django-mellon - Development #43193

absence de décodage XML des attributs

21 mai 2020 19:20 - Frédéric Péters

Statut:	Fermé	**Début:**	21 mai 2020
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Si mon prénom est Frédéric, ça se trouve bien dans l'assertion avec l'encodage nécessaire :

```
<saml:Attribute Name="first_name" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" F
riendlyName=""><saml:AttributeValue>&lt;i&gt;Frédéric&lt;/i&gt;</saml:AttributeValue></saml:Attrib
ute>
```

mais cette forme encodée persiste à la lecture de l'assertion, dans le saml_attributes passé à provision* etc., exemple dans les logs :

```
trying to authenticate with attributes {'username': ['fred'], 'first_name': ['&lt;i&gt;Frédéric&lt
;/i&gt;'], ...
```

et c'est cette forme qui se trouve mise en db, et mon nom apparait alors comme Frédéric.

Révisions associées

Révision c05f4a31 - 21 mai 2020 21:04 - Benjamin Dauvergne

views: ignore XML content in SAML attributes (#43193)

Historique

#1 - 21 mai 2020 19:21 - Frédéric Péters

- Fichier 0001-misc-don-t-treat-basic-attributes-as-XML-43193.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Ma proposition basique/conservatrice serait de juste prendre le texte tel quel en situation de lasso.SAML2_ATTRIBUTE_NAME_FORMAT_BASIC.

#2 - 21 mai 2020 20:39 - Benjamin Dauvergne

Le NameFormat n'a pas vraiment de rapport avec le contenu; je vais proposer autre chose.

#3 - 21 mai 2020 20:40 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#4 - 21 mai 2020 20:41 - Benjamin Dauvergne

- Assigné à Benjamin Dauvergne supprimé

#5 - 21 mai 2020 21:03 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#6 - 21 mai 2020 21:03 - Benjamin Dauvergne

- Fichier 0001-views-ignore-XML-content-in-SAML-attributes-43193.patch ajouté

- Tracker changé de Bug à Development

#7 - 21 mai 2020 21:05 - Benjamin Dauvergne

- Fichier 0001-views-ignore-XML-content-in-SAML-attributes-43193.patch ajouté

J'ai changé pour logger tout l'attribut en cas d'erreur c'est plus simple.

#8 - 25 mai 2020 17:10 - Paul Marillonnet

- Statut changé de Solution proposée à Solution validée

Du détail, mais j'aurais bien vu une modif du genre

```
diff --git a/mellon/views.py b/mellon/views.py
index 899989f..222edc2 100644
--- a/mellon/views.py
+++ b/mellon/views.py
@@ -212,7 +212,7 @@ class LoginView(ProfileMixin, LogMixin, View):
     def get_attribute_value(self, attribute, attribute_value):
         # check attribute_value contains only text
         for node in attribute_value.any:
-            if not isinstance(node, lasso.MiscTextNode) or not node.textChild:
+            if not isinstance(node, lasso.MiscTextNode) or node.content is None:
                 self.log.warning('unsupported attribute %s', attribute.exportToXml())
                 return None
         return ''.join(lasso_decode(node.content) for node in attribute_value.any)
```

parce qu'après une lecture rapide j'ai l'impression que c'est équivalent, mais aussi parce que ce patch y gagne en lisibilité.

Sinon c'est ok pour moi.

#9 - 26 mai 2020 19:33 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Du détail, mais j'aurais bien vu une modif du genre
> [...]
> parce qu'après une lecture rapide j'ai l'impression que c'est équivalent, mais aussi parce que ce patch y gagne en lisibilité.

Je ne crois pas que node.content puisse être à None, il me semble que libxml2 contrairement à ElementTree ne retourne jamais NULL comme contenu texte d'un noeud même s'il est vide (mais je me trompe peut-être, je n'ai pas tenté l'expérience, c'est vague un souvenir sur libxml2 qui est assez cohérent en général).

Si t'as <saml:AttributeValue><trucmuche></trucmuche></saml:AttributeValue> ça donne :

```
atv.any = [mtn]
mtn.textChild = False
mtn.content = ''
mtn.name = 'trucmuche'
```

<saml:AttributeValue></saml:AttributeValue> ça donne :

```
atv.any = [mtn]
mtn.textChild = True
mtn.content = ''
mtn.name = None
```

le but étant de supporter un contenu mixte (texte/noeuds) qui n'arrive jamais.

#10 - 27 mai 2020 08:52 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> Je ne crois pas que node.content puisse être à None, il me semble que libxml2 contrairement à ElementTree ne retourne jamais NULL comme contenu texte d'un noeud même s'il est vide (mais je me trompe peut-être, je n'ai pas tenté l'expérience, c'est vague un souvenir sur libxml2 qui est assez cohérent en général).

