

Re: HTTPSConnection script fails, but only on some servers (long)

Re: HTTPSConnection script fails, but only on some servers (long)

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2005-04/msg02097.html>

- *From:* andreas@xxxxxxxxxxxxx
 - *Date:* Wed, 13 Apr 2005 12:18:32 +0200
-

Well HTTPSConnection does not support proxies. (HTTP/CONNECT + switch to HTTPS)

And it hasn't ever. Although the code seems to make sense there is no support for handling that switch. Probably a good thing to complain about (file a new bug report).

In the meantime you should take a look a cURL and pycurl, which do support all kind of more extreme HTTP (FTP, etc.) handling, like using https over an proxy.

Andreas

On Tue, Apr 12, 2005 at 03:37:33AM -0400, Steve Holden wrote:

> Paul Winkler wrote:

>> This is driving me up the wall... any help would be MUCH appreciated.

>> I have a module that I've whittled down into a 65-line script in

>> an attempt to isolate the cause of the problem.

>>

>> (Real domain names have been removed in everything below.)

>>

>> SYNOPSIS:

>>

>> I have 2 target servers, at <https://A.com> and <https://B.com>.

>> I have 2 clients, wget and my python script.

>> Both clients are sending GET requests with exactly the

>> same urls, parameters, and auth info.

>>

>> wget works fine with both servers.

>> The python script works with server A, but NOT with server B.

>> On Server B, it provoked a "Bad Gateway" error from Apache.

>> In other words, the problem seems to depend on both the client

>> and the server. Joy.

>>

>> Logs on server B show malformed URLs ONLY when the client

>> is my python script, which suggests the script is broken...

>> but logs on server A show no such problem, which suggests

>> the problem is elsewhere.

>>

Re: HTTPSConnection script fails, but only on some servers (long)

Re: HTTPSConnection script fails, but only on some servers (long)

>>DETAILS

>>

>>Note, the module was originally written for the express
>>purpose of working with B.com; A.com was added as a point of reference
>>to convince myself that the script was not totally insane.
>>Likewise, wget was tried when I wanted to see if it might be
>>a client problem.

>>

>>Note the servers are running different software and return different
>>headers. wget -S shows this when it (successfully) hits url A:

>>

>> 1 HTTP/1.1 200 OK
>> 2 Date: Tue, 12 Apr 2005 05:23:54 GMT
>> 3 Server: Zope/(unreleased version, python 2.3.3, linux2) ZServer/1.1
>> 4 Content-Length: 37471
>> 5 Etag:
>> 6 Content-Type: text/html;charset=iso-8859-1
>> 7 X-Cache: MISS from XXX.com
>> 8 Keep-Alive: timeout=15, max=100
>> 9 Connection: Keep-Alive

>>

>>... and this when it (successfully) hits url B:

>>

>> 1 HTTP/1.1 200 OK
>> 2 Date: Tue, 12 Apr 2005 04:51:30 GMT
>> 3 Server: Jetty/4.2.9 (Linux/2.4.26-g2-r5-cti i386 java/1.4.2_03)
>> 4 Via: 1.0 XXX.com
>> 5 Content-Length: 0
>> 6 Connection: close
>> 7 Content-Type: text/plain

>>

>>Only things notable to me, apart from the servers are the "Via:" and
>>"Connection:" headers. Also the "Content-Length: 0" from B is odd, but
>>that doesn't seem to be a problem when the client is wget.

>>

>>Sadly I don't grok HTTP well enough to spot anything really
>>suspicious.

>>

>>The apache ssl request log on server B is very interesting.

>>When my script hits it, the request logged is like:

>>

>>A.com -- [01/Apr/2005:17:04:46 -0500] "GET
>><https://A.com/SkinServlet/zopeskin?action=updateSkinId&facilityId=1466&skinId=406>
>>HTTP/1.1" 502 351

>>

>>... which apart from the 502, I thought reasonable until I realized
>>there's

>>not supposed to be a protocol or domain in there at all. So this is

>>clearly

>>wrong. When the client is wget, the log shows something more sensible

>>like:

Re: HTTPSConnection script fails, but only on some servers (long)

