Redmine - Feature #2034
Scramble passwords in database table "Repositories"

2008-10-15 20:20 - Alexey Chepurnykh

Status:	Closed	**Start date:**	2008-10-15
Priority:	High	**Due date:**	
Assignee:		**% Done:**	0%
Category:	SCM	**Estimated time:**	0.00 hour
Target version:			
Resolution:	Fixed		

Description

If I do

```
SELECT r.login, r.password, r.url FROM repositories r
```

I can see all passwords and repository resources, though I mustn't to. Better to scramble passwords.

Related issues:

Related to Redmine - Feature # 7411: Option to cipher LDAP ans SCM passwords ...	**Closed**	**2011-01-22**
Duplicated by Redmine - Defect # 3981: Scramble passwords in database table "...	**Closed**	**2008-10-15**

History

#1 - 2008-10-16 10:06 - Thomas Lecavelier

Not relevant from my point of view: database access is server side, only people with legitim access to database could eventually have access to this table. Last point: I don't think SCM accept password pre-scrambled as authentication.

#2 - 2008-10-16 10:37 - Alexey Chepurnykh

Thank you for your answer, Thomas.

I am not agree. Do you like if you trust your account to somebody, he can touch your soft spot :-) ?

1. Often people have the same password to access in the different systems.
   I am administering redmine installation. Now I can see open passwords of all our developers and customers, allowed to access our SVN.
2. As far as I did integration of passwords in SVN and Redmine, I know this are passwords of the users in Redmine :-)
   Fortunately I am not an intruder. But I could be him potentially. :-)
3. Also some other people could have access to database. We can know each other's passwords now. We have only mutual protection now. This is not well security protection.

#3 - 2008-10-16 10:38 - Alexey Chepurnykh

Alex Chep wrote:

> *Thank you for your answer, Thomas.*
>
> *I am not agree. Do you like if you trust your account to somebody, he can touch your soft spot :-) ?*
>
> *1. Often people have the same password to access in the different systems.*
>
> *I am administering redmine installation. Now I can see open passwords of all our developers and customers, allowed to access our SVN.*

> *2. As far as I did integration of passwords in SVN and Redmine, I know this are passwords of the users in Redmine :-)*
> *Fortunately I am not an intruder. But I could be him potentially. :-)*
> *3. Also some other people could have access to database. We can know each other's passwords now. We have only mutual protection now.*
> *This is not good security protection.*

#4 - 2008-10-16 10:58 - Thomas Lecavelier

"With great power comes great responsibility"

I'm running a donebox instance where people can drop their "todo lists": their password are hashed with SHA1, but their tasks are saved in clear. If (as myself) they drop important informations like meetings, phone contact, etc. they would run a big risk but... I just automatized backups and juste **never** even try to look at the tasks.description field. It's matter of **confidence**.

As far as your team has access to redmine database, which is a production one, I think your true security hole is in the administration: my redmine databases are only readable by "root@127.0.0.1" and "rdm@127.0.0.1" users, the last have a auto-generated pwd that I don't even try to learn. Production databases have to be black boxes.

Oh btw: don't you have any webmail account? How many account message have you received with your credential? Your mail provider know your passwords. Big Brother knows everything about you :)

But the last word is still true: how do you want to browse restricted svn repository if it's not stored in clear or in an easy to decipher way? :)

Every people running even the smallest service is potentially an intruder. But confidence make difference between potential risk and proven risk :)

(btw, I'm respecting your opinion about this security consideration, since I don't close the issue. I let over people to explain how they feel it ;))

#5 - 2008-10-16 11:25 - Alexey Chepurnykh

Thank you for your positive attitude to the IT colleagues. You are right person - you trust people. I do really respect this and I got your view. it is partially right but there is other side. Internally I am not satisfied because we have enough clear understanding why we prefer to use the strong protected algorithms and highly protected systems.

PS. SVN through redmine is write protected but not directly.

#6 - 2008-10-16 13:13 - Nicolas Chuche

If you crypt password in rdbms, redmine need to uncrypt them to connect to the SCM. Everyone with access to the redmine server should be able to uncrypt this password too.

