Hobo - Development #63523

gestion centralisée des accès aux API / Infrastructure minimale pour accès HTTP basic

04 avril 2022 16:48 - Emmanuel Cazenave

Statut:	Rejeté	**Début:**	04 avril 2022
Priorité:	Normal	**Echéance:**	
Assigné à:	Emmanuel Cazenave	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Des objets APIAccess (login/mot de passe) qui se propagent via hobo.json and co.

Une commande management pour créer ces objets.

Une classe d'authentification django-rest-framework qui les exploite.

Demandes liées:

Lié à Hobo - Support #66662: Accès API : deuxième round	**Rejeté**	**27 juin 2022**

Historique

#1 - 04 avril 2022 17:40 - Emmanuel Cazenave

Emmanuel Cazenave a écrit :

> Des objets APIAccess (login/mot de passe) qui se propagent via hobo.json and co.

D'un mail, Fred:

> Perso je serais pour introduire une méthode qui ne passe pas par
> rabbitmq; i.e. comme le provisionning HTTP pour les utilisateurs,
> sauf que ce serait hobo qui lancerait les notifs, (spooler etc.).
>
> Côté réception je serais même pour partager l'endpoint,
>
> +++ b/hobo/provisionning/middleware.py
> @ -50,6 +50,8 @ class ProvisionningMiddleware(MiddlewareMixin, NotificationProcessing):
> return HttpResponseBadRequest()
>
> object_type = notification['objects'].get('@type')
> + if object_type == 'api-access':
> + return self.process_moi_ça()
> issuer = notification.get('issuer')

#2 - 05 avril 2022 12:56 - Frédéric Péters

- Sujet changé de Infrastructure minimale pour accès HTTP basic à gestion centralisée des accès ayx API / Infrastructure minimale pour accès HTTP basic

#3 - 16 mai 2022 16:17 - Emmanuel Cazenave

- Sujet changé de gestion centralisée des accès ayx API / Infrastructure minimale pour accès HTTP basic à gestion centralisée des accès aux API / Infrastructure minimale pour accès HTTP basic

#4 - 16 mai 2022 16:19 - Emmanuel Cazenave

- Statut changé de Nouveau à En cours

- Assigné à mis à Emmanuel Cazenave

#5 - 01 juin 2022 14:28 - Emmanuel Cazenave

- Fichier 0001-start-apiaccess-infrastructure-63523.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

Niveau modèle de donnée, je me calque sur ce qui est fait coté wcs puisque ça devra converger.

Coté commande d'ajout d'un accès, je génère le champ identifiant, sur l'idée de ne pas donner la main là dessus, afin de pouvoir s'en servir comme donnée 'immutable' lors du provisionning.

#6 - 27 juin 2022 15:58 - Emmanuel Cazenave

- Lié à Support #66662: Accès API : deuxième round ajouté

#7 - 27 juin 2022 17:09 - Emmanuel Cazenave

- Fichier 0001-start-apiaccess-infrastructure-63523.patch ajouté

Rebase.

#8 - 29 juin 2022 17:06 - Nicolas Roche

- Statut changé de Solution proposée à En cours

Je suis bluffé de voir que ça fonctionne en l'état sur chrono !

Cependant, le provisionning n'atteint pas authentic.

```
$ hobo-manage tenant_command provision_apiaccess --domain hobo.dev.publik.love
https://passerelle.dev.publik.love:443 "PUT /__provision__/?...
https://combo.dev.publik.love:443 "PUT /__provision__/?...
https://agent-combo.dev.publik.love:443 "PUT /__provision__/?...
https://fargo.dev.publik.love:443 "PUT /__provision__/?...
https://chrono.dev.publik.love:443 "PUT /__provision__/?...
https://bijoe.dev.publik.love:443 "PUT /__provision__/?...
```

Je dirais parce que le mécanisme de provisionning est prévu pour diffuser depuis authentic vers les autres briques et donc qu'on ne précise pas de provisionning-url pour authentic dans la conf des services. J'imagine qu'ici il va y avoir des effets de bords à corriger si on veut ça.

Et sinon je vois quelques limitations :

- La description n'est pas accessible via la commande de création de clé.
- On a rien pour supprimer la clé (parce que je me dit que ça peut manquer en cas de divulgation d'une clé).

Remarque complètement anodine : dans test_apiaccess.py tu as 3 variables inutilisées.

