Redmine - Feature #29405

Support header Content Security Policy

2018-08-18 14:31 - Ludovic Andrieux

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Security	**Estimated time:**	0.00 hour
Target version:			
Resolution:			

Description

Hi,

According Google, this a basic Content Security Policy.

```
Content-Security-Policy: default-src https:; script-src https: 'unsafe-inline'; style-src https: 'unsafe-inline'
```

Redmine crash with it because there is some call to eval in javascript in some pages.

Regards,
Ludovic

History

#1 - 2019-04-02 12:23 - cam lafit

Hello

A workaround is to enable all via a **config/initializers/csp.rb**

```
Rails.application.config.content_security_policy do |policy|
  policy.default_src "*",  :data, :blob, "'unsafe-inline'", "'unsafe-eval'"
  policy.font_src    "*",  :data, :blob, "'unsafe-inline'", "'unsafe-eval'"
  policy.img_src     "*",  :data, :blob, "'unsafe-inline'", "'unsafe-eval'"
  policy.object_src  "*",  :data, :blob, "'unsafe-inline'", "'unsafe-eval'"
  policy.script_src  "*",  :data, :blob, "'unsafe-inline'", "'unsafe-eval'"
  policy.style_src   "*",  :data, :blob, "'unsafe-inline'", "'unsafe-eval'"

  # Specify URI for violation reports
  # policy.report_uri "/csp-violation-report-endpoint"
end

#Rails.application.config.content_security_policy_report_only = true
```

#2 - 2023-07-03 10:45 - Popa Marius

Any news on this patch ?

#3 - 2023-07-03 16:48 - Popa Marius

- File clipboard-202307031747-pojyg.png added

clipboard-202307031747-pojyg.png

#4 - 2023-07-03 16:49 - Popa Marius

- File clipboard-202307031749-tbv3n.png added

clipboard-202307031749-tbv3n.png

#5 - 2023-07-04 16:59 - Popa Marius

```
Changed policy.object_src  "none"
```

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/object-src

#6 - 2023-07-04 21:25 - Popa Marius

changed

```
policy.font_src    :self, :https, :data
```

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/font-src

#7 - 2023-07-04 21:46 - Popa Marius

changed

```
 policy.style_src   :self, :https , :unsafe_inline
```

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/style-sr

#8 - 2023-07-05 09:48 - Popa Marius

- File clipboard-202307051048-oevyb.png added

we need :unsafe_inline otherwise

clipboard-202307051048-oevyb.png

#9 - 2023-07-10 11:24 - Popa Marius

define policy.frame_ancestors :none https://content-security-policy.com/frame-ancestors/

#10 - 2023-07-25 12:01 - Jérôme Gallot

+1 for the feature.

:unsafe_inline must not be used, not secured so there's a lot to do in order to make redmine works like a charm with CSP and i don't speak of plugins ...

Interesting subject, a bit tricky

#11 - 2024-03-11 09:01 - Paul Takemura

Hello, I am using the Bitnami distribution of Redmine 5.0.6-3-r06 on Debian 11 (Bullseye). The Bitnami provided changelog can be seen at https://bitnami.com/stack/redmine/amidebian-x64-hvm-ebs-nami/changelog.txt

I believe that the lack of a proper CSP header is preventing the display of PDF attachments from within the Safari browser (Safari 17.3.1 on macOS Sonoma 14.3.1).

The default headers_module is as follows:

```
<IfModule headers_module>
    #
    # Avoid passing HTTP_PROXY environment to CGI's on this or any proxied
    # backend servers which have lingering "httpoxy" defects.
    # 'Proxy' request header is undefined by the IETF, not listed by IANA
    #
    RequestHeader unset Proxy early
</IfModule>
```

With the above configuration, PDF file attachments are not displayed.

I revised the block by adding a CSP header:

```
<IfModule headers_module>
    #
    # Avoid passing HTTP_PROXY environment to CGI's on this or any proxied
    # backend servers which have lingering "httpoxy" defects.
    # 'Proxy' request header is undefined by the IETF, not listed by IANA
    #
    RequestHeader unset Proxy early
    # 2024-03-10 Paul added CSP
    Header set Content-Security-Policy "default-src 'self'; script-src 'self' 'unsafe-inline'; style-src 'self
```

```
' 'unsafe-inline'"
</IfModule>
```

After making this addition, the PDF is not displayed on the first try. But after returning to the download screen and clicking on the link again, the PDF is displayed perfectly. Are there any ideas on how to remedy this?

#12 - 2024-03-18 11:49 - Paul Takemura

I think I solved this (at least for the problem I raised), but I really should have the change vetted by the Redmine developers, because it probably weakens security.

I removed the word “sandbox” from ~/stack/redmine/app/controllers/attachments_controller.rb:

headers['content-security-policy'] = "default-src 'none'; style-src 'unsafe-inline'; sandbox"

I also reverted the change I had made to Apache's https.conf back to the original form. (Edit: You may in fact need the CSP in Apache too. Caching and other factors may be flustering my tests.)

Now, in Safari, PDF files are displayed within the browser from the very first try.

To do a complete test, restart the whole stack, not just Apache, and refresh the page to which you are adding the PDF file.

#13 - 2024-03-20 10:53 - L S

Jérôme Gallot wrote in #note-10:

> +1 for the feature.
>
> :unsafe_inline must not be used, not secured so there's a lot to do in order to make redmine works like a charm with CSP and i don't speak of plugins ...
>
> Interesting subject, a bit tricky

We've doing some tests due to Rails having integrated CSP features. Here you have a roadmap should somebody can be assigned:

The better way to carry implement CSP improvement would be to use nonces in order to not require recalculation of CSS and JS hashes for every change made. Some changes to code will be required to have it running:

- Creating style classes to replace inline styles applied everywhere. For example, the projects drop-down lists projects indented depending on their levels, but sets indentation inline by multiplying 16px by the project level; this should be changed to make use of classes like:

  ```
  indented-1-level {padding-left:16px;}
  ```

- Adding a CSP configuration in intializers:

  ```
  # redmine/config/initializers/csp.rb

  Rails.application.configure do
    config.content_security_policy do |policy|
      policy.default_src :self, :https
      policy.font_src    :self, :https, :data
      policy.img_src     :self, :https, :data
      policy.object_src  :none
      policy.script_src  :self, :https
      policy.style_src   :self, :https
      policy.base_uri    :self
    end

    # Generate session nonces for permitted inline scripts and styles.
    config.content_security_policy_nonce_generator = ->(request) { SecureRandom.base64(24) }
  # request.session.id.to_s is also suggested by authors
    config.content_security_policy_nonce_directives = %w(script_src style-src)

    # policy.report_uri should/could also be set
  end
  ```

- Replacing all JavaScript tags in ERB files:

  ```
  <%= javascript_tag do -%>
  <%= javascript_include_tag "script" %>
  ```

  to

```
<%= javascript_tag nonce: content_security_policy_nonce do -%>
<%= javascript_include_tag "script", nonce: content_security_policy_nonce %>
```

We tried by setting content_security_policy_nonce_directives just for scripts and, temporarily, :unsafe-inline to style_src, but, at the moment nonce was used, inline styles weren't allowed...

Files

2018-08-18_142722.png	76.2 KB	2018-08-18	Ludovic Andrieux
clipboard-202307031747-pojyg.png	62.8 KB	2023-07-03	Popa Marius
clipboard-202307031749-tbv3n.png	14.8 KB	2023-07-03	Popa Marius
clipboard-202307051048-oevyb.png	30.5 KB	2023-07-05	Popa Marius