Ruby - Bug #14630

DON'T IGNORE ME!!! Uncaught exception: SSL_connect returned=1 errno=0 state=error: certificate verify failed (error number 1)

03/25/2018 03:54 PM - paul_coppinger (Paul Coppinger)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 2.5.0p0 (2017-12-25 revision 61468) [x86_64-darwin17]	**Backport:**	2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN

Description

I'm attempting to connect to a server with verify_mode = OpenSSL::SSL::VERIFY_PEER. It fails with the error:

Uncaught exception: SSL_connect returned=1 errno=0 state=error: certificate verify failed (error number 1)
/usr/local/Cellar/ruby/2.5.0_2/lib/ruby/2.5.0/net/protocol.rb:44:in connect_nonblock'
/usr/local/Cellar/ruby/2.5.0_2/lib/ruby/2.5.0/net/protocol.rb:44:in ssl_socket_connect'
/usr/local/Cellar/ruby/2.5.0_2/lib/ruby/2.5.0/net/http.rb:981:in connect' /usr/local/Cellar/ruby/2.5.0_2/lib/ruby/2.5.0/net/http.rb:920:in do_start'
/usr/local/Cellar/ruby/2.5.0_2/lib/ruby/2.5.0/net/http.rb:909:in start' /usr/local/Cellar/ruby/2.5.0_2/lib/ruby/2.5.0/net/http.rb:609:in start'
/Users/pc/work/unipagos/node/test/tester.rb:93:in request' /Users/pc/work/unipagos/node/test/unipagos_tester.rb:79:in authenticate_user'
/Users/pc/work/unipagos/node/test/authenticate.rb:111:in `<top (required)>'

Strange thing is, if I attempt to verify the certificates directly there is no problem.

This is two-level PKI with a root CA (root.pem) that has two sub-CAs. One sub-CA (admin.pem) is used to issue server certificates (such as mobile.pem) and the other sub-CA (user.pem) is used to issue user certificates. I have included the PEM files for all four certificates.

Here's the code to setup the options for the connection:

```
store = OpenSSL::X509::Store.new
store.add_cert(OpenSSL::X509::Certificate.new(File.read('./root.pem')))
store.add_cert(OpenSSL::X509::Certificate.new(File.read('./admin.pem')))
store.add_cert(OpenSSL::X509::Certificate.new(File.read('./user.pem')))
@options = {
    use_ssl: true,
    ssl_version: :TLSv1_2,
    verify_mode: OpenSSL::SSL::VERIFY_PEER,
    store: store,
    keep_alive_timeout: 30,
    cert: config[:cert].nil? ? nil : OpenSSL::X509::Certificate.new(File.read(config[:cert])),
    key: config[:key].nil? ? nil : OpenSSL::PKey::EC.new(File.read(config[:key]))
}
result = store.verify(@options[:cert])
puts result
result = store.verify(OpenSSL::X509::Certificate.new(File.read('./mobile.pem')))
puts result
```

The output of the above is:
true
true

However, I get the above error when I try to connect to the server (identified by mobile.pem) as follows:

```
req = Net::HTTP::Get.new(uri, headers)
res = Net::HTTP::start(req.uri.hostname, req.uri.port, @options) do |http|
    http.request(req)
end
```

Any ideas?

History

#1 - 03/25/2018 03:58 PM - paul_coppinger (Paul Coppinger)

- Description updated

#2 - 03/29/2018 01:10 AM - paul_coppinger (Paul Coppinger)

- Subject changed from Uncaught exception: SSL_connect returned=1 errno=0 state=error: certificate verify failed (error number 1) to BUG!! Uncaught exception: SSL_connect returned=1 errno=0 state=error: certificate verify failed (error number 1)

#3 - 03/29/2018 12:54 PM - paul_coppinger (Paul Coppinger)

- Subject changed from BUG!! Uncaught exception: SSL_connect returned=1 errno=0 state=error: certificate verify failed (error number 1) to HELP!! Uncaught exception: SSL_connect returned=1 errno=0 state=error: certificate verify failed (error number 1)

#4 - 05/16/2018 11:30 AM - paul_coppinger (Paul Coppinger)

- Subject changed from HELP!! Uncaught exception: SSL_connect returned=1 errno=0 state=error: certificate verify failed (error number 1) to DON'T IGNORE ME!!! Uncaught exception: SSL_connect returned=1 errno=0 state=error: certificate verify failed (error number 1)

#5 - 06/29/2018 02:53 PM - dsh0416 (Delton Ding)

The option hash passing to Net::HTTP::start seems to contain misspelling key.

```
@options = {
    use_ssl: true,
    ssl_version: :TLSv1_2,
    verify_mode: OpenSSL::SSL::VERIFY_PEER,
    store: store,
    keep_alive_timeout: 30,
    cert: config[:cert].nil? ? nil : OpenSSL::X509::Certificate.new(File.read(config[:cert])),
    key: config[:key].nil? ? nil : OpenSSL::PKey::EC.new(File.read(config[:key]))
}
```

This should be

```
@options = {
    use_ssl: true,
    ssl_version: :TLSv1_2,
    verify_mode: OpenSSL::SSL::VERIFY_PEER,
    cert_store: store,
    keep_alive_timeout: 30,
    cert: config[:cert].nil? ? nil : OpenSSL::X509::Certificate.new(File.read(config[:cert])),
    key: config[:key].nil? ? nil : OpenSSL::PKey::EC.new(File.read(config[:key]))
}
```

You'd better double check ruby-doc described here:

opt sets following values by its accessor. The keys are #ca_file, #ca_path, cert, **#cert_store**, ciphers, #close_on_empty_response, key, #open_timeout, #read_timeout, #ssl_timeout, #ssl_version, use_ssl, #verify_callback, #verify_depth and verify_mode. If you set :use_ssl as true, you can use https and default value of #verify_mode is set as OpenSSL::SSL::VERIFY_PEER.

#6 - 06/17/2019 11:34 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed

Files

admin.pem	835 Bytes	03/25/2018	paul_coppinger (Paul Coppinger)
mobile.pem	955 Bytes	03/25/2018	paul_coppinger (Paul Coppinger)
user.pem	830 Bytes	03/25/2018	paul_coppinger (Paul Coppinger)
root.pem	782 Bytes	03/25/2018	paul_coppinger (Paul Coppinger)