Heap-based Buffer Overflow in vim/vim

✓ Valid Reported on Nov 18th 2021

0

Description

Greetings,
A Heap-based Buffer Overflow issue was discovered in Vim.
The POC file is reduced to the absolute minimum to reproduce the problem. Please see sanitizer output and the "trimmed" POC file link below.

System info OS version : Ubuntu 20.04.2 LTS + Clang 12 with ASan Vim Version : master(3cad470) - Thu Nov 18 15:37:29 2021 +0000

Steps to reproduce:

```
git clone https://github.com/vim/vim
```

```
LD=lld-12 AS=llvm-as-12 AR=llvm-ar-12 RANLIB=llvm-ranlib-12 CC=clang-12 CX
```

Download POC from This URL

```
./vim -u NONE -X -Z -e -s -S POC -c :qa!
```

Sanitizer output:

```
=================================================================
==135953==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x621000
READ of size 1 at 0x621000012918 thread T0
    #0 0x139d8cc in findmatchlimit /src/fuzzer11/triage_yeni/vim/src/search
    #1 0x60e8c5 in find_start_brace /src/fuzzer11/triage_yeni/vim/src/cinde
    #2 0x5c1385 in get_c_indent /src/fuzzer11/triage_yeni/vim/src/cindent.c
    #3 0xc0c7b8 in op_reindent /src/fuzzer11/triage_yeni/vim/src/indent.c:1
    #4 0xf389b7 in do_pending_operator /src/fuzzer11/triage_yeni/vim/src/op
    #5 0xe84da1 in normal_cmd /src/fuzzer11/triage_yeni/vim/src/normal.c:11
    #6 0x9aefb4 in exec_normal /src/fuzzer11/triage_yeni/vim/src/ex_docmd.c
    #7 0x9ad0aa in exec_normal_cmd /src/fuzzer11/triage_yeni/vim/src/ex_doc
    #8 0x9ad0aa in ex_normal /src/fuzzer11/triage_yeni/vim/src/ex_docmd.c:8
    #9 0x94ff7b in do_one_cmd /src/fuzzer11/triage_yeni/vim/src/ex_docmd.c:
    #10 0x94ff7b in do_cmdline /src/fuzzer11/triage_yeni/vim/src/ex_docmd.c
    #11 0x136cde4 in do_source /src/fuzzer11/triage_yeni/vim/src/scriptfile
    #12 0x13699e1 in cmd_source /src/fuzzer11/triage_yeni/vim/src/scriptfil
    #13 0x13699e1 in ex_source /src/fuzzer11/triage_yeni/vim/src/scriptfile
    #14 0x94ff7b in do_one_cmd /src/fuzzer11/triage_yeni/vim/src/ex_docmd.c
    #15 0x94ff7b in do_cmdline /src/fuzzer11/triage_yeni/vim/src/ex_docmd.c
    #16 0x1bcecfc in exe_commands /src/fuzzer11/triage_yeni/vim/src/main.c:
    #17 0x1bcecfc in vim_main2 /src/fuzzer11/triage_yeni/vim/src/main.c:773
    #18 0x1bc5a8f in main /src/fuzzer11/triage_yeni/vim/src/main.c:425:12
    #19 0x7f43168c50b2 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.
    #20 0x41f64d in _start (/src/fuzzer11/triage_yeni/vim/src/vim+0x41f64d)

0x621000012918 is located 24 bytes to the right of 4096-byte region [0x6210
allocated by thread T0 here:
    #0 0x49a8ad in malloc (/src/fuzzer11/triage_yeni/vim/src/vim+0x49a8ad)
    #1 0x4cc2cb in lalloc /src/fuzzer11/triage_yeni/vim/src/alloc.c:244:11

SUMMARY: AddressSanitizer: heap-buffer-overflow /src/fuzzer11/triage_yeni/v
Shadow bytes around the buggy address:
  0x0c427fffa4d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffa4e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffa4f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffa500: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffa510: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c427fffa520: fa fa fa[fa]fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa530: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa540: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa550: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa560: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa570: 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
  Shadow gap:              cc
==135953==ABORTING
```

s:

Heap-based Buffer Overflow - https://cwe.mitre.org/data/definitions/122.html

ability is capable of crashing software, bypass protection mechanism, modify of memory, and successful exploitation may lead to code execution

References

at Karagun

(Published)

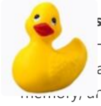
Vulnerability Type
CWE-122: Heap-based Buffer Overflow

Severity
High (7.5)

Visibility
Public

Status
Fixed

Found by



cem
@cemonatk
unranked

Fixed by

Bram Moolenaar
@brammool
maintainer

This report was seen 702 times.

We are processing your report and will contact the **vim** team within 24 hours. a year ago

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

Bram Moolenaar validated this vulnerability a year ago

cem has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

Bram Moolenaar a year ago

I wrote a test that reproduces the problem and a patch that fixes it: v8.2.3625.

Bram Moolenaar marked this as fixed in **8.2.3625** with commit **2de9b7** a year ago

Bram Moolenaar has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

Jamie Slome a year ago Admin

@cemonatk 👋 - same situation here! A bug caused the reward to erroneously be set to $355. I have reset the reward to the one shown at the point of disclosure ($0). Apologies for the confusion again.

Jamie Slome a year ago Admin

CVE published! 🎊

Sign in to join this conversation

privacy policy