w.c.s. - Development #35533

stockage des mots de passe hashés

22 août 2019 10:55 - Thomas Noël

Statut:	Rejeté	**Début:**	22 août 2019
Priorité:	Bas	**Echéance:**	
Assigné à:	Nicolas Roche	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Actuellement wcs sait stocker les champs mot de passe hashés en md5 et sha1 seulement ; c'est assez faible.

On pourrait ajouter les hashers "bien connus" présents dans Django.

Demandes liées:

Lié à Authentic 2 - Development #35482: passer le HASH du mot de passe des no...	**Fermé**	**20 août 2019**

Historique

#1 - 08 septembre 2019 22:04 - Frédéric Péters

1/ on dépend de django 1.11, pas besoin de if django.VERSION >= (1, 10, 0):. 2/ je ne serais pas pour l'exhaustivité, uniquement inclure les formats pertinents, voire même uniquement ajouter le hashage par défaut, PBKDF2PasswordHasher (c'est le cas en 1.11, c'est toujours le cas en 2.1).

#2 - 08 septembre 2019 22:14 - Nicolas Roche

- Fichier 0001-fields-use-django-hashers-to-support-more-hashed-pas.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Merci, tu as lu dans mes pensées !

Je met quand même ici mon patch et mon laïus (au moins pour moi pour m'y retrouver plus tard), mais oui ça me va très bien de n'intégrer que PBKDF2PasswordHasher. On pourra facilement en rajouter d'autres si besoin.

Voici tous les algos disponibles (via Django) en théorie :

```
$ grep 'algorithm = ' ~/src/django/django/contrib/auth/hashers.py
...
    algorithm = "pbkdf2_sha256"
    algorithm = "pbkdf2_sha1"
    algorithm = 'argon2'
    algorithm = "bcrypt_sha256"
    algorithm = "bcrypt"
    algorithm = "sha1"
    algorithm = "md5"
    algorithm = "unsalted_sha1"
    algorithm = "unsalted_md5"
    algorithm = "crypt"
```

L'API ne retournent que les algos qui sont déclarés dans les settings :

```
> from django.conf import global_settings
> global_settings.PASSWORD_HASHERS
[u'django.contrib.auth.hashers.PBKDF2PasswordHasher',
 u'django.contrib.auth.hashers.PBKDF2SHA1PasswordHasher',
 u'django.contrib.auth.hashers.Argon2PasswordHasher',
 u'django.contrib.auth.hashers.BCryptSHA256PasswordHasher',
 u'django.contrib.auth.hashers.BCryptPasswordHasher']

> from django.contrib.auth.hashers import get_hashers_by_algorithm
> get_hashers_by_algorithm().keys()
[u'argon2', u'bcrypt', u'pbkdf2_sha1', u'pbkdf2_sha256', u'bcrypt_sha256']
```

Dans les faits, les algos accessibles via l'API ne sont pas forcément installés :

```
> get_hasher('argon2')
<django.contrib.auth.hashers.Argon2PasswordHasher object at 0x7f19836be510>
> get_hasher('argon2').encode('secret', hasher.salt())
*** ValueError: Couldn't load 'Argon2PasswordHasher' algorithm library: No module named argon2

$ pip install argon2
> get_hasher('argon2').encode('secret', hasher.salt())
*** AttributeError: 'module' object has no attribute 'low_level'

$ pip install django[argon2]
> get_hasher('argon2').encode('secret', hasher.salt())
(ok)
```

Par ailleurs, on peut utiliser les algos disponibles sans passer nécessairement par l'API :

```
> from django.contrib.auth.hashers import SHA1PasswordHasher
> hasher = SHA1PasswordHasher()
> hasher.encode('secret', hasher.salt())
```

Le patch ci-dessous ajoute tous les algos disponibles dans django.
Pour info, j'ai dû modifier l'encodage du salt générique pour que les hasher salted md5 et sha1 autorisent les mots de passe non ascii.

```
*** UnicodeDecodeError: 'ascii' codec can't decode byte 0xe2 in position 0: ordinal not in range(128)'
```

Le hasher argon2 n'est disponible que depuis Django 1.10.
Il n'est pas installé par défaut et est succeptible de lever 2 exceptions.
Je simule un encodage en amont pour détecter par avance ces erreurs afin de ne pas le proposer en option dans le back-office.