#7 - 2008-10-16 15:58 - Jean-Philippe Lang

- Target version deleted (0.7.3)

> *Now I can see open passwords of all our developers and customers, allowed to access our SVN*

No. 1. You can see the passwords that Redmine uses to access your repositories (the one you enter when setting up the repository in Redmine).
2. But you can **not** see the redmine users' passwords (SHA-1 applied in the users table).

Concerning 1., as Nicolas said, Redmine would have to unscramble them if they were scramble in the database (using some kind of key in the application configuration). So someone having access to your redmine server would see this key and would be able to unscramble passwords.

If you have a better solution concerning 1., please share it.

#8 - 2008-10-16 15:59 - Jean-Philippe Lang

- Subject changed from Security hole in database table "Repository" to Scramble passwords in database table "Repositories"

#9 - 2008-10-16 15:59 - Jean-Philippe Lang

- Tracker changed from Defect to Feature

#10 - 2008-10-16 16:09 - Alexey Chepurnykh

- Target version set to 0.7.3

Yes. If the encription key will be hold inside the redmine application this will solve the problem.

#11 - 2008-10-16 16:38 - Nicolas Chuche

Alex Chep wrote:

> *Yes. If the encription key will be hold inside the redmine application this will solve the problem.*

Where in redmine :

- In the database ? bad idea...
- On the filesystem ? You can read it if you have server access so it's quite weak.
- In memory ? Do you really want to have to enter the password each time you restart rails ?

#12 - 2008-10-16 16:42 - Alexey Chepurnykh

Jean-Philippe Lang wrote:

> > *Now I can see open passwords of all our developers and customers, allowed to access our SVN*
>
> *No.*

Excuse me, but YES :-) . This is in my case only. Because as I said above: "I did integration of passwords in SVN and Redmine". SVN server takes passwords from redmine database.
I can not see passwords of redmine users. But the passwords for repositories are exactly the same. Such strange irony :-)

> 1. *You can see the passwords that Redmine uses to access your repositories (the one you enter when setting up the repository in Redmine).*
> 2. *But you can **not** see the redmine users' passwords (SHA-1 applied in the users table).*

> *Concerning 1., as Nicolas said, Redmine would have to unscramble them if they were scramble in the database (using some kind of key in the application configuration). So someone having access to your redmine server would see this key and would be able to unscramble passwords.*

Yes. But at list this is the next wall. He also can get access to the database if he did not have it. ;-). There is no perfection :-(

I like your creation. One of the most convenient solutions of open source.

#13 - 2008-10-16 16:51 - Alexey Chepurnykh

Nicolas Chuche wrote:

> *Alex Chep wrote:*
>
> > *Yes. If the encription key will be hold inside the redmine application this will solve the problem.*
>
> *Where in redmine :*
>
> - *In the database ? bad idea...*

Yes, bad idea :-)

> - *On the filesystem ? You can read it if you have server access so it's quite weak.*

Everything is relative. Let compare all system components and release the soft spots. This one is very-very soft. :-) At list it makes me scare. To keep the key in configuration is a good compromise.

> - *In memory ? Do you really want to have to enter the password each time you restart rails ?*

No. Memory is a bad idea :-)

#14 - 2008-10-16 20:40 - Jean-Philippe Lang

- Target version deleted (0.7.3)

Please do not target 0.7.3 since it's already released.

#15 - 2008-10-16 20:45 - Jean-Philippe Lang

Can you explain in more details how your svn/redmine password integration works ? I still don't get it.

#16 - 2008-10-16 21:11 - Arnaud Martel

Jean-Philippe, I guess that Alex is using Redmine.pm (extra/svn/Redmine.pm). In this case, SVN passwords are crypted using SHA-1 and stored inside the redmine database...

