Closed · Issue created 2 years ago by le3d1nq

[dot]heap-over-flow(off-by-null) in lib/common/shapes.c

tested on f8b9e035 / ubuntu 16.04 / clang / ASAN

Hi, I've came upon an heap-over-flow

parse_reclbl(lib/common/shapes.c:3311), you could trigger it by loading the content below with: dot filename

```
digraph structs {
struct [shape=record,label="<"];
}
```

ASAN report:

```
==14380==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6020000001f2 at pc 0x7ffff79fa166 bp 0x7fffffffdc60 sp
WRITE of size 1 at 0x6020000001f2 thread T0
    #0 0x7ffff79fa165 in parse_reclbl /root/graphviz/lib/common/shapes.c:3311:8
    #1 0x7ffff79ecd77 in record_init /root/graphviz/lib/common/shapes.c:3556:9
    #2 0x7ffff7a1a9e9 in common_init_node /root/graphviz/lib/common/utils.c:653:5
    #3 0x7ffff2b13c1d in dot_init_node /root/graphviz/lib/dotgen/dotinit.c:40:5
    #4 0x7ffff2b13c1d in dot_init_node_edge /root/graphviz/lib/dotgen/dotinit.c:81
    #5 0x7ffff2b1a4a4 in dotLayout /root/graphviz/lib/dotgen/dotinit.c:308:5
    #6 0x7ffff2b17517 in doDot /root/graphviz/lib/dotgen/dotinit.c:463:2
    #7 0x7ffff2b17517 in dot_layout /root/graphviz/lib/dotgen/dotinit.c:509
    #8 0x7ffff784e153 in gvLayoutJobs /root/graphviz/lib/gvc/gvlayout.c:85:2
    #9 0x516f3b in main /root/graphviz/cmd/dot/dot.c:132:6
    #10 0x7ffff65ff82f in __libc_start_main /build/glibc-LK5gWL/glibc-2.23/csu/../csu/libc-start.c:291
    #11 0x41aa58 in _start (/usr/local/bin/dot+0x41aa58)

0x6020000001f2 is located 0 bytes to the right of 2-byte region [0x6020000001f0,0x6020000001f2)
allocated by thread T0 here:
    #0 0x4dec88 in malloc (/usr/local/bin/dot+0x4dec88)
    #1 0x7ffff790efe4 in gmalloc /root/graphviz/lib/common/memory.c:47:10
    #2 0x7ffff790efe4 in zmalloc /root/graphviz/lib/common/memory.c:25

SUMMARY: AddressSanitizer: heap-buffer-overflow /root/graphviz/lib/common/shapes.c:3311:8 in parse_reclbl
Shadow bytes around the buggy address:
  0x0c047fff7fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff8000: fa fa 05 fa fa fa 00 03 fa fa 00 05 fa fa 00 fa
  0x0c047fff8010: fa fa 00 00 fa fa 00 00 fa fa fd fd fa fa fd fd
  0x0c047fff8020: fa fa 07 fa fa fa fd fd fa fa 00 01 fa fa 00 00
=>0x0c047fff8030: fa fa 00 fa fa fa 02 fa fa fa 02 fa fa fa[02]fa
  0x0c047fff8040: fa fa fd fa fa fa 00 fa fa fa fa fa fa fa fa fa
  0x0c047fff8050: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8060: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8070: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8080: 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
==14380==ABORTING
```

Drag your designs here or click to upload.

Tasks 0

No tasks are currently assigned. Use tasks to break down this issue into smaller parts.

Linked items 0

Link issues together to show that they're related or that one is blocking others. Learn more.

Related merge requests 1

fix: out-of-bounds write on invalid label
!1480

Activity

Stephen C. North @truenorth · 2 years ago — Owner

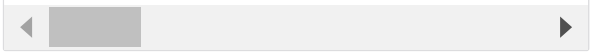
That seems like such low hanging fruit, I couldn't pass it up.

OK, How are you building with -fsanitize? After struggling for an hour, I can't get past the first library (cdt) as in

```
  CCLD     libcdt.la
Undefined symbols for architecture x86_64:
  "___asan_init", referenced from:
      _asan.module_ctor in dtclose.o
      _asan.module_ctor in dtdisc.o
      _asan.module_ctor in dtextract.o
      _asan.module_ctor in dtflatten.o
      _asan.module_ctor in dthash.o
      _asan.module_ctor in dtlist.o
      _asan.module_ctor in dtmethod.o
```

I configure with

```
CC=clang CXX=clang++ LIBTOOL=glibtool CXXFLAGS='-shared-libasan -fsanitize=address -fno-omit-frame-pointer -fno-c
```



Anyway, my hat's off for providing such a concise example and getting us the full stack trace. Maybe I can work with that.

