huntr

Use After Free in radareorg/radare2

0

✓ Valid Reported on Dec 30th 2021

Description

This vulnerability is of use-after-free. The bug exists in latest stable release (radare2-5.5.4). Specifically, the vulnerable code is picked out as follows (`libr/io/io_bank.c`):

```
//  ./libr/io/io_bank.c line 229
        // the entry->data is a freed pointer address
        while (entry && r_io_submap_to (((RIOSubMap *)entry->data)) <= r_io
                //delete all submaps that are completly included in sm
                RRBNode *next = r_rbnode_next (entry);
                // this can be optimized, there is no need to do search her
                r_crbtree_delete (bank->submaps, entry->data, _find_sm_by_
                entry = next;
        }
```

Proof of Concept

Build the radare2 5.5.4 with address sanitizer, download the POC_FILE. Then run

```
# disable some features of address sanitizer to avoid false positive
export ASAN_OPTIONS=detect_leaks=0:abort_on_error=1:symbolize=0:allocator_
# trigger the crash
radare2 -A -q POC_FILE
```

The crash stack information is:

```
=================================================================
==7874==ERROR: AddressSanitizer: heap-use-after-free on address 0x604001aa8
READ of size 8 at 0x604001aa8d30 thread T0

    #0 0x7ffff70e69e5  (/src/projects/radare2-5.5.4/radare2/install-asan/li
    #1 0x7ffff70b8d9c  (/src/projects/radare2-5.5.4/radare2/install-asan/li
    #2 0x7ffff381a410  (/src/projects/radare2-5.5.4/radare2/install-asan/li
    #3 0x7ffff37e2b3a  (/src/projects/radare2-5.5.4/radare2/install-asan/li
    #4 0x7ffff37e159d  (/src/projects/radare2-5.5.4/radare2/install-asan/li
    #5 0x7ffff36a9556  (/src/projects/radare2-5.5.4/radare2/install-asan/li
    #6 0x7ffff763b6f3  (/src/projects/radare2-5.5.4/radare2/install-asan/li
    #7 0x7ffff73b80b2  (/lib/x86_64-linux-gnu/libc.so.6+0x270b2)
    #8 0x55555557239d  (/src/projects/radare2-5.5.4/radare2/install-asan/bi

0x604001aa8d30 is located 32 bytes inside of 40-byte region [0x604001aa8d1
freed by thread T0 here:
    #0 0x5555555ed392  (/src/projects/radare2-5.5.4/radare2/install-asan/bi
    #1 0x7ffff7b37d39  (/src/projects/radare2-5.5.4/radare2/install-asan/li
    #2 0x7ffff7b38fb3  (/src/projects/radare2-5.5.4/radare2/install-asan/li

previously allocated by thread T0 here:
    #0 0x5555555ed772  (/src/projects/radare2-5.5.4/radare2/install-asan/bi
    #1 0x7ffff7b3368a  (/src/projects/radare2-5.5.4/radare2/install-asan/li
    #2 0x7ffff70e5700  (/src/projects/radare2-5.5.4/radare2/install-asan/li

SUMMARY: AddressSanitizer: heap-use-after-free (/src/projects/radare2-5.5.
Shadow bytes around the buggy address:
  0x0c088034d150: fa fa fd fd fd fd fd fa fa fa fd fd fd fd fd fa
  0x0c088034d160: fa fa fd fd fd fd fd fa fa fa fd fd fd fd fd fa
  0x0c088034d170: fa fa fd fd fd fd fd fa fa fa fd fd fd fd fd fa
  0x0c088034d180: fa fa 00 00 00 00 00 fa fa fa 00 00 00 00 00 fa
  0x0c088034d190: fa fa 00 00 00 00 00 fa fa fa 00 00 00 00 00 fa
=>0x0c088034d1a0: fa fa fd fd fd fd[fd]fa fa fa fd fd fd fd fd fa
  0x0c088034d1b0: fa fa fd fd fd fd fd fa fa fa fd fd fd fd fd fa
  0x0c088034d1c0: fa fa 00 00 00 00 00 fa fa fa fd fd fd fd fd fa
  0x0c088034d1d0: fa fa 00 00 00 00 00 fa fa fa fd fd fd fd fd fa
  0x0c088034d1e0: fa fa 00 00 00 00 00 fa fa fa fd fd fd fd fd fa
  0x0c088034d1f0: fa fa 00 00 00 00 00 fa fa fa 00 00 00 00 00 fa
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1

  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
  Shadow gap:              cc
==7874==ABORTING

#0  0x00007ffff73d718b in raise () from /lib/x86_64-linux-gnu/libc.so.6
#1  0x00007ffff73b6859 in abort () from /lib/x86_64-linux-gnu/libc.so.6
#2  0x000055555560ba77 in __sanitizer::Abort() ()
#3  0x0000555555609fa1 in __sanitizer::Die() ()
#4  0x00005555555f14e4 in __asan::ScopedInErrorReport::~ScopedInErrorReport
#5  0x00005555555f30aa in __asan::ReportGenericError(unsigned long, unsigne
#6  0x00005555555f3948 in __asan_report_load8 ()
#7  0x00007ffff70e69e6 in r_io_bank_map_add_top (io=<optimized out>, bankic
#8  0x00007ffff70b8d9d in r_io_map_add (io=0x61b000001c80, fd=<optimized ou
#9  0x00007ffff381a411 in add_section (core=0x7fffec26a800, sec=0x60200037f
#10 bin_sections (r=0x7fffec26a800, pj=<optimized out>, mode=<optimized out
#11 0x00007ffff37e2b3b in r_core_bin_info (core=0x7fffec26a800, action=<opt
#12 0x00007ffff37e159e in r_core_bin_set_env (r=0x7fffec26a800, binfile=<op
#13 0x00007ffff36a9557 in r_core_file_do_load_for_io_plugin (r=0x7fffec26a8
#14 r_core_bin_load (r=0x7fffec26a800, filenameuri=<optimized out>, baddr=<
#15 0x00007ffff763b6f4 in r_main_radare2 (argc=<optimized out>, argv=<optin
#16 0x00007ffff73b80b3 in __libc_start_main () from /lib/x86_64-linux-gnu/l
#17 0x000055555557239e in _start ()
```

