Authentic 2 - Development #40989

validation d'adresse email avec des caractères pas autorisés dans le domaine

24 mars 2020 20:30 - Frédéric Péters

Statut:	Fermé	**Début:**	24 mars 2020
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Genre email = u'foo@bar\x00', ça va finir par arriver à

```
File "/usr/lib/python2.7/dist-packages/authentic2/forms/fields.py" in validate
  120.         email_validator(value)

File "/usr/lib/python2.7/dist-packages/authentic2/validators.py" in __call__
  65.         mxs = self.check_mxs(hostname)

File "/usr/lib/python2.7/dist-packages/authentic2/validators.py" in check_mxs
  51.                 socket.gethostbyname(idna_encoded)

Exception Type: TypeError at /accounts/register/
Exception Value: gethostbyname() argument 1 must be string without null bytes, not str
```

Révisions associées

Révision 702feb4d - 27 mars 2020 15:33 - Frédéric Péters

validators: use only dnspython to resolve domains (#40989)

Also fix SMTP error value check when testing email adresses with an RCPT check.

Historique

#1 - 24 mars 2020 20:34 - Frédéric Péters

- Fichier 0001-validators-don-t-crash-checking-MX-of-invalid-domain.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Au plus simple, (aussi tapé sur le SaaS de recette)

```
-            except socket.error:
+            except (TypeError, socket.error):
```

mais si quelqu'un veut aller vers plus de formalisme, c'est bien aussi.

#2 - 25 mars 2020 10:46 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#3 - 25 mars 2020 10:46 - Benjamin Dauvergne

- Fichier 0001-validators-don-t-crash-checking-MX-of-invalid-domain.patch ajouté

- Fichier 0004-tests-test-email-validator-40989.patch ajouté

- Fichier 0002-validators-piggyback-on-django-EmailValidator-40989.patch ajouté

- Fichier 0003-validators-check-SMTP-RCPT-error-is-5xx-40989.patch ajouté

- Tracker changé de Bug à Development

J'ai ajouté l'utilisation du EmailValidator classique et des tests sur la validation de domaine et la validation RCPT.

#4 - 27 mars 2020 12:22 - Paul Marillonnet

- Je pense qu'on pourrait carrément hériter la classe du EmailValidator django, au lieu du 'piggyback' qui saute moins aux yeux.
- L'usage du validateur django rend redondantes deux parties du validateur a2 :

```
diff --git a/src/authentic2/validators.py b/src/authentic2/validators.py
index cb2c4d9c3..afecac390 100644
--- a/src/authentic2/validators.py
+++ b/src/authentic2/validators.py
@@ -43,10 +43,6 @@ class EmailValidator(object):
             mxs = [str(mx.exchange).rstrip('.') for mx in mxs]
             return mxs
         except dns.exception.DNSException:
-            try:
-                idna_encoded = force_text(domain).encode('idna')
-            except UnicodeError:
-                return []
             try:
                 socket.gethostbyname(idna_encoded)
                 return [domain]
@@ -56,10 +52,6 @@ class EmailValidator(object):

     def __call__(self, value):
         DjangoEmailValidator()(value)
-        try:
-            hostname = value.split('@')[-1]
-        except KeyError:
-            raise ValidationError(_('Enter a valid email address.'), code='invalid-email')
         if not app_settings.A2_VALIDATE_EMAIL_DOMAIN:
             return True
```

- Je suis dubitatif sur l'usage du paramètre A2_VALIDATE_EMAIL_DOMAIN, car avec l'introduction du validateur django, et en l'état actuel du code, une validation syntaxique du domaine est effectuée même lorsque ce paramètre est à False.

#5 - 27 mars 2020 12:28 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> - Je pense qu'on pourrait carrément hériter la classe du EmailValidator django, au lieu du 'piggyback' qui saute moins aux yeux.

Les arguments du *init*.py matche pas je voulais pas en dépendre, je continue à penser que c'est plus simple d'utiliser la composition que l'héritage ici vu qu'on a pas besoin de passer du paramétrage à DjangoEmailValidator.

> - L'usage du validateur django rend redondant deux parties du validateur a2 :
> [...]

Pour le idna non, c'est nécessaire.

