Redmine - Feature #6597
Configurable session lifetime and timeout

2010-10-07 13:43 - Frank Helk

Status:	Closed	**Start date:**	2010-10-07
Priority:	Normal	**Due date:**	
Assignee:	Jean-Philippe Lang	**% Done:**	0%
Category:	Accounts / authentication	**Estimated time:**	0.00 hour
Target version:	2.1.0		
Resolution:	Fixed		

Description

I've observed that even when I let the browser open overnight, the redmine session does never time out even when no action is taken.

I suspect that this is intended ("it's no bug, it's a feature"), but I consider this to be a security risk. Maybe I've missed something in the docs, but I'm not aware of such a setting.

Maybe that could be made configurable ?
If it is already, the default timeout value should be set not to be infinite.

Associated revisions

Revision 9797 - 2012-06-10 15:16 - Jean-Philippe Lang

Configurable session lifetime and timeout (#6597).

History

#1 - 2010-10-07 23:16 - Felix Schäfer

Have you activated autologin? See [[RedmineSettings#Autologin]].

#2 - 2010-10-08 09:32 - Frank Helk

- File AuthSettings.png added

No. Autologin is disabled.

See attached settings snippet.

#3 - 2010-10-08 18:41 - Felix Schäfer

IIRC the rails default is not to expire cookies, you will want to put something like :expire_after => 2.hours in your config/initializers/session_inirializer.rb (or whatever it was called) next to the session key.

#4 - 2010-10-12 11:53 - Frank Helk

I've found some conversation about that on the web, but I havn't been able to find something detailled ...

I tried to add the directive to the *redmine/config/initializers/session_store.rb*, which then looked like this (comment lines stripped):

```
ActionController::Base.session = {
  :session_key => '_redmine_session',
  :secret => 'aa5010c9a255c1b60dbec7b81e8f3f42de1baa8760677724e81c3e425c5b1fa916d422931d99a2f2',
  :expire_after => 2.hours
}
```

but that rendered my redmine installation inaccessible after restart of redmine. Any hint what I did wrong, or where to find related documentation ?

#5 - 2010-10-12 22:17 - Felix Schäfer

This looks ok, maybe you have some problems with the local time on the server, or something is skewed there. Could you hit your redmine install with curl -v to see what the expire date of the cookie is, and if that date translated to localtime is still somewhere in the future? That's what it looks like on my dev machine with :expire_after => 2.hours:

```
$ date -u && curl -v http://localhost:3000
Di 12 Okt 2010 20:21:18 UTC
…
Set-Cookie: _redmine_session=SomeCookie; path=/; expires=Tue, 12-Oct-2010 22:21:18 GMT; HttpOnly
```

Which looks ok to me.

#6 - 2012-05-12 14:02 - Jean-Philippe Lang

- Tracker changed from Defect to Feature
- Subject changed from Session timeout to Configurable session lifetime and timeout
- Category changed from UI to Accounts / authentication
- Assignee set to Jean-Philippe Lang
- Target version set to 2.1.0

Setting :expire_after option on cookie store does not invalidate the session, it just prevents the browser from sending the cookie after an amount of time which is not what we want.

#7 - 2012-05-12 19:42 - Terence Mill

Jean-Philippe Lang wrote:

> *- Sessions: there's no way to expire sessions if you're using the default cookie store for sessions. You can have a look at the active record session store that should support that. I agree that being able to control session life time from within Redmine is a desirable feature (#6597).*
> *- Autologin: the autologin duration is controlled on the server side that's why the cookie "expires" attribute is meaningless (set to 1 year so that it's greater than the actual autologin duration)*
> *- API key: it does not expire but you can regenerate a new one (and thus invalid the previous one) in "My account"*

It would be usefuly to be able to set session exppiration timeout in case of inactivity and general session timeout. First one shall be smaller than last one, of course.
Rest api expiration timeout shall be configurable also. Standard could be "never expires" like is now.
Is there any work around to force relogin after some time, beyond server restart? Maybe if running on jruby on tomcat via web.xml?
The problem (secuity risk) is that if someone leaves pc and browser open, anyone can take over the session.

#8 - 2012-05-13 10:24 - Jean-Philippe Lang

Terence Mill wrote:

> *It would be usefuly to be able to set session exppiration timeout in case of inactivity and general session timeout.*

Yes, that's what I will add. Are these terms correct: *Session inactivity timeout* and *Session maximum lifetime*?

> *Is there any work around to force relogin after some time, beyond server restart?*

Changing the secret token of your cookie store will kill all the sessions, that's the only solution that I see.
Even restarting the server does not invalidate sessions if you're using the cookie store.

