Bug 3392707 - heap-use-after-free in saa_wbytes nasmlib/saa.c:132

Status: CLOSED FIXED

Alias: None

Product: NASM
Component: Assembler (show other bugs)
Version: 2.15.xx
Hardware: All All

Importance: Medium normal
Assignee: nobody

URL:

Depends on:
Blocks:

Reported: 2020-07-28 04:01 PDT by Suhwan
Modified: 2020-07-30 17:08 PDT (History)
CC List: 5 users (show)

Obtained from: Build from source archive using configure

Attachments	
poc (301 bytes, text/plain) 2020-07-28 04:01 PDT, Suhwan	Details
Add an attachment (proposed patch, testcase, etc.)	

Note
You need to log in before you can comment on or make changes to this bug.

Suhwan 2020-07-28 04:01:17 PDT Description

```
Created attachment 411796 [details]
poc

Hi,
I found a heap-use-after-free in saa_wbytes nasmlib/saa.c:132
It is triggered in nasm version 2.15rc10.

Please run following command
`nasm -f win64 -o tmp.o $PoC`


==32342==ERROR: AddressSanitizer: heap-use-after-free on address 0x6060000098f0 at
pc 0x7f61c67b5733 bp 0x7ffeb12d40b0 sp 0x7ffeb12d3858
READ of size 1 at 0x6060000098f0 thread T0
    #0 0x7f61c67b5732  (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x79732)
    #1 0x55ca40ce4396 in memcpy /usr/include/x86_64-linux-
gnu/bits/string_fortified.h:34
    #2 0x55ca40ce4396 in saa_wbytes nasmlib/saa.c:132
    #3 0x55ca40eede30 in coff_sect_write output/outcoff.c:687
    #4 0x55ca40eede30 in coff_out output/outcoff.c:615
    #5 0x55ca40d0dcf2 in out asm/assemble.c:434
    #6 0x55ca40d14d03 in out_rawdata asm/assemble.c:462
    #7 0x55ca40d14d03 in out_eops asm/assemble.c:663
    #8 0x55ca40d4e387 in assemble asm/assemble.c:700
    #9 0x55ca40ca7f6f in process_insn asm/nasm.c:1605
    #10 0x55ca40ca7f6f in assemble_file asm/nasm.c:1720
    #11 0x55ca40c9c056 in main asm/nasm.c:712
    #12 0x7f61c636cb96 in __libc_start_main (/lib/x86_64-linux-
gnu/libc.so.6+0x21b96)
    #13 0x55ca40c9f129 in _start
(/mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/nasm-
2.15rc10/install_dir/bin/nasm+0x124129)

0x6060000098f0 is located 48 bytes inside of 49-byte region
[0x6060000098c0,0x6060000098f1)
freed by thread T0 here:
    #0 0x7f61c681a7a8 in __interceptor_free (/usr/lib/x86_64-linux-
gnu/libasan.so.4+0xde7a8)
    #1 0x55ca40d90c69 in parse_eops asm/parser.c:464

previously allocated by thread T0 here:
    #0 0x7f61c681af30 in realloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xdef30)
    #1 0x55ca40ccca50 in nasm_realloc nasmlib/alloc.c:101

SUMMARY: AddressSanitizer: heap-use-after-free (/usr/lib/x86_64-linux-
gnu/libasan.so.4+0x79732)
Shadow bytes around the buggy address:
  0x0c0c7fff92c0: fa fa fa fa 00 00 00 00 00 00 00 fa fa fa fa fa
  0x0c0c7fff92d0: 00 00 00 00 00 00 00 fa fa fa fa fa 00 00 00 00
  0x0c0c7fff92e0: 00 00 00 fa fa fa fa fa 00 00 00 00 00 00 00 fa
  0x0c0c7fff92f0: fa fa fa fa 00 00 00 00 00 00 00 fa fa fa fa fa
  0x0c0c7fff9300: 00 00 00 00 00 00 00 fa fa fa fa fa 00 00 00 00
=>0x0c0c7fff9310: 00 00 00 fa fa fa fa fa fd fd fd fd fd fd[fd]fa
  0x0c0c7fff9320: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0c7fff9330: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0c7fff9340: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0c7fff9350: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0c7fff9360: 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
==32342==ABORTING
```

Suhwan 2020-07-28 23:16:47 PDT Comment 1

```
This is found by Agency for Defense Development (ADD) in South Korea.
```

H. Peter Anvin 2020-07-30 15:49:58 PDT Comment 2

```
The original problem fixed in checkin 6ac6ac57e3d01ea8ed4ea47706eb724b59176461, but
this PoC triggers another sanitizer violation when using -gcv8, so leave open for
now.
```

H. Peter Anvin 2020-07-30 17:08:29 PDT Comment 3

```
Secondary bug fixed in checkin 78df8828a0a5d8e2d8ff3dced562bf1778ce2e6c.
```

Format For Printing - XML - Clone This Bug - Top of page