Re: HTTPSConnection script fails, but only on some servers (long)

```
>>
>>A.com -- [01/Apr/2005:17:11:04 -0500] "GET
>>/SkinServlet/zopeskin?action=updateSkinId&facilityId=1466&skinId=406
>>HTTP/1.0" 200 -
>>
>>... which looks identical except for not including the spurious
>>protocol and domain, and the response looks as expected (200 with size
>>0).
>>
>>So, that log appears to be strong evidence that the problem is in my
>>client
>>script, right? The failing request is coming in with some bad crap in
>>the path, which Jboss can't handle so it barfs and Apache responds with
>>
>>Bad Gateway. Right?
>>
>>So why does the same exact client code work when hitting server B??
>>No extra gunk in the logs there. AFAICT there is nothing in the script
>>that could lead to such an odd request only on server A.
>>
>>
>>THE SCRIPT
>>
>>#!/usr/bin/python2.3
>>
>>from httplib import HTTPSConnection
>>from urllib import urlencode
>>import re
>>import base64
>>
>>url_re = re.compile(r'^([a-z]+)://([A-Za-z0-9._-]+)(:[0-9]+)?')
>>
>>target_urls = {
>> 'B': 'https://B/SkinServlet/zopeskin',
>> 'A': 'https://A/zope/manage_main',
>>}
>>
>>auth_info= {'B': ('userXXX', 'passXXX'),
>> 'A': ('userXXX', 'passXXX'),
>> }
>>
>>def doRequest(target, **kw):
>> """Provide a trivial interface for doing remote calls.
>> Keyword args are passed as query parameters.
>> """
>> url = target_urls[target]
>> user, passwd = auth_info[target]
>> proto,host,port=url_re.match(url).groups()
>> if port:
>> port = int(port[1:]) # remove the ':' ...
>> else:
```

Re: HTTPSConnection script fails, but only on some servers (long)

Re: HTTPSConnection script fails, but only on some servers (long)

```
>> port = 443
>> creds = base64.encodestring("%s:%s" % (user, passwd))
>> headers = {"Authorization": "Basic %s" % creds }
>> params = urlencode(kw).strip()
>> if params:
>> url = '%s?%s' % (url, params)
>> body = None # only needed for POST
>> args = ('GET', url, body, headers)
>> print "ARGS: %s" % str(args)
>> conn = HTTPSConnection(host)
>> conn.request(*args)
>> response = conn.getresponse()
>> data = response.read()
>> if response.status >= 300:
>> print
>> msg = '%i ERROR reported by remote system %s\n' %
>>(response.status,
>> url)
>> msg += data
>> raise IOError, msg
>> print "OK!"
>> return data
>>
>> if __name__ == '__main__':
>> print "attempting to connect..."
>> result1 = doRequest('A', skey='id', rkey='id')
>> result2 = doRequest('B', action='updateSkinId',
>> skinId='406', facilityId='1466')
>> print "done!"
>>
>>
>># EOF
>>
>>
>>So... what the heck is wrong here?
>>
>>at-wits-end-ly y'rs,
>>
>>Paul Winkler
>>
> Paul:
>
> I don't claim to have analyzed exactly what's going on here, but the
> most significant difference between the two is that you are accessing
> site B using HTTP 1.1 via an HTTP 1.0 proxy (as indicated by the "Via:"
> header).
>
> Whether this is a clue or a red herring time alone will tell.
>
> It's possible that wget and your client code aren't using the same proxy
> settings, for example.
```

Re: HTTPSConnection script fails, but only on some servers (long)

Re: HTTPSConnection script fails, but only on some servers (long)

>
> regards
> Steve
> --
> Steve Holden +1 703 861 4237 +1 800 494 3119
> Holden Web LLC <http://www.holdenweb.com/>
> Python Web Programming <http://pydish.holdenweb.com/>
>
> --
> <http://mail.python.org/mailman/listinfo/python-list>
> .

• *Follow-Ups:*

- ◆ ***Re: HTTPSConnection script fails, but only on some servers (long)***
 ◇ *From:* Paul Winkler
- ◆ ***Re: HTTPSConnection script fails, but only on some servers (long)***
 ◇ *From:* pyguy2
- ◆ ***Re: HTTPSConnection script fails, but only on some servers (long)***
 ◇ *From:* Paul Winkler

• *References:*

- ◆ ***HTTPSConnection script fails, but only on some servers (long)***
 ◇ *From:* Paul Winkler

- Prev by Date: ***Re: how to explain such codes, python's bug or mine?***
- Next by Date: ***Re: Compute pi to base 12 using Python?***
- Previous by thread: ***Re: HTTPSConnection script fails, but only on some servers (long)***
- Next by thread: ***Re: HTTPSConnection script fails, but only on some servers (long)***
- Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***