> - Je suis dubitatif sur l'usage du paramètre A2_VALIDATE_EMAIL_DOMAIN, car avec l'introduction du validateur django, et en l'état actuel du code, une validation syntaxique du domaine est effectuée même lorsque ce paramètre est à False.

Ça devrait être renommé en A2_VALIDATE_EMAIL_DOMAIN_BY_RCPT_CHECK parce que c'est ça que ça contrôle.

#6 - 27 mars 2020 12:30 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

> Ça devrait être renommé en A2_VALIDATE_EMAIL_DOMAIN_BY_RCPT_CHECK parce que c'est ça que ça contrôle.

Non je dis une connerie, ça dit bien ce que ça veut dire, on a une validation juste du domaine via le DNS puis sir A2_VALIDA_EMAIL une validation du mail lui même via tentative de RCPT; sans A2_VALIDATE_DOMAIN on a juste la validation syntaxique classique.

#7 - 27 mars 2020 12:37 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> Pour le idna non, c'est nécessaire.

Ok oui, lu trop vite le code de django, ce test d'encodage idna ne s'exécute pas à tous les coups.

#8 - 27 mars 2020 12:38 - Paul Marillonnet

Par contre je pense vraiment que tu peux retirer le try/except dans le deuxième bout de code que je pointais, c'est déjà fait ici.

#9 - 27 mars 2020 12:51 - Benjamin Dauvergne

- Fichier 0002-tests-test-email-validator-40989.patch ajouté

- Fichier 0001-validators-use-only-dnspython-to-resolve-domains-409.patch ajouté

Nouvelle série, j'ai viré complètement l'utilisation de gethostbyname, dnspython fournit déjà un service supérieur (l'encodage IDNA est implicite quand on passe une chaîne unicode comme domaine). J'ai aussi simplifié le control flow autour des settings, c'est plus simple à lire.

Je me suis inspiré de https://gist.github.com/dwinston/eaa57dc198923ef7bcf0ba30c2e3a39f pour simplifier check_mxs().

#10 - 27 mars 2020 14:55 - Paul Marillonnet

Je ne comprends plus ce que tu veux faire dans check_mxs, il y a un souci je crois, à partir du

```
    for record_type in ('AAAA', 'A'):
        […]
```

(record_type qui n'est d'ailleurs pas utilisé dans l'itération.)
J'ai l'impression que cette boucle est en trop, et qu'on ne souhaite pas vérifier les entrées AAAA et A qui ne nous seront pas utiles lors de la tentative de connection dans check_rcpt.

D'ailleurs je serais aussi pour renommer check_mxs en retrieve_mxs ou get_mxs, car (contrairement à check_rcpt) cette méthode n'est pas une vérification (silencieuse en cas de cas de succès, levant une erreur sinon) mais une récupération de données.

Une dernière question (même si ça concerne du code déjà existant), dans check_rcpt :

```
        for server in mxs:
            try:
                # […]
            except smtplib.SMTPServerDisconnected:
                break
            except smtplib.SMTPConnectError:
                continue
```

Pourquoi la première des deux exceptions engendre un break ? Pour moi ça devrait continue pour les deux, non ?

#11 - 27 mars 2020 15:01 - Benjamin Dauvergne

- Fichier 0002-tests-test-email-validator-40989.patch ajouté

- Fichier 0001-validators-use-only-dnspython-to-resolve-domains-409.patch ajouté

Voilà (et la boucle est bonne, si pas de MX on doit utiliser AAAA ou A pour envoyer un mail) :

```
diff --git src/authentic2/validators.py src/authentic2/validators.py
index 8856f2fc..40bc47fc 100644
--- src/authentic2/validators.py
+++ src/authentic2/validators.py
@@ -35,7 +35,7 @@ class EmailValidator(object):
     def __init__(self, rcpt_check=False):
         self.rcpt_check = rcpt_check

-    def check_mxs(self, domain):
+    def query_mxs(self, domain):
         try:
             mxs = dns.resolver.query(domain, 'MX')
             mxs = [str(mx.exchange).rstrip('.') for mx in mxs]
@@ -49,7 +49,7 @@ class EmailValidator(object):

         for record_type in ('AAAA', 'A'):
             try:
-                mxs = dns.resolver.query(domain, 'MX')
+                mxs = dns.resolver.query(domain, record_type)
                 mxs = [str(mx.address).rstrip('.') for mx in mxs]
                 return mxs
             except dns.resolver.NXDOMAIN:
```

```
@@ -74,7 +74,7 @@ class EmailValidator(object):
                     raise ValidationError(_('Invalid email address.'), code='rcpt-check-failed')
                 break
             except smtplib.SMTPServerDisconnected:
-                break
+                continue
             except smtplib.SMTPConnectError:
                 continue

@@ -83,7 +83,7 @@ class EmailValidator(object):

         localpart, hostname = value.split('@', 1)
         if app_settings.A2_VALIDATE_EMAIL_DOMAIN:
-            mxs = self.check_mxs(hostname)
+            mxs = self.query_mxs(hostname)
             if not mxs:
                 raise ValidationError(_('Email domain is invalid'), code='invalid-domain')
             if self.rcpt_check and app_settings.A2_VALIDATE_EMAIL:
```

