Authentic 2 - Development #71823

idp_oidc : faire que l'accès des clients OIDC à l'API se configure dans le BO via les écrans de configuration des clients d'API

29 novembre 2022 11:25 - Paul Marillonnet

Statut:	Rejeté	**Début:**	29 novembre 2022
Priorité:	Normal	**Echéance:**	
Assigné à:	Paul Marillonnet	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	Non

Description

i.e. ça revient à dire qu'un client OIDC peut aussi être un client d'API.
Par défaut (puisque par défaut OIDCClient.has_api_access := False), les clients OIDC pourraient apparaître dans /manage/api-clients/ désactivés, en gris, et on irait les activer au cas par cas.
Il faut garder en tête que l'accès à l'API authentic par les clients OIDC fait quelques transformations via des hooks, par exemple pour que l'uuid des usagers servis soient le sub que le client reçoit au SSO.

Demandes liées:

Lié à Authentic 2 - Bug #71820: /manage/services/ : ajout les flags d'accès à...	**Fermé**	**29 novembre 2022**
Lié à Authentic 2 - Development #71506: api/rbac : faire que l'endpoint de sy...	**Fermé**	**21 novembre 2022**
Lié à Authentic 2 - Development #71905: /manage/ : sur l'écran de configurati...	**Fermé**	**01 décembre 2022**

Historique

#1 - 29 novembre 2022 11:26 - Paul Marillonnet

- Lié à Bug #71820: /manage/services/ : ajout les flags d'accès à l'API et de gestion des profils à la configuration des clients OIDC ajouté

#2 - 29 novembre 2022 11:42 - Thomas Noël

> i.e. ça revient à dire qu'un client OIDC peut aussi être un client d'API.

Pour moi maintenant qu'on a des clients d'API explicites, ce mélange des genres n'a plus lieu d'être.

#3 - 29 novembre 2022 11:51 - Paul Marillonnet

Thomas Noël a écrit :

> Pour moi maintenant qu'on a des clients d'API explicites, ce mélange des genres n'a plus lieu d'être.

Aucun souci à décider de ça, simplement en l'état du code actuel, retirer l'accès à l'API aux clients OIDC et créer des clients d'API à la place va casser des choses, car les hooks spécifiques à l'API appelée par un client OIDC ne seront plus exécutés :
https://git.entrouvert.org/authentic.git/tree/src/authentic2_idp_oidc/apps.py

Edit: et des choses plus spécifiques encore dans le plugin GLC.

#4 - 29 novembre 2022 12:09 - Paul Marillonnet

Paul Marillonnet a écrit :

> Aucun souci à décider de ça, simplement en l'état du code actuel, retirer l'accès à l'API aux clients OIDC et créer des clients d'API à la place va casser des choses, car les hooks spécifiques à l'API appelée par un client OIDC ne seront plus exécutés :
> https://git.entrouvert.org/authentic.git/tree/src/authentic2_idp_oidc/apps.py
>
> Edit: et des choses plus spécifiques encore dans le plugin GLC.

Donc une idée de plan pour ne pas casser l'existant :
· ajouter une FK de client OIDC associé au modèle de client d'API, avec les modifications d'interface adéquates dans /manage/api-clients/,

· faire que les hooks OIDC, qui s'exécutaient jusque là si

`hasattr(request.user, 'oidc_client')`

s'exécutent maintenant si

`isinstance(request.user, APIClient) and request.user.oidc_client is not None`

· écrire une migration qui :
(1) pour chaque OIDCClient tel que OIDCClient.has_api_access := True, crée le client d'API associé
(2) retire cette colonne OIDCClient.has_api_access.

#5 - 29 novembre 2022 13:57 - Paul Marillonnet

- Statut changé de Nouveau à En cours

- Assigné à mis à Paul Marillonnet

Paul Marillonnet a écrit :

> Donc une idée de plan pour ne pas casser l'existant […]

Sauf avis contraire je vais faire ça.

#6 - 29 novembre 2022 14:11 - Frédéric Péters

Moi ça m'irait qu'il ne soit rien fait dans une précipitation que je trouve inutile ici. Déjà le ticket #71820 pour moi aurait du être refusé, qu'on n'emcombre pas ces interfaces avec cette partie legacy bizarre.

#7 - 29 novembre 2022 14:12 - Thomas Noël

Je vais parler sans trop savoir : pas moyen de faire une migration qui prend les comptes oidc et les envoie dans des APIClientUser ? Puis qui supprime l'existance complète de l'accès aux API via les comptes OIDC ?

#8 - 29 novembre 2022 14:19 - Paul Marillonnet

Thomas Noël a écrit :

> Je vais parler sans trop savoir : pas moyen de faire une migration qui prend les comptes oidc et les envoie dans des APIClientUser ? Puis qui supprime l'existance complète de l'accès aux API via les comptes OIDC ?

Je ne sais pas trop si on pourrait alors conserver les deux cas différents :
· l'appelant est un client d'API, on se contente de retourner les données inchangées ;
· l'appelant est par ailleurs un client OIDC, il faut modifier les données (notamment dans le cas où celles-ci concernent les usagers, changer l'uuid en sub pour prévenir le croisement des données entre clients etc.).

