Redmine - Feature #387

Support for Linux passwords

2007-07-05 11:27 - Rick Daley

Status:	New		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:			**% Done:**	0%
Category:	Accounts / authentication		**Estimated time:**	0.00 hour
Target version:				
Resolution:				

Description
In our environment, all of the people that use use redMine have Linux accounts on the machine that runs redMine. It would be nice to be able to configure redMine to use passwords from these user's Linux accounts, rather than having them maintain separate passwords.

History

#1 - 2007-07-12 03:47 - Jérôme Fenal

In general, this shouldn't be done by redMine, but done by the
web server, with mod_auth_pam, mod_auth_ldap, etc. and redMine
using the credentials passed to it by the web server.
This way, authentication code can stay clear, without specifics
to one or another auth mean.

#2 - 2007-07-20 08:45 - Jean-Philippe Lang

LDAP auth is commonly used, that's why it's natively supported
in Redmine.
But JÃƒÂ©rÃƒÂ´me is right, Redmine should also provide a way to rely
on the web server to authenticate users and thus allow any auth
mean to be used.

A simple hook (based on HTTP headers) in the Redmine's authentication
mechanism should do the trick.

#3 - 2008-12-18 21:58 - Scott Serr

Has anyone hacked this together? How hard would it be? I imagine by the time someone told me how to do it, they could do it.

Trac had buggy support for mod_auth_pam. (had you restrict that area of the site in your apache.conf) So I suspect it's not dead simple. I rather it not be buggy. I'm looking into bringing up an LDAP server and populating it nightly with ypcat... A hack, but I would know where to start.

Anyone?

#4 - 2008-12-19 18:34 - Scott Serr

After reading what I posted, that sounded whiny.

I would be happy to research this and code it up. I need some direction... possibly an example of this hook in another project (ruby or otherwise).

The whining sound came from thinking that people willing to help would rather just code it themselves.

Here is what I think Jean-Philippe was thinking with the simple hook based on HTTP headers. In Apache, you would make some dummy restricted area location that would use PAM. This would not be directly accessed by the user web browser. But the Login page's backend would go "hit" the server in this location with the user/pass that was entered in the login page. If it gets a ??? error in the HTTP headers, then don't authenticate.

Is this how it should be done? Thanks!

#5 - 2010-11-18 10:39 - Sebastian Roth

There is a ruby PAM module, but it's not even included in Debian.

- http://ruby-pam.sourceforge.net/ruby-pam.html
- http://ruby-pam.sourceforge.net/pam-ruby.html

#6 - 2011-03-23 11:59 - Toshi MARUYAMA

- Category set to Accounts / authentication