

enhanced mod_perl module for apache

2022/05/09 14:09 - Admin Redmine

ステータス:	New	開始日:	2009/08/05
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:	SCM extra_46	予定工数:	0.00時間
対象バージョン:		作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/3712	status_id:	1
category_id:	46	tracker_id:	3
version_id:	0	plus1:	0
issue_org_id:	3712	affected_version:	
author_id:	443	closed_on:	
assigned_to_id:	0	affected_version_id:	
comments:	21		

説明

Redmine.pm is a very good module and I used it for a very long time but, recently, I needed to have more configuration options and I decided to rewrite it. This new version adds the following things:

4 new directives:

- *RedmineAuthenticationOnly*: when defined, the module only checks the credentials with redmine (database or LDAP, depending the user's settings). It allows a mix, for example, between redmine authentication and subversion access management using AuthzSVNAccessFile directive)
- *RedmineProjectId*: when defined to a project identifier, the module will check user's permissions based on a specific project. It allows, for example, to link a redmine project with its documentation in a DokuWiki instance and manage access rights in DokuWiki from redmine.
- *RedmineReadPermissions*: one or more permissions used to allow read access (ie: GET PROPFIND REPORT and OPTIONS). Default value is :browse_repository
- *RedmineWritePermissions*: one or more permissions used to allow others accesses (ie: everything except GET PROPFIND REPORT and OPTIONS). Default value is :commit_access

and more integration with redmine:

- anonymous access is denied if "Authentication required" is checked in redmine (Administration->Settings/Authentication).
- The module will use the permissions defined for *Anonymous* et *Non member* roles if the project is public.
- read and write accesses are checked using redmine permissions

perl is not my preferred language so, please, feel free to make any comment or modify the source code...

Arnaud

journals

I like your patch very much as it adds cool new features and cleans up the code. Thank you very much for sharing! :-)
Some time ago I added a patch to Redmine to authenticate against an LDAP server as user (see #1913).

To make this work with your enhanced version of Redmine.pm a patch is required that scans the LDAP binddn for the occurrence of the string "\$login" and if it is there the string will be replaced by the username of the current user (and used together with his password to authenticate against the LDAP server). The modification is minor and completely backwards compatible with unpatched Redmine. So you might safely add this patch to RedmineAdvanced.pm and it will just work... :-)

Thnx, very useful patch!

I updated the query at line 199 to cope with the removal of the member_roles table in latest 0.8 stable.

Updated query result checks to be compatible with sqlite.

The diff is applied on Arnaud's file.

The script gives open access when the project stated in "RedmineProjectId" does not exist in redmine.

Lluís Vilanova wrote:

The script gives open access when the project stated in "RedmineProjectId" does not exist in redmine.

Yes, you're right but only when one of following conditions is satisfied:

- READ access is requested and "Authentication required" is not checked (in Administration -> Settings -> Authentication)
- current user is an administrator of the redmine site
- *RedmineAuthenticationOnly* is defined

In my opinion, only the first condition may introduce a security hole and administrators should be aware of it..

How do I use these files?

I have 0.8-stable as-is.

Thanks.

I replaced Redmine.pm for this RedmineAdvanced.pm, and copied it to perl5/apache folders.

Am I doing it correctly?

Well, I replaced all Redmine.pm for RedmineAdvanced.pm, and updated dav_svn.conf with the new name.

How do I use the new features?

Tks.

Well here are some examples:

Example1 : Most used case. Users will be authenticated with redmine and you will give them read and write authorization based on roles/permissions. In this example, user will have read access when having a role with the "Browse repository" permission (and commit/write access when having the "Commit access" permission).

PerlLoadModule Apache::Authn::RedmineAdvanced

```
DAV svn
SVNParentPath /var/svn
```

```
AuthType Basic
AuthName "REDMINE area"
Require valid-user
```

```
PerlAccessHandler Apache::Authn::RedmineAdvanced::access_handler
PerlAuthenHandler Apache::Authn::RedmineAdvanced::authen_handler
RedmineDSN "DBI:mysql:database=redmine;host=localhost"
RedmineDbUser "redmine"
RedmineDbPass "XXXXXX"
RedmineReadPermissions :browse_repository
RedmineWritePermissions :commit_access
```

Example2 : For a specific repository, you want to manage the rights granted to people with the AuthzSVNAccessFile directive. In this case, you only want to authenticate users (authorization will be defined in the file used the AuthzSVNAccessFile directive).

```
PerlLoadModule Apache::Authn::RedmineAdvanced
```

```
DAV svn
SVNPath /mnt/subversion/myrepo
AuthzSVNAccessFile /mnt/subversion/accesslist
```

```
AuthType Basic
AuthName "REDMINE area"
Require valid-user
```

```
PerlAccessHandler Apache::Authn::RedmineAdvanced::access_handler
PerlAuthenHandler Apache::Authn::RedmineAdvanced::authen_handler
RedmineDSN "DBI:mysql:database=redmine;host=localhost"
RedmineDbUser "redmine"
RedmineDbPass "XXXXXX"
RedmineAuthenticationOnly "On"
```