#3 - 08 septembre 2019 22:20 - Nicolas Roche

- Statut changé de Solution proposée à En cours

#4 - 09 septembre 2019 12:03 - Nicolas Roche

- Fichier 0001-fields-use-django-hashers-to-support-more-hashed-pas.patch ajouté

- Statut changé de En cours à Solution proposée

Ce patch ajoute le hashage par défaut, PBKDF2PasswordHasher.
Dites-moi si vous en voulez d'autres.

edit: j'ai oublié de modifier le message du commit dans mon patch :

```
fields: add PBKDF2 hashed password format (#35533)
```

#5 - 09 septembre 2019 12:10 - Frédéric Péters

Ok ça vient d'avant mais le code make_encoder() etc. complique les choses, ça pourrait juste être une simple fonction et dessous 'pbkdf2': lambda x: pbkdf2_encode(x).

Aussi ça serait pas mal de lier ce ticket au besoin fonctionnel qui est à l'origine.

#6 - 09 septembre 2019 13:16 - Nicolas Roche

- Lié à Development #35482: passer le HASH du mot de passe des nouveaux utilisateurs aux service web d'A2 ajouté

#7 - 09 septembre 2019 14:04 - Nicolas Roche

- Fichier 0001-fields-add-PBKDF2-hashed-password-format-35533.patch ajouté

> le code make_encoder() etc. complique les choses

oui, excès de zèle

> besoin fonctionnel qui est à l'origine

il n'y en a pas : évoqué (mais non requis) pour signal-publik, et à priori n'y sera finalement pas utilisé.

#8 - 09 septembre 2019 14:13 - Frédéric Péters

Pas très fan d'intégrer du code inutilisé.

Mais à part ça, ok pour le code, mais tant qu'à faire, également modifier le paramétrage par défaut du champ pour exploiter cet algo.

#9 - 09 septembre 2019 14:21 - Nicolas Roche

> Pas très fan d'intégrer du code inutilisé.

aucun problème, je l'ai fait parque Thomas me l'avais demandé (on peut laisser au chaud ici)

> paramétrage par défaut du champ

arf, je ne sais pas où c'est

#10 - 09 septembre 2019 14:23 - Frédéric Péters

> (on peut laisser au chaud ici)

Non, on intègre ou on rejette le ticket.

C'est le

```
    formats = ['sha1']
```

dans les attributs de la classe.

#11 - 09 septembre 2019 15:34 - Nicolas Roche

- Fichier 0001-fields-add-PBKDF2-hashed-password-format-35533.patch ajouté

(merci)

#12 - 10 septembre 2019 18:00 - Nicolas Roche

- Assigné à mis à Nicolas Roche

#13 - 15 mai 2020 15:04 - Thomas Noël

- Statut changé de Solution proposée à Rejeté

Finalement ça ne sera pas utilisé (et certainement jamais) ; annulons cette affaire.

Fichiers

0001-fields-use-django-hashers-to-support-more-hashed-pas.patch	7,93 ko 08 septembre 2019	Nicolas Roche
0001-fields-use-django-hashers-to-support-more-hashed-pas.patch	3,52 ko 09 septembre 2019	Nicolas Roche
0001-fields-add-PBKDF2-hashed-password-format-35533.patch	3,21 ko 09 septembre 2019	Nicolas Roche
0001-fields-add-PBKDF2-hashed-password-format-35533.patch	4,06 ko 09 septembre 2019	Nicolas Roche