Ruby - Bug #18945

node_id is not initialized but it is used leading to UB

07/28/2022 10:52 PM - graywolf (Gray Wolf)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	master	**Backport:**	2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN

Description

I am trying to compile ruby in a reproducible way, but libruby always has a different hash. After two days of digging I've reached the conclusion that node_id is used when not initialized, leading to an undefined behaviour. In my case it manifested as a garbage value of node_id (for some nodes only) leading to this:

```
$ ./miniruby -e 'puts RubyVM.enum_for(:each_builtin).to_a.select { |k, v| k == "array" }[0][1].to_
binary' | sha256sum
436c0866ec18ac217cb220ee8c40c8d1b495d275cad85800bd151e091019586c  -
$ ./miniruby -e 'puts RubyVM.enum_for(:each_builtin).to_a.select { |k, v| k == "array" }[0][1].to_
binary' | sha256sum
3ac523233f8360aa355fa41d8b5b71da94732c8a6d5267b1408bdcf1f847bf6a  -
```

Seems to be sensitive to a build environment, I suspect gcc version. I've tried in ubuntu 21.04 (good) and in ubuntu 21.10 (bad).

I have two possible patches:

```
diff --git a/node.c b/node.c
index a10d5122c3..483e7fa8fb 100644
--- a/node.c
+++ b/node.c
@@ -1138,6 +1138,7 @@ rb_node_init(NODE *n, enum node_type type, VALUE a0, VALUE a1, VALUE a2)
     n->nd_loc.beg_pos.column = 0;
     n->nd_loc.end_pos.lineno = 0;
     n->nd_loc.end_pos.column = 0;
+    n->node_id = -1;
 }

 typedef struct node_buffer_elem_struct {
```

I'm not sure about the -1 here and if it has any special meaning or not. Second one is

```
diff --git a/compile.c b/compile.c
index 6a9ed2a5d0..0108eccc0a 100644
--- a/compile.c
+++ b/compile.c
@@ -8012,7 +8012,7 @@ compile_builtin_mandatory_only_method(rb_iseq_t *iseq, const NODE *node,
const N
     struct rb_args_info args = {
         .pre_args_num = ISEQ_BODY(iseq)->param.lead_num,
     };
-    NODE args_node;
+    NODE args_node = {0};
     rb_node_init(&args_node, NODE_ARGS, 0, 0, (VALUE)&args);

     // local table without non-mandatory parameters
@@ -8034,7 +8034,7 @@ compile_builtin_mandatory_only_method(rb_iseq_t *iseq, const NODE *node,
const N
         tbl->ids[i] = ISEQ_BODY(iseq)->local_table[i + skip_local_size];
```

```
     }

-    NODE scope_node;
+    NODE scope_node = {0};
     rb_node_init(&scope_node, NODE_SCOPE, (VALUE)tbl, (VALUE)mandatory_node(iseq, node), (VALU
E)&args_node);

     rb_ast_body_t ast = {
```

Both are sufficient to fix the issue at hand. I think both of them should be applied (since the second one correctly initializes the whole structure and the first one should cover other call places as well).

Please let me know what you think about this. Thank you.

History

#1 - 07/29/2022 02:34 PM - mame (Yusuke Endoh)

Good catch! Your first patch looks great to me. Can you send a PR to https://github.com/ruby/ruby/? Thank you!

#2 - 07/29/2022 06:14 PM - graywolf (Gray Wolf)

I hope I did everything correctly: https://github.com/ruby/ruby/pull/6202

Sidenote: It's bit shame one has to read and understand this https://docs.github.com/en/site-policy/github-terms/github-terms-of-service fairly long document before sending a patch.

#3 - 08/01/2022 01:41 AM - mame (Yusuke Endoh)

Thank you very much, merged!

> Sidenote: It's bit shame one has to read and understand this https://docs.github.com/en/site-policy/github-terms/github-terms-of-service fairly long document before sending a patch.

Thank you for your time. I should have made a PR on your behalf.

#4 - 08/24/2022 04:58 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed

Pull requested merged at c69ad738dc7c713df547a51607917ca78df6b793