Redmine - Feature #2647

Repository browsing shall respect ACLs in repository

2009-02-02 17:18 - Mathias Kühn

Status:	New	**Start date:**	2009-02-02
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	SCM extra	**Estimated time:**	0.00 hour
Target version:			
Resolution:			

Description

The current implementation of the repository browser connects to a repository using a provided login and password. In our company we're running subversion with .authz access control where we nut just limit write access but also hide complete trees in the repository from some users.

Since the repository browser does not change to the appropriate user account, we must either us a very low privileged user account to allow browsing, or we allow free read access for all registered members of the project.

We implemented a simple solution that extends the repository settings (so far only for subversion) with a checkbox that enables logon using the currently logged in user. Since we don't have the credentials at that point, we're using a separate vhost on the server that 'authenticates' users by just grabbing their login and ignoring their credentials. If no user is logged in, the provided user account is taken from the repository settings. This solution works very nicely, still it may need some tweaks. We're considering to have an additional token along with that setting that would be sent as password to close that theoretical breach.

If anyone is interested in that solution, please let me know. The described solution was done against trunk @ rev 2335.

Related issues:

Related to Redmine - Defect #13484: restricted access folder in a redmine pro... **New**

History

#1 - 2009-02-05 09:04 - Thomas Pihl

Sounds interesting. Any chance you could add a patch here for us to see? I can see good use for this (also with subversion in current case).

/T

#2 - 2009-02-05 18:44 - Mathias Kühn

- File repository_auth.patch added

Hi, here's the patch. Actually it works the following way. In addition to provide just the username and password for repository access, an authentication method can be chosen.

- Use Username/Password: That's the current implementation
- Use current users name: Pass the name of the current user as username and the securitytoken as password.
- Use current users role: Use current users role in the project as username and the securitytoken as password.

On the server side, the Redmine.pm script has been extended to provide additional handlers

```
   PerlAccessHandler Apache::Authn::Redmine::redmine_access_handler
   PerlAuthenHandler Apache::Authn::Redmine::redmine_authen_handler
```

which take the provided username and allow access. If the optional parameter **RedmineSecurityToken** is provided, it must match the security token provided by the client, otherwise the authentication attempt is rejected.

The following example shows an example config providing a public access to subversion as already shown and the secured private access point for redmines repository browser:

```
<VirtualHost *>

 PerlLoadModule Apache::Authn::Redmine
 <Location /svn>
   DAV svn
   SVNParentPath "/srv/svn"
   AuthzSVNAccessFile "/srv/svn/.authz"

   AuthType Basic
   AuthName "Repository"

   Require valid-user

   PerlAccessHandler Apache::Authn::Redmine::access_handler
   PerlAuthenHandler Apache::Authn::Redmine::authen_handler

   RedmineDSN DBI:mysql:database=redmine;host=127.0.0.1
   RedmineDbUser root
   RedmineDbPass root
 </Location>

 <Location /svn-redmine>

   DAV svn
   SVNParentPath "/srv/svn"
   AuthzSVNAccessFile "/srv/svn/.authz"

   Order deny,allow
   Deny from all

   AuthType Basic
   AuthName "Private Repository"

   Require valid-user

   PerlAccessHandler Apache::Authn::Redmine::redmine_access_handler
   PerlAuthenHandler Apache::Authn::Redmine::redmine_authen_handler

   RedmineDSN DBI:mysql:database=redmine;host=127.0.0.1
   RedmineDbUser root
   RedmineDbPass root

   RedmineSecurityToken secureThing

   <Limit GET PROPFIND OPTIONS REPORT>
     Allow from localhost
   </Limit>

 </Location>

 DocumentRoot /srv/redmine/public
 ServerName subversion.internal.lan
 ServerAdmin admin@internal.lan

 <Directory /srv/redmine/public>
  AllowOverride None
  Order allow,deny
  Allow from all
 </Directory>
</VirtualHost>
```

#3 - 2009-02-21 13:28 - Jean-Philippe Lang

- Category set to SCM

#4 - 2009-04-17 03:47 - Kiall Mac Innes

I would love to see this committed!

but... being able to admin the authz file from within redmine would be great addition as well...

#5 - 2009-10-04 23:55 - Lluís Vilanova

Duplicates #2315.

This would mostly eliminate the need for specifying a username/passowrd in the project settings (solving as a side effect #2034), which is the main reason why I'm using a file:/// URL.

#6 - 2009-11-02 17:45 - Nikita Pustovoytov

Lluís Vilanova wrote:

> Duplicates #2315.
>
> This would mostly eliminate the need for specifying a username/passowrd in the project settings (solving as a side effect #2034), which is the main reason why I'm using a file:/// URL.

