How To Create Security Rules

▼ Details

This tutorial explains how to create security rules into ERP5. This is important in ERP systems since you might not want everyone to be able to perform all actions into the system. In our example we will define three roles, the administrator of the forum, the user and the visitors.

Agenda

- Set up different "functions" for forum users
- Change security settings on modules
- Configure a workflow

▼ Details

A basic forum workflow diagram

▼ Details

The forum is functional so far, but only the site administrators can use it properly. You will now learn how to configure the security for your application.

It's important that you understand how security is modelled in ERP5 models, as such you should read these documentation items before proceeding:
ERP5 5A Security Model
How to Design Security

Security is usually modelled in terms of **"who" can "do what" under which "circumstances"**. From the explanation of the workflow states we created before, three kinds of **"actors"** or "functions" (as we will configure later) are going to interact with the Forum:
● A **forum Visitor**, who can read public threads and their posts.
● A **forum Editor or User , who can create new threads, edit them, post replies to them and make them public.**
●A forum Administrator, which is not necessarily a site administrator, and is responsible for moderating the threads: closing them for comments or making them private or sticky.

Add "Forum" category in Function Base Category

▼ Details

In ERP5, when creating new "functions" to be performed by users in a new application, these functions are usually implemented through Categories in the Base Category called Function.

To create them, select **Configure Categories** in the **My Favorites** menu. Then look for the **Function** Base Category. If you can't see it at first, it might be in one of the following pages.

You can also search for it by typing Function in the Title field or function in the ID field in the line with the “cog wheel”, then clicking that wheel.

Once inside the Function Base Category, select **Add Category** in the **Action** menu, then fill in at least the following properties for your new Category:
● **Id:** forum ● **Title:** Forum ● **Codification:** FRM ● **Description:** Function Category for forum users

Don't forget to save your changes.

Forum subcategory: Administrator

▼ Details

From this newly created **Function category**, **subcategories** can be created that represent the actual functions of users in the Forum application.

In the page of **Forum category**, select **Add Category** in the **Action menu** to create our **Forum Administrator function**. Fill in the fields with these values:
● **Id:** administrator
● **Title:** Administrator

● **Codification:** ADM
● **Description:** Forum Administrator

Other Forum subcategories

▼ Details

Go back to the **Forum category** to add two other subcategories:
● Id: user
● Title: User
● Codification: USR
● Description: Forum User that can create new threads

And
● Id: visitor
● Title: Visitor
● Codification: VST
● Description: Forum Visitor, can replies on public threads

Be careful to add the above categories to the **Forum category only**. At the end, going back to the Forum category you should see your newly created Forum functions as in the illustration above.

Create and validate Persons

▼ Details

Now we have created functions for Forum module, we can now create Person objects from **Person module**, and configure them to access the site.

Go to **Main Page >> Persons** and select **Add Person** in the **Action menu**.

In the **View tab**, fill in the First and Last names of the person you just added.

Then in the **Assignment tab**, make sure that the user has credentials for logging into the site by filling in the fields for login, password and password confirmation.

After set login and password, you must validate the person and start his assignment. Otherwise, for the final tests, it will be impossible for us to connect with his login/ password.

To validate, go to the **Action bar** and select **Validate**.

In next slide, we will assign function to the user, then start the assignment.

Assign Functions to Persons

▼ Details

After validating the new Person, we have to assign a forum function to him.

While looking at the Person object, select **Add Assignment** from the **Action menu**. Then select the **Forum/Administrator** value for the **Function** field, and type in a title for this assignment. Also, you must settle a **period for the assignment**.

After you save the form, select **Start Assignment** from the **Action menu**. You should then see the **State** field of this assignment should read **Started**.

Repeat the steps on the previous slide and this one to create two more Persons assigning respectively each of the other two Function categories: **Forum/User** and **Forum/Visitor**. If you want, you can create even more Person objects with these assignments.