Matthew Fernandez @smattr · 2 years ago — Owner

I can reproduce this on 970dd1a9 using the following steps:

```
$ CFLAGS="-g -fsanitize=address" CXXFLAGS="-g -fsanitize=address" ./configure && make clean && make && make insta
...
$ dot -Tsvg test.dot
Error: bad label format <
=================================================================
==6491==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x602000000212 at pc 0x7f42b94324fc bp 0x7ffcde3
WRITE of size 1 at 0x602000000212 thread T0
```

```
    #0 0x7f42b94324fb in parse_reclbl /home/matthew/graphviz/lib/common/shapes.c:3311
    #1 0x7f42b94347be in record_init /home/matthew/graphviz/lib/common/shapes.c:3556
    #2 0x7f42b94412e6 in common_init_node /home/matthew/graphviz/lib/common/utils.c:653
    #3 0x7f42b6266959 in dot_init_node /home/matthew/graphviz/lib/dotgen/dotinit.c:40
    #4 0x7f42b6267461 in dot_init_node_edge /home/matthew/graphviz/lib/dotgen/dotinit.c:81
    #5 0x7f42b6268ed0 in dotLayout /home/matthew/graphviz/lib/dotgen/dotinit.c:308
    #6 0x7f42b626a62d in doDot /home/matthew/graphviz/lib/dotgen/dotinit.c:463
    #7 0x7f42b626ac08 in dot_Layout /home/matthew/graphviz/lib/dotgen/dotinit.c:509
    #8 0x7f42b935d77c in gvLayoutJobs /home/matthew/graphviz/lib/gvc/gvlayout.c:85
    #9 0x5587e6f2f960 in main /home/matthew/graphviz/cmd/dot/dot.c:132
    #10 0x7f42b910009a in __libc_start_main ../csu/libc-start.c:308
    #11 0x5587e6f2f269 in _start (/tmp/tmp.j2w4UW3ksm/bin/dot+0x2269)

0x602000000212 is located 0 bytes to the right of 2-byte region [0x602000000210,0x602000000212)
allocated by thread T0 here:
    #0 0x7f42b9656330 in __interceptor_malloc (/lib/x86_64-linux-gnu/libasan.so.5+0xe9330)
    #1 0x7f42b93b0c02 in gmalloc /home/matthew/graphviz/lib/common/memory.c:47
    #2 0x7f42b93b0aeb in zmalloc /home/matthew/graphviz/lib/common/memory.c:25
    #3 0x7f42b94346b8 in record_init /home/matthew/graphviz/lib/common/shapes.c:3552
    #4 0x7f42b94412e6 in common_init_node /home/matthew/graphviz/lib/common/utils.c:653
    #5 0x7f42b6266959 in dot_init_node /home/matthew/graphviz/lib/dotgen/dotinit.c:40
    #6 0x7f42b6267461 in dot_init_node_edge /home/matthew/graphviz/lib/dotgen/dotinit.c:81
    #7 0x7f42b6268ed0 in dotLayout /home/matthew/graphviz/lib/dotgen/dotinit.c:308
    #8 0x7f42b626a62d in doDot /home/matthew/graphviz/lib/dotgen/dotinit.c:463
    #9 0x7f42b626ac08 in dot_Layout /home/matthew/graphviz/lib/dotgen/dotinit.c:509
    #10 0x7f42b935d77c in gvLayoutJobs /home/matthew/graphviz/lib/gvc/gvlayout.c:85
    #11 0x5587e6f2f960 in main /home/matthew/graphviz/cmd/dot/dot.c:132
    #12 0x7f42b910009a in __libc_start_main ../csu/libc-start.c:308

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/matthew/graphviz/lib/common/shapes.c:3311 in parse_reclbl
Shadow bytes around the buggy address:
  0x0c047fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff8000: fa fa 05 fa fa fa 00 03 fa fa 00 05 fa fa 06 fa
  0x0c047fff8010: fa fa 00 00 fa fa 00 00 fa fa 00 00 fa fa fd fd
  0x0c047fff8020: fa fa fd fd fa fa 07 fa fa fa fd fd fa fa 00 01
  0x0c047fff8030: fa fa 00 00 fa fa 00 fa fa fa 02 fa fa fa 02 fa
=>0x0c047fff8040: fa fa[02]fa fa fa fd fa fa fa 00 fa fa fa fa fa
  0x0c047fff8050: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8060: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8070: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8080: fa 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 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
==6491==ABORTING
```

```
$ gcc --version
gcc (Debian 8.3.0-6) 8.3.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
$ uname -rms
Linux 4.19.0-8-amd64 x86_64
$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 10 (buster)
Release:        10
Codename:       buster
```

@truenorth , let me know if you want me to debug this and come up with a fix. Otherwise I'll assume you're working on it. Thanks!

Matthew Fernandez mentioned in merge request !1480 (merged) 2 years ago

Matthew Fernandez @smattr · 2 years ago — Owner

This was fixed in 784411ca, but apparently my commit message format did not align with the criteria Gitlab uses to auto-close issues.

Matthew Fernandez closed 2 years ago

Matthew Fernandez mentioned in issue #1675 (closed) 2 years ago

Matthew Fernandez @smattr · 1 year ago — Owner

I just discovered somebody apparently requested a CVE for this: https://security-tracker.debian.org/tracker/CVE-2020-18032.