Redmine - Defect #39862

Attachments functionality for (custom) plugins broken since fix for CVE-2022-44030

2023-12-16 22:16 - Naha Sapimapethilon

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Marius BĂLTEANU	**% Done:**	0%
Category:	Plugin API	**Estimated time:**	0.00 hour
Target version:	5.0.8		
Resolution:	Fixed	**Affected version:**	5.0.0

Description

I notice this in current 5.1-stable branch, but should be all the way back to defect #37772 if I tracked it right.

The problem is with the new constraints for some attachments routes, *when used by a plugin*. My plugin makes use of acts_as_attachable in its model and :partial=>'attachments/form' in its view, just like described here.

This is now broken with an error from app/helpers/attachments_helper.rb:23:in `container_attachments_edit_path':

```
No route matches {:action=>"edit_all", :controller=>"attachments", :id=>"138026",
 :object_id=>138026, :object_type=>"myplugin",
 :project_id=>"1"}, possible unmatched constraints: [:object_type]
```

its actually coming from this block in config/routes.rb (finding that took me a while):

```
constraints object_type: /(issues|versions|news|messages|wiki_pages|projects|documents|journals)/
do
   get 'attachments/:object_type/:object_id/edit', :to => 'attachments#edit_all', :as =>
:object_attachments_edit
   patch 'attachments/:object_type/:object_id', :to => 'attachments#update_all', :as =>
:object_attachments
   get 'attachments/:object_type/:object_id/download', :to => 'attachments#download_all', :as =>
:object_attachments_download
end
```

the list of constraints on object_type needs myplugin in it, so it gets permitted to use these routes.

Since plugin routes get loaded at the very end of config/routes.rb I can't just overwrite/redefine since it already exists at the time I get loaded. Also I spot no functionality in the routing code of rails that allows modification from within an included routes file or at runtime via Rails.application.routes.routes... looks all read-only.

My workaround so far is to modify the release by

```
sed -i config/routes.rb -e '/constraints object_type:/ s/documents|journals/documents|journals|myp
lugin/'
```

right before starting up Redmine.

I think a proper solution would be to have this list be expandable somehow, perhaps via myplugin/init.rb?

Am a little lost here solving it on my own.

Related issues:

Precedes Redmine - Feature #39948: Add Redmine::Plugin proxy method for Redmi...	**Closed**

Associated revisions

Revision 22551 - 2023-12-22 03:08 - Marius BĂLTEANU

Fixes attachments functionality for (custom) plugins broken since fix for CVE-2022-44030 by adding a dynamic routing constraint which can be modified by plugins (#39862).

Patch by @jkraemer.

Revision 22552 - 2023-12-22 03:13 - Marius BĂLTEANU

Merge r22551 from trunk to 5.1-stable (#39862).

Revision 22553 - 2023-12-22 03:25 - Marius BĂLTEANU

Backport r22551 from trunk to 5.0-stable (#39862).

Revision 22554 - 2023-12-22 03:33 - Marius BĂLTEANU

Add missing change for #39862 to 5.1-stable.

Revision 22566 - 2023-12-27 07:07 - Go MAEDA

Fix RuboCop offense Style/HashSyntax (Don't mix styles in the same hash) introduced in r22551 (#39862).

Revision 22569 - 2023-12-27 16:40 - Marius BĂLTEANU

Fixes exception in acts_as_attachable when running on Ruby 2.7 (#39862).

Patch by @maeda.

Revision 22570 - 2023-12-27 16:43 - Marius BĂLTEANU

Merge r22569 from trunk to 5.1-stable (#39862).

Revision 22571 - 2023-12-27 16:43 - Marius BĂLTEANU

Merge r22569 from trunk to 5.0-stable (#39862).

Revision 22747 - 2024-02-27 08:06 - Marius BĂLTEANU

Introduces Redmine::Plugin#attachment_object_type to provide better API for registering plugin models having attachments (#39948, #39862).

Patch by Jens Krämer (@jkraemer).

