Access of Memory Location Before Start of Buffer in radareorg/radare2

0

Valid Reported on Jan 22nd 2022

Description

This vulnerability is of out-of-bound read which is caused by negative buffer index. 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 is highlighted out as follows:

```
// libr/anal/p/anal_arm_cs.c
...
#define VEC64_MASK(sh, sz) (bitmask_by_width[63]^(bitmask_by_width[sz-1]<<s
...
static void vector64_dst_append(RStrBuf *sb, csh *handle, cs_insn *insn, ir
...
            // line 1087
            r_strbuf_appendf (sb, "0x%"PFMT64x",&,%s%s,0x%"PFMT64x",&,|,%s%
                // under crafted inputs, the dimsize can be 0, causes the i
                mask, REG64 (n), regc, VEC64_MASK (shift, dimsize), REG64 (
...
```

As shown above, the `bitmask_by_width[sz-1]` can be `bitmask_by_width[-1]` under crafted inputs.

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 positives

export ASAN_OPTIONS=detect_leaks=0:abort_on_error=1:symbolize=0:allocator_m
# trigger the crash
./radare2 -A -q POC_FILE
```

The crash stack info is:

```
=================================================================
==17883==ERROR: AddressSanitizer: global-buffer-overflow on address 0x7fff
READ of size 8 at 0x7ffff1516f78 thread T0
    #0 0x7ffff028908c  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #1 0x7ffff0289967  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #2 0x7ffff024f321  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #3 0x7ffff023273c  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #4 0x7ffff0bc9ce8  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #5 0x7ffff0bdaf28  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #6 0x7ffff0bf4278  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #7 0x7ffff3a575e5  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #8 0x7ffff37c98fa  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #9 0x7ffff37c7b66  (/src/projects/radare2-5.5.4/lastest-radare2/install
    #10 0x7ffff37b7b0c  (/src/projects/radare2-5.5.4/lastest-radare2/instal
    #11 0x7ffff3594f1c  (/src/projects/radare2-5.5.4/lastest-radare2/instal
    #12 0x7ffff373c041  (/src/projects/radare2-5.5.4/lastest-radare2/instal
    #13 0x7ffff372dcb0  (/src/projects/radare2-5.5.4/lastest-radare2/instal
    #14 0x7ffff352c392  (/src/projects/radare2-5.5.4/lastest-radare2/instal
    #15 0x7ffff7634c5e  (/src/projects/radare2-5.5.4/lastest-radare2/instal
    #16 0x7ffff73a50b2  (/lib/x86_64-linux-gnu/libc.so.6+0x270b2)
    #17 0x55555557239d  (/src/projects/radare2-5.5.4/lastest-radare2/instal

0x7ffff1516f78 is located 8 bytes to the left of global variable 'bitmask_b
0x7ffff1516f78 is located 48 bytes to the right of global variable '<string
  '<string literal>' is ascii string 'lr,pc,='
SUMMARY: AddressSanitizer: global-buffer-overflow (/src/projects/radare2-5.
Shadow bytes around the buggy address:
  0x10007e29ad90: f9 f9 f9 f9 00 00 04 f9 f9 f9 f9 f9 00 00
  0x10007e29ada0: f9 f9 f9 f9 00 00 01 f9 f9 f9 f9 f9 00 00 00 00
```

```
  0x10007e29adb0: f9 f9 f9 f9 00 00 03 f9 f9 f9 f9 f9 00 00 00 00
  0x10007e29adc0: 03 f9 f9 f9 f9 f9 f9 f9 00 00 06 f9 f9 f9 f9 f9
  0x10007e29add0: 00 00 00 02 f9 f9 f9 f9 00 07 f9 f9 f9 f9 f9 f9

