w.c.s. - Development #41766

auth http basic, réponse initiale

15 avril 2020 20:26 - Frédéric Péters

Statut:	Fermé	**Début:**	15 avril 2020
Priorité:	Normal	**Echéance:**	
Assigné à:	Frédéric Péters	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Dans un appel API qui autorise l'authentification HTTP, si elle n'est pas présente ou échoue, répondre avec 401 et un entête WWW-Authenticate.

Révisions associées

Révision 84fe1caa - 21 avril 2020 09:18 - Frédéric Péters

api: raise 401 on authenticated API access where basic auth is allowed (#41766)

Historique

#1 - 15 avril 2020 20:35 - Frédéric Péters

- Fichier 0001-api-raise-401-on-authenticated-API-access-where-basi.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

#2 - 16 avril 2020 00:39 - Thomas Noël

get_user_from_api_query_string peut avoir répondu False s'il y a la bonne entête Authorization mais ni email ni NameID dans la query_string pour chercher le user cible. Dans ce cas il ne faut pas répondre 401 mais bien 403.

Il faudrait donc plutôt avoir un if api_user is None and api_name: raise HttpResponse401Error(api_name)

#4 - 16 avril 2020 09:52 - Nicolas Roche

- Statut changé de Solution proposée à Solution validée

Ça marche !

J'obtient bien une 401 avec l'entête qui invite au challenge :

```
Www-Authenticate: Basic realm="ics"
```

Que ce soit dans les navigateurs ou dans thunderbird, L'URL qui précise les accès fonctionne directement, et avec l'URL qui ne les précise pas ils m'invitent à les transmettre.
Pour configurer l'agenda dans thunderbird :

- Nouveau > Agenda
- Sur le réseau
- iCalendar (ICS)

Concernant la remarque de Thomas, je constate que l'on obtient aussi user à None à la requête initiale (quand il n'y a pas encore le header 'Authorization' de passé) parce que les requêtes ne sont pas signées. Donc la correction du test pour renvoyer 403 aux utilisateurs qui ne font pas l'effort de s'identifier comme anonyme fait que le patch ne fonctionne plus.

```
if auth_header and api_name:
    ...
elif not is_url_signed():
    return None
```

Peut-être juste considérer comme anonyme tous les utilisateurs qui ne fournissent pas d'email ou de NameId vide ou non.
Et donc retirer le dernier test :

```
<<<
user = None
..
    elif 'email' in get_request().form or 'NameID' in get_request().form:
        # email or NameID were given as empty to the query string, this maps
        # the anonymous user case.
        return False
return user
---
user = None  # this maps the anonymous user case.
...
return user
>>>
```

#5 - 16 avril 2020 12:21 - Nicolas Roche

- Statut changé de Solution validée à Solution proposée

Je "dévalide" le temps que la remarque de Thomas soit traitée.

#6 - 16 avril 2020 12:28 - Benjamin Dauvergne

Il faudrait découper get_user_from_api_query_string() en deux (en conservant la méthode actuelle quand même) :

- is_api_authenticated(api_name=None) : fait le check d'auth HTTP et de signature Publik
- get_api_user_from_request(request) : regarde les champs Email/NameID

Ainsi on peut gérer tous les cas correctement ici (on pourrait aussi renvoyer un WWW-Authenticate: X-Publik pour la forme, c'est juste pour faire joli).

```
if not is_authenticated(request):
    if get_request().user and get_request().user.is_admin:
        return # grant access to admins, to ease debug
    raise HttpResponse401Error(api_name) <- gérer le cas api_name=None
api_user = get_user_from_request(request)
if not api_user:
    raise AccessForbiddenError('user not authenticated')
```

#7 - 16 avril 2020 13:14 - Frédéric Péters

Je ne compte pas faire évoluer beaucoup le patch, s'il ne plait pas il y aura possibilité de tickets pour gérer les cas qui ne m'auront pas intéressé. Merci.

#8 - 16 avril 2020 13:22 - Thomas Noël

> Je ne compte pas faire évoluer beaucoup le patch

Et ma toute petite proposition en note 2 ? (qui me semble vraiment pas grand chose et suffisante pour que je acke fortement)

#9 - 16 avril 2020 13:56 - Frédéric Péters

Bien sûr, j'allais lire ~~attentivement~~ et je comptais bien faire évoluer le patch, mais je trouvais que ce ticket commençait à être un peu trop chargé en commentaires.

#10 - 16 avril 2020 13:57 - Nicolas Roche

Avec la proposition en note 2 le patch ne fonctionne plus (je l'ai également testée).

#11 - 16 avril 2020 13:58 - Frédéric Péters

Laissez ce ticket tranquille, merci.

#12 - 20 avril 2020 21:01 - Frédéric Péters

- Fichier 0001-api-raise-401-on-authenticated-API-access-where-basi.patch ajouté

Version révisée du patch, qui découpe la fonction get_user_from_api_query_string presque comme évoqué par Benjamin (je ne mêle pas vérification auth basic et signature).

(je ne suis pas fan de la manière dont le court-circuitage admin se fait et il n'était pas testé, maintenant il fonctionne sûr).

#13 - 21 avril 2020 00:45 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

```
    username, password = force_text(base64.decodestring(force_bytes(auth_header))).split(':', 1)
```

Si on nous envoie une chaîne sans ':' ou dont l'encodage base64 est mauvais ça va faire une trace, il faudrait faire ça en deux fois et intercepter ValueError.

#14 - 21 avril 2020 09:18 - Frédéric Péters

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

Modifié pour intercepter ValueError mais sans distinction base64/manque un :.

```
commit 84fe1caa5342076285555180dcdfd2afba988bc2
Author: Frédéric Péters <fpeters@entrouvert.com>
Date:   Wed Apr 15 20:34:27 2020 +0200

    api: raise 401 on authenticated API access where basic auth is allowed (#41766)
```

#15 - 21 avril 2020 12:16 - Frédéric Péters

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

Fichiers

0001-api-raise-401-on-authenticated-API-access-where-basi.patch	3,62 ko	15 avril 2020	Frédéric Péters
0001-api-raise-401-on-authenticated-API-access-where-basi.patch	9,12 ko	20 avril 2020	Frédéric Péters