Ruby - Bug #8680

ruby crashes when built with AddressSanitizer

07/24/2013 08:41 PM - halfie (Ruby Guy)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	6398f79853f73fd307f5ecd266a94644ecf625b9	**Backport:**	1.9.3: UNKNOWN, 2.0.0: UNKNOWN

Description

ruby crashes when built with AddressSanitizer (http://clang.llvm.org/docs/AddressSanitizer.html).

To reproduce the problem, try to build latest ruby from trunk.

$ gcc --version
gcc (GCC) 4.8.1 20130603 (Red Hat 4.8.1-1)
...

$ export CFLAGS="-fsanitize=address -O2 -g"
$ export LDFLAGS="-fsanitize=address"
$ ./configure
$ make
...
processing probes in object files
rm -f probes.o
linking miniruby

==13084== ERROR: AddressSanitizer: stack-buffer-underflow on address 0x7fffaa2c43e0 at pc 0x7f3c76db0ac2 bp 0x7fffaa2c3ee0 sp 0x7fffaa2c3ed8
READ of size 8 at 0x7fffaa2c43e0 thread T0
#0 0x7f3c76db0ac1 (/scratch/repos/ruby/miniruby+0x11eac1)
#1 0x7f3c76db0cba (/scratch/repos/ruby/miniruby+0x11ecba)
#2 0x7f3c76db2ab0 (/scratch/repos/ruby/miniruby+0x120ab0)
#3 0x7f3c76db3a8e (/scratch/repos/ruby/miniruby+0x121a8e)
#4 0x7f3c76db52ab (/scratch/repos/ruby/miniruby+0x1232ab)
#5 0x7f3c76f1a668 (/scratch/repos/ruby/miniruby+0x288668)
#6 0x7f3c76dd1d82 (/scratch/repos/ruby/miniruby+0x13fd82)
#7 0x7f3c76e50df4 (/scratch/repos/ruby/miniruby+0x1bedf4)
#8 0x7f3c76e6fb3d (/scratch/repos/ruby/miniruby+0x1ddb3d)
#9 0x7f3c7700b86b (/scratch/repos/ruby/miniruby+0x37986b)
#10 0x7f3c76e2e3a4 (/scratch/repos/ruby/miniruby+0x19c3a4)
#11 0x7f3c76efa3f0 (/scratch/repos/ruby/miniruby+0x2683f0)
#12 0x7f3c76d8a68d (/scratch/repos/ruby/miniruby+0xf868d)
#13 0x7f3c76ef9fae (/scratch/repos/ruby/miniruby+0x267fae)
#14 0x7f3c76d8c44e (/scratch/repos/ruby/miniruby+0xfa44e)
#15 0x7f3c76d8f816 (/scratch/repos/ruby/miniruby+0xfd816)
#16 0x7f3c76fc7b37 (/scratch/repos/ruby/miniruby+0x335b37)
#17 0x7f3c76fff525 (/scratch/repos/ruby/miniruby+0x36d525)
#18 0x7f3c76fd9f69 (/scratch/repos/ruby/miniruby+0x347f69)
#19 0x7f3c76fe7986 (/scratch/repos/ruby/miniruby+0x355986)
#20 0x7f3c76fec907 (/scratch/repos/ruby/miniruby+0x35a907)
#21 0x7f3c76d83de1 (/scratch/repos/ruby/miniruby+0xf1de1)
#22 0x7f3c76d88dec (/scratch/repos/ruby/miniruby+0xf6dec)
#23 0x7f3c76d0ad98 (/scratch/repos/ruby/miniruby+0x78d98)
#24 0x7f3c725a8b74 (/usr/lib64/libc-2.17.so+0x21b74)
#25 0x7f3c76d0b6d0 (/scratch/repos/ruby/miniruby+0x796d0)

Address 0x7fffaa2c43e0 is located at offset 0 in frame <rb_io_getline_1> of T0's stack:
This frame has 1 object(s):
[32, 36) 'cr'
...
make: *** [.rbconfig.time] Error 1

