Infinite recursive function calls result in stack overflow in vim/vim

2

Valid Reported on May 16th 2022

Description

When providing certain input, the program will enter an infinite loop where it continually calls:
get_expr_register ->
cmdline_handle_backslash_key ->
getcmdline ->
getcmdline_int ->
cmdline_handle_backslash_key ->
get_expr_register ->
etc.

GDB

```
[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.
0x00005555556284c5 in getcmdline_int (firstc=61, count=0, indent=0, clear_c
1618            save_cmdline(&save_ccline);
```

```
#0  0x00005555556284c5 in getcmdline_int (firstc=61, count=0, indent=0, cle
#1  0x000055555562842d in getcmdline (firstc=61, count=0, indent=0, do_conc
#2  0x000055555572ef97 in get_expr_register () at register.c:104
#3  0x00005555556270cc in cmdline_handle_backslash_key (c=101, gotesc=0x7ff
#4  0x0000555555628c8f in getcmdline_int (firstc=61, count=0, indent=0, cle
#5  0x000055555562842d in getcmdline (firstc=61, count=0, i
#6  0x000055555572ef97 in get_expr_register () at register.c
#7  0x00005555556270cc in cmdline_handle_backslash_key (c=101, gotesc=0x7ff
```

```
#8  0x0000555555628c8f in getcmdline_int (firstc=61, count=0, indent=0, cle
#9  0x000055555562842d in getcmdline (firstc=61, count=0, indent=0, do_conc
#10 0x000055555572ef97 in get_expr_register () at register.c:104
```

...

```
#2052 0x000055555572ef97 in get_expr_register () at register.c:104
#2053 0x00005555556270cc in cmdline_handle_backslash_key (c=101, gotesc=0x7
#2054 0x0000555555628c8f in getcmdline_int (firstc=61, count=0, indent=0, c
#2055 0x000055555562842d in getcmdline (firstc=61, count=0, indent=0, do_cc
#2056 0x000055555572ef97 in get_expr_register () at register.c:104
#2057 0x00005555556270cc in cmdline_handle_backslash_key (c=101, gotesc=0x7
#2058 0x0000555555628c8f in getcmdline_int (firstc=61, count=0, indent=0, c
#2059 0x000055555562842d in getcmdline (firstc=61, count=0, indent=0, do_cc
#2060 0x000055555572ef97 in get_expr_register () at register.c:104
#2061 0x00005555556270cc in cmdline_handle_backslash_key (c=101, gotesc=0x7
#2062 0x0000555555628c8f in getcmdline_int (firstc=61, count=0, indent=0, c
#2063 0x000055555562842d in getcmdline (firstc=61, count=0, indent=0, do_cc
#2064 0x000055555572ef97 in get_expr_register () at register.c:104
#2065 0x00005555556270cc in cmdline_handle_backslash_key (c=101, gotesc=0x7
#2066 0x0000555555628c8f in getcmdline_int (firstc=61, count=0, indent=0, c
#2067 0x000055555562842d in getcmdline (firstc=61, count=0, indent=0, do_cc
#2068 0x000055555572ef97 in get_expr_register () at register.c:104
#2069 0x00005555556270cc in cmdline_handle_backslash_key (c=101, gotesc=0x7
#2070 0x0000555555628c8f in getcmdline_int (firstc=61, count=0, indent=0, c
#2071 0x000055555562842d in getcmdline (firstc=61, count=0, indent=0, do_cc
#2072 0x000055555572ef97 in get_expr_register () at register.c:104
```

Valgrind

```
==99366== Memcheck, a memory error detector
==99366== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==99366== Using Valgrind-3.14.0 and LibVEX; rerun with -h for copyright inf
==99366== Command: ./vim -u NONE -X -Z -e -s -S id:000000,si
==99366==
==99366== Stack overflow in thread #1: can't grow stack to 0x1ffe801000
```

```
==99366==
==99366== Process terminating with default action of signal 11 (SIGSEGV)
==99366==    at 0x4E31A97: kill (syscall-template.S:78)

==99366==    by 0x28B7C9: may_core_dump (os_unix.c:3529)
==99366==    by 0x28B781: mch_exit (os_unix.c:3495)
==99366==    by 0x3FADEC: getout (main.c:1726)
==99366==    by 0x24F54D: preserve_exit (misc1.c:2217)
==99366==    by 0x289482: deathtrap (os_unix.c:1175)
==99366==    by 0x4E3183F: ??? (in /usr/lib/x86_64-linux-gnu/libc-2.28.so)
==99366==    by 0x483573D: malloc (vg_replace_malloc.c:299)
```

Proof of Concept

```
./vim -u NONE -e -s -S crash_input
Segmentation fault
```

https://github.com/GreaterGoodest/vim-pocs/blob/master/crash_input

Impact

This could cause a denial of service due to crashing the process.

CVE
CVE-2022-1771
(Published)

Vulnerability Type
CWE-674: Uncontrolled Recursion

Severity
Medium (5.5)

Registry
Other

Affected Version
8.2.4963

Visibility
Public

Status
Fixed

Found by

Ryan Good
@greatergoodest
legend

Fixed by

Bram Moolenaar
@brammool
maintainer

This report was seen 1,511 times.

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

Ryan Good modified the report 6 months ago

Ryan Good modified the report 6 months ago

Ryan Good modified the report 6 months ago

Ryan Good modified the report 6 months ago

Ryan Good modified the report 6 months ago

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

Ryan Good modified the report 6 months ago

Ryan Good 6 months ago Researcher

Note: I have this set to "stack based buffer overflow" due to the lack of an appr
should really be Uncontrolled Recursion (https://cwe.mitre.org/data/definitio

Bram Moolenaar validated this vulnerability 6 months ago

I can reproduce the problem. The POC can be used for a regression test.

Ryan Good 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 51f0bf 6 months ago

Bram Moolenaar has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Bram Moolenaar 6 months ago

Fixed in patch 8.2.4975

Jamie Slome 6 months ago Admin

I have updated the report to CWE-674 as requested by the researcher above, as this CWE was missing from our weakness selection at the point of submission 👍

Sign in to join this conversation