

Heap-based Buffer Overflow in vim/vim

 Valid Reported on Nov 23rd 2021

0

🔥 Description

When fuzzing vim commit [3c19b5050](#) (works with latest build and latest commit [65259b5c6](#) per this time of this report) [v8.2.3635](#) with clang 12 and ASan, I discovered a heap buffer overflow.

Proof of Concept

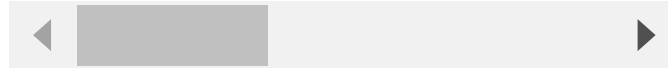
Here is the poc download link

https://drive.google.com/file/d/16KrhaQ_Feps6uhQGwuc_l8KETMz0eTyz/view?usp=sharing



How to build

```
LD=lld AS=llvm-as AR=llvm-ar RANLIB=llvm-ranlib CC=clang CXX=clang++ CFLAGS=
make -j2
```



Proof of Concept

Run crafted file with this command

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

ASan stack trace:

```
aldo@vps:~/vim-8.2.3635/src$ ASAN_OPTIONS=symbolize=1 ASAN_SYMBOLIZER_PATH=
=====
==2485525==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x61900
WRITE of size 1023 at 0x619000000e81 thread T0
#0 0x486717 in strcpy (/home/aldo/vim/src/vim+0x486717)
#1 0xc80184 in find_help_tags /home/aldo/vim/src/help.c:426:6
#2 0xc7b823 in ex_help /home/aldo/vim/src/help.c:88:9
#3 0xa06f9d in do_one_cmd /home/aldo/vim/src/ex_docmd.c:2614:2
#4 0x9eabea in do_cmdline /home/aldo/vim/src/ex_docmd.c:1000:17
#5 0x1592d2d in do_source /home/aldo/vim/src/scriptfile.c:1406:5
#6 0x158dd61 in cmd_source /home/aldo/vim/src/scriptfile.c:971:14
#7 0x158d66e in ex_source /home/aldo/vim/src/scriptfile.c:997:2
#8 0xa06f9d in do_one_cmd /home/aldo/vim/src/ex_docmd.c:2614:2
#9 0x9eabea in do_cmdline /home/aldo/vim/src/ex_docmd.c:1000:17
#10 0x9f1fd1 in do_cmdline_cmd /home/aldo/vim/src/ex_docmd.c:594:12
#11 0x1f239a1 in exe_commands /home/aldo/vim/src/main.c:3081:2
#12 0x1f1cd23 in vim_main2 /home/aldo/vim/src/main.c:773:2
#13 0x1f081ed in main /home/aldo/vim/src/main.c:425:12
#14 0x7ffff78260b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/
#15 0x41fe2d in _start (/home/aldo/vim/src/vim+0x41fe2d)

0x619000000e81 is located 0 bytes to the right of 1025-byte region [0x61900
allocated by thread T0 here:
#0 0x49aced in malloc (/home/aldo/vim/src/vim+0x49aced)
#1 0x4cc661 in lalloc /home/aldo/vim/src/alloc.c:244:11
#2 0x4cc4ba in alloc /home/aldo/vim/src/alloc.c:151:12
#3 0x1f08292 in common_init /home/aldo/vim/src/main.c:910:19
#4 0x1f068bb in main /home/aldo/vim/src/main.c:179:5
#5 0x7ffff78260b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/c
```

```
SUMMARY: AddressSanitizer: heap-buffer-overflow (/home/aldo/vim/src/vim+0x4
Shadow bytes around the buggy address:
```

```
0x0c327fff8180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c327fff8190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c327fff81a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c327fff81b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c327fff81c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c327fff81d0:[01]fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c327fff81e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c327fff81f0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x0c327fff8200: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x0c327fff8210: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x0c327fff8220: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
```

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
```