#17 - 2008-10-17 08:28 - Alexey Chepurnykh

Jean-Philippe Lang wrote:

> *Can you explain in more details how your svn/redmine password integration works ? I still don't get it.*

I use Apache server for svn interface over http or https
Here is a part of this config.

```
LoadModule dav_svn_module     modules/mod_dav_svn.so
LoadModule authz_svn_module   modules/mod_authz_svn.so

<VirtualHost *:80>
  ServerAdmin admin@domain.ru
  DocumentRoot /opt/svn
  ServerName svn.domain.ru
  ErrorLog logs/svn-error_log
  CustomLog logs/svn-access_log common

  <Location /svn>
   DAV  svn
   SVNParentPath "/opt/svn/"
   SVNListParentPath on
   SVNAutoversioning On
   ModMimeUsePathInfo On

   # The problem now is that SVNIndexXSLT option does not work if we put <Location />. Only if <Location /svn>
   SVNIndexXSLT "/svnindex.xsl"

   #SSLRequireSSL

   # our access control policy
   AuthzSVNAccessFile /etc/svn/authz.ini

   # try anonymous access first
   Satisfy Any

   # only authenticated users may access the repository
   Require valid-user

   # how to authenticate a user
   AuthType Basic
   AuthName "Subversion repository"

   AuthMySQLEnable on
   AuthMySQLDB redmine_db
   AuthMySQLUser redmine_user
   AuthMySQLPassword redmine_db_password

   AuthMySQLNameField login
   AuthMySQLPasswordField hashed_password
   AuthMySQLUserTable "users,members"

   AuthMySQLUserCondition "users.id=members.user_id and users.status=1"
   AuthMySQLPwEncryption sha1
```

```
    </Location>
</VirtualHost>
```

The group access is set separately in /etc/svn/authz.ini
If I want only internal users to connect to SVN, I add a custom field "Company" to Users view. And then improve select for AuthMySQLUserCondition directive.

#18 - 2008-10-17 08:53 - Alexey Chepurnykh

Ups, something is wrong with this part :-)

```
AuthMySQLUserTable "users,members"
AuthMySQLUserCondition "users.id=members.user_id and users.status=1"
```

Better I will change this to:

```
AuthMySQLUserTable "users"
AuthMySQLUserCondition "users.status=1"
```

A fresh idea. If I create an abstract project with identifier "svn-access" for only those who uses Subversion I can change condition.

```
AuthMySQLUserCondition "projects.identifier='svn-access' and users.id=members.user_id and projects.id=members.project_id"
```

This solution I adopted from http://yabumaru.jp/archives/10

#19 - 2009-08-26 08:51 - Vinod Singh

Jean-Philippe Lang wrote:

> *Can you explain in more details how your svn/redmine password integration works ? I still don't get it.*

Take a scenario where redmine and svn both are integrated with LDAP. Now users will have same passwords for both and in redmine their passwords will be stored as plain text. Whoever is having access to redmine database can access all corporate services for any user whose passwords are saved in redmine. This looks quite scary.

#20 - 2009-09-01 03:02 - Eric Davis

Vinod Singh wrote:

> *Take a scenario where redmine and svn both are integrated with LDAP. Now users will have same passwords for both and in redmine their passwords will be stored as plain text. Whoever is having access to redmine database can access all corporate services for any user whose passwords are saved in redmine. This looks quite scary.*

Let me correct this in case someone doesn't follow the full conversation. Redmine **user** passwords are encrypted. The username and passwords entered for the **repository** access are not encrypted because Redmine needs to be able to read that password to get the repository.

To address your concerns, are you using LDAP user accounts for the repository module in Redmine? If so, why don't you create a read-only LDAP user for the svn repository that Redmine can use. That way the repository account can have restricted permissions and no user authentication will be stored in the repositories table.

#21 - 2009-09-01 03:39 - Vinod Singh

> *To address your concerns, are you using LDAP user accounts for the repository module in Redmine? If so, why don't you create a read-only LDAP user for the svn repository that Redmine can use. That way the repository account can have restricted permissions and no user authentication will be stored in the repositories table.*

Exactly that is what I did after discovering plain text passwords being stored.

