Xfig Tickets

Xfig is a diagramming tool
Brought to you by: tklxfiguser

#74 stack-buffer-overflow in put_arrow at genpict2e.c:1191

Milestone: fig2dev **Status:** closed **Owner:** nobody **Labels:** None
Updated: 2020-12-21 **Created:** 2019-12-28 **Creator:** Suhwan Song **Private:** No

Hi,
I found a stack-buffer-overflow in put_arrow at genpict2e.c:1191
Please run following command to reproduce it,

```
fig2dev -L pict2e $PoC
```

ASAN LOG

```
==39178==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7fffc3316ec8 at pc 0x0
READ of size 4 at 0x7fffc3316ec8 thread T0
    #0 0x77150c in put_arrow /home/tmp/mcj-fig2dev/fig2dev/dev/genpict2e.c:1191:39
    #1 0x767f69 in genpict2e_arc /home/tmp/mcj-fig2dev/fig2dev/dev/genpict2e.c:2575:6
    #2 0x54b8bb in gendev_objects /home/tmp/mcj-fig2dev/fig2dev/fig2dev.c:1003:6
    #3 0x54b8bb in main /home/tmp/mcj-fig2dev/fig2dev/fig2dev.c:480
    #4 0x7f8b1eb44b96 in __libc_start_main /build/glibc-OTsEL5/glibc-2.27/csu/../csu/libc-s
    #5 0x41b3a9 in _start (/home/tmp/fig2dev+0x41b3a9)

Address 0x7fffc3316ec8 is located in stack of thread T0 at offset 72 in frame
    #0 0x76d4cf in put_arrow /home/tmp/mcj-fig2dev/fig2dev/dev/genpict2e.c:1152

  This frame has 9 object(s):
    [32, 36) 'sx' (line 1153)
    [48, 52) 'sy' (line 1153)
    [64, 68) 'lx' (line 1153) <== Memory access at offset 72 overflows this variable
    [80, 480) 'points' (line 1154) <== Memory access at offset 72 underflows this variable
    [544, 944) 'fillpoints' (line 1154)
    [1008, 1408) 'clippts' (line 1154)
    [1472, 1476) 'npoints' (line 1155)
    [1488, 1492) 'nfillpoints' (line 1155)
    [1504, 1508) 'nclippts' (line 1155)
HINT: this may be a false positive if your program uses some custom stack unwind mechanism
      (longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow /home/tmp/mcj-fig2dev/fig2dev/dev/genpict2
Shadow bytes around the buggy address:
  0x10007865ad80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007865ad90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007865ada0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007865adb0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007865adc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x10007865add0: f1 f1 f1 f1 04 f2 04 f2 04[f2]00 00 00 00 00 00
  0x10007865ade0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007865adf0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007865ae00: 00 00 00 00 00 00 00 00 00 00 00 00 f2 f2 f2 f2
  0x10007865ae10: f2 f2 f2 f2 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007865ae20: 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
  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
==39178==ABORTING
```

fig2dev Version 3.2.7b
I also tested this in git Commit [3065ab] and can reproduce it.

1 Attachments

id:000007,sig:06,src:000000,op:havoc,rep:2

Related

Commit: [3065ab]

Discussion

Dr. Werner Fink - *2020-01-29*

Seems that the POC here cause that `npoints` become `0` which leads to an ilegal access within the array `points[50]` within `put_arrow()` of `fig2dev/dev/genpict2e.c` ... remains the the question if `calc_arrow()` can return `npoints` larger than `50`.

issue74.patch

tkl - *2020-02-05*

The cause for the error is, that an arrow head with zero length (or height, as called internally in fig2dev) is tried to be drawn on an arc. But the direction of the arrow is computed as the chord, not the tangent, between the tip and the back of the arrow head. With zero length, the start- and the end points of the arrow coincide. Then, calc_arrow() detects this co-incidence and returns prematurely, but not gracefully, i.e., having set npoints to an invalid number, namely 0. The fix in commit [3165d8] is to compute the chord for an arrow height equal to the line thickness. Although, on second thought, the tangent would have been the correct choice, since this is the asymptotic limit for the arrow length approaching zero.

Related

Commit: [3165d8]

tkl - *2020-02-16*

- **status**: open --> pending

tkl - *2020-02-16*

With commit [100e27], arcs with arrow heads that have a length of zero are now drawn using the tangent, not a secant to the arc.

Related

Commit: [100e27]

tkl - *2020-12-21*

- **status**: pending --> closed

Log in to post a comment.