Authentic 2 - Development #35205

OIDC, implémenter Resource Owner Password Credential Grant

02 août 2019 11:04 - Benjamin Dauvergne

Statut:	Fermé	**Début:**	02 août 2019
Priorité:	Normal	**Echéance:**	
Assigné à:	Paul Marillonnet	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Voir :

https://tools.ietf.org/html/rfc6749#section-4.3

Il doit être étendu pour prendre en plus le slug ou le nom d'une OU.

La réponse doit contenir seulement un id_token.

Le web-service doit être protégé par du throttling pour éviter de servir aux attaques de vérification de mot de passe par force brute.

Révisions associées

Révision dda27fe4 - 24 janvier 2020 14:38 - Paul Marillonnet

idp_oidc: support oauth2 resource owner password credential grant (#35205)

Révision d6c8741f - 27 janvier 2020 18:12 - Paul Marillonnet

idp_oidc: fix typo in authz view warning msg, introduced in #35205

Historique

#1 - 02 août 2019 11:15 - Benjamin Dauvergne

On ajoutera un client_id et un client_secret à ce que demande la spec, un appel typique ressemblera à :

https://idp.com/idp/oidc/token/?grant_type=password&username=jean.dupont&password=1234&ou-slug=saint-michel-les-coquillages&client_id=1234&client_secret=4567

#4 - 15 octobre 2019 17:52 - Paul Marillonnet

- Assigné à mis à Paul Marillonnet

Ok.

#5 - 15 octobre 2019 18:04 - Benjamin Dauvergne

Du calme c'est pas financé, Mike ?

#6 - 23 décembre 2019 14:58 - Paul Marillonnet

- Fichier 0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Je verrais bien quelque comme ça, une lecture très directe de la rfc, modulo la condition qu'on s'était fixée au début à savoir ne délivrer qu'un jeton ID.

#7 - 24 décembre 2019 16:43 - Benjamin Dauvergne

- utiliser django-ratelimit plutôt que exponentialretrytimeout (c'est au client de vérifier si il y a trop d'essais sur un username pas à nous) simplement sur le client_id avec une valeur assez haute ; c'est juste pour limiter le déni de service,
- utiliser exponentialretrytimeout uniquement sur le username+client_id (la vrai IP du client final on ne l'a pas, pas la peine de s'en préoccuper, le

client qui nous interroge devra le faire lui même),
- plutôt que d'augmenter l'indentation dans token() faire 2 méthodes
- quel est le besoin d'avoir une fixture cleartext_pw_user ? on a déjà plein de fixture d'utilisateurs (et leur mot de passe est systématiquement égal au username)

- [redacted]
  ```
  'client authentication failed')
  ```

  c'est sur-indenté

- [redacted]

  on peut arrêter d'utiliser json.dumps il y a un JsonResponse en django 1.11[1]

- [redacted]

  ce sera toujours 0 pour du login/mdp (et ce profil ne supporte que ça)

[1]: https://docs.djangoproject.com/fr/1.11/ref/request-response/#jsonresponse-objects

#8 - 24 décembre 2019 16:43 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

#9 - 08 janvier 2020 15:17 - Paul Marillonnet

Tu as déjà pu tester django-ratelimit? En mode TDD je n'arrive pas du tout au résultat escompté, je ne parviens tout bonnement pas à faire fonctionner le module (pas de positifs à la rate-limitation, que des (faux ou vrais) négatifs).
Je vais creuser un peu dans le code de django-ratelimit pour comprendre ce à côté de quoi je passe, parce qu'à la lecture de la doc mon code m'a l'air convenable.

#10 - 08 janvier 2020 15:23 - Frédéric Péters

django-ratelimit est utilisé dans combo et w.c.s.

#11 - 08 janvier 2020 15:24 - Paul Marillonnet

Merci, je regarde ça.

#12 - 09 janvier 2020 18:22 - Paul Marillonnet

- Fichier 0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch ajouté

- Statut changé de En cours à Solution proposée

Je pense qu'on peut faire quelque chose comme ça.