#22 - 2009-09-01 12:47 - Alexey Chepurnykh

> *To address your concerns, are you using LDAP user accounts for the repository module in Redmine? If so, why don't you create a read-only LDAP user for the svn repository that Redmine can use. That way the repository account can have restricted permissions and no user authentication will be stored in the repositories table.*

This is a simple solution. But what if different developers are not supposed to read sourcecode from other projects?

#23 - 2009-10-06 00:04 - Lorenzo Pisani

I can't believe you guys actually think it's ok to store passwords in plain text on the database... just because you need to get it back?
the password should still be encrypted and the key should be in a config file... that is just the 'safer' way to do it...
data in a database is much more insecure!

#24 - 2009-10-06 01:00 - Anonymous

Lorenzo Pisani wrote:

> *I can't believe you guys actually think it's ok to store passwords in plain text on the database... just because you need to get it back?*
> *the password should still be encrypted and the key should be in a config file... that is just the 'safer' way to do it...*
> *data in a database is much more insecure!*

As has already been said many times - the user passwords are encrypted, while the repository passwords are not, and cannot be. Everyone here agrees that access to the database gains them (at least) read access to each repository, but the exact same problem will arise if someone can get access to the key in the config file (as in your example). What makes you believe that the application server is more secure than the database? Even more so, what makes you believe the password file for your repository is more secure?

If the user account passwords were being stored in plain text, then this issue would be valid. The requirement for the repository passwords to be available in plain text for Redmine to authenticate with is something that cannot be worked around. Whether it is encoded in the database or not means nothing - it still needs to be decrypted into plain-text for use, and if Redmine can do it, then anybody with access does. The solution proposed by Alex requires that that everyone changes their SVN implementations to sit behind Apache. This solution also only works for SVN, where as

Redmine supports others repository types.

Moral of the story: create a read-only SVN account that is only used by Redmine, and restrict logins by this user to the IP address of your application server. Nobody is saying that plain-text is good, but there is a lack of anything better.

#25 - 2009-10-06 17:24 - Alexey Chepurnykh

Dear, Nick,

I did not get you. Why do you think that there is a lack of anything better than keeping plain-text passwords?

As Lorenzo wrote,

> *the password should still be encrypted and the key should be in a config file...*

This is a good solution, I am 100% agree with Lorenzo.

#26 - 2009-10-07 01:34 - Anonymous

Alex Chep wrote:

> *I did not get you. Why do you think that there is a lack of anything better than keeping plain-text passwords?*
>
> *As Lorenzo wrote,*
>
> > *the password should still be encrypted and the key should be in a config file...*
>
> *This is a good solution, I am 100% agree with Lorenzo.*

How do you secure the config file in a way that is more secure than your database?

#27 - 2009-10-07 11:37 - Alexey Chepurnykh

What we have in the case now:
1. Insecure passwords in database.

In the offered case:
1. Secure passwords in database
2. Insecure config file with the key

If intruder needs to have passwords which way is more complex to get it?
If the intruder is a developer of the system he can get the passwords in either case.
But what if he is not a developer?
What case in proper perspective has less chances to open the passwords to the intruder?

#28 - 2009-10-07 14:56 - Lluís Vilanova

Don't know for other SCMs, but subversion access is served on three flavours: 1. file: Full and unconstrained access (this is what I use to avoid

storing and arbitrary username/password in redmine

2. svn+ssh: Needs an account in the machine
3. http: Whatever security policy you might come up with

On both the last two cases, redmine can be modified in a similar way to what is explained in #2647 and #3712, such that the client's ID (or even a role) is checked on every repository browse operation from the redmine interface.

In such a case, the security policy applied when browsing code from the redmine interface naturally follows from what has been configured to access the repository outside redmine (at least this is the case for http, which is the one I know of).

This approach completely eliminates the threat of storing passwords in plain as it is now the case for the discussed remote repository access. Even more, all password storage is centralized to wherever the administrator decides (aka redmine DDBB, LDAP, whatever it has been configured in redmine for day-to-day authentication).

#29 - 2011-02-26 14:26 - Jean-Philippe Lang

- Status changed from New to Closed

- Resolution set to Fixed

Encryption support for passwords in the database is now added, see #7411.