NULL Pointer Dereference in function vim_regexec_string in vim/vim

0

Valid Reported on May 13th 2022

Description

NULL Pointer Dereference in function vim_regexec_string at regexp.c:2733 allows attackers to cause a denial of service (application crash) via a crafted input.

vim version

```
git log
commit 31ad32a325cc31f0f2bdd530c68bfb856a2187c5 (HEAD -> master, tag: v8.2.
```

History

Very similar to what was fixed in 8.2.4901 and 8.2.4938, but another different code path. How about check for NULL regprog right at regexp.c:2733 in function vim_regexec_string?

POC

```
./vim -u NONE -i NONE -n -m -X -Z -e -s -S /mnt/share/max/fuzz/poc/vim/poc_
Segmentation fault
```

poc_n3_s.dat

GDB

```
── Output/messages ──────────────────────────────────────────────────────
[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.
0x0000000000d24b92 in vim_regexec_string (rmp=0x7fffffff8980, line=0x602006
2733        if (rmp->regprog->re_in_use)
── Assembly ─────────────────────────────────────────────────────────────
 0x0000000000d24b77  vim_regexec_string+567 cmp    %cl,%al
 0x0000000000d24b79  vim_regexec_string+569 jl     0xd24b8b <vim_regexec_st
 0x0000000000d24b7f  vim_regexec_string+575 mov    0x118(%rbx),%rdi
 0x0000000000d24b86  vim_regexec_string+582 callq  0x4a1350 <__asan_report_
 0x0000000000d24b8b  vim_regexec_string+587 mov    0x118(%rbx),%rax
 0x0000000000d24b92  vim_regexec_string+594 cmpl   $0x0,(%rax)
 0x0000000000d24b95  vim_regexec_string+597 je     0xd24bfc <vim_regexec_st
 0x0000000000d24b9b  vim_regexec_string+603 mov    0x176770c,%ecx
 0x0000000000d24ba2  vim_regexec_string+610 mov    $0x172e1e0,%rax
 0x0000000000d24ba9  vim_regexec_string+617 mov    (%rax),%rax
── Breakpoints ──────────────────────────────────────────────────────────
── Expressions ──────────────────────────────────────────────────────────
── History ──────────────────────────────────────────────────────────────
── Memory ───────────────────────────────────────────────────────────────
── Registers ────────────────────────────────────────────────────────────
    rax 0x0000000000000014     rbx 0x00007fffffff8660     rcx 0x0000000000
    rbp 0x00007fffffff8860     rsp 0x00007fffffff8560      r8 0x0000000020
    r12 0x000000000041fe30     r13 0x00007fffffffe3f0     r14 0x0000000000
     cs 0x00000033              ss 0x0000002b              ds 0x00000000
── Source ───────────────────────────────────────────────────────────────
 2728      int         result;
 2729      regexec_T    rex_save;
 2730      int         rex_in_use_save = rex_in_use;
 2731
 2732      // Cannot use the same prog recursively, it contains state.
 2733      if (rmp->regprog->re_in_use)
 2734      {
 2735      emsg(_(e_cannot_use_pattern_recursively));
 2736      return FALSE;
 2737      }
── Stack ────────────────────────────────────────────────────────────────
[0] from 0x0000000000d24b92 in vim_regexec_string+594 at regexp.c:2733
```

```
[1] from 0x000000000042564a in vim_regexec+90 at regexp.c:2816
[2] from 0x000000000053f286 in fname_match+454 at buffer.c:2958
[3] from 0x000000000051af2b in buflist_match+139 at buffer.c:2934

[4] from 0x0000000000515845 in buflist_findpat+4053 at buffer.c:2656
[5] from 0x00000000007f821e in do_one_cmd+50910 at ex_docmd.c:2532
[6] from 0x00000000007e5826 in do_cmdline+14134 at ex_docmd.c:992
[7] from 0x0000000000e8c39d in do_source_ext+13725 at scriptfile.c:1674
[8] from 0x0000000000e88df7 in do_source+103 at scriptfile.c:1801
[9] from 0x0000000000e8872d in cmd_source+2317 at scriptfile.c:1174
[+]
── Threads ──────────────────────────────────────────────────────────────
[1] id 3841802 name vim from 0x0000000000d24b92 in vim_regexec_string+594 a
── Variables ────────────────────────────────────────────────────────────
arg rmp = 0x7fffffff8980: {regprog = 0x0,startp = {[0] = 0x7fffffff8d08 "\0
loc result = -1, rex_save = {reg_match = 0x7fffffff7840,reg_mmatch = 0x7fff
──────────────────────────────────────────────────────────────────────────

>>> p rmp->regprog
$1 = (regprog_T *) 0x0
>>>
```

Impact

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

References

- 8.2.4901
- 8.2.4938

CVE
CVE-2022-1725
(Published)

Vulnerability Type
CWE-476: NULL Pointer Dereference

Severity
Medium (6.6)

Registry
Other

Affected Version
*

Visibility
Public

Status
Fixed

Found by

TDHX ICS Security
@jieyongma
pro

Fixed by

Bram Moolenaar
@brammool
maintainer

This report was seen 990 times.

We are processing your report and will contact the **vim** team within 24 hours. 6 months ago

We have contacted a member of the **vim** team and are waiting to hear back 6 months ago

Bram Moolenaar 6 months ago Maintainer

I can reproduce it. I'll add some extra checks for regprog becoming NULL.

Bram Moolenaar validated this vulnerability 6 months ago

TDHX ICS Security has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

Bram Moolenaar marked this as fixed in 8.2 with commit b62dc5 6 months ago

Bram Moolenaar has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Bram Moolenaar 6 months ago Maintainer

Fixed in patch 8.2.4959

Sign in to join this conversation