Redmine - Feature #29660

Add Referrer-Policy header to prevent browsers from sending private data to external sites

2018-09-22 11:38 - Ebrahim Mohammadi

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

Description

Currently Redmine sets no HTTP Referrer configuration, so full URL of Redmine page containing a link to an external website is sent as HTTP Referrer header to the target website. This could be a source of securiy and/or privacy issues. See https://moz.com/blog/meta-referrer-tag for more information.

I suggest setting the referrer meta tag to "origin-when-crossorigin" as the default value. It would be as easy as adding this line to header section of main layout:

```
<meta name="referrer" content="origin-when-crossorigin" />
```

It would also be great if admin could change the referrer setting to other possible values.

Related issues:

Related to Redmine - Feature #24583: Remove HTTP Referer	**Closed**
Related to Redmine - Feature #23630: Migrate to Rails 5.2	**Closed**
Related to Redmine - Feature #14648: Add a link dispatcher to textile texts	**Closed**

History

#1 - 2018-09-22 11:58 - Go MAEDA

- Related to Feature #24583: Remove HTTP Referer added

#2 - 2018-09-22 12:13 - Go MAEDA

It is not a perfect solution because some browsers such as Edge 18 and Safari on iOS 12 don't support "origin-when-crossorigin" value for "referrer" meta tag.
https://caniuse.com/#search=referer

However, adding the meta tag is effective to mitigate the security risk if users in the organization use Chrome, Firefox, or Safari.

```
Index: app/views/layouts/base.html.erb
===================================================================
--- app/views/layouts/base.html.erb    (revision 17495)
+++ app/views/layouts/base.html.erb    (working copy)
@@ -7,6 +7,7 @@
 <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
 <meta name="description" content="<%= Redmine::Info.app_name %>" />
 <meta name="keywords" content="issue,bug,tracker" />
+<meta name="referrer" content="origin-when-crossorigin" />
 <%= csrf_meta_tag %>
 <%= favicon %>
 <%= stylesheet_link_tag 'jquery/jquery-ui-1.11.0', 'application', 'responsive', :media => 'all' %>
```

#3 - 2018-09-22 13:23 - Go MAEDA

- Target version set to Candidate for next major release

#4 - 2018-09-25 23:26 - Go MAEDA

- Target version changed from Candidate for next major release to 4.1.0

Setting target version to 4.1.0.

#5 - 2018-09-29 04:16 - Go MAEDA

Maybe adding Referrer-Policy header is preferable rather than adding a meta-tag for the following reasons.

- We can ensure that the header is always set
- Sysadmins can easily override the header by configuring web servers if necessary

```
diff --git a/config/application.rb b/config/application.rb
index d77d37e70..3e014f480 100644
--- a/config/application.rb
+++ b/config/application.rb
@@ -53,6 +53,8 @@ module RedmineApp
     # Sets the Content-Length header on responses with fixed-length bodies
     config.middleware.insert_after Rack::Sendfile, Rack::ContentLength
 
+    config.action_dispatch.default_headers['Referrer-Policy'] = 'origin-when-cross-origin'
+
     # Verify validity of user sessions
     config.redmine_verify_sessions = true
```

#6 - 2018-10-01 10:05 - Gregor Schmidt

This seems to be a sensible improvement. It does not impact the behavior for users on IE/Edge, but it improves privacy for users of Chrome, Firefox, Safari.

I also agree, that using an HTTP header is the better option.

However, I am not sure whether same-origin or origin-when-cross-origin would be the better default.

#7 - 2018-10-01 17:31 - Holger Just

The most secure variant would probably be strict-origin-when-cross-origin which seems to be supported by the same current set of commonly used browsers which support origin-when-cross-origin. Compared to the shorter option, the strict one ensures that we don't sent any referrer from a secure page to an unsecure page, similar to the default no-referrer-when-downgrade.

As such, I vote for strict-origin-when-cross-origin. That way, we can still leverage the use of referrers locally (e.g. through log analysis) but don't leak potentially private data to external sites.

If we also want to specifically support IE, we should use origin. However, this would impact most local log analyzer software people might use on their Redmine installations as user flows can not be followed anymore. Thus, I think it's a reasonable tradeoff to use origin-when-cross-origin with most users and to accept the potential leaks for IE/Edge users (until Microsoft and Apple implement the current spec in Edge and Safari/iOS).