J'ai trituré un peu le patch parce qu'il me semblait d'abord, à tort, qu'avoir le “if not isinstance(node, lasso.MiscTextNode):” était suffisant. Mais dans notre cas au moins, les node qui vérifient “not node.textChild” ont l'attribut content valant None, ce qui casserait le join quelques lignes plus bas si on n'ajoute ni “or not node.textChild” ni “or node.content is None” au if.

#11 - 27 mai 2020 11:23 - Benjamin Dauvergne

Je ne comprends toujours pas, tu peux me donner un exemple de code et contenu XML qui donnerait .content is None quand .textChild is True ?

#12 - 27 mai 2020 11:43 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> Je ne comprends toujours pas, tu peux me donner un exemple de code et contenu XML qui donnerait .content is None quand .textChild is True ?

Non justement mon impression à la relecture c'était qu'il y a équivalence entre .content is None et not .textChild. Je vais vérifier.

#13 - 27 mai 2020 12:18 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Benjamin Dauvergne a écrit :
>
> > Je ne comprends toujours pas, tu peux me donner un exemple de code et contenu XML qui donnerait .content is None quand .textChild is True ?
>
> Non justement mon impression à la relecture c'était qu'il y a équivalence entre .content is None et not .textChild. Je vais vérifier.

Ben non, je viens de le montrer plus haut.

```
In [5]: mtn = lasso.MiscTextNode.newWithString('coucou')

In [6]: mtn.name = 'coin'

In [7]: mtn.exportToXml()

Out[7]: '<coin>coucou</coin>'

In [8]: mtn.textChild

Out[8]: False

In [9]: mtn.content

Out[9]: 'coucou'
```

Après je suis d'accord que la classe MiscTextNode est super mal nommée, mais c'est historique ©.

#14 - 27 mai 2020 12:33 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> Ben non, je viens de le montrer plus haut.
> [...]
>
> Après je suis d'accord que la classe MiscTextNode est super mal nommée, mais c'est historique ©.

Ah pardon, je n'ai pas vu que c'était ce que tu montrais plus haut. Il faudrait alors s'assurer qu'on n'a pas de cas possible où un node peut vérifier “isinstance(node, lasso.MiscTextNode) and node.textChild and node.content is None", auquel cas le ''.join(…) casserait.

Tu disais “il me semble que libxml2 contrairement à ElementTree ne retourne jamais NULL comme contenu texte d'un noeud même s'il est vide […]” et pourtant en appliquant :

```
--- a/mellon/views.py
+++ b/mellon/views.py
@@ -212,7 +212,7 @@ class LoginView(ProfileMixin, LogMixin, View):
     def get_attribute_value(self, attribute, attribute_value):
         # check attribute_value contains only text
         for node in attribute_value.any:
-            if not isinstance(node, lasso.MiscTextNode) or not node.textChild:
+            if not isinstance(node, lasso.MiscTextNode):
                 self.log.warning('unsupported attribute %s', attribute.exportToXml())
                 return None
         return ''.join(lasso_decode(node.content) for node in attribute_value.any)>
```

les tests cassent pour cette raison :

```
[…]
mellon/views.py:416: in get
    return self.continue_sso_artifact(request, lasso.HTTP_METHOD_ARTIFACT_GET)
mellon/views.py:394: in continue_sso_artifact
    return self.sso_success(request, login)
mellon/views.py:227: in sso_success
    content = self.get_attribute_value(at, attribute_value)
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

self = <mellon.views.LoginView object at 0x7f1ed8feca90>, attribute = <lasso.Saml2Attribute object at 0x7f1ed9
10e7c0>, attribute_value = <lasso.Saml2AttributeValue object at 0x7f1ed910ebb0>

    def get_attribute_value(self, attribute, attribute_value):
        # check attribute_value contains only text
        for node in attribute_value.any:
            if not isinstance(node, lasso.MiscTextNode):
                self.log.warning('unsupported attribute %s', attribute.exportToXml())
                return None
>       return ''.join(lasso_decode(node.content) for node in attribute_value.any)
E       TypeError: sequence item 1: expected str instance, NoneType found
```

#15 - 02 juin 2020 06:14 - Benjamin Dauvergne

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

```
commit c05f4a3129ee85388c29d7170f3e7f52d0425a95
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Thu May 21 21:01:49 2020 +0200

    views: ignore XML content in SAML attributes (#43193)
```

#16 - 10 juin 2020 12:16 - Frédéric Péters

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

Fichiers

0001-misc-don-t-treat-basic-attributes-as-XML-43193.patch	1,18 ko	21 mai 2020	Frédéric Péters
0001-views-ignore-XML-content-in-SAML-attributes-43193.patch	4,11 ko	21 mai 2020	Benjamin Dauvergne
0001-views-ignore-XML-content-in-SAML-attributes-43193.patch	4,08 ko	21 mai 2020	Benjamin Dauvergne