Redmine - Defect #36958
Crafted input breaks CommonMark Markdown formatter

2022-04-14 12:05 - Go MAEDA

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Marius BALTEANU	**% Done:**	0%
Category:	Text formatting	**Estimated time:**	0.00 hour
Target version:	5.0.1		
Resolution:	Fixed	**Affected version:**	5.0.0

Description

If you create an issue or a Wiki page contains specific data, the CommonMark Markdown formatter raises an exception when rendering the object. Malicious users can use this bug for DoS attacks.

Steps to reproduce:

1. Set the text formatting to "CommonMark Markdown"
2. Create an issue that contains a string http://example.com/foo#bar#
3. Access the newly created issue. You will see "Internal Error"

```
ActionView::Template::Error (bad URI(is not URI?): "http://example.com/foo#bar#"):
  88:
  89:   <p><strong><%=l(:field_description)%></strong></p>
  90:   <div class="wiki">
  91:   <%= textilizable @issue, :description, :attachments => @issue.attachments %>
  92:   </div>
  93: </div>
  94: <% end %>

lib/redmine/wiki_formatting/common_mark/external_links_filter.rb:34:in `block in call'
lib/redmine/wiki_formatting/common_mark/external_links_filter.rb:29:in `call'
lib/redmine/wiki_formatting/common_mark/formatter.rb:66:in `to_html'
lib/redmine/wiki_formatting.rb:96:in `to_html'
app/helpers/application_helper.rb:868:in `textilizable'
app/views/issues/show.html.erb:91
app/controllers/issues_controller.rb:118:in `block (2 levels) in show'
app/controllers/issues_controller.rb:110:in `show'
lib/redmine/sudo_mode.rb:61:in `sudo_mode'
```

Associated revisions

Revision 21558 - 2022-05-03 18:50 - Marius BALTEANU

Fix rendering invalid URI fails with exception in CommonMark Markdown (#36958).

Patch by Holger Just.

Revision 21559 - 2022-05-03 18:50 - Marius BALTEANU

Add a test for #36958.

Patch by Go MAEDA.

Revision 21570 - 2022-05-11 22:31 - Marius BALTEANU

Merged r21558 and r21559 to 5.0-stable (#36958).

History

#1 - 2022-04-14 14:50 - Go MAEDA

- File 36958.patch added

the attached patch fixes the issue.

#2 - 2022-04-19 10:53 - Go MAEDA

- Target version set to 5.0.1

Setting the target version to 5.0.1.

#3 - 2022-04-19 12:41 - Holger Just

I can confirm this issue. However, I don't believe this can be used as an actual DoS of the application itself. This issue might be abused however to cause errors on many pages by including the invalid URI in issues / journals / wiki pages, ...

With that being said, I think we could also just use the following code instead of rescuing the specific exception. If you are confident that URI::InvalidURIError is the only exception being thrown for invalid URIs, then your code is fine too.

```
scheme = URI.parse(url).scheme rescue nil
next if scheme.blank?
```

#4 - 2022-04-27 23:22 - Marius BALTEANU

- Assignee set to Go MAEDA

Go MAEDA, can you handle this?

#5 - 2022-05-03 18:52 - Marius BALTEANU

- Status changed from New to Resolved
- Assignee changed from Go MAEDA to Marius BALTEANU
- Resolution set to Fixed

Patches committed with the recommendation from Holger. I think it's safe to not consider this a security issue.

#6 - 2022-05-11 22:32 - Marius BALTEANU

- Status changed from Resolved to Closed

#7 - 2022-05-16 23:54 - Marius BALTEANU

- Project changed from Security to Redmine

#8 - 2022-05-16 23:54 - Marius BALTEANU

- Category set to Text formatting

Files

36958.patch	1.55 KB	2022-04-14	Go MAEDA