Authentic 2 - Development #5262

Manage authorizations to connect to a service provider

12 août 2014 15:15 - Frédéric Péters

Statut:	Rejeté	**Début:**	12 août 2014
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:	2.2.0		
Patch proposed:	Non	**Planning:**	

Description

At the moment all users are allowed to connect to all service providers; in many cases it would be useful to only present some service providers to some users. (the cdg59 extension to authentic1 did that, and it has again been discussed last week at Certivox).

Demandes liées:

Lié à Publik - Project management #10245: Gérer les autorisations de se conne...	**Fermé**	**09 mars 2016**
Lié à Authentic 2 - Development #15456: Contrôle d'accès au SSO basé sur les ...	**Fermé**	**16 mars 2017**
Bloqué par Authentic 2 - Development #751: Improve the manager based on RBAC	**Fermé**	**12 octobre 2011**

Historique

#1 - 12 août 2014 16:04 - Benjamin Dauvergne

I don't think cdg59 is a good inspiration for this feature and there is already another ticket open on the CUD projet for exactly the same thinkg #4775 and there is the very general and old ticket #751.

My current idea woul be to have some kind of permission framework a bit like the one from Django but stored permissions would be by objects and objects concerned would be specified by backends. Permission could be attached to roles (a real role object). Roles could be attached to user or groups. Algebraic data types could be:

```
Permission = (Action, Object) # ex. (Login, SAML provider#01)

Role = (Name, Slug, [Permission])

UserRoleMapping = (User, Role)
GroupRoleMapping = (Group, Role)
```

IdP backends would provide basic actions for their service providers 'login/logout', 'defederation', etc.. but also maybe later specific actions created by administrators that would be more like roles on the sp side like 'administer', 'access-back-office'.

I'm gonna link this ticket to #751 where general discussion should continue and Mickaël should get involved since I think he has a lot of ideas on all of this.

#2 - 12 août 2014 16:24 - Frédéric Péters

> I don't think cdg59 is a good inspiration for this feature [...]

It represents a use case that should be possible, however the feature is implemented.

> My current idea woul be to have some kind of permission framework [...]

I referenced the permission framework in an email, I also think it would be an appropriate direction. (I'll skip details here).

#3 - 12 août 2014 17:47 - Benjamin Dauvergne

Frédéric Péters a écrit :

> I don't think cdg59 is a good inspiration for this feature [...]

> It represents a use case that should be possible, however the feature is implemented.

CDG59 data model is hierarchical (but with a little bit flat hierarchy) I think an ORBAC approach would cover the CDG59 use case, but Mike could confirm this better than me.

#4 - 06 mars 2015 15:28 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

- Priorité changé de Normal à Haut

#5 - 06 mars 2015 16:37 - Benjamin Dauvergne

- Version cible mis à future

#6 - 06 mars 2015 16:37 - Benjamin Dauvergne

- Version cible changé de future à 2.1.12

#8 - 17 mars 2015 16:58 - Benjamin Dauvergne

- Version cible changé de 2.1.12 à 2.1.13

#9 - 23 mars 2015 16:31 - Benjamin Dauvergne

- Version cible changé de 2.1.13 à 2.2.0

#10 - 22 juin 2015 15:44 - Benjamin Dauvergne

- Priorité changé de Haut à Normal

#11 - 05 octobre 2015 10:33 - Benjamin Dauvergne

New plan for this ticket, in a first time we should add an authorize(user) method to the Service class, it should throw PermissionDenied with a translatable status message if it fails (raise PermissionDenied(_('whatever'))).

A authorized_roles M2M field should be added to service, when not empty the authorize() method should check that the given user is part of those groupes, if not it raises.
A boolean should be shown stating if in case of permission denied, SSO should continue with an error response of if the user should be returned to the homepage with an error message shown.

On SAML 2.0 side the authorize method should be called in the sso code path and handled correctly.

#12 - 05 octobre 2015 10:33 - Benjamin Dauvergne

It should also be integrated in the CAS IdP.

