Stack-based Buffer Overflow in function spell_dump_compl in vim/vim

0

Valid Reported on Jun 29th 2022

Description

Stack-based Buffer Overflow in function spell_dump_compl at spell.c:4038

vim version

```
git log
commit 75417d960bd17a5b701cfb625b8864dacaf0cc39 (HEAD -> master, tag: v9.0.
```

POC

```
./afl/src/vim -u NONE -i NONE -n -m -X -Z -e -s -S ./poc_sbo1_s.dat -c :qa!
=================================================================
==622487==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7fff
WRITE of size 4 at 0x7fffffff1e58 thread T0
    #0 0xf0f298 in spell_dump_compl /home/fuzz/fuzz/vim/afl/src/spell.c:403
    #1 0x9d23ff in ins_compl_dictionaries /home/fuzz/fuzz/vim/afl/src/insex
    #2 0x9cd232 in get_next_dict_tsr_completion /home/fuzz/fuzz/vim/afl/src
    #3 0x9cbec7 in get_next_completion_match /home/fuzz/fuzz/vim/afl/src/in
    #4 0x9c928e in ins_compl_get_exp /home/fuzz/fuzz/vim/afl/src/insexpand.
    #5 0x9c7c48 in find_next_completion_match /home/fuzz/fuzz/vim/afl/src/i
    #6 0x9c0714 in ins_compl_next /home/fuzz/fuzz/vim/afl/src/insexpand.c:4
    #7 0x9c118c in ins_complete /home/fuzz/fuzz/vim/afl/src/insexpand.c:494
    #8 0x674409 in edit /home/fuzz/fuzz/vim/afl/src/edit.c:1
    #9 0xb6a9cc in invoke_edit /home/fuzz/fuzz/vim/afl/src/
    #10 0xb4d9bd in nv_edit /home/fuzz/fuzz/vim/afl/src/normal.c:7005:2
```

```
    #11 0xb1fe8f in normal_cmd /home/fuzz/fuzz/vim/afl/src/normal.c:939:5
    #12 0x81539e in exec_normal /home/fuzz/fuzz/vim/afl/src/ex_docmd.c:881
    #13 0x814bc8 in exec_normal_cmd /home/fuzz/fuzz/vim/afl/src/ex_docmd.c:

    #14 0x814779 in ex_normal /home/fuzz/fuzz/vim/afl/src/ex_docmd.c:8693:6
    #15 0x7dd6f9 in do_one_cmd /home/fuzz/fuzz/vim/afl/src/ex_docmd.c:2570:
    #16 0x7ca5b5 in do_cmdline /home/fuzz/fuzz/vim/afl/src/ex_docmd.c:992:1
    #17 0x1159f0c in call_user_func /home/fuzz/fuzz/vim/afl/src/userfunc.c:
    #18 0x1155ffd in call_user_func_check /home/fuzz/fuzz/vim/afl/src/userf
    #19 0x11503a4 in call_func /home/fuzz/fuzz/vim/afl/src/userfunc.c:3613:
    #20 0x114d743 in get_func_tv /home/fuzz/fuzz/vim/afl/src/userfunc.c:183
    #21 0x1180e6a in ex_call /home/fuzz/fuzz/vim/afl/src/userfunc.c:5594:6
    #22 0x7dd6f9 in do_one_cmd /home/fuzz/fuzz/vim/afl/src/ex_docmd.c:2570:
    #23 0x7ca5b5 in do_cmdline /home/fuzz/fuzz/vim/afl/src/ex_docmd.c:992:1
    #24 0xe59ece in do_source_ext /home/fuzz/fuzz/vim/afl/src/scriptfile.c:
    #25 0xe56966 in do_source /home/fuzz/fuzz/vim/afl/src/scriptfile.c:1801
    #26 0xe562a3 in cmd_source /home/fuzz/fuzz/vim/afl/src/scriptfile.c:117
    #27 0xe559ae in ex_source /home/fuzz/fuzz/vim/afl/src/scriptfile.c:1200
    #28 0x7dd6f9 in do_one_cmd /home/fuzz/fuzz/vim/afl/src/ex_docmd.c:2570:
    #29 0x7ca5b5 in do_cmdline /home/fuzz/fuzz/vim/afl/src/ex_docmd.c:992:1
    #30 0x7cf231 in do_cmdline_cmd /home/fuzz/fuzz/vim/afl/src/ex_docmd.c:5
    #31 0x1424092 in exe_commands /home/fuzz/fuzz/vim/afl/src/main.c:3133:2
    #32 0x142022b in vim_main2 /home/fuzz/fuzz/vim/afl/src/main.c:780:2
    #33 0x141573d in main /home/fuzz/fuzz/vim/afl/src/main.c:432:12
    #34 0x7ffff7bee082 in __libc_start_main /build/glibc-SzIz7B/glibc-2.31/
    #35 0x41ea5d in _start (/home/fuzz/fuzz/vim/afl/src/vim+0x41ea5d)

Address 0x7fffffff1e58 is located in stack of thread T0 at offset 1048 in f
    #0 0xf0d48f in spell_dump_compl /home/fuzz/fuzz/vim/afl/src/spell.c:387

  This frame has 3 object(s):
    [32, 1048) 'arridx' (line 3875) <== Memory access at offset 1048 overfl
    [1184, 2200) 'curi' (line 3876)
    [2336, 2590) 'word' (line 3877)
HINT: this may be a false positive if your program uses some custom stack u
      (longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow /home/fuzz/fuzz/vim/afl/sr
Shadow bytes around the buggy address:
  0x10007fff6370: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff6380: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff6390: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
  0x10007fff63a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff63b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x10007fff63c0: 00 00 00 00 00 00 00 00 00 00 00[f2]f2 f2 f2 f2

  0x10007fff63d0: f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 00 00 00 00
  0x10007fff63e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff63f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff6400: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007fff6410: 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
==622487==ABORTING
```