#9 - 2012-05-13 12:57 - Terence Mill

> *Yes, that's what I will add. Are these terms correct: Session inactivity timeout* and *Session maximum lifetime*?

I think thats ok, "session timeout" and "session lifetime" would be ok also.

> *Is there any work around to force relogin after some time, beyond server restart?*
>
> *Changing the secret token of your cookie store will kill all the sessions, that's the only solution that I see.*
> *Even restarting the server does not invalidate sessions if you're using the cookie store.*

Brbb.. thats bad. Please provide hints for downwards compatibility for this ehancement (best would be a patch) for redmine 1.4/1.2 also, if thats possible.
For secuity reasons we need this feature as soon as possible (running redmine 1.2 and on the way to 1.3/maybe 1.4), and i am not sure if we will upgrade towards 2.x soon. There are to many espectable plugin incompatibilities for this major upgrade and then migration works will last some time.

Can u make clear what the autologin setting controls in as for this feature? After your comments i don't even undertstand what its good for. I thiught first if autologin is off, its not possible to keep users authenticated beyond browser/server lifetime.

Tx for support.

#10 - 2012-05-13 13:53 - Jean-Philippe Lang

Terence Mill wrote:

> *Brbb.. thats bad. Please provide hints for downwards compatibility for this ehancement (best would be a patch) for redmine 1.4/1.2 also, if thats possible.*
> *For secuity reasons we need this feature as soon as possible (running redmine 1.2 and on the way to 1.3/maybe 1.4), and i am not sure if we will upgrade towards 2.x soon. There are to many espectable plugin incompatibilities for this major upgrade and then migration works will last some time.*

Maybe this feature will be backported to 1.4-stable. But I've just tested active record session store with Redmine 1.4 and it will let you control sessions lifetime. Here are the steps (should work with 1.2 as well):

1. apply r9690
2. run `rake db:sessions:create` to create the migration for the sessions table
3. run `rake db:migrate RAILS_ENV=xxx` to actually create the sessions table
4. in config/environment.rb, add the following line to the configuration block:

```
config.action_controller.session_store = :active_record_store
```

After restart, Redmine now uses the sessions table to store sessions and you can kill sessions periodically with a simple SQL query that deletes old sessions.

> *Can u make clear what the autologin setting controls in as for this feature? After your comments i don't even undertstand what its good for. I thiught first if autologin is off, its not possible to keep users authenticated beyond browser/server lifetime.*

The cookie that holds your session_id is a session cookie which is not preserved when you close your browser. The autologin feature keeps you logged in even after you closed your browser. It uses a persistent cookie that contains a random token. Redmine knows when this token was generated and if it's still valid (based on the autologin duration defined in application settings). This is the common "Remember me" feature that can be seen on many sites when logging in.

#11 - 2012-05-14 09:56 - Frank Helk

Just another aspect on how to solve that ... inspired by my online banking site.

Include a script into every (!) page that does a simple countdown (preferably shown somhere in the page header, sth. like "AutoLogout in hh:mm:ss") along with a button to "Reset" to start. If the countdown is over, the script pops up a warning that leaves an additional minute or so (configurable) to reset the Countdown. If no action is taken, the schript automagically calls the logout sequence.

If very sophisticated behaviour is desired, the script could catch user actions on the page like editing, scrolling, etc. to reset the counter as long as the user is actively working on the page

Any timeout values should be configurable, along with the option to leave out the entire timeout thing.

Looks to me like not that much of coding, and it would be nearly bullet proof.

#12 - 2012-05-29 17:36 - Prof. Dr. YoMan

+1 for the idea of Frank.
Session timeout with visual feedback as usual web-banking-apps have is a really userfriendly solution.

#13 - 2012-05-29 20:16 - Terence Mill

The problem with this client side script is that it can be abused to keep connection alive, till ever. So in every case this is only ok for Session inactivity timeout and never for Session maximum lifetime. This shall be fixed on server config.

#14 - 2012-06-10 15:21 - Jean-Philippe Lang

- Status changed from New to Closed
- Resolution set to Fixed

Feature added in r9797. 2 settings added to control session lifetime and timeout, they are turned off by default. No fancy behaviour on the client side.

#15 - 2012-06-10 19:16 - Terence Mill

JP, can you please commit this also to 1.4.4.
We can jump to 2.x because we need some time to update plugins. On the other hand this is a really importnat issue for security concerns.
Tx!

#16 - 2012-06-10 22:32 - Jean-Philippe Lang

Terence Mill wrote:

> *JP, can you please commit this also to 1.4.4.*

It's backported to 1.4-stable in r9810.

#17 - 2012-06-11 08:58 - Terence Mill

GREAT!

Files

AuthSettings.png	6.19 KB	2010-10-08	Frank Helk