#13 - 12 octobre 2015 11:43 - Benjamin Dauvergne

- Assigné à changé de Benjamin Dauvergne à Josué Kouka

#14 - 16 octobre 2015 15:58 - Josué Kouka

- Statut changé de Nouveau à En cours

#15 - 20 octobre 2015 12:22 - Josué Kouka

- Fichier 0001-handling-of-connection-through-a-service-provide-5262.patch ajouté

- Fichier 0001-update-french-translation-5262.patch ajouté

- Patch proposed changé de Non à Oui

Patch submission

- feature
- test
- translation

#16 - 21 octobre 2015 11:02 - Benjamin Dauvergne

- service.roles is not related to authorization, it links services to roles visible only to them, it's what i call "service's roles", you really need to add a new m2m field named for example "authorization_roles"
- this new field must be editable from form on /manager/services/<id>/
- as a general rule do not use the signals for internal business rule it should be used only for ad-hoc extension, i.e. django is right to use signals, an extension app for Django is right to do it, but an application like Authentic should not
- authorization should be done by a direct call from SAML and CAS code to the service.check_authorized() method and it should throw

PermissionDenied with a translatable message as sole argument (as stated in my previous comment)
- idp_signals.authorize_service is deprecated as design is wrong and interface is complicated
- user in role.members.all() does not work as expected as it ignores inherited members, you must do somehthing like (user.roles_and_parents() & service.authorized_roles.all()).exists()
- I would like CAS support to be implemented as part of this ticket (code is a lot simpler than for SAML 2.0); for CAS message should be shown to user using django.contrib.messages as there is no way to report an error through the CAS response
- An option on the service object should allow to force showing the message directly on the IdP instead of using a protocol way of returning it to the SP (as for example mod_mellon does not show statusMessage to users)
- Another option should allow not responding at all to the user but returning to the IdP homepage (some SP does not handle at all any error returned by SAML)

#17 - 21 octobre 2015 17:32 - Josué Kouka

- Fichier example.png ajouté

Benjamin Dauvergne a écrit :

- service.roles is not related to authorization, it links services to roles visible only to them, it's what i call "service's roles", you really need to add a new m2m field named for example "authorization_roles"
- this new field must be editable from form on /manager/services/<id>/

Adding m2m fied in **Service** related to **role** won't work since in **authentic2.a2_rabc.models.Role** there's a **ForeignKey** related to **Service**. So, my suggestion is to add a **BooleanField** to **Role Model** such as **authorized**.

example.png
Since the relationship between Roles and Service is defined by a **Foreign Key**, i guess that the only roles which can be **authorized** are Roles related to the service and not roles from other services. Thus all roles from a services will be listed there and those with the **authorized** attribute checked should have a their member allowed to connect from this SP

#18 - 22 octobre 2015 12:04 - Benjamin Dauvergne

Josué Kouka a écrit :

> Benjamin Dauvergne a écrit :
>
> - service.roles is not related to authorization, it links services to roles visible only to them, it's what i call "service's roles", you really need to add a new m2m field named for example "authorization_roles"
> - this new field must be editable from form on /manager/services/<id>/
>
> Adding m2m fied in **Service** related to **role** won't work since in **authentic2.a2_rabc.models.Role** there's a **ForeignKey** related to **Service**. So, my suggestion is to add a **BooleanField** to **Role Model** such as **authorized**.

I don't see any problem here, you can add as much relations as you want between two models, just fix the related name[1] if there is a collision.

[1]: https://docs.djangoproject.com/en/1.8/ref/models/fields/#django.db.models.ForeignKey.related_name

> Since the relationship between Roles and Service is defined by a **Foreign Key**, i guess that the only roles which can be **authorized** are Roles related to the service and not roles from other services. Thus all roles from a services will be listed there and those with the **authorized** attribute checked should have a their member allowed to connect from this SP

No, authorized is not a property of the role but a relation between a role and a service. I don't want to limit authorized roles arbitrarily to those linked to the service, any role can be authorized for a service.

#19 - 09 mars 2016 10:17 - Benjamin Dauvergne

- Lié à Project management #10245: Gérer les autorisations de se connecter à un service dans authentic ajouté

#20 - 14 septembre 2016 05:11 - Pierre Cros

Needed in Belgium to allow different authentication levels (eID)

#21 - 14 septembre 2016 07:48 - Frédéric Péters

Nope, it's not related.

#22 - 15 septembre 2016 10:56 - Benjamin Dauvergne

- Assigné à changé de Josué Kouka à Benjamin Dauvergne

Je reprends ce ticket.

#23 - 01 avril 2017 13:15 - Frédéric Péters

- Patch proposed changé de Oui à Non

#24 - 03 mai 2017 13:39 - Frédéric Péters

- Lié à Development #15456: Contrôle d'accès au SSO basé sur les rôles ajouté

#25 - 25 juin 2017 19:19 - Frédéric Péters

- Statut changé de En cours à Rejeté

Remplacé par #15456

Fichiers

0001-handling-of-connection-through-a-service-provide-5262.patch	5,88 ko	20 octobre 2015	Josué Kouka
0001-update-french-translation-5262.patch	49,4 ko	20 octobre 2015	Josué Kouka
example.png	191 ko	21 octobre 2015	Josué Kouka