943b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#7 0x55d1b064fd19 in _start (/home/user/linlhq/libredwg/asan_build/programs/dwg2SVG+0x27ad19)
```

0x60300000725c is located 0 bytes to the right of 28-byte region [0x603000007240,0x60300000725c)
allocated by thread T0 here:

```
#0 0x7f769718fb50 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xdeb50)
#1 0x55d1b06543f2 in htmlwescapce ../../programs/escape.c:79
```

SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86\_64-linux-gnu/libasan.so.4+0x5166d)

Shadow bytes around the buggy address:

```
0x0c067fff88d0: 00 00 fa fa 00 00 00 00 fa fa 00 00 00 fa fa
0x0c067fff88e0: 00 00 00 00 fa 00 00 00 00 fa 00 00 00 fa
0x0c067fff88f0: fa fa 00 00 00 00 fa fa 00 00 00 00 fa 00 00
0x0c067fff8900: 00 fa fa 00 00 00 00 fa fa 00 00 00 00 fa fa
0x0c067fff8910: 00 00 00 fa fa fa 00 00 00 00 fa 00 00 00 00
=>0x0c067fff8920: fa fa fd fd fd fd fa 00 00 00[04]fa fa fa fa
0x0c067fff8930: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c067fff8940: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c067fff8950: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c067fff8960: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c067fff8970: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c067fff8980: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c067fff8990: 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
==19465==ABORTING
```

linlhq commented on Jan 10, 2020

Author

```
==9632==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60400000abd at pc 0x7f1b8c9f066e bp 0x7ffe890ea740 sp 0x7ffe890e9ee8
READ of size 10 at 0x60400000abd thread T0
#0 0x7f1b8c9f066d (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x5166d)
#1 0x55ff72e70067 in strcat /usr/include/x86_64-linux-gnu/bits/string_fortified.h:128
#2 0x55ff72e70067 in htmlEscape ../../programs/escape.c:48
#3 0x55ff72e6dbb5 in output_TEXT ../../programs/dwg2SVG.c:113
#4 0x55ff72e6dbb5 in output_object ../../programs/dwg2SVG.c:312
#5 0x55ff72e6dbb5 in output_BLOCK_HEADER ../../programs/dwg2SVG.c:371
#6 0x55ff72e6b544 in output_SVG ../../programs/dwg2SVG.c:418
#7 0x55ff72e6b544 in main ../../programs/dwg2SVG.c:525
#8 0x7f1b8c231b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#9 0x55ff72e6bd19 in _start (/home/user/linlhq/libredwg/asan_build/programs/dwg2SVG+0x27ad19)
```

0x60400000abd is located 0 bytes to the right of 45-byte region [0x60400000a90,0x60400000abd) allocated by thread T0 here:

```
#0 0x7f1b8ca7df40 in realloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xdef40)
#1 0x55ff72e6ffa1 in htmlEscape ../../programs/escape.c:39
```

SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86\_64-linux-gnu/libasan.so.4+0x5166d)

Shadow bytes around the buggy address:

```
0x0c087fff8100: fa fa fd fd fd fd fa fa 00 00 00 00 fa
0x0c087fff8110: fa fa fd fd fd fd fa fa fa fd fd fd fd fa
0x0c087fff8120: fa fa 00 00 00 00 fa fa fa 00 00 00 00 fa
0x0c087fff8130: fa fa 00 00 00 00 fa fa fa 00 00 00 05 fa
0x0c087fff8140: fa fa 00 00 00 00 fa fa fd fd fd fd fa
=>0x0c087fff8150: fa fa 00 00 00 00 00[05]fa fa fa fa fa fa
0x0c087fff8160: fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c087fff8170: fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c087fff8180: fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c087fff8190: fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c087fff81a0: 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  
==9632==ABORTING

linlhq commented on Jan 10, 2020

Author

## 8.heap-buffer-overflow in htmlwescape ../../programs/escape.c:97

POC: [https://github.com/linlhq/research/blob/master/PoCs/libreDWG\\_7b9cb829/id:000099%2Csig:06%2Csrc:005125%2B003017%2Ccop:splice%2Crep:8](https://github.com/linlhq/research/blob/master/PoCs/libreDWG_7b9cb829/id:000099%2Csig:06%2Csrc:005125%2B003017%2Ccop:splice%2Crep:8)