I'm with Maeda-san and Greger to use a header instead of a meta tag.

#8 - 2018-10-02 09:36 - Ebrahim Mohammadi

Using Referrer-Policy HTTP header is a great idea.

#9 - 2018-10-02 10:14 - Gregor Schmidt

Holger Just wrote:

> If we also want to specifically support IE, we should use origin. However, this would impact most local log analyzer software people might use on their Redmine installations as user flows can not be followed anymore. Thus, I think it's a reasonable tradeoff to use origin-when-cross-origin with most users and to accept the potential leaks for IE/Edge users (until Microsoft and Apple implement the current spec in Edge and Safari/iOS).

Whatever you decide to use in the end, please **don't** use origin, since it would not only affect logs, but Redmine itself relies on the Referer header, whenever it uses redirect :back.

When using origin, this would result in a redirect to the main page. The fallback specified for redirect_back_or_default would not be used, since a Referer header is present. This would result in a serious negative impact on the user flow.

#10 - 2018-10-02 11:24 - Ludovic Andrieux

Hi,

If you look at HTTP Headers, could you have a look to #29405

Best regards

#11 - 2018-10-02 11:49 - Go MAEDA

Holger Just wrote:

> The most secure variant would probably be strict-origin-when-cross-origin which seems to be supported by the same current set of commonly

> used browsers which support origin-when-cross-origin. Compared to the shorter option, the strict one ensures that we don't sent any referrer from a secure page to an unsecure page, similar to the default no-referrer-when-downgrade.

Thank you for the detailed explanation. Now I think strict-origin-when-cross-origin is the best. We should not use looser option than default no-referrer-when-downgrade.

```
Index: config/application.rb
===================================================================
--- config/application.rb    (revision 17559)
+++ config/application.rb    (working copy)
@@ -53,6 +53,10 @@
     # Sets the Content-Length header on responses with fixed-length bodies
     config.middleware.insert_after Rack::Sendfile, Rack::ContentLength
 
+    # Strip path information in the Referer header to prevent sending
+    # private data to external sites
+    config.action_dispatch.default_headers['Referrer-Policy'] = 'strict-origin-when-cross-origin'
+
     # Verify validity of user sessions
     config.redmine_verify_sessions = true
```

#12 - 2018-10-02 12:48 - Go MAEDA

- Subject changed from Control over HTTP Referrer Configuration to Add Referrer-Policy header to prevent browsers from sending private data to external sites

- Assignee set to Jean-Philippe Lang

#13 - 2019-01-01 17:21 - Alexander Meindl

It looks like *strict-origin-when-cross-origin* is already set by Rails 5 default settings, see https://guides.rubyonrails.org/security.html#content-security-policy

#14 - 2019-01-02 02:13 - Go MAEDA

- Status changed from New to Closed

- Assignee deleted (Jean-Philippe Lang)

- Target version deleted (4.1.0)

- Resolution set to Fixed

Alexander Meindl wrote:

> It looks like *strict-origin-when-cross-origin* is already set by Rails 5 default settings, see https://guides.rubyonrails.org/security.html#content-security-policy

You are right. Thank you for pointing it out. Since Redmine 4.0.0 adds the header by default, we can close this issue.

I confirmed that "Referrer-Policy"=>"strict-origin-when-cross-origin" is included in default_headers and the header sent by Redmine 4.0.0 has Referrer-Policy field.

```
$ bin/rails r 'p Rails.application.config.action_dispatch.default_headers["Referrer-Policy"]'
"strict-origin-when-cross-origin"
```

#15 - 2019-01-02 06:22 - Marius BĂLTEANU

- Related to Patch #28933: Migrate to Rails 5.2 added

#16 - 2019-01-02 06:22 - Marius BĂLTEANU

- Related to deleted (Patch #28933: Migrate to Rails 5.2)

#17 - 2019-01-02 06:23 - Marius BĂLTEANU

- Related to Feature #23630: Migrate to Rails 5.2 added

#18 - 2019-08-10 08:42 - Go MAEDA

- Related to Feature #14648: Add a link dispatcher to textile texts added