Maybe I'm not very familiar with SVN, but where can I get security token? I'm using pure Collabnet Subversion without Apache and it is impossible to install Apache along with SVN. Some repos use path-based authentification to restrict some users. So this patch is very important for me. But how can I get the token??

#7 - 2011-03-29 06:32 - Toshi MARUYAMA

- Category changed from SCM to SCM extra

#8 - 2011-09-10 04:20 - Robert Rath

- File repository_auth.7077.patch added

This patch is still relevant and necessary when doing fine grain subversion access control managed by Apache.

Added update patch against current stable branch 1.2-stable (revision 7077)

... Robert

#9 - 2012-04-01 13:04 - Robert Rath

- File redmine-1.3-9291_repository_auth.patch added

This patch is still relevant and necessary when doing fine grain subversion access control managed by Apache in redmine-1.3-stable.

Added update patch against current stable branch 1.3-stable (revision 9291)

... Robert

#10 - 2012-04-23 09:49 - Pramod kumbhar

Robert Rath wrote:

> This patch is still relevant and necessary when doing fine grain subversion access control managed by Apache in redmine-1.3-stable.
>
> Added update patch against current stable branch 1.3-stable (revision 9291)
>
> ... Robert

Robert, Can you please produce patch for current release i.e. 1.4.1? I tried to use your patch but failed.

Thanks!

#11 - 2013-09-03 07:39 - Robert Rath

My patch against Redmine 1.3 r9291 was missing a db/migrate file which prevented the patch from working.

Here is a a copy of the missing file's contents './db/migrate/109_add_special_repository_security.rb'

```
class AddSpecialRepositorySecurity < ActiveRecord::Migration
  def self.down
    remove_column :repositories, :login_method
    remove_column :repositories, :security_token
  end

  def self.up
    add_column :repositories, :login_method, :int, :default => 0, :null => true
    add_column :repositories, :security_token, :string, :limit => 60, :default => "", :null => true
  end
end
```

#12 - 2013-09-03 07:45 - Robert Rath

- File redmine-2.3-12119_repository_auth.patch added

This patch is still relevant and necessary when doing fine grain subversion access control managed by Apache in redmine-2.3-stable.

Added update patch against current stable branch 2.3-stable (revision 12119)

... Robert

#13 - 2013-09-04 06:24 - Robert Rath

- File redmine-2.3-12119_repository_auth-2.patch added

My updated 2.3 patch fell prey to the new 'safe_attributes' feature which protects the database against changes not defined in the model.
I have provided a new update patch which correctly allows edits of the new security fields.

... Robert Rath

#14 - 2013-09-04 23:03 - Tilo Mey

This doesn't work with redmine and subversion on different servers, even with mysql-DB-acces, does it?
Background: if running on the same server, the project-admins have acces to ALL repos via file:// via redmine :(

#15 - 2013-09-05 03:24 - Robert Rath

Tilo Mey wrote:

> This doesn't work with redmine and subversion on different servers, even with mysql-DB-acces, does it?
> Background: if running on the same server, the project-admins have acces to ALL repos via file:// via redmine :(

Hi Tilo,

I use this all the time with external subversion servers via HTTPS urls, in fact the only reason the security model works is through the Apache2 authz module.
I also run some servers with all services locally as well but for these project-admin security is no issue.

Local file system assess as you described bypasses it all as you point out. Perhaps one way around this is to use file system permissions to block filesystem access and then Apache to serve SVN back to Redmine to control the security. This may not be possible however with Apache2 serving Redmine as they need to share the same user account. You could use a different service for Redmine, ie webbrick and Apache2 on a different port for SVN. This way these services can run under different user accounts which will prevent Redmine direct file system access to svn. You could also use obscurity and place the SVN files in a very hard to guess location on the filesystem (poor security).

Or just put Redmine and SVN on different servers. If you use virtual machines this solution will be the simplest.

...Robert

edit...

In all cases to implement this fine grained authz security you need the pearl module installed and configured for Apache2 on the remote subversion server. This is a problem for accessing third party servers. It may be possible to work around this by using a local Apache2 proxy but you would have to implement some form of local caching of credentials to be presented to the remote server which may or may not be acceptable.

#16 - 2013-10-24 14:55 - Daniel Hger

Hi Robert,
Im getting internal error on Redmine 2.3.3 while loading the add-svn-repo site.