History

#1 - 2023-12-18 06:40 - Jens Krämer

- File 0001-dynamic-object_type-routing-constraint-39862.patch added

One way to fix this would be with a dynamic routing constraint which can be modified by plugins as in the attached patch

#2 - 2023-12-18 20:10 - Naha Sapimapethilon

that patch did not apply to my 5.1-stable branch. However, I finagled the changes into config/routes.rb and acts_as_attachable.rb and that works.

also I am fine with myplugin/init.rb having

```
Redmine::Plugin.register :myplugin do
  ...
  Redmine::Acts::Attachable::ObjectTypeConstraint.register_object_type(Expense.name.underscore.pluralize)
  ...
end
```

thanks!

#3 - 2023-12-19 21:07 - Mischa The Evil

- Target version set to 5.0.8

- Affected version changed from 5.1.1 to 5.0.0

#4 - 2023-12-22 03:35 - Marius BĂLTEANU

- Status changed from New to Resolved

- Assignee set to Marius BĂLTEANU

- Resolution set to Fixed

Committed the fix on trunk and backported it to stable branches. On 5.0-stable, the fix is without the test because the plugins routing test was

introduced in 5.1.

#5 - 2023-12-22 06:40 - Alexander Meindl

Hi,

this change breaks all redmineup plugins with Redmine 5.1-stable branch (I suppose same with 5.0-stable branch). They use redmine_crm gem in all plugins with this compatibility issue.

```
bundle exec rake db:migrate
rake aborted!
NoMethodError: undefined method `[]' for Redmine::Acts::Attachable::ObjectTypeConstraint:Class

          options[:object_type] = /.+/ if options[:object_type] && options[:object_type].is_a?(Regexp)
                                                  ^^^^^^^^^^^^^^
/srv/redmine/.local/share/gem/ruby/3.1.0/gems/redmine_crm-0.0.62/lib/redmine_crm/compatibility/routing_mapper_
patch.rb:15:in `constraints_with_redmine_crm'
/srv/redmine/redmine/config/routes.rb:322:in `block in <top (required)>'
/srv/redmine/.local/share/gem/ruby/3.1.0/gems/actionpack-6.1.7.6/lib/action_dispatch/routing/route_set.rb:427:
in `instance_exec'
/srv/redmine/.local/share/gem/ruby/3.1.0/gems/actionpack-6.1.7.6/lib/action_dispatch/routing/route_set.rb:427:
in `eval_block'
/srv/redmine/.local/share/gem/ruby/3.1.0/gems/actionpack-6.1.7.6/lib/action_dispatch/routing/route_set.rb:409:
in `draw'
/srv/redmine/redmine/config/routes.rb:20:in `<top (required)>'
```

I am not sure this change should go to stable branch, if it breaks existing plugins.

#6 - 2023-12-22 09:38 - Marius BĂLTEANU

Alexander Meindl wrote in #note-5:

> Hi,
>
> this change breaks all redmineup plugins with Redmine 5.1-stable branch (I suppose same with 5.0-stable branch). They use redmine_crm gem in all plugins with this compatibility issue.
>
> [...]
>
> I am not sure this change should go to stable branch, if it breaks existing plugins.

In this case, yes, I think we cannot deliver the change in maintenance releases. Jens, Mischa, what do you think?

#7 - 2023-12-24 13:28 - ashraf alzyoud

All redmineup plugins broken after update

#9 - 2023-12-24 15:24 - Holger Just

Well, the patch in the redmine_crm gem (here in version 0.0.62 licensed under GPL 2) is as follows:

```
# lib/redmine_crm/compatibility/routing_mapper_patch.rb

module RedmineCrm
  module Patches
    module RoutingMapperPatch
      def self.included(base)
        base.send(:include, InstanceMethods)

        base.class_eval do
          alias_method :constraints_without_redmine_crm, :constraints
          alias_method :constraints, :constraints_with_redmine_crm
        end
      end

      module InstanceMethods
        def constraints_with_redmine_crm(options = {}, &block)
          options[:object_type] = /.+/ if options[:object_type] && options[:object_type].is_a?(Regexp)
          constraints_without_redmine_crm(options, &block)
        end
      end
    end
  end
end
```

```
unless ActionDispatch::Routing::Mapper.included_modules.include?(RedmineCrm::Patches::RoutingMapperPatch)
  ActionDispatch::Routing::Mapper.send(:include, RedmineCrm::Patches::RoutingMapperPatch)
end
```

This patch thus circumvents/removes one of the security fixes we have introduced in #37772 for CVE-2022-44030. The patch in the plugin gem thus reduces the security of the entire Redmine the plugins are installed to.

