

Authentic 2 - Bug #37515

crash sur "update metadata" saml

07 novembre 2019 10:25 - Frédéric Péters

Statut:	Fermé	Début:	07 novembre 2019
Priorité:	Normal	Echéance:	
Assigné à:	Valentin Deniaud	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
<pre>File "/usr/lib/python2.7/dist-packages/django/contrib/admin/options.py" in changelist_view 1590. response = self.response_action(request, queryset=cl.get_queryset(request)) File "/usr/lib/python2.7/dist-packages/django/contrib/admin/options.py" in response_action 1287. response = func(self, request, queryset) File "/usr/lib/python2.7/dist-packages/authentic2/saml/admin.py" in update_metadata 105. provider.update_metadata() File "/usr/lib/python2.7/dist-packages/authentic2/saml/models.py" in update_metadata 410. self.clean() File "/usr/lib/python2.7/dist-packages/authentic2/saml/models.py" in clean 388. p = lasso.Provider.newFromBuffer(lasso.PROVIDER_ROLE_ANY, self.metadata.encode('utf 8')) Exception Type: UnicodeDecodeError at /admin/saml/libertyprovider/ Exception Value: 'ascii' codec can't decode byte 0xc3 in position 2571: ordinal not in range(128) >/pre></pre>			

Historique

#1 - 12 novembre 2019 10:56 - Benjamin Dauvergne

- Assigné à mis à Valentin Deniaud

#2 - 13 novembre 2019 16:21 - Valentin Deniaud

J'ai beau mettre des é partout dans un fichier de métadonnées initialement valides, je ne reproduis pas (le provider est instancié quoiqu'il arrive, ce qui paraît normal).

Et à regarder le code auto-généré dans lasso.py je ne vois rien qui puisse lever cette exception.

Par contre

```
383             self.entity_id_shal = hashlib.shal(self.entity_id.encode('ascii')).hexdigest()
```

va provoquer un crash similaire si l'id a un caractère non ascii (mais ce n'est pas un id valide d'après la norme, donc bon).

#4 - 13 novembre 2019 16:44 - Benjamin Dauvergne

Juste mettre force_bytes() et/ou force_str() (pour des raisons du au changement de comportement entre py2/py3 et le binding lasso) et tout ira bien.

#5 - 14 novembre 2019 10:46 - Valentin Deniaud

OK merci Fred je vois le bug.

changement de comportement entre py2/py3 et le binding lasso

En fait ce n'est pas le binding lasso qui lève l'exception, c'est self.metadata.encode('utf8') (moi j'avais pas idée qu'un encode pouvait lever une exception de décodage, mais maintenant j'ai compris).

Ce qu'il se passe c'est que seulement dans le cas où on force la mise à jour des métadonnées on a self.metadata = response.content, donc des données déjà encodées, et le réencodage plante.

Et donc je me dis qu'on pourrait ajouter une condition pour ne pas encoder dans ce cas (ou laisser `force_bytes` faire la même chose). Mais je vois aussi que dans le binding `lasso` on a `str2lasso` qui fait exactement ça (encoder en utf8 si metadata est unicode). Donc le bon fix ce ne serait pas simplement d'enlever ce `.encode('utf8')`, qui est redondant avec le code dans `lasso` ?

#6 - 14 novembre 2019 11:11 - Benjamin Dauvergne

Valentin Deniaud a écrit :

En fait ce n'est pas le binding `lasso` qui lève l'exception, c'est `self.metadata.encode('utf8')` (moi j'avais pas idée qu'un encode pouvait lever une exception de décodage, mais maintenant j'ai compris).

Ce que je disais c'est qu'en python2 `lasso` prend des `str()` encodés en UTF-8 et en python3 des `str` (qui ne sont donc pas encodées) il faut donc utiliser `force_str()` pour couvrir les deux (et pas `force_bytes()` je disais une connerie, ça ne marcherait plus en python3).

Ce qu'il se passe c'est que seulement dans le cas où on force la mise à jour des métadonnées on a `self.metadata = response.content`, donc des données déjà encodées, et le réencodage plante.

Le plus simple :

```
try:
    self.metadata = response.text
except UnicodeError:
    # gestion des erreurs
```

#7 - 14 novembre 2019 12:02 - Valentin Deniaud

- Fichier `0001-saml-ensure-self.metadata-is-always-unicode.patch` ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Joli. Par contre `request.text` fait du `unicode(..., errors='replace')`, pas d'exceptions à rattraper du coup.

#8 - 14 novembre 2019 12:28 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Valentin Deniaud a écrit :

Joli. Par contre `request.text` fait du `unicode(..., errors='replace')`, pas d'exceptions à rattraper du coup.

Tant mieux, ça foirera juste des intitulés qu'on n'utilise de toute façon pas.

#9 - 14 novembre 2019 14:18 - Valentin Deniaud

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

```
commit 55ec200c40a316255373a049d40798f5b5fc0c68
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Thu Nov 14 11:59:46 2019 +0100
```

```
saml: ensure LibertyProvider.metadata is always unicode
```

#10 - 15 février 2020 20:17 - Frédéric Péters

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

Fichiers

0001-saml-ensure-self.metadata-is-always-unicode.patch

837 octets 14 novembre 2019

Valentin Deniaud