Python requests gives me "bad handshake" error

Ask Question

Using Python [requests](#) like this

```
import requests;
requests.get('https://internal.site.no')
```

gives me an error [many](#) [have](#) [had](#);

```
SSLError: ("bad handshake: Error([('SSL routines', 'SSL23_GET_SERVER_HELLO', 'sslv3 alert
handshake failure')],)",)
```

however, none of the missing package they suggest works. Even setting `verify=False` gives me the same error. Curl gives me no error trying to access the same site.

Versions:

- Alpine 3.4
- requests 2.12.1 (it works in 2.11.1)
- OpenSSL 1.0.2j 26 sep 2016

`python` `ssl` `python-requests`

asked Nov 22 '16 at 12:03

[xeor](#)
2,656   2   22   52

2 Answers

The most likely error is that `requests` and the server are not able to negotiate a cipher to use.

Check what curl uses;

```
curl --verbose https://internal.site.r
```

It will give you a lot of output, but the

Looking at the diff from [2.11.1 to 2.12.0](#) of requests, shows a new version of urllib3 (to version 1.19). Maybe it's the [removal of 3des](#) that bites you here?

If you check your `curl --verbose ...` output used cipher against [this](#) usefull list of cipher name mapping. You can try adding the `openssl` name of the name to what `requests` accept, example (you can do this in the beginning of your app/script):

```
import requests
requests.packages.urllib3.util.ssl_.DE
```

if curl shows you that it is using `TLS_DH_anon_WITH_AES_128_CBC_SHA256` (as an example).

Another handy tip is to use the nmap script [ssl-enum-ciphers](#), like this:

```
nmap --script ssl-enum-ciphers -p 443
```

to get a list of what it finds as supported ciphers (note, script might be noisy)...

answered Nov 22 '16 at 12:03

xeor
2,656   2   22   52

I also had the same issue. Check which version of requests you are using.

```
import requests
print requests.__version__
```

You should try downgrading to version 2.11.1. I did this, and it fixed my problem. To do this, issue the following commands in the terminal

```
pip uninstall requests
pip install requests==2.11.1
```

Hope this helps.

answered Nov 14 '17 at 20:38

CPSuperstore
121   8

Then it is probably the removal of
3des from urllib3 that was biting you
as well. The the link in my answer for
diff. I think it is a better solution to
monkeypatch the `DEFAULT_CIPHERS`
in urllib3 than being stuck on an old
version. The problem lies in the
`https` server we are talking to tho.
It's a reason why 3des is out.. :) –
xeor Nov 14 '17 at 21:08

This should be the answer. – Bhargav
Dec 7 '17 at 4:37

Only this solution worked for me after
trying so much different things. –
Tarun Sapra May 25 at 12:27