#9 - 29 novembre 2022 14:22 - Paul Marillonnet

Frédéric Péters a écrit :

> Moi ça m'irait qu'il ne soit rien fait dans une précipitation que je trouve inutile ici. Déjà le ticket #71820 pour moi aurait du être refusé, qu'on n'emcombre pas ces interfaces avec cette partie legacy bizarre.

J'ai passé #71820 en considérant que ce n'est pas legacy dans la mesure où, actuellement, configurer un client d'API pour une entité qui est aussi un client OIDC vis-à-vis d'authentic ne fonctionnera pas, il y aura une variation irréconciliable entre les données servies au SSO et celles obtenues via l'API.

#10 - 29 novembre 2022 16:49 - Paul Marillonnet

- Lié à Development #71506: api/rbac : faire que l'endpoint de synchronisation s'adapte aux permissions par OU de l'appelant ajouté

#11 - 30 novembre 2022 15:26 - Paul Marillonnet

J'ai l'impression qu'il n'y a pas consensus ici parce que le plan de #71823-4 est un peu vague, je vais taper quelques lignes et prendre quelque captures d'écran pour que ça prenne une allure plus concrète, ensuite on verra si on poursuit comme ça ou pas.

#12 - 30 novembre 2022 15:54 - Frédéric Péters

C'est l'idée même de mélanger API et OIDC qui ne devrait pas apparaitre dans les interfaces, pour moi. (j'entends qu'il y a du legacy, que ça amène des changements de comportements etc. et là-dessus je n'ai aucun problème à ce que ça reste limité à /admin/).

#13 - 30 novembre 2022 16:12 - Paul Marillonnet

Frédéric Péters a écrit :

> (j'entends qu'il y a du legacy, que ça amène des changements de comportements etc. et là-dessus je n'ai aucun problème à ce que ça reste limité à /admin/).

Perso pas dingue de la situation actuelle où seuls les clients qui ont accès à l'/admin/ au petit bonheur la chance peuvent faire une config client OIDC entièrement fonctionnelle d'eux-mêmes, et où ceux qui jouent le jeu de s'emparer des écrans de /manage/ récemment ajoutés se heurtent à des lacunes de ce côté là.

> C'est l'idée même de mélanger API et OIDC qui ne devrait pas apparaitre dans les interfaces, pour moi.

J'entends l'argument, je suis à cours d'idées de proposition qui iraient dans ce sens. Pour moi il y a la nécessité :
· pour les clients OIDC d'accéder à l'API,
· pour l'API a2 de savoir qu'il faut interposer le système de pseudonymat propre à OIDC sur les données renvoyées lorsque c'est un client OIDC qui appelle.
Ce mélange intrinsèque va forcément se répercuter dans les interfaces, je ne vois pas comment il pourrait en être autrement. Je suis preneur d'idées qui pourraient me détromper.

#14 - 30 novembre 2022 16:20 - Frédéric Péters

> une config client OIDC entièrement fonctionnelle d'eux-mêmes

Mais pour moi une config client OIDC entièrement fonctionnelle existe déjà, sans avoir à y mélanger des bouts d'API.

J'entends (maintenant) que quand je parle de "config client OIDC" je parle juste d'un service OIDC relié à authentic pour faire du SSO, et que tu entends quelque chose d'autre, des trucs d'API pour lesquels je ne sais pas faire la part des choses entre ce qui serait quelque chose officiel OIDC et quelque chose qui a pu être un "bricolage" spécifique Authentic à un moment.

#15 - 30 novembre 2022 16:26 - Paul Marillonnet

Frédéric Péters a écrit :

> > une config client OIDC entièrement fonctionnelle d'eux-mêmes
>
> Mais pour moi une config client OIDC entièrement fonctionnelle existe déjà, sans avoir à y mélanger des bouts d'API.
>
> J'entends (maintenant) que quand je parle de "config client OIDC" je parle juste d'un service OIDC relié à authentic pour faire du SSO, et que tu entends quelque chose d'autre, des trucs d'API pour lesquels je ne sais pas faire la part des choses entre ce qui serait quelque chose officiel OIDC et quelque chose qui a pu être un "bricolage" spécifique Authentic à un moment.

Non, tu as raison, c'est moi qui imaginais ce flag has_api_access nécessaire à l'accès à l'endpoint UserInfo (et dans le cas où il aurait été à False ça aurait voulu dire "ce client ne reçoit que des infos de l'idtoken, il n'est pas autorisé à interroger le fournisseur OIDC indépendamment des séquences de connexion de l'usager"), je viens de vérifier et ce n'est pas le cas, mes excuses.

En fait ce flag n'est utilisé que pour nos bricolages pas du tout officiels OIDC. Je vais revert #71820.

#16 - 30 novembre 2022 16:31 - Paul Marillonnet

- Statut changé de En cours à Rejeté

Ticket fondé sur une vision erronée de ma part quant aux droits d'accès à l'endpoint UserInfo par les clients OIDC, je le rejette.

#17 - 01 décembre 2022 10:24 - Paul Marillonnet

- Lié à Development #71905: /manage/ : sur l'écran de configuration OIDC ne faire apparaître les champs hors-spéc qui pour les superusers ajouté