Role Mapping: Author and Auditor on Module

▼ Details

Once the **function categories** have been defined, they need to be **mapped to actual roles on the objects**. This is done with **role mapping rules** defined centrally in the **Portal Types** of objects.

The Role mapping infrastructure of Portal Types in ERP5 enables arbitrarily complex rules to be applied when **mapping categories to the "5A" roles of the ERP5 security infrastructure**. In this tutorial, the security mapping rules of the **erp5_dms Business Template** provide for a simple mapping of category functions to security roles.

Go to **Portal Types >> Discussion Thread Module** and click on the **Action tab** select **"Add Role Information"** . Then fill the form with the following information:
Title: User
Roles:
Author
Auditor
Description: Forum Administrators and Users are allowed to access the module and create threads.
Categories:
function/forum/administrator
function/forum/user
This maps the **Author and Auditor roles** to the **Administrator and User function categories**.

Role Mapping: Auditor on Module

▼ Details

Repeat the previous step and add the **Visitor** category.

Title: Visitor
Role: Auditor
Description: Forum Visitors can view the discussion thread module and the public threads inside it.
Category: function/forum/visitor

Don't forget to click on **"Update Local Roles"** back in **Action menu** in **Discussion Thread Module** after you've added all the role mappings.

Role Mapping: Assignor on Thread

▼ Details

Besides setting up role mapping on the **Discussion Thread Module**, we also need to set up role mapping on the **Discussion Thread** objects themselves.

Navigate to **Portal Types >> Discussion Thread** and click on the **Action tab** select **"Add Role Information"**. Then fill the form with the following information:
● **Title:** Administrator
● **Role:** Assignor
● **Description:** Forum Administrators are allowed to modify any thread and change its status.
● **Category:** function/forum/administrator

Role Mapping: Author and Auditor on Thread

▼ Details

Use the same form to add role mapping for the **User** category in **thread**:

● **Title:** User
● **Roles:**
Author
Auditor
● **Description:** Forum users can reply to thread posts.
● **Category:** function/forum/user

Role Mapping: Auditor on Thread

▼ Details

For the last role mapping on **Discussion Thread**, **Visitors** need to be able to read the threads, even if they can't do anything else. So add the following role mapping to Discussion Thread as well:

● **Name:** Visitor
● **Role:** Auditor
● **Description:** Forum visitors read threads with their posts.

● **Category:** function/forum/visitor

If you have existing Threads and don't intend to delete them, don't forget to click on **"Update Local Roles"** on this portal type as well to synchronize the role mapping on those instances after you've added all the role mappings.

Role Mapping on Post: Enable Local Role Acquisition

▼ Details

We could also add role mappings for the **Discussion Post portal type**, but since it is basically an embedded object into discussion threads and it is not manipulated individually, it can inherit the role mappings of its container, which will be a **Discussion Thread**.

Navigate back to **Portal Types >> Discussion Post**. Then enable the **"Acquire Local Roles"** checkbox present in Properties.

Don't forget to click on **"Update Local Roles"** afterwards.

Map Roles for Person and its Module

▼ Details

Since the discussion forum displays **information on the Persons** that posted threads or replies, we need to enable **access for the Person objects** to the users of the Discussion Forum. And since Person objects are contained inside the **Person Module**, access to it also needs to be enabled.

This could be accomplished in many ways, but the most convenient for now is to map the **Auditor role** to the **forum function categories** in the Portal Types of both **Person and Person Module** to make it easier to export these settings later.

Go to **Portal Types >> Persons** (if you can't find it, put Person on the ID field) and go to **Action tab** and select **"Add Role Information"**. After that:

● **Title:** Forum User
● **Role:** Auditor
● **Description:** Forum users can see information on Persons.
● **Category:**
function/forum/visitor
function/forum/user
function/forum/administrator

Do not forget to do the same with **Person Module**.