Seule chose qui m'embête : utiliser l'identifiant de client comme clé pour la rate limitation implique de tenter d'authentifier ce client (si on ne le fait pas, il suffit pour un attaquant de connaître l'identifiant valide d'un client sans pour autant connaître son secret pour le bloquer complètement). Est-ce que ça ne défait pas déjà l'objectif de prévention d'une attaque par déni de service.
Si ce n'est pas le cas, une amélioration possible du patch en l'état serait de mettre en cache le résultat d'authentification obtenu lors de la détermination de la clé pour ensuite le réutiliser, sans avoir à authentifier le client une seconde fois – mais je ne suis pas certain de la pertinence de cette amélioration (le dilemme se résume à cette question : est-ce que (i) une authn + une mise en cache de l'id + une requête dans le cache pour retrouver l'id + une requête ORM pour retrouver le client c'est mieux que (ii) deux authn ?).

#13 - 09 janvier 2020 18:32 - Paul Marillonnet

Paul Marillonnet a écrit :

> mais je ne suis pas certain de la pertinence de cette amélioration

Et si on me le demande à la relecture, je peux aussi taper un script qui compare en temps d'exécution et en nombre de requêtes sql les versions avec et sans "amélioration".

#14 - 09 janvier 2020 18:43 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

(Des effets de bord entre les tests dûs à l'usage de mêmes clés de limitation, je revois ma copie.)

#15 - 09 janvier 2020 22:00 - Benjamin Dauvergne

Il me semble que dans les tests on utilise le backend locmem, donc un simple cache.clear() avant et après le test devraient suffire.

#16 - 10 janvier 2020 10:49 - Paul Marillonnet

- Fichier 0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch ajouté

- Statut changé de En cours à Solution proposée

Benjamin Dauvergne a écrit :

> Il me semble que dans les tests on utilise le backend locmem, donc un simple cache.clear() avant et après le test devraient suffire.

Bien vu, merci.
Nouveau patch ici (avec ta précédente relecture prise en compte).

#17 - 10 janvier 2020 14:37 - Benjamin Dauvergne

- il reste encore des HttpResponse(json.dumps...
- en cas d'erreur la description doit être dans une clé 'error_description', pas 'desc', voir https://openid.net/specs/openid-connect-core-1_0.html#AuthError
- le cas d'erreur ligne 125 n'est pas testé
- il n'y a pas de validation des scopes (à factoriser avec le code d'autorisation)
- tu ne retournes qu'un id_token et pas d'access token, comme on ne propose pas de politique à sujet pour l'instant je renverrai tout
- je m'aperçois au passage que le token_type est faux dans le code actuel, c'est Bearer dans le code et bearer dans la spec

#18 - 10 janvier 2020 14:37 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

#19 - 13 janvier 2020 08:58 - Paul Marillonnet

> - tu ne retournes qu'un id_token et pas d'access token, comme on ne propose pas de politique à sujet pour l'instant je renverrai tout

Oui ok. Le descriptif du ticket indique le contraire, et dans mes souvenirs ça me paraissait logique mais je ne me souviens plus pourquoi :)
Je fais la modification (ainsi que toutes celles demandées dans ta relecture) et je reviens ici.

#20 - 13 janvier 2020 11:02 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Oui ok. Le descriptif du ticket indique le contraire, et dans mes souvenirs ça me paraissait logique mais je ne me souviens plus pourquoi :)

Ouaip c'est vrai, « La réponse doit contenir seulement un id_token » et pareil j'ai oublié la justification alors oublions, c'est juste qu'un access token est plus simple à consommer, mais on pourra imaginer une politique dans le paramétrage pour ce mode (pour les modes "navigateur" c'est dès la demande d'autorisation qu'on précise ce qu'on souhaite comme jeton, le problème c'est que la spécification openidconnect n'est pas très complète sur ce point, en credential grant elle ne précise pas ce qu'on peut ou pas renvoyer ou si ça devrait être configurable).