#12 - 27 mars 2020 15:15 - Paul Marillonnet

- Statut changé de Solution proposée à Solution validée

Benjamin Dauvergne a écrit :

> Voilà (et la boucle est bonne, si pas de MX on doit utiliser AAAA ou A pour envoyer un mail) :

Ok, j'aurais appris un truc :)

Du coup si la boucle est bonne, peut-être renommer query_mxs en quelque chose de plus générique puisqu'on ne cherche pas nécessairement que les entrées MX ? (Je te laisse décider.)

#13 - 27 mars 2020 15:34 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Benjamin Dauvergne a écrit :
>
> > Voilà (et la boucle est bonne, si pas de MX on doit utiliser AAAA ou A pour envoyer un mail) :
>
> Ok, j'aurais appris un truc :)
>
> Du coup si la boucle est bonne, peut-être renommer query_mxs en quelque chose de plus générique puisqu'on ne cherche pas nécessairement que les entrées MX ? (Je te laisse décider.)

Les A sont des MX aussi dans ce cas, MX ça veut dire Mail eXchanger, serveur mail quoi; le nom est très bien, je vais corriger les petits glitchs qui restent.

#14 - 27 mars 2020 15:58 - Benjamin Dauvergne

- Fichier 0001-validators-use-only-dnspython-to-resolve-domains-409.patch ajouté

- Statut changé de Solution validée à Solution proposée

#15 - 27 mars 2020 16:25 - Paul Marillonnet

- Statut changé de Solution proposée à Solution validée

Ok, ack.

#16 - 27 mars 2020 16:47 - Benjamin Dauvergne

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

```
commit 702feb4d95bc593d494dfc4e70a17191f72f061a
Author: Frédéric Péters <fpeters@entrouvert.com>
Date:   Tue Mar 24 20:31:25 2020 +0100

    validators: use only dnspython to resolve domains (#40989)

    Also fix SMTP error value check when testing email adresses with an RCPT
```

```
    check.
```

#17 - 01 avril 2020 20:14 - Frédéric Péters

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

Fichiers

0001-validators-don-t-crash-checking-MX-of-invalid-domain.patch	823 octets	24 mars 2020	Frédéric Péters
0001-validators-don-t-crash-checking-MX-of-invalid-domain.patch	827 octets	25 mars 2020	Benjamin Dauvergne
0004-tests-test-email-validator-40989.patch	5,43 ko	25 mars 2020	Benjamin Dauvergne
0002-validators-piggyback-on-django-EmailValidator-40989.patch	1,08 ko	25 mars 2020	Benjamin Dauvergne
0003-validators-check-SMTP-RCPT-error-is-5xx-40989.patch	952 octets	25 mars 2020	Benjamin Dauvergne
0002-tests-test-email-validator-40989.patch	5,43 ko	27 mars 2020	Benjamin Dauvergne
0001-validators-use-only-dnspython-to-resolve-domains-409.patch	4,64 ko	27 mars 2020	Benjamin Dauvergne
0002-tests-test-email-validator-40989.patch	5,43 ko	27 mars 2020	Benjamin Dauvergne
0001-validators-use-only-dnspython-to-resolve-domains-409.patch	4,81 ko	27 mars 2020	Benjamin Dauvergne
0001-validators-use-only-dnspython-to-resolve-domains-409.patch	12,1 ko	27 mars 2020	Benjamin Dauvergne