poc_sbo1_s.dat

Impact

This vulnerability is capable of crashing software, modify memory, and possible remote execution.

CVE
CVE-2022-2304
(Published)

Vulnerability Type
CWE-121: Stack-based Buffer Overflow

Severity
High (7.8)

Registry
Other

Affected Version
*

Visibility
Public

Status
Fixed

Found by

TDHX ICS Security
@jieyongma
pro

Fixed by

Bram Moolenaar
@brammool
maintainer

This report was seen 919 times.

We are processing your report and will contact the **vim** team within 24 hours. 5 months ago

We have contacted a member of the **vim** team and are waiting to hear back 5 months ago

Bram Moolenaar 5 months ago Maintainer

I cannot reproduce the problem. It may already have been fixed by patch 9.0.

We have sent a follow up to the vim team. We will try again in 7 days. 5 months ago

TDHX 5 months ago Researcher

Verified with latest codebase (v9.0.0032). The issue still exsits.

SAN

```
./vim/src/vim -u NONE -i NONE -n -m -X -Z -e -s -S ./poc_sbo1_s.dat -c :qa!
=================================================================
==17799==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7fff5f460988 at p
WRITE of size 4 at 0x7fff5f460988 thread T0
    #0 0x56387552d841 in spell_dump_compl /home/fuzz/fuzz/vim/vim/src/spell.c:4038
    #1 0x5638752b2157 in ins_compl_dictionaries /home/fuzz/fuzz/vim/vim/src/insexpand.
    #2 0x5638752b98bc in get_next_dict_tsr_completion /home/fuzz/fuzz/vim/vim/src/inse
    #3 0x5638752bbc61 in get_next_completion_match /home/fuzz/fuzz/vim/vim/src/insexpa
    #4 0x5638752bc1ec in ins_compl_get_exp /home/fuzz/fuzz/vim/vim/src/insexpand.c:376
    #5 0x5638752bce6e in find_next_completion_match /home/fuzz/fuzz/vim/vim/src/insexp
    #6 0x5638752bd23b in ins_compl_next /home/fuzz/fuzz/vim/vim/src/insexpand.c:4103
    #7 0x5638752c0367 in ins_complete /home/fuzz/fuzz/vim/vim/src/insexpand.c:4954
    #8 0x56387511ae4c in edit /home/fuzz/fuzz/vim/vim/src/edit.c:1281
    #9 0x56387538e67e in invoke_edit /home/fuzz/fuzz/vim/vim/src/normal.c:7037
    #10 0x56387538e4c6 in nv_edit /home/fuzz/fuzz/vim/vim/src/normal.c:7007
    #11 0x563875366b4e in normal_cmd /home/fuzz/fuzz/vim/vim/src/normal.c:939
    #12 0x5638751ebe54 in exec_normal /home/fuzz/fuzz/vim/vim/src/ex_docmd.c:8814
    #13 0x5638751ebc13 in exec_normal_cmd /home/fuzz/fuzz/vim/vim/src/ex_docmd.c:8777
    #14 0x5638751eb4b7 in ex_normal /home/fuzz/fuzz/vim/vim/src/ex_docmd.c:8695
    #15 0x5638751c7e1e in do_one_cmd /home/fuzz/fuzz/vim/vim/src/ex_docmd.c:2570
    #16 0x5638751bf0c1 in do_cmdline /home/fuzz/fuzz/vim/vim/src/ex_docmd.c:992
    #17 0x56387565a779 in call_user_func /home/fuzz/fuzz/vim/vim/src/userfunc.c:2901
    #18 0x56387565b9c7 in call_user_func_check /home/fuzz/fuzz/vim/vim/src/userfunc.c:
    #19 0x56387565e27b in call_func /home/fuzz/fuzz/vim/vim/src/userfunc.c:3614
    #20 0x563875654c76 in get_func_tv /home/fuzz/fuzz/vim/vim/src/userfunc.c:1834
    #21 0x56387566a859 in ex_call /home/fuzz/fuzz/vim/vim/src/userfunc.c:5595
    #22 0x5638751c7e1e in do_one_cmd /home/fuzz/fuzz/vim/vim/src/ex_docmd.c:2570
    #23 0x5638751bf0c1 in do_cmdline /home/fuzz/fuzz/vim/vim/src/ex_docmd.c:992
    #24 0x5638754de7b2 in do_source_ext /home/fuzz/fuzz/vim/vim/src/scriptfile.c:1674
    #25 0x5638754df8e4 in do_source /home/fuzz/fuzz/vim/vim/src/scriptfile.c:1801
    #26 0x5638754dc473 in cmd_source /home/fuzz/fuzz/vim/vim/src/scriptfile.c:1174
    #27 0x5638754dc4d8 in ex_source /home/fuzz/fuzz/vim/vim/src/scriptfile.c:1200
    #28 0x5638751c7e1e in do_one_cmd /home/fuzz/fuzz/vim/vim/src/ex_d