Impact

The bug is of Heap-use-after-free. The POC attached here can be directly used to launch DoS attack. Besides, it is very possible for the attacker to finally accomplish RCE (Remote Code Execution).

References

- PoC file

CVE
CVE-2022-0139
(Published)

Vulnerability Type
CWE-416: Use After Free

Severity
High (7.1)

Visibility
Public

Status
Fixed

Found by

Cen Zhang
@occia
unranked

Fixed by

pancake
@trufae
maintainer

This report was seen 456 times.

We are processing your report and will contact the **radareorg/radare2** team within 24 hours.

a year ago

We have contacted a member of the **radareorg/radare2** team and are waiting to hear back

a year ago

We have sent a follow up to the **radareorg/radare2** team. We will try again in 7 days. a year ago

A **radareorg/radare2** maintainer validated this vulnerability a year ago

Cen Zhang has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

A **radareorg/radare2** maintainer a year ago Maintainer

Fixed in https://github.com/radareorg/radare2/pull/19549 thanks for reporting

Cen Zhang a year ago Researcher

hi, thank you for the fix! Just to mention that there is a bounty for patcher. And you can complete the "Fix Submission" process of this report and get the bounty (though not much).

Cen Zhang 10 months ago Researcher

@admin , hi, I think this bug can be published since it has already been fixed in the above link. (Though the developer didn't submit a fix in huntr website).

Cen Zhang 10 months ago Researcher

Hi, I've tested the radare2 with latest commit (ed2030b79e68986bf04f3a6279463ab989fe400f) , the use-after-free bug can still be triggered. The fix didn't really helps since the use-after-free trigger point is at L229 of `io_bank.c` while the patch is at line 233.

Cen Zhang 10 months ago Researcher

Sorry, I messed up the radare2 binaries in my environment, the bug has been fixed in latest commit~ Sorry for the caused inconvenience!

pancake 10 months ago Maintainer

Then it's good to go?

Cen Zhang 10 months ago Researcher

Yes, it is fixed.

Jamie Slome 10 months ago Admin

@maintainer - are we able to submit a fix against this report using the `confirm fix` button?

pancake marked this as fixed in **5.6.0** with commit **378972** 10 months ago

pancake has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

Sign in to join this conversation