Use After Free in radareorg/radare2

Valid Reported on Jan 22nd 2022

0

Description

This vulnerability is of type use-after-free. And after quick investigation I think it is very likely to be successfully exploited to remote code execution. The bug exists in latest stable release (radare2-5.5.4) and lastest master branch (ed2030b79e68986bf04f3a6279463ab989fe400f, updated in Jan 22, 2022). Specifically, the vulnerable code (located at `libr/bin/format/pyc/marshal.c`) and the bug's basic explanation are highlighted as follows:

```
// libr/bin/format/pyc/marshal.c
static pyc_object *get_object(RBuffer *buffer) {
...
// line 1114
// the ref_idx->data points to an already freed memory block!
        if (ref_idx->data != ret) {
            // this bug might be exploited since program will call dangerou
            // in a released version of radare2 (no address sanitizer)
            free_object (ref_idx->data);
```

Proof of Concept

Build the radare2 (5.5.4 or latest commit ed2030b79e68986bf04f3a6279463ab989fe400f) and run it using the input POC.

```
# build the radare2 with address sanitizer
export CFLAGS=" -fsanitize=address "; export CXXFLAGS=" -fsanitize=address
CFGARG=" --enable-shared=no " PREFIX=`realpath install` bash sys/build.sh
# disable some features of address sanitizer to avoid false
export ASAN_OPTIONS=detect_leaks=0:abort_on_error=1:symboli
# trigger the crash
```

```
./radare2 -A -q POC_FILE
```

The crash stack is:

```
=================================================================
==18153==ERROR: AddressSanitizer: heap-use-after-free on address 0x60300008
READ of size 8 at 0x60300008b3f0 thread T0
    #0 0x7ffff2c29c24  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #1 0x7ffff2c1dc21  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #2 0x7ffff2c23427  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #3 0x7ffff2c204db  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #4 0x7ffff2c1b7b3  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #5 0x7ffff2599d94  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #6 0x7ffff2598054  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #7 0x7ffff257df9e  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #8 0x7ffff252179b  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #9 0x7ffff2520876  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #10 0x7ffff386facc  (/src/projects/radare2-5.5.4/lastest-radare2/instal
    #11 0x7ffff76312ae  (/src/projects/radare2-5.5.4/lastest-radare2/instal
    #12 0x7ffff73a50b2  (/lib/x86_64-linux-gnu/libc.so.6+0x270b2)
    #13 0x55555557239d  (/src/projects/radare2-5.5.4/lastest-radare2/instal

0x60300008b3f0 is located 0 bytes inside of 24-byte region [0x60300008b3f0,
freed by thread T0 here:
    #0 0x5555555ed392  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #1 0x7ffff78af77d  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #2 0x7ffff2c1d50a  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #3 0x7ffff2c227b5  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #4 0x7ffff2c1dc21  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #5 0x7ffff2c23427  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #6 0x7ffff2c204db  (/src/projects/radare2-5.5.4/lastest-radare2/install

previously allocated by thread T0 here:
    #0 0x5555555ed772  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #1 0x7ffff78a8889  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #2 0x7ffff2c1dc21  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #3 0x7ffff2c23427  (/src/projects/radare2-5.5.4/lastest
    #4 0x7ffff2c204db  (/src/projects/radare2-5.5.4/lastest
```

```
SUMMARY: AddressSanitizer: heap-use-after-free (/src/projects/radare2-5.5.4
Shadow bytes around the buggy address:
  0x0c0680009620: 00 00 fa fa 00 00 00 fa fa fa 00 00 00 00 fa fa

  0x0c0680009630: 00 00 00 00 fa fa 00 00 00 00 fa fa 00 00 00 fa
  0x0c0680009640: fa fa 00 00 00 fa fa fa 00 00 00 00 fa fa 00 00
  0x0c0680009650: 00 00 fa fa 00 00 00 00 fa fa fd fd fd fa fa fa
  0x0c0680009660: fd fd fd fa fa fa fd fd fd fa fa fa fd fd fd fa
=>0x0c0680009670: fa fa fd fd fd fa fa fa fd fd fd fa fa fa[fd]fd
  0x0c0680009680: fd fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c0680009690: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c06800096a0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c06800096b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c06800096c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa 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
  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
==18153==ABORTING

Program received signal SIGABRT, Aborted.
0x00007ffff73c418b in raise () from /lib/x86_64-linux-gnu/libc.so.6
(gdb) bt
#0  0x00007ffff73c418b in raise () from /lib/x86_64-linux-g
#1  0x00007ffff73a3859 in abort () from /lib/x86_64-linux-gnu/libc.so.6
```

```
#2  0x00005555560ba77 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  0x00007ffff2c29c25 in get_object (buffer=<optimized out>) at /src/proje
#8  0x00007ffff2c1dc22 in get_code_object (buffer=<optimized out>) at /src/
#9  0x00007ffff2c23428 in get_object (buffer=<optimized out>) at /src/proje
#10 0x00007ffff2c204dc in get_sections_symbols_from_code_objects (buffer=<c
    magic=<optimized out>) at /src/projects/radare2-5.5.4/lastest-radare2/l
#11 0x00007ffff2c2e582 in pyc_get_sections_symbols (sections=0x7fffffffc136
    at /src/projects/radare2-5.5.4/lastest-radare2/libr/../libr/bin/p/../fc
#12 0x00007ffff2c1b7b4 in symbols (arch=<optimized out>) at /src/projects/r
#13 0x00007ffff2599d95 in r_bin_object_set_items (bf=<optimized out>, o=<op
#14 0x00007ffff2598055 in r_bin_object_new (bf=<optimized out>, plugin=<opt
    sz=<optimized out>) at bobj.c:168
#15 0x00007ffff257df9f in r_bin_file_new_from_buffer (bin=0x616000000980, f
    loadaddr=<optimized out>, fd=<optimized out>, pluginname=<optimized out
#16 0x00007ffff252179c in r_bin_open_buf (bin=<optimized out>, buf=<optimiz
#17 0x00007ffff2520877 in r_bin_open_io (bin=0x616000000980, opt=<optimized
#18 0x00007ffff386facd in r_core_file_do_load_for_io_plugin (r=0x7fffec3328
#19 r_core_bin_load (r=0x7fffec332800, filenameuri=<optimized out>, baddr=<
#20 0x00007ffff76312af in r_main_radare2 (argc=<optimized out>, argv=<optim
#21 0x00007ffff73a50b3 in __libc_start_main () from /lib/x86_64-linux-gnu/l
#22 0x000055555557239e in _start ()
```

Impact

If address sanitizer is disabled during the compiling, the program should executes into the `free_object` function with an already freed memory block. Therefore I think it is very likely to be exploited by carefully manipulating the wild pointer `ref_idx->data`. For more general description of use-after-free, see CWE.

References

- poc file

CVE-2022-0520
(Published)

Vulnerability Type

CWE-416: Use After Free

Severity
High (8.8)

Visibility
Public

Status
Fixed

Found by

Cen Zhang

@occia

unranked

Fixed by

pancake

@trufae

maintainer

This report was seen 408 times.

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

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

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

We have sent a second follow up to the **radareorg/radare2** team. We will try again in 10 days.
10 months ago

pancake validated this vulnerability 10 months ago

Cen Zhang has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

pancake 10 months ago Maintainer

Fixed in https://github.com/radareorg/radare2/pull/19667
https://github.com/radareorg/radare2/pull/19667/commits/1c29d4b20de505dad408c4ab3af3309083a80685

pancake marked this as fixed in 5.6.2 with commit 8525ad 10 months ago

pancake has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Sign in to join this conversation