ASAN says:

```
==682==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6040000019fa at pc 0x7ff79912866e bp 0x7ffdc846b10 sp 0x7ffdc8462b8
READ of size 28 at 0x6040000019fa thread T0
#0 0x7ff79912866d (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x5166d)
#1 0x5586562889e7 in strcat /usr/include/x86_64-linux-gnu/bits/string_fortified.h:128
#2 0x5586562889e7 in htmlwescape ../../programs/escape.c:97
#3 0x5586562848de in output_TEXT ../../programs/dwg2SVG.c:111
#4 0x5586562848de in output_object ../../programs/dwg2SVG.c:312
#5 0x5586562848de in output_BLOCK_HEADER ../../programs/dwg2SVG.c:371
#6 0x558656283477 in output_SVG ../../programs/dwg2SVG.c:411
#7 0x558656283477 in main ../../programs/dwg2SVG.c:525
#8 0x7ff79909b996 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
#9 0x558656283d19 in _start (/home/user/linlhq/libredwg/asan_build/programs/dwg2SVG+0x27ad19)
```

0x6040000019fa is located 0 bytes to the right of 42-byte region [0x6040000019d0,0x6040000019fa) allocated by thread T0 here:

```
#0 0x7ff7991b5b50 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xdeb50)
#1 0x5586562883f2 in htmlwescape ../../programs/escape.c:79
```

SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86\_64-linux-gnu/libasan.so.4+0x5166d)

Shadow bytes around the buggy address:

```
0x0c087fff82e0: fa fa 00 00 00 00 fa fa fa 00 00 00 00 00
0x0c087fff82f0: fa fa fd fd fd fd fa fa 00 00 00 00 fa
0x0c087fff8300: fa fa 00 00 00 00 fa fa fa fd fd fd fd fa
0x0c087fff8310: fa fa 00 00 00 00 fa fa 00 00 00 00 fa
0x0c087fff8320: fa fa 00 00 00 00 fa fa fa fd fd fd fd fa
=>0x0c087fff8330: fa fa fd fd fd fd fa fa 00 00 00 00 00[02]
0x0c087fff8340: fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c087fff8350: fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c087fff8360: fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c087fff8370: fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c087fff8380: 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

```
ASAN:inernal. re
Left alloca redzone: ca
Right alloca redzone: cb
==682==ABORTING
```

 **rurban** self-assigned this on Jan 10, 2020


 **rurban** added the `bug` label on Jan 10, 2020

**rurban** commented on Jan 10, 2020

Contributor

Nice! I haven't fuzzed that yet.



 **rurban** added a commit that referenced this issue on Jan 10, 2020

 `skip NULL text_value` ...

b7fa220

 **rurban** added a commit that referenced this issue on Jan 10, 2020

 `fix off-by-one and overflow` ...

7e4c59b

 **rurban** added a commit that referenced this issue on Jan 10, 2020

 `fix off-by-one and overflow` ...

e8c3edb

 **rurban** added a commit that referenced this issue on Jan 10, 2020

 `protect from wrong style` ...


f2b5adb


**linhlhq** commented on Jan 10, 2020

Author

ya, I really like fuzz libreDWG because you are very hard working and dedicated to this libreDWG. I hope one day it will be safer.


 **rurban** closed this as completed on Jan 10, 2020

 **rurban** added this to the **0.11** milestone on Jan 10, 2020

 **rurban** added a commit that referenced this issue on Jan 12, 2020


 `Release 0.10.1` ...

3d9e2d4

 **rurban** added a commit that referenced this issue on Jan 12, 2020

 `Release 0.10.1` ...

828cb38

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

Assignees

 **rurban**

Labels

`bug` `fuzzing`

Projects

None yet

Milestone

0.11

Development

No branches or pull requests

2 participants

