New issue

Jump to bottom

AddressSanitizer: heap-use-after-free in yasm_intnum_destroy() libyasm/intnum.c:415 #161

Open **Clingto** opened this issue on May 19, 2021 · 2 comments

Clingto commented on May 19, 2021

System info:
Ubuntu 16.04.6 LTS, X64, gcc 5.4.0, yasm (latest master `009450c`)
I think it is probably a similar issue as #149
Compile Command:

```
$ ./autogen.sh
make distclean

CC=gcc CXX=g++ CFLAGS="-fsanitize=address -fno-omit-frame-pointer -g" CXXFLAGS="-fsanitize=address -fno-omit-frame-pointer -g" ./configure --prefix=$PWD/build --disable-shared
make -j
make install
```

Run Command:

```
$ yasm  $POC
```

POC file:
https://github.com/Clingto/POC/blob/master/MSA/yasm/yasm-415-yasm_intnum_destroy-UAF

ASAN info:

```
yasm: file name already has no extension: output will be in `yasm.out'
=================================================================
==16102==ERROR: AddressSanitizer: heap-use-after-free on address 0x602000007098 at pc 0x7ffa04efacb7 bp 0x7fff5b056900 sp 0x7fff5b0568f0
READ of size 4 at 0x602000007098 thread T0
    #0 0x7ffa04efacb6 in yasm_intnum_destroy  test/yasm-uaf/SRC_asan/libyasm/intnum.c:415
    #1 0x7ffa04ee7e69 in expr_delete_term  test/yasm-uaf/SRC_asan/libyasm/expr.c:1017
    #2 0x7ffa04ee7e69 in expr_simplify_identity  test/yasm-uaf/SRC_asan/libyasm/expr.c:582
    #3 0x7ffa04ee8e3c in expr_level_op  test/yasm-uaf/SRC_asan/libyasm/expr.c:700
    #4 0x7ffa04eea5d1 in expr_level_tree  test/yasm-uaf/SRC_asan/libyasm/expr.c:880
    #5 0x7ffa04eea546 in expr_level_tree  test/yasm-uaf/SRC_asan/libyasm/expr.c:864
    #6 0x7ffa04eea546 in expr_level_tree  test/yasm-uaf/SRC_asan/libyasm/expr.c:864
    #7 0x7ffa04eeb686 in yasm_expr__level_tree  test/yasm-uaf/SRC_asan/libyasm/expr.c:906
    #8 0x7ffa04eeeb52 in yasm_expr_get_intnum  test/yasm-uaf/SRC_asan/libyasm/expr.c:1261
    #9 0x7ffa04ed9c03 in yasm_bc_create_data  test/yasm-uaf/SRC_asan/libyasm/bc-data.c:292
    #10 0x7ffa01795e6e in parse_exp  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parse.c:669
    #11 0x7ffa0179b89f in parse_exp  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parse.c:258
    #12 0x7ffa0179b89f in parse_line  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parse.c:289
    #13 0x7ffa0179b89f in nasm_parser_parse  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parse.c:231
    #14 0x7ffa0178f36b in nasm_do_parse  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parser.c:66
    #15 0x7ffa0178f36b in nasm_parser_do_parse  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parser.c:83
    #16 0x402c84 in do_assemble  test/yasm-uaf/SRC_asan/frontends/yasm/yasm.c:521
    #17 0x402c84 in main  test/yasm-uaf/SRC_asan/frontends/yasm/yasm.c:753
    #18 0x7ffa0490b82f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)
    #19 0x403ee8 in _start ( test/yasm-uaf/bin_asan/bin/yasm+0x403ee8)

0x602000007098 is located 8 bytes inside of 16-byte region [0x602000007090,0x6020000070a0)
freed by thread T0 here:
    #0 0x7ffa051c52ca in __interceptor_free (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x982ca)
    #1 0x7ffa04ee887c in expr_level_op  test/yasm-uaf/SRC_asan/libyasm/expr.c:689
    #2 0x7ffa04eea5d1 in expr_level_tree  test/yasm-uaf/SRC_asan/libyasm/expr.c:880
    #3 0x7ffa04eea546 in expr_level_tree  test/yasm-uaf/SRC_asan/libyasm/expr.c:864
    #4 0x7ffa04eea546 in expr_level_tree  test/yasm-uaf/SRC_asan/libyasm/expr.c:864
    #5 0x7ffa04eeb686 in yasm_expr__level_tree  test/yasm-uaf/SRC_asan/libyasm/expr.c:906
    #6 0x7ffa04eeeb52 in yasm_expr_get_intnum  test/yasm-uaf/SRC_asan/libyasm/expr.c:1261
    #7 0x7ffa04ed9c03 in yasm_bc_create_data  test/yasm-uaf/SRC_asan/libyasm/bc-data.c:292
    #8 0x7ffa01795e6e in parse_exp  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parse.c:669
    #9 0x7ffa0179b89f in parse_exp  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parse.c:258
    #10 0x7ffa0179b89f in parse_line  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parse.c:289
    #11 0x7ffa0179b89f in nasm_parser_parse  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parse.c:231
    #12 0x7ffa0178f36b in nasm_do_parse  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parser.c:66
    #13 0x7ffa0178f36b in nasm_parser_do_parse  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parser.c:83
    #14 0x402c84 in do_assemble  test/yasm-uaf/SRC_asan/frontends/yasm/yasm.c:521
    #15 0x402c84 in main  test/yasm-uaf/SRC_asan/frontends/yasm/yasm.c:753
    #16 0x7ffa0490b82f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)

previously allocated by thread T0 here:
    #0 0x7ffa051c5602 in malloc (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x98602)
    #1 0x7ffa04f16769 in def_xmalloc  test/yasm-uaf/SRC_asan/libyasm/xmalloc.c:69
    #2 0x7ffa04efab26 in yasm_intnum_copy  test/yasm-uaf/SRC_asan/libyasm/intnum.c:397
    #3 0x7ffa04ee33e4 in expr_item_copy  test/yasm-uaf/SRC_asan/libyasm/expr.c:975
    #4 0x7ffa04ee33e4 in yasm_expr__copy_except  test/yasm-uaf/SRC_asan/libyasm/expr.c:1006
    #5 0x7ffa04eebc13 in expr_expand_equ  test/yasm-uaf/SRC_asan/libyasm/expr.c:834
    #6 0x7ffa04eebc13 in expr_expand_equ  test/yasm-uaf/SRC_asan/libyasm/expr.c:843
    #7 0x7ffa04eebc13 in expr_expand_equ  test/yasm-uaf/SRC_asan/libyasm/expr.c:839
    #8 0x7ffa04eebc13 in yasm_expr__level_tree  test/yasm-uaf/SRC_asan/libyasm/expr.c:905
    #9 0x7ffa04eeeb52 in yasm_expr_get_intnum  test/yasm-uaf/SRC_asan/libyasm/expr.c:1261
    #10 0x7ffa04ed9c03 in yasm_bc_create_data  test/yasm-uaf/SRC_asan/libyasm/bc-data.c:292
    #11 0x7ffa01795e6e in parse_exp  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parse.c:669
    #12 0x7ffa0179b89f in parse_exp  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parse.c:258
    #13 0x7ffa0179b89f in parse_line  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parse.c:289
    #14 0x7ffa0179b89f in nasm_parser_parse  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parse.c:231
    #15 0x7ffa0178f36b in nasm_do_parse  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parser.c:66
    #16 0x7ffa0178f36b in nasm_parser_do_parse  test/yasm-uaf/SRC_asan/modules/parsers/nasm/nasm-parser.c:83
    #17 0x402c84 in do_assemble  test/yasm-uaf/SRC_asan/frontends/yasm/yasm.c:521
    #18 0x402c84 in main  test/yasm-uaf/SRC_asan/frontends/yasm/yasm.c:753
    #19 0x7ffa0490b82f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)
```

```
SUMMARY: AddressSanitizer: heap-use-after-free  test/yasm-uaf/SRC_asan/libyasm/intnum.c:415 yasm_intnum_destroy
Shadow bytes around the buggy address:
  0x0c047fff8dc0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8dd0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8de0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8df0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8e00: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c047fff8e10: fa fa fd[fd]fa fa 00 00 fa fa 00 00 fa fa fd fa
  0x0c047fff8e20: fa fa fd fa fa fa 07 fa fa fa fd fa fa fa fd fa
  0x0c047fff8e30: fa fa 07 fa fa fa 00 00 fa fa 00 00 fa fa 07 fa
  0x0c047fff8e40: fa fa fd fa fa fa fd fd fa fa fd fa fa fa fd fa
  0x0c047fff8e50: fa fa fd fa fa fa 03 fa fa fa 00 00 fa fa 00 00
  0x0c047fff8e60: fa fa fd fa fa fa fd fa fa fa fd fa fa fa fd fd
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
  Heap right redzone:      fb
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack partial redzone:   f4
  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
==16102==ABORTING
```

