

Redmine - Defect #15560

RJS leaking

2013-11-27 17:23 - egor homakov

Status: Closed	Start date:
Priority: Urgent	Due date:
Assignee:	% Done: 0%
Category: Security	Estimated time: 0.00 hour
Target version:	Affected version:
Resolution: Fixed	
Description	
example - http://www.redmine.org/boards/2/topics/quote/5682.js	
all files that respond with JS with private data for GET requests are vulnerable to homakov.blogspot.com/2013/05/do-not-use-rjs-like-techniques.html	
in redmine we should remove:	
attachments/destroy.js.erb members/create.js.erb attachments/upload.js.erb members/destroy.js.erb custom_fields/new.js.erb members/update.js.erb groups/add_users.js.erb messages/quote.js.erb groups/autocomplete_for_user.js.erb repositories/add_related_issue.js.erb groups/destroy_membership.js.erb repositories/new.js.erb groups/edit_membership.js.erb repositories/remove_related_issue.js.erb groups/remove_user.js.erb users/destroy_membership.js.erb issue_categories/create.js.erb users/edit_membership.js.erb issue_categories/new.js.erb versions/create.js.erb issue_relations/create.js.erb versions/new.js.erb issue_relations/destroy.js.erb versions/status_by.js.erb issues/bulk_edit.js.erb watchers/_set_watcher.js.erb issues/update_form.js.erb watchers/append.js.erb journals/edit.js.erb watchers/create.js.erb journals/new.js.erb watchers/destroy.js.erb journals/update.js.erb watchers/new.js.erb members/autocomplete.js.erb wikis/edit.js.erb	
Related issues:	
Related to Redmine - Defect #17770: very simple fix: that causes many sites... New	

History

#1 - 2013-11-28 15:56 - Etienne Massip

- Status changed from New to Needs feedback

I'm not fond of RJS neither but I can't see how an attacker will get access to the private data without first getting access to an authenticated user loaded page?

#2 - 2013-11-28 15:57 - Etienne Massip

For example, there's no sensible data exposed by the server in your example?

#3 - 2013-11-28 16:27 - egor homakov

i gave a link to my blog post above: <http://homakov.blogspot.com/2013/05/do-not-use-rjs-like-techniques.html>

When redmine user visits 3rd party website, that website can include something like `<script src="http://www.redmine.org/boards/2/topics/quote/5682.js"></script>` or iterate all comments, or any other GET-accessible actions (check the files I listed above, some of them suit).

Also he redefines

```
function $(){ return {val: function(){ console.log('LEAKED',arguments)}};
document.write('<script src="http://www.redmine.org/boards/2/topics/quote/5682.js"></script>')
```

#4 - 2013-11-28 17:17 - egor homakov

btw this route is has no CSRF protection

```
match 'sys/projects/:id/repository', :to => 'sys#create_project_repository', :via => :post
```

because no protect_from_forgery in SysController

#5 - 2013-11-28 17:28 - Etienne Massip

egor homakov wrote:

btw this route is has no CSRF protection

```
match 'sys/projects/:id/repository', :to => 'sys#create_project_repository', :via => :post
```

because no protect_from_forgery in SysController

It's normal behavior, this controller is called for system task by passing a key as param.

#6 - 2013-11-28 17:44 - egor homakov

I see, thanks.

i can't reproduce `get 'watchers/new', :to => 'watchers#new'`

but i think most of routes above work for admin users only. I don't have redmine installation to test, so quote-link is only example i have so far. Please check if there are other JS-responding GET routes, non-GET are fine.

#7 - 2017-12-03 19:26 - Toshi MARUYAMA

- Related to Defect #17770: very simple fix: that causes many sites to break, and much confusion - incorrect use of .js suffix added

#8 - 2024-01-03 16:18 - Holger Just

- Status changed from Needs feedback to Closed

- Resolution set to Fixed

This should be addressed for some time now with Rails' builtin XHR CSRF handling. The endpoints which return javascript all require that the request sets a X-Requested-With: XMLHttpRequest header. If such a header is not set in the request, the response is blocked by Rails' builtin ActionController::RequestForgeryProtection middleware.

Such a header is not (and can not) be set for "normal" included requests such as an inclusion of the resource in a <script> tag. It can only be set by a JavaScript AJAX client (as is done by rails-ujs which is used by Redmine to requests these endpoints). Here, the clients are restricted by the same origin policy. As such, an external website can not request these endpoints via AJAX.

I believe it was, Egor, you who nudged Rails towards this solution in <https://github.com/rails/rails/pull/13345> :)