Ruby master - Feature #5072

Avoid inadvertent symbol creation in reflection methods

07/22/2011 08:02 AM - jeremyevans0 (Jeremy Evans)

Status:	Closed
Priority:	Normal
Assignee:	nobu (Nobuyoshi Nakada)
Target version:	

Description

I recently discovered a denial of service vulnerability in ActiveRecord's mass assignment methods related to the insecure use of ruby's reflection methods (e.g. respond_to?). Because these methods take strings and automatically create symbols from them, they are not safe to call with a string coming from the user. Because they create the symbol internally, they look safe, but if you pass user-created strings to these methods, you open yourself up to denial of service through memory exhaustion (see http://sequel.heroku.com/2011/07/16/dangerous-reflection/).

This could be fixed using a fairly simple observation, which is that if you do:

respond_to?("foo")

and "foo" is not already in the symbol table, no method named "foo" can exist. So this code provides a patch that changes the reflection methods to return false immediately if given a string which doesn't already exist in the symbol table. There should be no performance impact from this, since the symbol table lookup has to be done anyway.

I'm also adding an earlier patch I wrote that adds String#interned?, for checking if a string is already interned. There was an internal method for this added in r10932, but it must have been removed while the prototype was left in intern.h. String#interned? allows a user to check if a string is already in the symbol table, and can be used by user code to ensure that symbols are not created inadvertently.

Related issues:

Related to Ruby master - Feature #5112: Remove inadvertent symbol creation fr...	**Closed**	**07/28/2011**
Follows Ruby master - Feature #5079: More removal of inadvertent symbol creation	**Closed**	**07/23/2011**
Follows Ruby master - Feature #5089: Even More Inadvertent Symbol Removal, An...	**Closed**	**07/24/2011**

Associated revisions

Revision 34918aa8 - 07/22/2011 12:06 PM - nobu (Nobuyoshi Nakada)

- object.c (rb_mod_{const,cvar}_defined, rb_obj_ivar_defined): avoid inadvertent symbol creation in reflection methods. based on a patch by Jeremy Evans at [ruby-core:38367]. [Feature #5072]
- vm_method.c (rb_mod_method_defined) (rb_mod_{public,private,protected}_method_defined) (obj_respond_to): ditto.
- parse.y (rb_check_id): new function returns already interned ID or 0.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@32621 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 32621 - 07/22/2011 12:06 PM - nobu (Nobuyoshi Nakada)

- object.c (rb_mod_{const,cvar}_defined, rb_obj_ivar_defined): avoid inadvertent symbol creation in reflection methods. based on a patch by Jeremy Evans at [ruby-core:38367]. [Feature #5072]
- vm_method.c (rb_mod_method_defined) (rb_mod_{public,private,protected}_method_defined) (obj_respond_to): ditto.
- parse.y (rb_check_id): new function returns already interned ID or 0.

Revision 32621 - 07/22/2011 12:06 PM - nobu (Nobuyoshi Nakada)

- object.c (rb_mod_{const,cvar}_defined, rb_obj_ivar_defined): avoid inadvertent symbol creation in reflection methods. based on a patch by Jeremy Evans at [ruby-core:38367]. [Feature #5072]
- vm_method.c (rb_mod_method_defined) (rb_mod_{public,private,protected}_method_defined) (obj_respond_to): ditto.
- parse.y (rb_check_id): new function returns already interned ID or 0.

Revision 32621 - 07/22/2011 12:06 PM - nobu (Nobuyoshi Nakada)

- object.c (rb_mod_{const,cvar}_defined, rb_obj_ivar_defined): avoid inadvertent symbol creation in reflection methods. based on a patch by Jeremy Evans at [ruby-core:38367]. [Feature #5072]
- vm_method.c (rb_mod_method_defined) (rb_mod_{public,private,protected}_method_defined) (obj_respond_to): ditto.

- parse.y (rb_check_id): new function returns already interned ID or 0.

Revision 32621 - 07/22/2011 12:06 PM - nobu (Nobuyoshi Nakada)

- object.c (rb_mod_{const,cvar}_defined, rb_obj_ivar_defined): avoid inadvertent symbol creation in reflection methods. based on a patch by Jeremy Evans at [ruby-core:38367]. [Feature #5072]
- vm_method.c (rb_mod_method_defined) (rb_mod_{public,private,protected}_method_defined) (obj_respond_to): ditto.
- parse.y (rb_check_id): new function returns already interned ID or 0.

Revision 32621 - 07/22/2011 12:06 PM - nobu (Nobuyoshi Nakada)

- object.c (rb_mod_{const,cvar}_defined, rb_obj_ivar_defined): avoid inadvertent symbol creation in reflection methods. based on a patch by Jeremy Evans at [ruby-core:38367]. [Feature #5072]
- vm_method.c (rb_mod_method_defined) (rb_mod_{public,private,protected}_method_defined) (obj_respond_to): ditto.
- parse.y (rb_check_id): new function returns already interned ID or 0.

Revision 32621 - 07/22/2011 12:06 PM - nobu (Nobuyoshi Nakada)

- object.c (rb_mod_{const,cvar}_defined, rb_obj_ivar_defined): avoid inadvertent symbol creation in reflection methods. based on a patch by Jeremy Evans at [ruby-core:38367]. [Feature #5072]
- vm_method.c (rb_mod_method_defined) (rb_mod_{public,private,protected}_method_defined) (obj_respond_to): ditto.
- parse.y (rb_check_id): new function returns already interned ID or 0.

