huntr

Out-of-bound write in function ml_append_int in vim/vim

0

✓ Valid Reported on Jun 25th 2022

Description

Out-of-bound write in function `ml_append_int` at memline.c:2895

Version

```
commit 8eba2bd291b347e3008aa9e565652d51ad638cfa (HEAD, tag: v8.2.5151)
```

Proof of Concept

```
guest@elk:~/trung$ valgrind ./vim2/src/vim -u NONE -i NONE -n -m -X -Z -e -
==28900== Memcheck, a memory error detector
==28900== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==28900== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright inf
==28900== Command: ./vim2/src/vim -u NONE -i NONE -n -m -X -Z -e -s -S /hom
==28900==
  debug=  define=^\s*#\s*define  dictionary=  diffexpr=  diffopt=internal,f
==28900== Invalid read of size 1
==28900==    at 0x4C38796: memmove (in /usr/lib/valgrind/vgpreload_memcheck
==28900==    by 0x21594C: memmove (string_fortified.h:40)
==28900==    by 0x21594C: ml_append_int (memline.c:2895)
==28900==    by 0x218AED: ml_flush_line (memline.c:4054)
==28900==    by 0x21949C: ml_append_flush.part.11 (memline.c:3315)
==28900==    by 0x313727: u_undoredo (undo.c:2820)
==28900==    by 0x31403F: u_doit.part.9 (undo.c:2272)
==28900==    by 0x230E8F: nv_kundo (normal.c:4756)
==28900==    by 0x2385B4: normal_cmd (normal.c:939)
==28900==    by 0x1B671C: exec_normal (ex_docmd.c:8807)
==28900==    by 0x1B697F: ex_normal (ex_docmd.c:8693)
==28900==    by 0x1BB29D: do_one_cmd (ex_docmd.c:2570)
```

```
==28900==    by 0x1BB29D: do_cmdline (ex_docmd.c:992)
==28900==    by 0x2ABF00: do_source_ext (scriptfile.c:1674)
==28900==  Address 0x6ab8270 is 0 bytes after a block of size 4,096 alloc'

==28900==    at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-
==28900==    by 0x140C60: lalloc (alloc.c:246)
==28900==    by 0x38122A: mf_alloc_bhdr.isra.3 (memfile.c:884)
==28900==    by 0x382006: mf_new (memfile.c:375)
==28900==    by 0x2147DF: ml_new_data (memline.c:4080)
==28900==    by 0x21769C: ml_open (memline.c:394)
==28900==    by 0x150EA4: open_buffer (buffer.c:186)
==28900==    by 0x1A8096: do_ecmd (ex_cmds.c:3029)
==28900==    by 0x1BC226: do_exedit (ex_docmd.c:7158)
==28900==    by 0x1BC835: ex_splitview (ex_docmd.c:6774)
==28900==    by 0x1BB29D: do_one_cmd (ex_docmd.c:2570)
==28900==    by 0x1BB29D: do_cmdline (ex_docmd.c:992)
==28900==    by 0x2ABF00: do_source_ext (scriptfile.c:1674)
==28900==
==28900== Invalid write of size 1
==28900==    at 0x4C3878B: memmove (in /usr/lib/valgrind/vgpreload_memcheck
==28900==    by 0x21594C: memmove (string_fortified.h:40)
==28900==    by 0x21594C: ml_append_int (memline.c:2895)
==28900==    by 0x218AED: ml_flush_line (memline.c:4054)
==28900==    by 0x21949C: ml_append_flush.part.11 (memline.c:3315)
==28900==    by 0x313727: u_undoredo (undo.c:2820)
==28900==    by 0x31403F: u_doit.part.9 (undo.c:2272)
==28900==    by 0x230E8F: nv_kundo (normal.c:4756)
==28900==    by 0x2385B4: normal_cmd (normal.c:939)
==28900==    by 0x1B671C: exec_normal (ex_docmd.c:8807)
==28900==    by 0x1B697F: ex_normal (ex_docmd.c:8693)
==28900==    by 0x1BB29D: do_one_cmd (ex_docmd.c:2570)
==28900==    by 0x1BB29D: do_cmdline (ex_docmd.c:992)
==28900==    by 0x2ABF00: do_source_ext (scriptfile.c:1674)
==28900==  Address 0x6ab8270 is 0 bytes after a block of size 4,096 alloc'
==28900==    at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-
==28900==    by 0x140C60: lalloc (alloc.c:246)
==28900==    by 0x38122A: mf_alloc_bhdr.isra.3 (memfile.c:884)
==28900==    by 0x382006: mf_new (memfile.c:375)
==28900==    by 0x2147DF: ml_new_data (memline.c:4080)
==28900==    by 0x21769C: ml_open (memline.c:394)
==28900==    by 0x150EA4: open_buffer (buffer.c:186)
==28900==    by 0x1A8096: do_ecmd (ex_cmds.c:3029)
```

```
==28900==    by 0x1A8096: do_ecmd (ex_cmds.c:3029)
==28900==    by 0x1BC226: do_exedit (ex_docmd.c:7158)
==28900==    by 0x1BC835: ex_splitview (ex_docmd.c:6774)

==28900==    by 0x1BB29D: do_one_cmd (ex_docmd.c:2570)
==28900==    by 0x1BB29D: do_cmdline (ex_docmd.c:992)
==28900==    by 0x2ABF00: do_source_ext (scriptfile.c:1674)
==28900==