1

arizvisa commented on Nov 1, 2021

UGh. I know you're bugs w/ a fuzzer, and you're probably super proud of finding them and stuff....which is great and all that, and probably gives you massive amounts of content for your blog and other social media... It makes me wonder, where's your CVE? Where's the link for your advisory and who the "vendor" should credit?

But like seriously, as a coder, these types of "reports" are pretty much the absolute worst kind. You're literally dumping a backtrace, dropping your "poc", and not even highlighting the relevant code in the software's parser. You're not even suggesting a potential fix... Like seriously? These 20ish bugs that you're submitting are probably something that you can even fix yourself and submit a PR for, if you really (and truly) want to get them fixed.

You need to keep in mind that yasm developers are not a security vendor or composed of a massive team or anything with extraordinary amounts of funding for a support team to triage crappy bugs in order to convert them into actionable reports. It's literally a small group of coders that are maintaining an assembler for free...and for our use, which we should be pretty damned grateful for.

Some people have real bugs and real feature requests because they actually use this tool. When you flood a bug tracker with a bunch of bugs and provide zero effort at all to get them fixed...

2

natalie13m mentioned this issue on Nov 1, 2021

Stack overflow in parse_expr6(5,4,3,2,1) modules/parsers/nasm/nasm-parse.c #152

Open

Clingto commented on Jul 23 • edited

Author

> vendor

Thank you for your suggestion. We just do the research and mean to help make the software work more stable, and we're doing research to speed up proposing potential fixes.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants