

w.c.s. - Bug #20737

appel webservice et erreur SSL

15 décembre 2017 11:07 - Thomas Noël

Statut:	Fermé	Début:	15 décembre 2017
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Non		

Description

En cas de pépin dans un requests() on n'intercepte par leur soucis SSL :

Exception:

```
type = '<class 'OpenSSL.SSL.WantWriteError'>', value = ''
```

Stack trace (most recent call first):

```
File "/usr/lib/python2.7/dist-packages/OpenSSL/SSL.py", line 849, in _raise_ssl_error
  847         raise WantReadError()
  848     elif error == _lib.SSL_ERROR_WANT_WRITE:
>  849         raise WantWriteError()
  850     elif error == _lib.SSL_ERROR_ZERO_RETURN:
  851         raise ZeroReturnError()
```

locals:

```
ssl = <cdata 'SSL *' 0x1038cc0>
self = <OpenSSL.SSL.Connection object at 0x7f0ada8f7290>
result = -1
error = 3
```

```
File "/usr/lib/python2.7/dist-packages/OpenSSL/SSL.py", line 977, in sendall
  975     while left_to_send:
  976         result = _lib.SSL_write(self._ssl, data + total_sent, left_to_send)
>  977         self._raise_ssl_error(self._ssl, result)
  978         total_sent += result
  979         left_to_send -= result
```

...

```
File "/usr/lib/python2.7/dist-packages/wcs/qommon/misc.py", line 299, in _http_request
  297         raise ConnectionError('connection timed out while fetching the page')
  298     except requests.RequestException as err:
>  299         raise ConnectionError('error in HTTP request to %s (%s)' % (hostname, err))
  300     else:
  301         data = response.content
```

Historique

#1 - 15 décembre 2017 11:09 - Thomas Noël

- Description mis à jour

#2 - 15 décembre 2017 11:45 - Benjamin Dauvergne

Là on a surtout un soucis de version de requests parce que normalement il devrait toujours wrapper les exceptions d'origine dans une requests.exceptions.SSLError, ex.: <https://github.com/requests/requests/issues/1294>

#3 - 15 décembre 2017 13:32 - Thomas Noël

De fait en recette nous tournons avec les backports pour toutes les parties en présence dans l'affaire : requests, urllib3 et python-openssl

Package: python-openssl

Pin: release a=jessie-backports

Pin-Priority: 900

```
Package: python-requests
Pin: release a=jessie-backports
Pin-Priority: 900
```

```
Package: python-urllib3
Pin: release a=jessie-backports
Pin-Priority: 900
```

#4 - 15 décembre 2017 13:50 - Benjamin Dauvergne

Oui justement on a le souci qu'en prod où on utilise pas ces backports je viens de checker la version 1.9.1-3 de urllib3 (celle en prod venant de jessie) ne catch pas les exceptions WantWriteError là où il faut et la version 1.16-1~bpo8+1 en recette (venant de jessie-backports) le fait. Gogogo backports.

#6 - 19 décembre 2017 16:16 - Thomas Noël

Ca semble ok ; est-ce qu'on impose les versions dans les dépendances de wcs ?

#7 - 26 janvier 2020 21:59 - Frédéric Péters

- *Statut changé de Nouveau à Fermé*

On est depuis à une version suffisante de requests.