Hello,

We experienced a segfault in our application. It seems that it happened right after we've upgraded from ruby 2.6 to ruby 2.7.2. The faulting ruby code is actionpack-6.0.3.4/lib/action_controller/log_subscriber.rb:16:

```
info " Parameters: #{params.inspect}" unless params.empty?
```

However, the problem seems to be due to string.c:1072

Here is an extract of the output from ruby (full output attached):

```
-- Control frame information -----------------------------------------------
   c:0073 p:---- s:0500 e:000499 CFUNC :inspect
   c:0072 p:---- s:0497 e:000496 CFUNC :inspect
   c:0071 p:0137 s:0493 e:000490 METHOD /usr/app/rubygems/ruby/2.7.0/gems/actionpack-6.0.3.4/lib/action_controller/log_subscriber.rb:16
   c:0070 p:0038 s:0483 e:000482 METHOD /usr/app/rubygems/ruby/2.7.0/gems/active_support-6.0.3.4/lib/active_support/supportsubscriber.rb:145

-- Ruby level backtrace information ----------------------------------------
/usr/app/rubygems/ruby/2.7.0/gems/puma-5.0.4.4/lib/puma/thread_pool.rb:145:in `block in spawn_thread'
/usr/app/rubygems/ruby/2.7.0/gems/puma-5.0.4.4/lib/puma/server.rb:430:in `process_client'
/usr/app/rubygems/ruby/2.7.0/gems/puma-5.0.4.4/lib/puma/request.rb:75:in `handle_request'
...
```

Thank you for the report. I think that this is the same issue as #17289 and #17362. Unfortunately, the core team has no reproducible code, so we haven't investigated the issue yet. Can you provide your code, or at least, the core dump file?
I've attached code that reproduces the issue, along with output and the macOS crash report from my machine. Just run bundle and then ruby segfault.rb. It reliably segfaults for me within a minute or two on Ruby 2.7.5. I tried it on Ruby 3.1.1, and it doesn't appear to have the same problem.

We started seeing this segfault after upgrading to Ruby 2.7, and continue to see it almost daily in production.

It seems that both rails and anyway_config need to be required to trigger the issue. I'm not sure why, but if I had to guess I would say that it's something related to either refinements or pattern matching.

With this new information, could we reopen the issue please?

I've attached code that reproduces the issue, along with output and the macOS crash report from my machine. Just run bundle and then ruby segfault.rb. It reliably segfaults for me within a minute or two on Ruby 2.7.5. I tried it on Ruby 3.1.1, and it doesn't appear to have the same problem.

We started seeing this segfault after upgrading to Ruby 2.7, and continue to see it almost daily in production.

It seems that both rails and anyway_config need to be required to trigger the issue. I'm not sure why, but if I had to guess I would say that it's something related to either refinements or pattern matching.

With this new information, could we reopen the issue please?

I've attached code that reproduces the issue, along with output and the macOS crash report from my machine. Just run bundle and then ruby segfault.rb. It reliably segfaults for me within a minute or two on Ruby 2.7.5. I tried it on Ruby 3.1.1, and it doesn't appear to have the same problem.

We started seeing this segfault after upgrading to Ruby 2.7, and continue to see it almost daily in production.

It seems that both rails and anyway_config need to be required to trigger the issue. I'm not sure why, but if I had to guess I would say that it's something related to either refinements or pattern matching.

With this new information, could we reopen the issue please?
h = [:foo] * 8
20.times.map { Thread.new { 10_000.times { |i| h.inspect } } }.map(&:join)
p :done

$ ./miniruby -v segfault.rb
ruby 2.7.6p209 (2021-12-31 revision 1034b6e7b0) [x86_64-linux]
seauth.rb:15: [BUG] Segmentation fault at 0x0000000000000000
(snip)

There are search results for git bisect.
- f2286925f08406bc857f7b03ad6779a5d61443ae will fix the issue but it is too large to backport.
- 66c644da5e80258bb3217941223d9292923c3548 is the first commit that creates a problem at least as far as I have observed.

And I have confirmed that the problem is suppressed by the revert commit.

diff --git a/vm_eval.c b/vm_eval.c
index 76e56fac8a..3ef6ff3cd9 100644
--- a/vm_eval.c
+++ b/vm_eval.c
@@ -1010,17 +1010,7 @@
     rb_funcallv_with_cc(struct rb_call_data *cd, VALUE recv, ID mid, int argc, const
     vm_search_method(cd, recv);

     if (LIKELY(! UNDEFINED_METHOD_ENTRY_P(cc->me))) {
         return RETURN_VALUE(ruby_array(1),
                 GET_EC(),
                 &(struct rb_calling_info) {
                 Qundef,
                 recv,
                 argc,
                 RB_NO_KEYWORDS,
                 },
                 cd,
                 argv
         +
         RETURN_VALUE(ruby_array(1),recv, mid, argc, argv, cc->me, RB_NO_KEYWORDS);
         }
     }

However, this is not a backport and I am not sure if it is acceptable from a policy standpoint.

#11 - 03/19/2022 02:21 PM - wanabe (_ wanabe)

Alternatively, the following seems to solve the problem.

diff --git a/vm_eval.c b/vm_eval.c
index 76e56fac8a..64af8815ea 100644
--- a/vm_eval.c
+++ b/vm_eval.c
@@ -1018,7 +1018,7 @@
     rb_funcallv_with_cc(struct rb_call_data *cd, VALUE recv, ID mid, int argc, const
     argc,
     RB_NO_KEYWORDS,
     ),
     cd,
     &(struct rb_call_data) { *cc, *ci },
     argv
 );

#12 - 03/24/2022 12:21 AM - shyouhei (Shyouhei Urabe)

@palkan (Vladimir Dementyev) thank you for the reproducer. That helps us a lot.

Also thank you @wanabe (_ wanabe). I'm +1: for the "alternative" approach in #note-11.

#13 - 03/25/2022 08:00 AM - wanabe (_ wanabe)

- Status changed from Open to Closed

Thank you for confirming.
This is not reproduced in 3.0 or later, and it would be better to close this issue, leaving the decision to the backport maintainer. I'm going to close it.

Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>ruby_segfault_log.txt</td>
<td>86.2 KB</td>
<td>12/09/2020</td>
<td>lionelperrin (Lionel Perrin)</td>
</tr>
<tr>
<td>segfault.rb</td>
<td>1.03 KB</td>
<td>02/25/2022</td>
<td>tom_dalling (Tom Dalling)</td>
</tr>
<tr>
<td>log.txt</td>
<td>40.3 KB</td>
<td>02/25/2022</td>
<td>tom_dalling (Tom Dalling)</td>
</tr>
<tr>
<td>repro.zip</td>
<td>17.5 KB</td>
<td>02/25/2022</td>
<td>tom_dalling (Tom Dalling)</td>
</tr>
</tbody>
</table>