Select Workflow Permissions

▼ Details

The security of the discussion threads will be handled by the **workflow** we defined before. For that we need to enhance it with security configuration. This is done in two steps: first, we select which **permissions** to control with the **workflow**, and then, for each workflow **state**, we define how those **permissions** are mapped to the **portal roles**.

Go to **/portal_workflow**, and click on the **discussion_thread_workflow**. From there, open the **Permissions tab**, and make sure the following permissions are listed and add them if they are not:

● **Access contents information** → Regulates who can access properties of an object.
● **Add portal content** → Regulates who can create new objects within another object. In this case, new Discussion Posts inside a Discussion Thread.
● **Delete objects** → Regulates who can delete objects from inside another object
● **Modify portal content** → Regulates who can modify an object
● **View** → Regulates who is allowed to visit an object directly with the browser

Permissions on States draft, public, sticky

▼ Details

The permissions we just select can now be configured in each workflow state. On the list of state at the bottom of the workflow main page, select the **draft** state and adjust the permissions as illustrated on the first illustration above. They

establish that only a manager and an owner can look at a thread and make changes to it.

Make sure to always **uncheck** the top checkboxes **"Acquire permission settings"**. When checked, the security configuration will be inherited from the parent object.

At this point it is important to remember that the **Owner** role is automatically granted to the user who created the thread.

Then go back to the **States section**, select the **public** state and adjust the permission mappings according to the second illustration above.

In this state, **Authors** can **"Add portal content"** meaning they can add Discussion Post objects to a Discussion Thread object.

Adjust the permission mappings for the **sticky** state the same way you did for the public state.

Since the Discussion Thread Module Portal Type has a role mapping giving **Author** role to the forum/user function category, this means that forum users can post replies to public threads.

Notice that **Auditors**, which are mapped to **forum/visitors** don't have that permission, and so they can't post replies in this configuration.

Permissions on States closed, hidden

▼ Details

Accordingly, adjust the permissions for states **closed** and **hidden** according to the illustrations above.

You will notice that, in the closed state, neither **Author** nor **Owner** can Add portal content any longer, whereas in the hidden state, the Auditor role loses his View and Access contents information privileges, while the Author role loses View, Access contents information and Add portal content rights.

Guard the Transitions

▼ Details

Of course, it's not much use configuring the permissions in the workflow if any user can publish a thread and then enable the other users to post to it.

So navigate back to the **discussion_thread_workflow**, click on the **Transitions tab** and select the **close** transition. In the Properties form there is a **Guard** area with fields for criteria for protecting a transition. If none of these criteria are defined, anyone can execute this transition. If more than one of criteria is defined, all defined criteria must be met for the transition to be allowed.

Due to the security model described earlier, the easiest way to protect the **close transition** is by restricting it to the **Owner role**, so that only the original authors put this value in the Role field. The same should be done for the **close_action** field.

An forum user will want to create a new thread from draft state, so **publish transition** must allow users to do this action.

As for the other transitions, only the forum administrator should perform them. Since we defined that forum administrators get the **Assignor role** inside the Discussion Threads. So now place the Assignor value this time in the Role field of **all the other transitions**.

Guard Publish Transition

▼ Details

We have set **Owner** to the publish action, so that a user who wants to create a new thread will be able to publish it.

However, a hidden thread must not be published by a user. A simple way to achieve this behavior is to add a new transition named "**unhide**" to let the **Assignor** publish the hidden one. Remember that you need to create two transitions, and set their properties correctly: unhide and unhide_action transitions.

After having created the transitions, do not forget to disallow the **Publish** transition an allow the **Unhide** transition on the **Hidden** state, as shown above.

Update security settings

▼ Details

After setting Role in all transitions, you should have a Transitions list as shown in the slide.

To make sure all existing thread objects acquire the permissions you have defined, navigate back to the **discussion_thread_workflow** and click on the **Update Security Roles** action.