Authentic 2 - Bug #1419

lacking SNI support (Server Name Indication, RFC 6066)

04 mai 2012 21:42 - Jean Christophe André

Statut:	Fermé	**Début:**	04 mai 2012
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	90%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:		**Planning:**	

Description

With the, now current, lack of IPv4 addresses, it is more and more common to find situations where there is only one IP address for multiple secured websites, using NamedVirtualHost behind SSL/TLS with SNI extension.

The current Authentic 2 code is using M2Crypto for HTTPS requests, which doesn't set the SNI extension and doesn't seam to support it either. It could instead use pyCurl, which does support the SNI extension and set it by default (I've tested it successfully with Debian Squeeze).

This is especially annoying when we try to directly update one (or multiple) SP metadata from the Liberty Providers list in the admin interface (/admin/saml/libertyprovider/)...

Ref:
- http://en.wikipedia.org/wiki/Server_Name_Indication
- http://tools.ietf.org/html/rfc6066#section-3

Révisions associées

Révision 03bc2069 - 05 mai 2012 16:20 - Benjamin Dauvergne

[http_utils] prefer using pycurl instead of M2Crypto to retrieve HTTPs URLS as it supports server name indication

Thanks to Jean Christophe André for the feature request and the patch.

Fixes #1419

Historique

#1 - 04 mai 2012 22:20 - Jean Christophe André

This bunch of code show how to add pycurl support and has been tested successfully against a SNI enabled Apache website.

Beware: it doesn't validate the SSL/TLS security! (it was not my goal here, it's only a proof of concept).

```
diff --git a/lib/python2.6/site-packages/authentic2/http_utils.py b/lib/python2.6/site-packages/authentic2/htt
p_utils.py
index 04ed0b0..84a87ac 100644
--- a/lib/python2.6/site-packages/authentic2/http_utils.py
+++ b/lib/python2.6/site-packages/authentic2/http_utils.py
@@ -1,4 +1,10 @@
 try:
+    import pycurl
+except ImportError:
+    pycurl = None
+import cStringIO
+
+try:
     import M2Crypto
 except ImportError:
     M2Crypto = None
@@ -22,8 +28,21 @@ def get_url(url):
        check the certificate'''

     if url.startswith('https'):
-        if not M2Crypto:
-            raise urllib2.URLError('https is unsupported without M2Crypto')
+        if not pycurl and not M2Crypto:
+            raise urllib2.URLError('https is unsupported without either pyCurl or M2Crypto')
```

```
+        if pycurl:
+            try:
+                buf = cStringIO.StringIO()
+                c = pycurl.Curl()
+                c.setopt(c.URL, str(url))
+                c.setopt(c.WRITEFUNCTION, buf.write)
+                c.perform()
+                r = buf.getvalue()
+                buf.close()
+                return r
+            except pycurl.error, e:
+                # Wrap error
+                raise urllib2.URLError('SSL access error %s' % e)
         try:
             return M2Crypto.m2urllib2.build_opener(get_ssl_context()).open(url).read()
         except M2Crypto.SSL.Checker.SSLVerificationError, e:
```

#2 - 05 mai 2012 13:14 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

I'm working on it.

#3 - 05 mai 2012 16:25 - Benjamin Dauvergne

- Statut changé de Nouveau à Solution déployée

- % réalisé changé de 0 à 90

Appliqué par commit 03bc206907cb1510147d01d6f2ba0a3a55960544.

#4 - 15 juin 2012 09:30 - Benjamin Dauvergne

- Assigné à Benjamin Dauvergne supprimé

#5 - 09 septembre 2014 14:25 - Benjamin Dauvergne

- Statut changé de Solution déployée à Fermé