Decoding the above trace gives us,

```
=================================================================
==12684== ERROR: AddressSanitizer: stack-buffer-underflow on address 0x7fff4ec9f180 at pc 0x7ff3ce5e6ac2 bp 0x7fff4ec9ec80
sp 0x7fff4ec9ec78
READ of size 8 at 0x7fff4ec9f180 thread T0
#0 0x7ff3ce5e6ac1 in mark_locations_array /scratch/repos/ruby/gc.c:2682
#1 0x7ff3ce5e6cba in gc_mark_locations /scratch/repos/ruby/gc.c:2693
#2 0x7ff3ce5e8ab0 in gc_marks_body /scratch/repos/ruby/gc.c:3419
#3 0x7ff3ce5e9a8e in gc_marks /scratch/repos/ruby/gc.c:3653
#4 0x7ff3ce5eb2ab in heap_prepare_freeslot /scratch/repos/ruby/gc.c:906
#5 0x7ff3ce750668 in str_alloc /scratch/repos/ruby/string.c:384
#6 0x7ff3ce607d82 in rb_io_getline_fast /scratch/repos/ruby/io.c:2902
#7 0x7ff3ce686df4 in lex_getline /scratch/repos/ruby/parse.y:5416
#8 0x7ff3ce6a5b3d in yycompile0 /scratch/repos/ruby/parse.y:5347
#9 0x7ff3ce84186b in rb_suppress_tracing /scratch/repos/ruby/vm_trace.c:367
#10 0x7ff3ce6643a4 in yycompile /scratch/repos/ruby/parse.y:5379
#11 0x7ff3ce7303f0 in load_file_internal /scratch/repos/ruby/ruby.c:1728
#12 0x7ff3ce5c068d in rb_ensure /scratch/repos/ruby/eval.c:820
#13 0x7ff3ce72ffae in load_file /scratch/repos/ruby/ruby.c:1765
#14 0x7ff3ce5c244e in rb_load_internal /scratch/repos/ruby/load.c:594 (discriminator 3)
#15 0x7ff3ce5c5816 in rb_require_safe /scratch/repos/ruby/load.c:958
#16 0x7ff3ce7fdb37 in vm_call_cfunc_with_frame /scratch/repos/ruby/vm_insnhelper.c:1493
#17 0x7ff3ce835525 in vm_call_cfunc /scratch/repos/ruby/vm_insnhelper.c:1583
#18 0x7ff3ce80ff69 in vm_exec_core /scratch/repos/ruby/insns.def:1017
#19 0x7ff3ce81d986 in vm_exec /scratch/repos/ruby/vm.c:1198
#20 0x7ff3ce822907 in rb_iseq_eval_main /scratch/repos/ruby/vm.c:1448
#21 0x7ff3ce5b9de1 in ruby_exec_internal /scratch/repos/ruby/eval.c:252 (discriminator 1)
#22 0x7ff3ce5bedec in ruby_exec_node /scratch/repos/ruby/eval.c:317
#23 0x7ff3ce540d98 in main /scratch/repos/ruby/main.c:36
#24 0x7ff3c9ddeb74 in ?? ??:0
#25 0x7ff3ce5416d0 in _start ??:?

...
```

Is this expected behaviour? Can we patch ruby to be a bit more friendly towards AddressSanitizer?

This "problem" exists in older versions too (e.g. ruby-2.0.0-p247).

Associated revisions

Revision 2bf5be1db569ed4f7d4633fe7d56e585ae6adf52 - 09/26/2013 05:46 AM - nobu (Nobuyoshi Nakada)

gc.c: disable AddressSanitizer

- gc.c (mark_locations_array): disable AddressSanitizer. based on a
  patch by halfie (Ruby Guy) at [ruby-core:57372].
  [ruby-core:56155] [Bug #8680]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@43047 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 2bf5be1d - 09/26/2013 05:46 AM - nobu (Nobuyoshi Nakada)

gc.c: disable AddressSanitizer

- gc.c (mark_locations_array): disable AddressSanitizer. based on a
  patch by halfie (Ruby Guy) at [ruby-core:57372].
  [ruby-core:56155] [Bug #8680]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@43047 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision cbb6a3a6aa40481c976afc4e26f90d1c0b0e2a05 - 11/06/2018 05:06 AM - shyouhei (Shyouhei Urabe)

annotate functions to blacklist MSAN

In these functions we are intentionally reading memory address
not owned by us. These reads should not be diagnosed.