Log:

Completed 500 Internal Server Error in 113ms

```
ActionView::Template::Error (undefined method `login_method' for #<Repository::Subversion:0x007f60bc9d3860>):
17: <% button_disabled = true >
18: < if Jack Zheng >
19: <   button_disabled = ! @repository.class.scm_available >
20: <=    repository_field_tags(f, Jack Zheng)%>
21: <% end %>
22: </div>
23:
lib/redmine/views/labelled_form_builder.rb:34:in `select'
app/helpers/repositories_helper.rb:164:in `subversion_field_tags'
app/helpers/repositories_helper.rb:126:in `repository_field_tags'
app/views/repositories/_form.html.erb:20:in `_app_views_repositories__form_html_erb__3600649647781905487_33101280'
app/views/repositories/new.html.erb:4:in `block in app_views_repositories_new_html_erb__1890930613153873159_32704720'
app/helpers/application_helper.rb:948:in `labelled_form_for'
app/views/repositories/new.html.erb:3:in `_app_views_repositories_new_html_erb___1890930613153873159_32704720'
```

Btw: is it possible to add this patch to the official redmine version?

#17 - 2013-10-30 14:26 - Robert Rath

Daniel Hger wrote:

> Hi Robert,
> Im getting internal error on Redmine 2.3.3 while loading the add-svn-repo site.
>
> Log:
>
> Completed 500 Internal Server Error in 113ms
>
> ```
> ActionView::Template::Error (undefined method `login_method' for #<Repository::Subversion:0x007f60bc9d3860>):
> 17: <% button_disabled = true >
> 18: < if Jack Zheng >
> 19: <   button_disabled = ! @repository.class.scm_available >
> 20: <=    repository_field_tags(f, Jack Zheng)%>
> 21: <% end %>
> 22: </div>
> 23:
> lib/redmine/views/labelled_form_builder.rb:34:in `select'
> app/helpers/repositories_helper.rb:164:in `subversion_field_tags'
> app/helpers/repositories_helper.rb:126:in `repository_field_tags'
> app/views/repositories/_form.html.erb:20:in `_app_views_repositories__form_html_erb__3600649647781905487_33101280'
> app/views/repositories/new.html.erb:4:in `block in app_views_repositories_new_html_erb__1890930613153873159_32704720'
> app/helpers/application_helper.rb:948:in `labelled_form_for'
> app/views/repositories/new.html.erb:3:in `_app_views_repositories_new_html_erb___1890930613153873159_32704720'
> ```
>
> Btw: is it possible to add this patch to the official redmine version?

Hi Daniel,

My patch is specific to the official Redmine version as at 2.3 r12119. Once the official version moves on the patch may no longer work. You may however be able to update back to r12119, apply the patch and then update back to the latest production release.

When I get a free moment I will see where it stands with respect to the current Redmine release.

Regards,
Robert Rath

#18 - 2014-01-30 20:22 - Daniel Hger

My error occured because i didn't applied db:migrate correctly.
This patch still works for Redmine v2.4.2 - just the file config/locales/en.yml changed a little bit.

I'm just worried about one thing: users who are not authorized to access the repository can still crawl over all revisions.
Redmine stores these informations in its database when the repository is accessed.
The goal is to prevent users who are not authorized from gaining informations about the repository.

edit: it is possible to achieve this via roles, but not based on the repo-rules

#19 - 2014-01-30 23:44 - Robert Rath

Daniel Hger wrote:

> My error occured because i didn't applied db:migrate correctly.
> This patch still works for Redmine v2.4.2 - just the file config/locales/en.yml changed a little bit.
>
> I'm just worried about one thing: users who are not authorized to access the repository can still crawl over all revisions.
> Redmine stores these informations in its database when the repository is accessed.
> The goal is to prevent users who are not authorized from gaining informations about the repository.
>
> edit: it is possible to achieve this via roles, but not based on the repo-rules

Hi Daniel,

This patch transfers all access control of SVN to Apache's authz module. This means that in addition to setting up individual user access to SVN you have fine grained access control throughout your entire repository tree simply by setting up and editing an Apache authz file, creating groups and R/W assignments as simply or as complex as you require. Unfortunately the authz will need to be manually edited (or managed with some other tool) on the server's file system.

... Robert Rath

As I previously mentioned, this access control mechanism is intimately bound to apache. Anyone with direct access to the repository's file system can bypass it all.

#20 - 2014-04-22 15:19 - Daniel Hger

The problem is that Redmine still copies the revisions into its database when a uses tries to access the repository. (Thats why the first access to a big repo needs more time, just look in the db/logs, a huge amount of data is getting stored there)

So someone who cant access the repo can still see the revisions (including path names, file names, actions).
Redmine needs the revisions stored in its db to be able to provide links to them in issues and forums.
We need a fix to hide the revision pages from users who cant access the repo.
So in the controller for this view we have to add a check if the current user can really read the information provided on the revision's page.

