Ruby master - Bug #10613

SNI is not optional when using TLS

12/17/2014 10:48 PM - edk750 (Eddy Kim)

Status:	Closed		
Priority:	Normal		
Assignee:	naruse (Yui NARUSE)		
Target version:			
ruby -v:	2.1	**Backport:**	2.0.0: UNKNOWN, 2.1: UNKNOWN

Description

If ruby is using openssl with TLS extensions, and we attempt to connect to a server which supports TLS, but not SNI, the connection fails.

e.g.:

```
uri = URI.parse("https://example.com") # a server that supports TLSv1 but not the TLS extensions
http = Net::HTTP.new(uri.host, uri.port)
http.use_ssl = true
http.ssl_version = :TLSv1
http.verify_mode = OpenSSL::SSL::VERIFY_PEER
response = http.get(url)

OpenSSL::SSL::SSLError: SSL_connect returned=1 errno=0 state=SSLv3 read server hello B: parse tlse
xt
```

If I patch the Net::HTTP#connect method to not assign the hostname to the socket (s), we can avoid this error.

History

#1 - 12/17/2014 11:02 PM - edk750 (Eddy Kim)

sorry, this was my first ticket, so didn't see the bug reporting guidelines until now:

ruby 1.9.3p484 (2013-11-22 revision 43786) x86_64-linux

And this should go to naruse, though I can't seem to edit the issue metadata to assign it to him. :(

#2 - 12/18/2014 03:56 AM - nobu (Nobuyoshi Nakada)

- Description updated

- Status changed from Open to Assigned

- Assignee set to naruse (Yui NARUSE)

#3 - 01/03/2015 01:17 AM - edk750 (Eddy Kim)

Hi, any feedback on this?

The patch adds the ability to turn off SNI triggering behavior, but by default it continues the previous behavior.

Not all SSL servers support SNI, and by forcing SNI without an option to disable it, makes it impossible to communicate with an conforming TLS implementation.

We're using this patch on our ruby installations, but I think this is something that would be widely useful to the community, especially since it's not obvious why a TLS negotiation would fail with some servers.

Please let me know if I need to do anything to help get this merged in.

Thanks!

#4 - 12/09/2016 06:14 PM - bschmeck (Ben Schmeckpeper)

Is there any chance of this patch getting merged?

We're encountering this bug and are currently monkeypatching the Net::HTTP#connect method to disable SNI for the connections where it's problematic.

#5 - 12/10/2016 07:01 AM - rhenium (Kazuki Yamaguchi)

Which server? TLS servers conforming to the TLS/TLS extensions specification should simply ignore the extension if it is not supported.

#6 - 12/15/2016 04:25 AM - bschmeck (Ben Schmeckpeper)

Unfortunately, I don't know any details about the server. It's not a box that we own.

I am connecting over an SSH tunnel, so the hostname being used for SNI is 127.0.0.1. The server simply returns a 400; my best guess is that it's not configured to use a default certificate for unknown hostnames. (I can make things work by editing /etc/hosts locally, but that's not a viable approach in our production environment.)

I am able to set the Host and Location headers, but have not been able to find a better solution than mokeypatching Net::HTTP#connect to disable SNI for connections to this server.

In my case, the cert on this server is expired, so I don't even verify it. For what it's worth, cURL does not perform SNI when certificate checking is disabled (using the -k or --insecure flag.)

(Sorry for the delay in getting back to you, I had just created my account and wasn't watching this issue.)

#7 - 12/17/2016 05:26 PM - rhenium (Kazuki Yamaguchi)

Ben Schmeckpeper wrote:

> I am connecting over an SSH tunnel, so the hostname being used for SNI is 127.0.0.1. The server simply returns a 400; my best guess is that it's not configured to use a default certificate for unknown hostnames. (I can make things work by editing /etc/hosts locally, but that's not a viable approach in our production environment.)

Connecting to HTTPS server with a raw IP address is not supported currently because the domain name is required for the SNI and also for the certificate verification on connection time.

There is a rejected ticket about specifying the IP address to connect, though in Japanese: [Feature #5180]. Editing /etc/hosts and overriding TCPSocket.open are the suggested workaround.

Not sure about what is the best way if net/http wants to support such usage, but I can say adding an option to turn off SNI is not on the right track.

#8 - 07/30/2020 07:01 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Assigned to Closed

I think the need for this is handled by Net::HTTP#ipaddr=, which was added in Ruby 2.4.

Files

optional-sni.patch	1019 Bytes	12/17/2014	edk750 (Eddy Kim)