Xfig Tickets

Xfig is a diagramming tool

Brought to you by: tklxfiguser

#58 global-buffer-overflow in get_line() function

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

Hi,
I found global-buffer-overflow in get_line at read.c:1528
Please run following command to reproduce it,

```
fig2dev -L box $PoC
```

Here's log

```
==23137==ERROR: AddressSanitizer: global-buffer-overflow on address 0x556239047e7f at pc 0x
READ of size 1 at 0x556239047e7f thread T0
    #1 0x556238c5e9fb in read_objects fig2dev-3.2.7b/fig2dev/read.c:278
    #2 0x556238c5e1d3 in readfp_fig fig2dev-3.2.7b/fig2dev/read.c:172
    #3 0x556238c5e0a9 in read_fig fig2dev-3.2.7b/fig2dev/read.c:142
    #4 0x556238c55ef3 in main fig2dev-3.2.7b/fig2dev/fig2dev.c:422
    #5 0x7f9d44bf4b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
    #6 0x556238c46979 in _start (fig2dev-3.2.7b+0x6e979)

0x556239047e7f is located 59 bytes to the right of global variable 'gif_colnum' defined in
0x556239047e7f is located 1 bytes to the left of global variable 'buf' defined in 'read.c:8
Shadow bytes around the buggy address:
  0x0aacc7200f70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aacc7200f80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aacc7200f90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aacc7200fa0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aacc7200fb0: 00 00 00 00 00 00 00 00 00 f9 f9 f9 f9 f9 f9 f9
=>0x0aacc7200fc0: 00 00 00 00 00 00 00 00 04 f9 f9 f9 f9 f9 f9[f9]
  0x0aacc7200fd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aacc7200fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aacc7200ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aacc7201000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aacc7201010: 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
==23137==ABORTING
```

fig2dev Version 3.2.7b
I also tested this in git master and can reproduce it.

1 Attachments

id:000000,sig:06,src:000000,op:havoc,rep:128

Discussion

Suhwan Song - *2019-12-12*

Log is incomplete, it's my mistake

Log in to post a comment

jkl 2020-01-06

- Status: pending → closed
- Fixed with commit [41b9bb]

Related

Commit: [41b9bb]

```
==7335==ERROR: AddressSanitizer: global-buffer-overflow on address 0x558be
READ of size 1 at 0x558bed4e1e7f thread T0
    #0 0x558bed103e35 in get_line fig2dev-3.2.7b/fig2dev/read.c:1512
    #1 0x558bed0f8[illegible] in read_objects fig2dev-3.2.7b/fig2dev/read.c:278
    #2 0x558bed0f81d3 in readfp_fig fig2dev-3.2.7b/fig2dev/read.c:172
    #3 0x558bed0f80a9 in read_fig fig2dev-3.2.7b/fig2dev/read.c:142
    #4 0x558bed0efef3 in main fig2dev-3.2.7b/fig2dev/fig2dev.c:422
    #5 0x7fdf0a08cb96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.
    #6 0x558bed0e0979 in _start (fig2dev-3.2.7b+0x6e979)

0x558bed4e1e7f is located 59 bytes to the right of global variable 'gif_co
0x558bed4e1e7f is located 1 bytes to the left of global variable 'buf' def
SUMMARY: AddressSanitizer: global-buffer-overflow fig2dev-3.2.7b/fig2dev/r
Shadow bytes around the buggy address:
  0x0ab1fda94370: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ab1fda94380: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ab1fda94390: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ab1fda943a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ab1fda943b0: 00 00 00 00 00 00 00 00 00 f9 f9 f9 f9 f9 f9 f9
=>0x0ab1fda943c0: 00 00 00 00 00 00 00 00 04 f9 f9 f9 f9 f9 f9[f9]
  0x0ab1fda943d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ab1fda943e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ab1fda943f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ab1fda94400: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ab1fda94410: 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
==7335==ABORTING
```