See also [Bug #8680]
See also https://travis-ci.org/ruby/ruby/jobs/451202718

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@65564 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision cbb6a3a6aa40481c976afc4e26f90d1c0b0e2a05 - 11/06/2018 05:06 AM - shyouhei (Shyouhei Urabe)

annotate functions to blacklist MSAN

In these functions we are intentionally reading memory address
not owned by us. These reads should not be diagnosed.

See also [Bug #8680]
See also https://travis-ci.org/ruby/ruby/jobs/451202718

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@65564 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision cbb6a3a6 - 11/06/2018 05:06 AM - shyouhei (Shyouhei Urabe)

annotate functions to blacklist MSAN

In these functions we are intentionally reading memory address
not owned by us. These reads should not be diagnosed.

See also [Bug #8680]
See also https://travis-ci.org/ruby/ruby/jobs/451202718

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@65564 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 07/24/2013 09:23 PM - akr (Akira Tanaka)

2013/7/24 halfie (Ruby Guy) assie181@gmail.com:

> Bug #8680: ruby crashes when built with AddressSanitizer
> https://bugs.ruby-lang.org/issues/8680
>
> Is this expected behaviour? Can we patch ruby to be a bit more friendly towards AddressSanitizer?

Ruby uses conservertive garbage collection.
I.e. Ruby scans stack area from beginning to end, including undefined area intentionally.
Is it related?

Tanaka Akira

#2 - 07/26/2013 01:18 AM - kosaki (Motohiro KOSAKI)

Does Boehm GC works on the AddressSanitizer? IOW, is this ruby specific?

#3 - 07/26/2013 03:30 PM - naruse (Yui NARUSE)

- Status changed from Open to Feedback

#4 - 07/31/2013 12:32 AM - halfie (Ruby Guy)

If we are 100% sure that the GC is fine, then the following patch can be applied to make CRuby ASAN friendly.

diff --git a/gc.c b/gc.c
index caaf00c..de7815b 100644
--- a/gc.c
+++ b/gc.c
@@ -31,6 +31,12 @@
#include <sys/types.h>
#include <assert.h>

+#if defined(**clang**) || defined (**GNUC**)
+# define ATTRIBUTE_NO_ADDRESS_SAFETY_ANALYSIS **attribute**((no_address_safety_analysis)) **attribute** ((noinline))

```
+#else
+# define ATTRIBUTE_NO_ADDRESS_SAFETY_ANALYSIS
+#endif
+
#ifdef HAVE_SYS_TIME_H
#include <sys/time.h>
#endif
@@ -2674,6 +2680,7 @@ ruby_stack_check(void)
#endif
}

+ATTRIBUTE_NO_ADDRESS_SAFETY_ANALYSIS
static void
mark_locations_array(rb_objspace_t *objspace, register VALUE *x, register long n)
{
```

#5 - 07/31/2013 12:33 AM - halfie (Ruby Guy)

- File 0001-enable-usage-of-AddressSanitizer-in-CRuby.patch added

#6 - 09/25/2013 10:27 PM - halfie (Ruby Guy)

- File 0001-Enable-usage-of-AddressSanitizer-in-CRuby.patch added

Attaching latest patch against tip. Feedback is welcome.

#7 - 09/26/2013 02:46 PM - nobu (Nobuyoshi Nakada)

- % Done changed from 0 to 100

- Status changed from Feedback to Closed

This issue was solved with changeset r43047.
Ruby, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

---

gc.c: disable AddressSanitizer

- gc.c (mark_locations_array): disable AddressSanitizer. based on a
  patch by halfie (Ruby Guy) at [ruby-core:57372].
  [ruby-core:56155] [Bug #8680]

Files

0001-enable-usage-of-AddressSanitizer-in-CRuby.patch	899 Bytes	07/31/2013	halfie (Ruby Guy)
0001-Enable-usage-of-AddressSanitizer-in-CRuby.patch	1.26 KB	09/25/2013	halfie (Ruby Guy)