Example 3: You have installed a web application on your server (a dokuWiki instance, for example) and you want to allow access only to members of a specific redmine's project (for example, the project "myproject"). Read and write access are given if users have "View wiki" or "Edit wiki pages" permissions...

```
PerlLoadModule Apache::Authn::RedmineAdvanced
```

```
AuthType Basic
AuthName "REDMINE area"
Require valid-user
```

```
PerlAccessHandler Apache::Authn::RedmineAdvanced::access_handler
PerlAuthenHandler Apache::Authn::RedmineAdvanced::authen_handler
RedmineDSN "DBI:mysql:database=redmine;host=localhost"
RedmineDbUser "redmine"
RedmineDbPass "XXXXXX"
RedmineProjectId myproject
RedmineReadPermissions :view_wiki_pages
RedmineWritePermissions :edit_wiki_pages
```

Hope this will help you....

This is perhaps a minor issue, but when a repository exists and a Redmine administrator authenticates, access is denied. I would say that a Redmine administrator should be granted access to all Redmine-managed assets. Looking at the MySQL general log, the following queries are issued:

```
SELECT hashed_password, auth_source_id
FROM users
WHERE users.status=1 AND login='bnordgren'
SELECT host,port,tls,account,account_password,base_dn,attr_login
from auth_sources
WHERE id = '1'
SELECT is_public FROM projects WHERE projects.identifier='private'
SELECT permissions
FROM members, projects, users, roles
WHERE projects.id=members.project_id AND
users.id=members.user_id AND
roles.id=members.role_id AND
users.status=1 AND
login='bnordgren' AND
identifier='private'
```

I made the same complaint against the "official version" of @Redmine.pm@ in #5070. :) If it is not always desirable to have redmine administrators be granted access, perhaps it could be an option? I'd do it myself but I don't know perl, and for the moment I can live with adding myself to the projects.

Also, it's probably best to make clear that the @member_roles@ table was apparently

added in 0.9.x, and not removed in the latest 0.8 stable as mentioned by Thimios Dimopulos above. So if you're running 0.9.x download the original version attached to this ticket. If you're running 0.8.x, download the version by Thimios Dimopulos. Your apache will thank you.

Bryce Nordgren wrote:

This is perhaps a minor issue, but when a repository exists and a Redmine administrator authenticates, access is denied. I would say that a Redmine administrator should be granted access to all Redmine-managed assets. Looking at the MySQL general log, the following queries are issued:
[...]

I made the same complaint against the "official version" of @Redmine.pm@ in #5070. :) If it is not always desirable to have redmine administrators be granted access, perhaps it could be an option? I'd do it myself but I don't know perl, and for the moment I can live with adding myself to the projects.

Well, I think you can patch my original file if you need. Just insert the following line at line number 321:

```
return OK if ( is_admin( $r->user, $r ) );
```

I didn't test the result but it should work as you like...

From my point of view, as an administrator, I don't want to bypass security access all the time but I understand your complaint...

Thanks for the patch. :) I'm not very familiar with perl and this gives me the head start I needed.

My requirement for some kind of "security bypass" has to do with creating mirrors of all the repositories. Redmine and the subversion repositories are outside the firewall, and I want to @svnsync@ all repositories to another machine nightly. Since I'm allowing my users to create non-public subversion repositories, I need a user that will always have access.

Does this still work in the latest versions of Redmine? I'm looking to allow some users to be able to do Example #2 on their project.

Yes, I'm using it with redmine 1.1.0

I reworked my own Redmine.pm yesterday and discovers this patch today. I have go the same way to enhance it but there are some differences :

- It is using the proper authentication (authn) and authorization (authz) handlers instead of access and authentication ones. They could theoretically work separately.
- The authentication phase only authenticates, e.g. checks login/password. It honors the 'login_required' settings. It returns "AUTH_REQUIRED" in case of password mismatch and "FORBIDDEN" for inactive accounts.
- The authorization phase checks if the authenticated user is allowed. It properly honor anonymous and non-member permissions on public projects. It returns "FORBIDDEN" if the user is not authorized, but for anonymous. In the latter case, it returns "AUTH_REQUIRED" to enforce login.
- The credential cache takes the required permission into account. With the shipped Redmine.pm, I think you can commit right after doing reading since the required permissions are not tested.
- I have added a RedmineCacheCredsMaxAge setting to define an expiration delay.
- The credential cache only works if you use both handlers. Credentials are recorded in the authz handler and they are tested in the authn handler during subsequent requests.

The whole thing needs to be tested and I think I will add some other features I have seen there. I am not sure how I should adapt the where-clause setting, as I have added some queries.

Hope this helps.

It's really sad this improved feature is still not planned. I actually found this after I made a thread request, that would be covered by this patch

<http://www.redmine.org/boards/2/topics/39274>

The RedmineAdvanced.pm is released in context of the webdav plugin.

https://github.com/amartel/redmine_webdav/blob/master/extra/svn/RedmineAdvanced.pm

see https://github.com/amartel/redmine_webdav

related issues

related, New, 5070, Redmine.pm does not allow Administrators to access svn repositories

履歴

#1 - 2022/05/10 17:25 - Admin Redmine

- カテゴリをSCM extra_46にセット