Revision 298349d0 - 07/26/2011 04:05 PM - nobu (Nobuyoshi Nakada)

- vm_method.c (obj_respond_to): fix the respond_to_missing? override case. based on the patch by Jeremy Evans at [ruby-core:38417]. [Feature #5072]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@32685 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 32685 - 07/26/2011 04:05 PM - nobu (Nobuyoshi Nakada)

- vm_method.c (obj_respond_to): fix the respond_to_missing? override case. based on the patch by Jeremy Evans at [ruby-core:38417]. [Feature #5072]

Revision 32685 - 07/26/2011 04:05 PM - nobu (Nobuyoshi Nakada)

- vm_method.c (obj_respond_to): fix the respond_to_missing? override case. based on the patch by Jeremy Evans at [ruby-core:38417]. [Feature #5072]

Revision 32685 - 07/26/2011 04:05 PM - nobu (Nobuyoshi Nakada)

- vm_method.c (obj_respond_to): fix the respond_to_missing? override case. based on the patch by Jeremy Evans at [ruby-core:38417]. [Feature #5072]

Revision 32685 - 07/26/2011 04:05 PM - nobu (Nobuyoshi Nakada)

- vm_method.c (obj_respond_to): fix the respond_to_missing? override case. based on the patch by Jeremy Evans at [ruby-core:38417]. [Feature #5072]

Revision 32685 - 07/26/2011 04:05 PM - nobu (Nobuyoshi Nakada)

- vm_method.c (obj_respond_to): fix the respond_to_missing? override case. based on the patch by Jeremy Evans at [ruby-core:38417]. [Feature #5072]

Revision 32685 - 07/26/2011 04:05 PM - nobu (Nobuyoshi Nakada)

- vm_method.c (obj_respond_to): fix the respond_to_missing? override case. based on the patch by Jeremy Evans at [ruby-core:38417]. [Feature #5072]

History

#1 - 07/22/2011 02:55 PM - matz (Yukihiro Matsumoto)

Quite nice idea! I am not going to make this specified behavior among implementations, but as an optimization it's great. I'd like to merge it, for 1.9.3, if the maintainer allows.

matz.

#2 - 07/22/2011 03:33 PM - jeremyevans0 (Jeremy Evans)

Great. I'll work on a patch that fixes the backwards compatibility issues mentioned in the 0002 patch (e.g. that instance_variable_defined?("foo") does not raise NameError).

Would you like this optimization ported to (class|instance)variable_get, remove(class|instance)variable, remove_const, and (public |)(instance_|)method? I don't think those methods are as likely to be used with user-provided strings, but it may be desired to handle those for consistency. I don't think send or const_get can use the optimization, since if the method or constant does not exist, they need to provide the symbol to method_missing or const_missing.

#3 - 07/22/2011 03:53 PM - nobu (Nobuyoshi Nakada)

Hi,

At Fri, 22 Jul 2011 14:55:40 +0900,
Yukihiro Matsumoto wrote in [ruby-core:38386]:

> Quite nice idea! I am not going to make this specified
> behavior among implementations, but as an optimization it's
> great. I'd like to merge it, for 1.9.3, if the maintainer allows.

I'm wondering who/when/why removed rb_sym_interned_p(). I was going to make a simillar change formerly, but have forgotten it long time.

--
Nobu Nakada

#4 - 07/22/2011 09:06 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r32621.
Jeremy, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- object.c (rb_mod_{const,cvar}_defined, rb_obj_ivar_defined): avoid inadvertent symbol creation in reflection methods. based on a patch by Jeremy Evans at [ruby-core:38367]. [Feature #5072]
- vm_method.c (rb_mod_method_defined) (rb_mod_{public,private,protected}_method_defined) (obj_respond_to): ditto.
- parse.y (rb_check_id): new function returns already interned ID or 0.

#5 - 07/23/2011 09:37 AM - mame (Yusuke Endoh)

- Status changed from Closed to Assigned

- Assignee set to nobu (Nobuyoshi Nakada)

Hello,

2011/7/22 Jeremy Evans merch-redmine@jeremyevans.net:

> This could be fixed using a fairly simple observation, which is that if you do:
>
> respond_to?("foo")
>
> and "foo" is not already in the symbol table, no method named "foo" can exist.

Nobu seemed to commit your patch, but unfortunately, the observation is false because of respond_to_missing?.

```
class Foo
def respond_to_missing?(mhd, flag)
true
end
end

p Foo.new.respond_to?("foo")  #=> true in 1.9.2, false in trunk
```

In this case, it will be ok to check if respond_to_missing? is defined before id check. But whether this feature should be included in 1.9.3 or not, should be discussed more carefully, I think.

--
Yusuke Endoh mame@tsg.ne.jp

#6 - 07/23/2011 02:36 PM - jeremyevans0 (Jeremy Evans)

- File 0002-Fix-handling-of-respond_to_missing-after-r32621.patch added

I guess the observation does fail for respond_to?, but it should hold for the other methods (both the ones in here #5072 and the ones in #5079). I've attached a patch here to fix the respond_to_missing? override case, but of course that would remove the protection for any class that overrides respond_to_missing?. One way to workaround this is to relax the spec so that respond_to_missing is allowed to receive a string instead of a symbol as the first argument.

#7 - 07/27/2011 01:05 AM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Closed

This issue was solved with changeset r32685.
Jeremy, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- vm_method.c (obj_respond_to): fix the respond_to_missing? override case. based on the patch by Jeremy Evans at [ruby-core:38417]. [Feature #5072]

Files

0001-Add-String-interned-for-checking-if-a-string-is-alre.patch	2.59 KB	07/22/2011	jeremyevans0 (Jeremy Evans)
0002-Avoid-inadvertent-symbol-creation-in-reflection-meth.patch	6.31 KB	07/22/2011	jeremyevans0 (Jeremy Evans)
0002-Fix-handling-of-respond_to_missing-after-r32621.patch	1.93 KB	07/23/2011	jeremyevans0 (Jeremy Evans)