As they do not use a supported Redmine extension API but effectively change code on assumptions they made at the time they created the patch, this was always prone to breakage in any release.

Accordingly, rather than postpone the patch in this issue here, we should roll it out in order to give the plugin authors a supported extension API so that they can fix their plugin in a way which does not impact Redmine's security and is not prone to arbitrary breakage anymore. The underlying issue must be fixed by the plugin authors in any case.

#10 - 2023-12-24 17:22 - Marius BĂLTEANU

- Status changed from Resolved to Closed

Thanks Holger, I agree with your arguments. Closing this.

#11 - 2023-12-26 14:47 - Kirill Bezrukov (RedmineUP)

Thank you Holger.

We just updated our gem redmine_crm v0.0.63 with routes fix

Holger Just wrote in #note-9:

> Well, the patch in the redmine_crm gem (here in version 0.0.62 licensed under GPL 2) is as follows:

#12 - 2023-12-27 07:55 - Mischa The Evil

- Precedes Feature #39948: Add Redmine::Plugin proxy method for Redmine::Acts::Attachable::ObjectTypeConstraint.register_object_type added

#13 - 2023-12-27 08:51 - Go MAEDA

- Status changed from Closed to Reopened

The method Redmine::Acts::Attachable::ObjectTypeConstraint.param_expression, introduced in r22551, is causing an exception when run with Ruby 2.7. This is due to the Set class in Ruby 2.7 not having a join method.

You can observe the error by running bin/rails test test/integration/attachments_test.rb.

```
Failure:
AttachmentsTest#test_download_all_with_wrong_container_type [/Users/maeda/redmines/trunk/test/integration/atta
chments_test.rb:242]:
Expected response to be a <404: missing>, but was a <500: Internal Server Error>.
Expected: 404
  Actual: 500
```

This can be fixed by converting the Set object to an array before applying the join method. The following patch should resolve the issue:

```
diff --git a/lib/plugins/acts_as_attachable/lib/acts_as_attachable.rb b/lib/plugins/acts_as_attachable/lib/act
s_as_attachable.rb
index 9c09a7870..43efe8cd3 100644
--- a/lib/plugins/acts_as_attachable/lib/acts_as_attachable.rb
+++ b/lib/plugins/acts_as_attachable/lib/acts_as_attachable.rb
@@ -39,7 +39,7 @@ module Redmine
           end

           def param_expression
-            @param_expression ||= Regexp.new("^(#{object_types.join("|")})$")
+            @param_expression ||= Regexp.new("^(#{object_types.to_a.join("|")})$")
           end
         end
       end
```

#14 - 2023-12-27 16:43 - Marius BĂLTEANU

- Status changed from Reopened to Closed

Committed and merged to stable branches, thanks!

#15 - 2024-03-15 18:14 - f0x Autumn

Alexander Meindl wrote in #note-5:

> Hi,
>
> this change breaks all redmineup plugins with Redmine 5.1-stable branch (I suppose same with 5.0-stable branch). They use redmine_crm gem in all plugins with this compatibility issue.
>
> [...]
>
> I am not sure this change should go to stable branch, if it breaks existing plugins.

Marius BĂLTEANU wrote in #note-14:

> Committed and merged to stable branches, thanks!

It was in Redmine 5.0.8

But I got the same error in the 5.1.2 (#40409)

Files

0001-dynamic-object_type-routing-constraint-39862.patch	3.83 KB	2023-12-18	Jens Krämer