Authentic 2 - Development #42086

L'opération d'auto-administration des rôles devrait être MANAGE_MEMBERS_OP pas CHANGE_OP

24 avril 2020 16:42 - Benjamin Dauvergne

Statut:	Fermé	**Début:**	24 avril 2020
Priorité:	Normal	**Echéance:**	
Assigné à:	Valentin Deniaud	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Les seuls qui doivent avoir CHANGE_OP sont les administrateurs de **tous** les rôles (soit globalement soit au niveau de chaque OU) et ceci via ADMIN_OP.

Demandes liées:

Lié à Authentic 2 - Development #20513: Ajouter une permission explicite pour...	**Fermé**	**08 décembre 2017**

Révisions associées

Révision 3e197664 - 21 août 2020 15:54 - Valentin Deniaud

a2_rbac: change self admin permission to manage_members (#42086)

Historique

#1 - 24 avril 2020 16:43 - Benjamin Dauvergne

- Lié à Development #20513: Ajouter une permission explicite pour gérer les membres d'un rôle ajouté

#2 - 24 avril 2020 16:43 - Benjamin Dauvergne

- Assigné à mis à Valentin Deniaud

#3 - 12 mai 2020 12:04 - Valentin Deniaud

- Fichier 0001-wip.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Je veux bien un avis sur l'approche, assez simple au demeurant, basée sur la description du ticket qui dit qu'une permission CHANGE d'un rôle sur lui même n'a jamais lieu d'exister.

#4 - 18 mai 2020 15:40 - Benjamin Dauvergne

Ça m'a l'air ok.

```
        new_perm = Permission.objects.filter(operation=new_op, target_ct=ct, target_id=role.pk).first()
```

Pas nécessaire si tu mets ou__isnull=True, à cause de l'index d'unicité, donc un get_or_create() suffira, je pense plus propre d'enlever l'ancienne et d'en créer une nouvelle, les permissions sont des objets partagés.

#5 - 19 mai 2020 17:43 - Valentin Deniaud

Benjamin Dauvergne a écrit :

> je pense plus propre d'enlever l'ancienne et d'en créer une nouvelle, les permissions sont des objets partagés.

Mais est-ce que role1 a le droit d'avoir CHANGE_OP sur role2 ? Le titre du ticket dit que peut-être et la description que non.

Aussi, un truc que j'aimerais bien comprendre, a2_rbac/models.py :

```
306         self.permissions.through.objects.get_or_create(role=self, permission=self_perm)
```

J'ai trouvé la doc, https://docs.djangoproject.com/en/1.11/ref/models/fields/#django.db.models.ManyToManyField.through

Mais elle ne me dit pas quel intérêt il y a à faire ça dans ce cas là, ni la différence par rapport à un self.permissions.add(self_perm).

#6 - 19 mai 2020 19:21 - Benjamin Dauvergne

Valentin Deniaud a écrit :

> Benjamin Dauvergne a écrit :
>
> > je pense plus propre d'enlever l'ancienne et d'en créer une nouvelle, les permissions sont des objets partagés.
>
> Mais est-ce que role1 a le droit d'avoir CHANGE_OP sur role2 ? Le titre du ticket dit que peut-être et la description que non.

Je vais simplifier alors peut-être : role.get_admin_role() ne doit pas avoir la permission CHANGE_OP sur role.

> Aussi, un truc que j'aimerais bien comprendre, a2_rbac/models.py :
> [...]
> J'ai trouvé la doc, https://docs.djangoproject.com/en/1.11/ref/models/fields/#django.db.models.ManyToManyField.through
> Mais elle ne me dit pas quel intérêt il y a à faire ça dans ce cas là, ni la différence par rapport à un self.permissions.add(self_perm).

Il n'y a pas de différence.

#7 - 26 mai 2020 10:20 - Valentin Deniaud

- Fichier 0001-a2_rbac-change-self-admin-permission-to-manage_membe.patch ajouté

Benjamin Dauvergne a écrit :

> Je vais simplifier alors peut-être : role.get_admin_role() ne doit pas avoir la permission CHANGE_OP sur role.

Oui ok ma question n'avait pas trop de sens, on retombe sur un cas déjà traité.
Et du coup je me demande si la migration ne pourrait pas juste être un Permission.objects.filter(operation=change_op, target_ct=role_ct).update(operation=manage_members_op) (mais dans le doute je reste sur mon approche en ciblant spécifiquement les rôles auto-administrés).

> Pas nécessaire si tu mets ou__isnull=True, à cause de l'index d'unicité, donc un get_or_create() suffira, je pense plus propre d'enlever l'ancienne et d'en créer une nouvelle, les permissions sont des objets partagés.

Un scénario où ça va casser (#42179 inside), c'est si la permission existe déjà avec l'ou. Dans ce cas, crash dans has_self_administration car la migration a créé la même permission sans ou, du coup le get sur (op=manage_members, target=role) renvoie deux permissions.

Donc soit attendre le fix qui fait que get_admin_role ne met plus l'ou de la permission + réparation des permissions existantes, soit modifier has_self_administration. Je pars sur cette deuxième option.

#8 - 20 août 2020 16:21 - Benjamin Dauvergne

Rebasé je vais relire quand aura buildé.

#9 - 21 août 2020 16:33 - Benjamin Dauvergne

- Statut changé de Solution proposée à Résolu (à déployer)

```
commit 3e197664ae314fd02bef8ab434af8cd04d249d30
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date:   Mon Apr 27 15:03:07 2020 +0200

    a2_rbac: change self admin permission to manage_members (#42086)
```

#10 - 24 août 2020 19:16 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Solution déployée

Fichiers

0001-wip.patch	4,3 ko	12 mai 2020	Valentin Deniaud
0001-a2_rbac-change-self-admin-permission-to-manage_membe.patch	11,5 ko	26 mai 2020	Valentin Deniaud