NULL Pointer Dereference in function generate_loadvar in vim/vim

0

✓ Valid Reported on Aug 16th 2022

Description

NULL Pointer Dereference in function generate_loadvar at vim9compile.c:1165 allows attackers to cause a denial of service (application crash) via a crafted input.

vim version

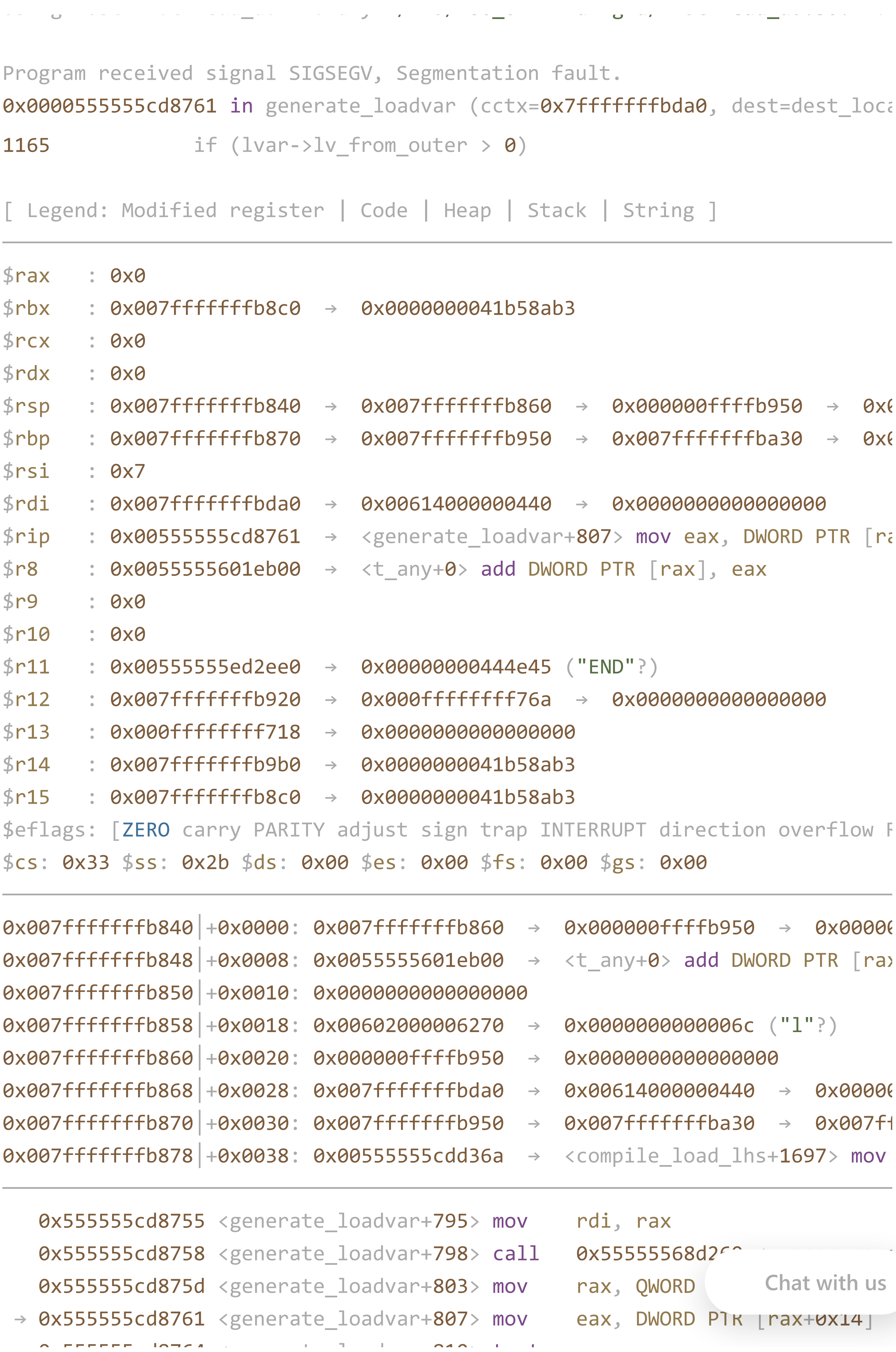
```
git log
commit e1f3fd1d02e3f5fe6d2b6d82687c6846b8e500f8 (HEAD -> master, origin/mas
Author: Bram Moolenaar <Bram@vim.org>
Date:   Mon Aug 15 18:51:32 2022 +0100

    Update runtime files
```

Proof of Concept

```
vim -u NONE -i NONE -n -m -X -Z -e -s -S /home/fuzz/test/poc1_null.dat -c
Segmentation fault (core dumped)
```