    #29 0x5638751bf0c1 in do_cmdline /home/fuzz/fuzz/vim/vim/src/ex_
    #30 0x5638751bd45b in do_cmdline_cmd /home/fuzz/fuzz/vim/vim/src/ex_docmd.c:586
    #31 0x5638757b15f2 in exe_commands /home/fuzz/fuzz/vim/vim/src/main.c:3133
```

```
    #32 0x5638757aa760 in vim_main2 /home/fuzz/fuzz/vim/vim/src/main.c:780
    #33 0x5638757aa018 in main /home/fuzz/fuzz/vim/vim/src/main.c:432
    #34 0x7fdac81c3082 in __libc_start_main ../csu/libc-start.c:308

    #35 0x563875046e2d in _start (/home/fuzz/fuzz/vim/vim/src/vim+0x139e2d)

Address 0x7fff5f460988 is located in stack of thread T0 at offset 1048 in frame
    #0 0x56387552c998 in spell_dump_compl /home/fuzz/fuzz/vim/vim/src/spell.c:3872

  This frame has 3 object(s):
    [32, 1048) 'arridx' (line 3875) <== Memory access at offset 1048 overflows this va
    [1184, 2200) 'curi' (line 3876)
    [2336, 2590) 'word' (line 3877)
HINT: this may be a false positive if your program uses some custom stack unwind mecha
      (longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow /home/fuzz/fuzz/vim/vim/src/spell.c:4
Shadow bytes around the buggy address:
  0x10006be840e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006be840f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006be84100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006be84110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006be84120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x10006be84130: 00[f2]f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2 f2
  0x10006be84140: f2 f2 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006be84150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006be84160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006be84170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10006be84180: 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
==17799==ABORTING
```

Valgrind

valgrind DOES NOT report this issue:

```
valgrind ./valgrind/src/vim -u NONE -i NONE -n -m -X -Z -e -s -S ./poc_sbo1_s.dat -c :
==17804== Memcheck, a memory error detector
==17804== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==17804== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==17804== Command: ./valgrind/src/vim -u NONE -i NONE -n -m -X -Z -e -s -S ./poc_sbo1_
==17804==
==17804==
==17804== HEAP SUMMARY:
==17804==     in use at exit: 77,614 bytes in 406 blocks
==17804==   total heap usage: 1,889 allocs, 1,483 frees, 1,165,812 bytes allocated
==17804==
==17804== LEAK SUMMARY:
==17804==    definitely lost: 0 bytes in 0 blocks
==17804==    indirectly lost: 0 bytes in 0 blocks
==17804==      possibly lost: 393 bytes in 8 blocks
==17804==    still reachable: 77,221 bytes in 398 blocks
==17804==         suppressed: 0 bytes in 0 blocks
==17804== Rerun with --leak-check=full to see details of leaked memory
==17804==
==17804== For lists of detected and suppressed errors, rerun with: -s
==17804== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Bram Moolenaar 5 months ago Maintainer

I see, this doesn't fail with valgrind because the invalid access is in the stack. I can reproduce it with ASAN.

Bram Moolenaar validated this vulnerability 5 months ago

TDHX ICS Security has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

Bram Moolenaar 5 months ago Maintainer

Fixed with patch 9.0.0035

Bram Moolenaar marked this as fixed in 9.0 with commit 54e5fe 5 months ago

Bram Moolenaar has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Sign in to join this conversation