#9 - 30 juin 2022 11:39 - Emmanuel Cazenave

- Fichier 0001-start-apiaccess-infrastructure-63523.patch ajouté

- Statut changé de En cours à Solution proposée

Nicolas Roche a écrit :

> Je dirais parce que le mécanisme de provisionning est prévu pour diffuser depuis authentic vers les autres briques et donc qu'on ne précise pas de provisionning-url pour authentic dans la conf des services. J'imagine qu'ici il va y avoir des effets de bords à corriger si on veut ça.

Je ne me suis pas soucié d'authentic. Si on veut s'en soucier alors effectivement pour l'instant pas d'appels tentés parce que pas de provisionning-url de dispo.

Au delà de ça ça soulève au moins un autre problème, plus bloquant : coté a2 le contrôle d'accès se fait via les rôles. Dans hobo on a accès à l'équivalent rôle qu'est hobo.agent.common.models.Group. Dans #66662, je rajoute la possibilité de lier un accès d'API à des rôles aka hobo.agent.common.models.Group, pour pourvoir gérer du contrôle d'accès.

Et donc si on veut que cette info arrive coté a2, ça va donner lieu à un meli melo Role a2 -> hobo.agent.common.models.Group -> Role a2. Bref je n'ai pas de plan pour qu'on puisse définir des accès aux API qui fonctionnent aussi coté a2 (sauf à éventuellement évacuer cette question de permissions, qu'un accès défini dans hobo permettent d'absolument tout faire dans a2).

> Et sinon je vois quelques limitations :
>
> - La description n'est pas accessible via la commande de création de clé.

- On a rien pour supprimer la clé (parce que je me dit que ça peut manquer en cas de divulgation d'une clé).

J'en resterais bien là, ces fonctionnalités arriveront via #66662.

> Remarque complètement anodine : dans test_apiaccess.py tu as 3 variables inutilisées.

Corrigé.

#10 - 30 juin 2022 12:13 - Frédéric Péters

> Je ne me suis pas soucié d'authentic. Si on veut s'en soucier alors effectivement pour l'instant pas d'appels tentés parce que pas de provisionning-url de dispo.

Pour moi il faut s'en soucier dès maintenant, l'uniformisation des appels, avant les abus récents sur les clés/signatures vers chrono, c'était pour les appels vers les API d'authentic.

#11 - 30 juin 2022 14:49 - Emmanuel Cazenave

Ok.

J'écrivais :

> (sauf à éventuellement évacuer cette question de permissions, qu'un accès défini dans hobo permettent d'absolument tout faire dans a2).

Et à lire le code de a2, c'est déjà cette approche qui est utilisée pour l'API dans le cas d'une authentification OIDC. Ça simplifie grandement les choses et je ne vois pas d'autre solution.

En partant là dessus, ce que vois comme ajustement pour ça fonctionne coté a2 :

- coté hobo que les appels de provisionning visent aussi a2
- coté a2 implémenter une mini classe façon authentic2.authentication.OIDCUser qui donne permission de tout faire, qui sera utilisée quand les identifiants d'une requête matchent avec un APIAccess

Je ferai des tickets dédiés, ce ticket me semble toujours bon pour relecture (sous réserve d'assentiment du developer council d'EO sur l'approche).

#12 - 30 juin 2022 15:27 - Frédéric Péters

> permission de tout faire

Je ne pense pas que créer un accès pour un tiers qui voudrait mettre un ics dans son outlook doive lui permettre de lister/supprimer tous les utilisateurs et rôles.

#13 - 30 juin 2022 16:03 - Emmanuel Cazenave

- Statut changé de Solution proposée à Nouveau

Ça s'entend mais je n'ai pas de plan pour arriver à ça en partant de hobo.

Je dirais qu'on peut mettre ce ticket et celui qui suit à la poubelle, et qu'il y a à élaborer un autre plan où les accès API se définissent dans a2 et sont diffusés vers les autres briques. Je vais faire un mail.

#14 - 06 juillet 2022 12:11 - Emmanuel Cazenave

- Statut changé de Nouveau à Rejeté

Changement de plan, dont #66985 est le début.

Fichiers

0001-start-apiaccess-infrastructure-63523.patch	22,6 ko	01 juin 2022	Emmanuel Cazenave
0001-start-apiaccess-infrastructure-63523.patch	22,6 ko	27 juin 2022	Emmanuel Cazenave
0001-start-apiaccess-infrastructure-63523.patch	22,5 ko	30 juin 2022	Emmanuel Cazenave