gdb debug info

```
[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.
0x0000555555cd8761 in generate_loadvar (cctx=0x7fffffffbda0, dest=dest_loca
1165            if (lvar->lv_from_outer > 0)

[ Legend: Modified register | Code | Heap | Stack | String ]
────────────────────────────────────────────────────────────────────────────
$rax   : 0x0
$rbx   : 0x007fffffffb8c0  →  0x0000000041b58ab3
$rcx   : 0x0
$rdx   : 0x0
$rsp   : 0x007fffffffb840  →  0x007fffffffb860  →  0x000000ffffb950  →  0x0
$rbp   : 0x007fffffffb870  →  0x007fffffffb950  →  0x007fffffffba30  →  0x0
$rsi   : 0x7
$rdi   : 0x007fffffffbda0  →  0x00614000000440  →  0x0000000000000000
$rip   : 0x00555555cd8761  →  <generate_loadvar+807> mov eax, DWORD PTR [ra
$r8    : 0x0055555601eb00  →  <t_any+0> add DWORD PTR [rax], eax
$r9    : 0x0
$r10   : 0x0
$r11   : 0x00555555ed2ee0  →  0x00000000444e45 ("END"?)
$r12   : 0x007fffffffb920  →  0x000fffffff76a  →  0x0000000000000000
$r13   : 0x000fffffff718  →  0x0000000000000000
$r14   : 0x007fffffffb9b0  →  0x0000000041b58ab3
$r15   : 0x007fffffffb8c0  →  0x0000000041b58ab3
$eflags: [ZERO carry PARITY adjust sign trap INTERRUPT direction overflow R
$cs: 0x33 $ss: 0x2b $ds: 0x00 $es: 0x00 $fs: 0x00 $gs: 0x00
────────────────────────────────────────────────────────────────────────────
0x007fffffffb840│+0x0000: 0x007fffffffb860  →  0x000000ffffb950  →  0x0000
0x007fffffffb848│+0x0008: 0x0055555601eb00  →  <t_any+0> add DWORD PTR [ra
0x007fffffffb850│+0x0010: 0x0000000000000000
0x007fffffffb858│+0x0018: 0x00602000006270  →  0x0000000000006c ("l"?)
0x007fffffffb860│+0x0020: 0x000000ffffb950  →  0x0000000000000000
0x007fffffffb868│+0x0028: 0x007fffffffbda0  →  0x00614000000440  →  0x0000
0x007fffffffb870│+0x0030: 0x007fffffffb950  →  0x007fffffffba30  →  0x007ff
0x007fffffffb878│+0x0038: 0x00555555cdd36a  →  <compile_load_lhs+1697> mov
────────────────────────────────────────────────────────────────────────────
   0x555555cd8755 <generate_loadvar+795> mov    rdi, rax
   0x555555cd8758 <generate_loadvar+798> call   0x55555568d2
   0x555555cd875d <generate_loadvar+803> mov    rax, QWORD
 → 0x555555cd8761 <generate_loadvar+807> mov    eax, DWORD PTR [rax+0x14]
```

```
   0x555555cd8764 <generate_loadvar+810> test   eax, eax
   0x555555cd8766 <generate_loadvar+812> jle    0x555555cd87bc <generate_lo
   0x555555cd8768 <generate_loadvar+814> mov    rax, QWORD PTR [rbp-0x20]

   0x555555cd876c <generate_loadvar+818> mov    edx, DWORD PTR [rax+0x14]
   0x555555cd876f <generate_loadvar+821> mov    rax, QWORD PTR [rbp-0x20]
──────────────────────────────────────────────────────────────────────────
   1160             break;
   1161         case dest_vimvar:
   1162             generate_LOADV(cctx, name + 2);
   1163             break;
   1164         case dest_local:
                  // lvar=0x007fffffffb850  →  0x0000000000000000
 → 1165             if (lvar->lv_from_outer > 0)
   1166             generate_LOADOUTER(cctx, lvar->lv_idx, lvar->lv_from_outer,
   1167                                                  type);
   1168             else
   1169             generate_LOAD(cctx, ISN_LOAD, lvar->lv_idx, NULL, type);
   1170             break;
──────────────────────────────────────────────────────────────────────────
[#0] Id 1, Name: "vim", stopped 0x555555cd8761 in generate_loadvar (), reas
──────────────────────────────────────────────────────────────────────────
[#0] 0x555555cd8761 → generate_loadvar(cctx=0x7fffffffbda0, dest=dest_local
[#1] 0x555555cdd36a → compile_load_lhs(lhs=0x7fffffffbe40, var_start=0x6020
[#2] 0x555555cddb2d → compile_assign_unlet(var_start=0x602000006290 "l[0]",
[#3] 0x555555cd1bde → compile_redir(line=0x6020000062b0 "redi END", eap=0x7
[#4] 0x555555ce58ee → compile_def_function(ufunc=0x614000000440, check_retu
[#5] 0x555555cbc9b3 → ex_defcompile(eap=0x7fffffffc270)
[#6] 0x555555817444 → do_one_cmd(cmdlinep=0x7fffffffc5d0, flags=0x7, cstack
[#7] 0x55555580e6e7 → do_cmdline(cmdline=0x6110000002c0 "def T()", fgetline
[#8] 0x555555b3186d → do_source_ext(fname=0x604000000213 "/home/fuzz/test/p
[#9] 0x555555b3299f → do_source(fname=0x604000000213 "/home/fuzz/test/poc1_
──────────────────────────────────────────────────────────────────────────
```

poc download: <p><a href="https://github.com/Janette88/vim/blob/main/poc1_null.dat">poc1_null.dat</a></p>

Impact

Chat with us

NULL Pointer Dereference in function generate_loadvar allows attackers to cause a denial of service (application crash) via a crafted input.

CVE
CVE-2022-2874
(Published)

Vulnerability Type
CWE-476: NULL Pointer Dereference

Severity
Medium (6.6)

Registry
Other

Affected Version
*

Visibility
Public

Status
Fixed

Found by

janette88
@janette88
master

Fixed by

Bram Moolenaar
@brammool
maintainer

This report was seen 618 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

I can reproduce it. The POC can be simplified by removing a couple of lines that don't matter.

janette88 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 9.0.0224

Bram Moolenaar marked this as fixed in **9.0.0223** with commit **4875d6** 3 months ago

Bram Moolenaar has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

Sign in to join this conversation

FAQ

contact us

terms

privacy policy

Chat with us