==28900== Invalid read of size 1
==28900==    at 0x4C38788: memmove (in /usr/lib/valgrind/vgpreload_memcheck
==28900==    by 0x21594C: memmove (string_fortified.h:40)
==28900==    by 0x21594C: ml_append_int (memline.c:2895)
==28900==    by 0x218AED: ml_flush_line (memline.c:4054)
==28900==    by 0x21949C: ml_append_flush.part.11 (memline.c:3315)
==28900==    by 0x313727: u_undoredo (undo.c:2820)
==28900==    by 0x31403F: u_doit.part.9 (undo.c:2272)
==28900==    by 0x230E8F: nv_kundo (normal.c:4756)
==28900==    by 0x2385B4: normal_cmd (normal.c:939)
==28900==    by 0x1B671C: exec_normal (ex_docmd.c:8807)
==28900==    by 0x1B697F: ex_normal (ex_docmd.c:8693)
==28900==    by 0x1BB29D: do_one_cmd (ex_docmd.c:2570)
==28900==    by 0x1BB29D: do_cmdline (ex_docmd.c:992)
==28900==    by 0x2ABF00: do_source_ext (scriptfile.c:1674)
==28900==  Address 0x6ab8273 is 3 bytes after a block of size 4,096 alloc'
==28900==    at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-
==28900==    by 0x140C60: lalloc (alloc.c:246)
==28900==    by 0x38122A: mf_alloc_bhdr.isra.3 (memfile.c:884)
==28900==    by 0x382006: mf_new (memfile.c:375)
==28900==    by 0x2147DF: ml_new_data (memline.c:4080)
==28900==    by 0x21769C: ml_open (memline.c:394)
==28900==    by 0x150EA4: open_buffer (buffer.c:186)
==28900==    by 0x1A8096: do_ecmd (ex_cmds.c:3029)
==28900==    by 0x1BC226: do_exedit (ex_docmd.c:7158)
==28900==    by 0x1BC835: ex_splitview (ex_docmd.c:6774)
==28900==    by 0x1BB29D: do_one_cmd (ex_docmd.c:2570)
==28900==    by 0x1BB29D: do_cmdline (ex_docmd.c:992)
==28900==    by 0x2ABF00: do_source_ext (scriptfile.c:1674)
==28900==
==28900==
==28900== Process terminating with default action of signal
==28900==    at 0x5851177: kill (syscall-template.S:78)
```

```
==28900==    by 0x254A47: may_core_dump (os_unix.c:3448)
==28900==    by 0x254A47: mch_exit (os_unix.c:3484)
==28900==    by 0x37FD2A: getout (main.c:1737)

==28900==    by 0x5850F0F: ??? (in /lib/x86_64-linux-gnu/libc-2.27.so)
==28900==    by 0x4C38795: memmove (in /usr/lib/valgrind/vgpreload_memcheck
==28900==    by 0x21594C: memmove (string_fortified.h:40)
==28900==    by 0x21594C: ml_append_int (memline.c:2895)
==28900==    by 0x218AED: ml_flush_line (memline.c:4054)
==28900==    by 0x21949C: ml_append_flush.part.11 (memline.c:3315)
==28900==    by 0x313727: u_undoredo (undo.c:2820)
==28900==    by 0x31403F: u_doit.part.9 (undo.c:2272)
==28900==    by 0x230E8F: nv_kundo (normal.c:4756)
==28900==    by 0x2385B4: normal_cmd (normal.c:939)
==28900==
==28900== HEAP SUMMARY:
==28900==     in use at exit: 129,159 bytes in 552 blocks
==28900==   total heap usage: 2,459 allocs, 1,907 frees, 4,494,485 bytes al
==28900==
==28900== LEAK SUMMARY:
==28900==    definitely lost: 9,947 bytes in 227 blocks
==28900==    indirectly lost: 0 bytes in 0 blocks
==28900==      possibly lost: 0 bytes in 0 blocks
==28900==    still reachable: 119,212 bytes in 325 blocks
==28900==         suppressed: 0 bytes in 0 blocks
==28900== Rerun with --leak-check=full to see details of leaked memory
==28900==
==28900== For counts of detected and suppressed errors, rerun with: -v
==28900== ERROR SUMMARY: 7643972 errors from 3 contexts (suppressed: 0 from
Segmentation fault
```

Attachment

poc35min

Impact

Typically, this can result in corruption of data, a crash, or code execution.

CVE

CVE-2022-2210
(Published)

Vulnerability Type

CWE-787: Out-of-bounds Write

Severity

High (7.8)

Registry

Other

Affected Version

8.2.5151

Visibility

Public

Status

Fixed

Found by

xikhud

@acquykhud

legend

Fixed by

Bram Moolenaar

@brammool

maintainer

This report was seen 917 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 can reproduce it. Took quite a while to find the root cause.

Bram Moolenaar validated this vulnerability 5 months ago

xikhud 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 8.2.5164

Bram Moolenaar marked this as fixed in 8.2 with commit c101ab 5 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

part of 418sec

company

about

team

Chat with us