For this check we need to access the svn-repo via a querry to the apache host (with user credentials) or evaluate the rights from ./conf/authz in the svn folder (filesystem) for each file mentioned on the rev's page.

Both solutions are dirty, but i dont see another way...

#21 - 2014-10-01 16:06 - Daniel Hger

I wrote a complete new patch which hides nearly everything from denied users.
In my company we use the same LDAP for Redmine and SVN, so all users have the same login.

Redmine has direct file access to the repository and compares ./conf/authz file for every request (which exposes svn related data) with the current user's name and the paths/files.
If the user is not permitted to see these paths, files, diffs, revisions, revision-info, etc. the user is redirected to an error page or the content is changed (e.g. tooltips).
If anyone is interested i can upload my code.

#22 - 2014-10-10 22:00 - Keith Johnson

Daniel Hger wrote:

> I wrote a complete new patch which hides nearly everything from denied users.
> In my company we use the same LDAP for Redmine and SVN, so all users have the same login.
>
> Redmine has direct file access to the repository and compares ./conf/authz file for every request (which exposes svn related data) with the current user's name and the paths/files.
> If the user is not permitted to see these paths, files, diffs, revisions, revision-info, etc. the user is redirected to an error page or the content is changed (e.g. tooltips).
> If anyone is interested i can upload my code.

Hi Daniel,
I would love to get a copy of your patches, even if they aren't perfect. We've got a project where this is necessary and it sounds like your changes are exactly what we need. If you don't want to upload them publicly my email is in my profile.

#23 - 2014-10-13 13:39 - Daniel Hger

- File svnpatch251_v3.patch added

Please let me know if you have further questions or improvements.
Currently every svn request is checked by svncheck. Maybe we should exclude web urls.

#24 - 2014-10-13 14:15 - Daniel Hger

Repositories have to be created and first accessed by admin. Afterwards Redmine copys the repo information into its database. This process can take a very long time for huge repos. Repos in sub directories are also supported.
My svn repos are mounted to /opt/svn/ (e.g. /opt/svn/reponame/conf/authz).
Im only a student and this code still needs improvements since it was just created to fit our company.

#25 - 2015-04-30 13:45 - Tilo Mey

Robert Rath wrote:

> I use this all the time with external subversion servers via HTTPS urls, in fact the only reason the security model works is through the Apache2 authz module.
> Or just put Redmine and SVN on different servers. If you use virtual machines this solution will be the simplest.
>
> ...Robert
>
> edit...
>
> In all cases to implement this fine grained authz security you need the pearl module installed and configured for Apache2 on the remote subversion server. This is a problem for accessing third party servers. It may be possible to work around this by using a local Apache2 proxy but

> you would have to implement some form of local caching of credentials to be presented to the remote server which may or may not be acceptable.

I've 2 servers:
one vor redmine, another for svn.
Direct svn via https with fine granular is working fine.

For redmine I'm using
redmine-2.3-12119_repository_auth-2.patch

Therefore I've something like this in /etc/apache/mods-eanbled/dav_svn.conf on svn

```
PerlLoadModule Apache::Redmine
<Location /svn-redmine>
     DAV svn
     SVNPath /srv/subversion/repo
     Order deny,allow
     Deny from all
     # only allow reading orders

     AuthType Basic
     AuthName redmine
     Require valid-user
     AuthUserFile /srv/subversion/authz
     PerlAccessHandler Apache::Redmine::redmine_access_handler
     PerlAuthenHandler Apache::Redmine::redmine_authen_handler

     RedmineSecurityToken "redminetoken" 

     <Limit GET PROPFIND OPTIONS REPORT>
       Allow from remine.blob.de
     </Limit>
</Location>
```

But there is an error on svn:
Can't connect to data source '' because I can't work out what driver to use (it doesn't seem to contain a 'dbi:driver:' prefix and the DBI_DRIVER env var is not set) at /usr/lib/perl5/Apache/Redmine.pm line 562.

On the repository-seting in redmine I've choosen "Name of current user"

Why does the svn is needing the database-connection to redmine?

#26 - 2021-06-29 07:36 - Go MAEDA

- Related to Defect #13484: restricted access folder in a redmine project added

Files

repository_auth.patch	9.51 KB	2009-02-05	Mathias Kühn
repository_auth.7077.patch	11.3 KB	2011-09-10	Robert Rath
redmine-1.3-9291_repository_auth.patch	13.3 KB	2012-04-01	Robert Rath
redmine-2.3-12119_repository_auth.patch	13 KB	2013-09-03	Robert Rath
redmine-2.3-12119_repository_auth-2.patch	14 KB	2013-09-04	Robert Rath
svnpatch251_v3.patch	13.7 KB	2014-10-13	Daniel Hger