#21 - 22 janvier 2020 23:49 - Benjamin Dauvergne

- Fichier 0002-ajustements.patch ajouté

- Fichier 0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch ajouté

- Statut changé de En cours à Solution proposée

J'ai fait les ajustements que je suggérais :

- finalement pour Bearer pas de souci, c'est insensible à la casse et c'est bien Bearer la valeur normative
- j'ai ajouté un schamp scopes vide par défaut pour limiter les scopes d'un client ; dans le cas d'un client credentials grant c'est aussi la valeur par défaut
- j'ai réindenté un peu tout

#22 - 23 janvier 2020 00:23 - Benjamin Dauvergne

- Fichier 0003-ajustements-2.patch ajouté

- Fichier 0002-ajustements.patch ajouté

- Fichier 0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch ajouté

- rajouté aux tests le scope minimal manquant "openid", pour avoir le sub
- ajout du test pour le cas ou le flow n'est pas le bon
- modifié create_user_info() pour prendre en compte le cas où openid n'est pas demandé (pas de sub)

#23 - 23 janvier 2020 01:33 - Benjamin Dauvergne

- Fichier 0003-ajustements-2.patch ajouté

- Fichier 0004-more-determinism-in-ratelimit-tests.patch ajouté

- Fichier 0002-ajustements.patch ajouté

- Fichier 0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch ajouté

Des ajustements pour rendre les tests sur les limites de fréquence moins instables.

#24 - 23 janvier 2020 09:37 - Paul Marillonnet

- Statut changé de Solution proposée à Solution validée

Benjamin Dauvergne a écrit :

> Des ajustements pour rendre les tests sur les limites de fréquence moins instables.

Ok (et j'avais déjà fait ce boulot dans ma branche avant-hier).
Merci pour tes modifs et corrections, ack :)

#25 - 24 janvier 2020 11:14 - Paul Marillonnet

- Fichier 0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch ajouté

Voilà, avec ajout d'un test sur tentative d'accès erroné sur l'endpoint d'authz, comme indiqué dans #35205#note-17.
Si c'est bon pour toi Benj, je pousse ça en début de semaine prochaine.

#26 - 24 janvier 2020 11:21 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Voilà, avec ajout d'un test sur tentative d'accès erroné sur l'endpoint d'authz, comme indiqué dans #35205#note-17.
> Si c'est bon pour toi Benj, je pousse ça en début de semaine prochaine.

Je l'avais déjà ajouté "test_credentials_grant_invalid_flow".

#27 - 24 janvier 2020 11:24 - Paul Marillonnet

- Fichier 0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch ajouté

Benjamin Dauvergne a écrit :

> Je l'avais déjà ajouté "test_credentials_grant_invalid_flow".

Pas vu, pardon. Le patch sans mon test redondant.

#28 - 24 janvier 2020 14:40 - Paul Marillonnet

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

```
commit dda27fe4880e800570a09f9531d430eaf7d01003
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date:   Mon Nov 18 17:09:54 2019 +0100

    idp_oidc: support oauth2 resource owner password credential grant (#35205)
```

#29 - 28 janvier 2020 15:18 - Frédéric Péters

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

Fichiers

0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch	18,9 ko	23 décembre 2019	Paul Marillonnet
0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch	19,6 ko	09 janvier 2020	Paul Marillonnet
0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch	19,9 ko	10 janvier 2020	Paul Marillonnet
0002-ajustements.patch	31,1 ko	22 janvier 2020	Benjamin Dauvergne
0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch	19,9 ko	22 janvier 2020	Benjamin Dauvergne
0003-ajustements-2.patch	4,41 ko	22 janvier 2020	Benjamin Dauvergne
0002-ajustements.patch	31,1 ko	22 janvier 2020	Benjamin Dauvergne
0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch	19,9 ko	22 janvier 2020	Benjamin Dauvergne
0003-ajustements-2.patch	4,41 ko	23 janvier 2020	Benjamin Dauvergne
0004-more-determinism-in-ratelimit-tests.patch	5,29 ko	23 janvier 2020	Benjamin Dauvergne
0002-ajustements.patch	31,1 ko	23 janvier 2020	Benjamin Dauvergne
0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch	19,9 ko	23 janvier 2020	Benjamin Dauvergne
0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch	32,5 ko	24 janvier 2020	Paul Marillonnet
0001-idp_oidc-support-oauth2-resource-owner-password-cred.patch	31,8 ko	24 janvier 2020	Paul Marillonnet