Redmine - Defect #11075

"Manage members" permission allows user to elevate own permissions

2012-06-01 16:36 - Rick Mason

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Permissions and roles	**Estimated time:**	0.00 hour
Target version:			
Resolution:		**Affected version:**	2.0.0

Description

1. Create a role which has only one permission, "Manage members".
2. Create a user who is not in any groups, but is in the new role on a single project.
3. Log in as that user.
4. Go to the only project you can see, and click settings.
5. Click "Edit" next to your username and grant yourself access to any role. This can include a role which has full permissions to the project and its settings.

Granting a user "Manage members" is therefore equivalent to granting them the most powerful role available on a project, because they can elevate their own permissions.

They shouldn't be able to modify their own permissions. Arguably they shouldn't be able to grant permissions higher than their own to anyone else either.

Related issues:

Related to Redmine - Feature #19707: Ability to limit member management to ce...	**Closed**

History

#1 - 2012-06-03 15:39 - Jean-Philippe Lang

> Arguably they shouldn't be able to grant permissions higher than their own to anyone else either.

What should we consider as higher permissions?

Say we have 2 roles with different permissions:

- Manager: add_issues, manage_members
- Developer: add_issues, commit_access

Should a manager not be able to give a developer role to anyone because he doesn't have the commit_access permission himself?

#2 - 2012-06-06 18:14 - Rick Mason

That's a good question and your scenario seems reasonable, so I'm not sure what the answer should be.

Here's the security scenario I was thinking of:

1. Alice wants to do x, but she can't because she doesn't have permission.
2. At the moment she can simply grant herself permission, assuming there's a suitable role set up.

To prevent that, you stop people from changing their own permissions.

1. Alice still wants to do x, but she can't.
2. She talks to Bob, who doesn't have permission to do x either.
3. She grants permission to Bob. Bob does x. Together they've got around the security.

Perhaps the answer could be that "manage members" could be changed to a series of permissions: "manage members of role a", "manage members of role b" and so on. The scenario above is still possible, but when an admin grants Alice the "manage members of x" permission it's clear what exactly is being allowed and Alice might be limited to granting Bob some minor permissions rather than everything available. Going back to your scenario, the Manager might be allowed to manage members of the "commit" role, but not of the "project admin" role.

A simpler answer might be to rename "manage members" to "grant permissions", again so it's clearer what's being allowed.

I'm not saying either one is right, they're just options to consider.

#3 - 2013-02-08 15:01 - Rajko Albrecht

I have the same problem with our redmine install(s), that a projekt manager can get more rights than wanted if he/she is able modify project members. We want some so called "supermanager" be able creating new toplevel projects, where the standard project manager is just able creating subprojects and managing project members. Due current system a standard project manager is able setting himself (or other project members) to the supermanager role.

I think following idea may help:

1. Order of roles may taken from the admin settings page, where you can give roles an order. In default install, the "manager" role has the highest order, 'cause is displayed on top of list.
   And now create a role "supermanager", which may create new top level projects, put it on top of this list (above "manager") so a project manager can never give a project member this role.
2. a project member with rights of mananging user member should never be able to upgrade its role to a higher level or to any other level.

I realy thought, the order of roles in admin page has some more meaning then just beeing a nice display order.

#4 - 2013-10-25 17:04 - Ribald Drobens

I encountered this issue as well and I would tend to Rajko Albrechts simple solution: take the enumeration order of the roles for the role, the manage_members-guy can provide at maximum.

Besides, I would like to add a few more thoughts:

1. Roles by definition describe the relation/permissions, a user (or group) has to/inside **a project**. However, some role-settings refer to **global settings**, like "add project" or "edit public filters". Exploiting this, a user with manage_members can promote himself to some role featuring this permissions and setup global projects as well as global filters everyone can see.

1. The "create subprojects" permission imho only makes sense using manage_members as well. So all roles able to create subprojects can use 1. to also create root-projects.

1. With manage_members set, someone can access the complete list of members and groups. Wouldn't it be better to restrict the visible users/groups, for example only to the members of the top-level project?

#5 - 2013-10-25 17:37 - Ching Ching

Rick Mason wrote:

1. Create a role which has only one permission, "Manage members".
2. Create a user who is not in any groups, but is in the new role on a single project.
3. Log in as that user.
4. Go to the only project you can see, and click settings.
5. Click "Edit" next to your username and grant yourself access to any role. This can include a role which has full permissions to the project and its settings.

Granting a user "Manage members" is therefore equivalent to granting them the most powerful role available on a project, because they can elevate their own permissions.

They shouldn't be able to modify their own permissions. Arguably they shouldn't be able to grant permissions higher than their own to anyone else either.

#6 - 2015-04-25 10:39 - Jean-Philippe Lang

- Related to Feature #19707: Ability to limit member management to certain roles added