=>0x10007e29ade0: 00 00 01 f9 f9 f9 f9 f9 00 f9 f9 f9 f9 f9 f9[f9]
  0x10007e29adf0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007e29ae00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007e29ae10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007e29ae20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007e29ae30: f9 f9 f9 f9 f9 f9 f9 f9 f9 f9 f9 f9 f9 f9 f9 f9
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
==17883==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-gnu/libc.so.6
#1  0x00007ffff73a3859 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::~Sco
#5  0x00005555555f30aa in __asan::ReportGenericError(unsigned long, unsigne
```

```
#6  0x000055555553f948 in __asan_report_load8 ()
#7  0x00007ffff028908d in vector64_dst_append (sb=<optimized out>, handle=<
    at /src/projects/radare2-5.5.4/lastest-radare2/libr/../libr/anal/p/anal

#8  0x00007ffff0289968 in arm64fpmath (a=<optimized out>, op=<optimized out
    insn=0x611000018b40, opchar=<optimized out>, negate=<optimized out>) at
#9  0x00007ffff024f322 in analop64_esil (a=<optimized out>, op=0x6150000064
    insn=0x611000018b40) at /src/projects/radare2-5.5.4/lastest-radare2/libr
#10 0x00007ffff023273d in analop (a=<optimized out>, op=<optimized out>, ad
    at /src/projects/radare2-5.5.4/lastest-radare2/libr/../libr/anal/p/anal
#11 0x00007ffff0bc9ce9 in r_anal_op (anal=<optimized out>, op=0x6150000064
#12 0x00007ffff0bdaf29 in fcn_recurse (anal=<optimized out>, fcn=<optimized
#13 0x00007ffff0bf4279 in r_anal_function_bb (anal=0x61a000000080, fcn=0x61
#14 r_anal_function (anal=0x61a000000080, fcn=<optimized out>, addr=284, le
#15 0x00007ffff3a575e6 in __core_anal_fcn (core=0x7fffec332800, at=<optimiz
#16 r_core_anal_fcn (core=<optimized out>, at=<optimized out>, from=<optimi
#17 0x00007ffff37c98fb in _anal_calls (core=0x7fffec332800, addr=184, addr_
#18 0x00007ffff37c7b67 in cmd_anal_calls (core=<optimized out>, input=<opti
#19 0x00007ffff37b7b0d in cmd_anal_all (core=<optimized out>, input=<optimi
#20 0x00007ffff3594f1d in cmd_anal (data=0x7fffec332800, input=<optimized o
#21 0x00007ffff373c042 in r_core_cmd_subst_i (core=<optimized out>, cmd=<op
#22 0x00007ffff372dcb1 in r_core_cmd_subst (core=<optimized out>, cmd=0x602
#23 0x00007ffff352c393 in run_cmd_depth (core=<optimized out>, cmd=<optimiz
#24 r_core_cmd (core=<optimized out>, cstr=<optimized out>, log=<optimized
#25 0x00007ffff34c0a15 in r_core_cmd0 (core=0x2, cmd=0x7fffffff8470 "") at
#26 0x00007ffff7634c5f in r_main_radare2 (argc=<optimized out>, argv=<optim
#27 0x00007ffff73a50b3 in __libc_start_main () from /lib/x86_64-linux-gnu/l
#28 0x000055555557239e in _start ()
```

Impact

The bug causes the program reads data before the beginning of the intented buffer. Typically, this can allow attackers to read sensitive information from other memory locations or cause a crash. More details see CWE-125: Out-of-bounds read.

References

- POC File

CVE
CVE-2022-0522
(Published)

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

Severity
Medium (6.3)

Visibility
Public

Status
Fixed

Found by

Cen Zhang
@occia
unranked

Fixed by

pancake
@trufae
maintainer

This report was seen 384 times.

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

Cen Zhang modified the report 10 months ago

Cen Zhang modified the report 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
10 months ago

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

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/commits/58eb66a051a2bf875617509578312521299272 94

pancake marked this as fixed in 5.6.2 with commit d17a7b 10 months ago

pancake has been awarded the fix bounty ✅

This vulnerability will not receive a CVE ❌

Sign in to join this conversation