Subject: SSL handshake error
Posted by bryan.js00 on Sun, 02 Mar 2014 02:08:22 GMT
View Forum Message <> Reply to Message

I'm just beginning to use U++, and I'm trying to learn how to use sockets and SSL. I have modified the HttpServer example to use SSL, but I'm getting the following error:

ERROR socket(256) / SSL handshake: SSL_ERROR_SSL

Here is the full code that I'm using:

```
#include <Core/Core.h>

using namespace Upp;

TcpSocket   server;

String cert;
String key;

void Server()
{
 for(;;) {
  TcpSocket socket;
  LOG("Waiting...");
  bool b = socket.Accept(server);
  if(b) {
   LOG("Connection accepted");

   socket.SSLCertificate(cert, key, FALSE);

   if( !socket.StartSSL() ) {
    LOG("Cannot start SSL\r\n");
    return;
   } else {
    LOG("SSL Started\r\n");
   }

   while( socket.SSLHandshake() ) { };

   LOG("Responding");

   HttpHeader http;
   http.Read(socket);
   String html;
```

```
  html << "<html>"
      << "<b>Method:</b> " << http.GetMethod() << "<br>"
      << "<b>URI:</b> " << http.GetURI() << "<br>";
  for(int i = 0; i < http.fields.GetCount(); i++)
   html << "<b>" << http.fields.GetKey(i) << ":</b> " << http.fields[i] << "<br>";
  int len = (int)http.GetContentLength();
  if(len > 0)
   socket.GetAll(len);
  html << "<b><i>Current time:</i></b> " << GetSysTime() << "</html>";
  HttpResponse(socket, http.scgi, 200, "OK", "text/html", html);

 }
 }
}

CONSOLE_APP_MAIN
{
 StdLogSetup(LOG_COUT|LOG_FILE);

 cert = LoadFile("D:/Develop/MyApps/ERPLib/erp.cert");
 key = LoadFile("D:/Develop/MyApps/ERPLib/erp.key");

 if(!server.Listen(4000, 10)) {
  LOG("Cannot open server port for listening\r\n");
  return;
 }

 Server();
}
```

The error occurs in the call to socket.StartSSL() and socket.StartSSL() returns FALSE.

Am I even using the SSL portion of sockets correctly? I'm kind of shooting in the dark.

Also, the 'client' portion of this test is FireFox web browser. I'm typing my computer's IP address plus the port 4000 into the address bar:
https://10.10.10.101:4000
Is there any problem with creating a connection that way?

Edit: forgot to mention I'm using OpenSSL 1.0.1f. Also, the cert and key information was generated using an online utility.

Subject: Re: SSL handshake error
Posted by bryan.js00 on Tue, 04 Mar 2014 04:20:52 GMT
View Forum Message <> Reply to Message

I made some progress...

I discovered OpenSSL was returning the following error:

error:140C5042:SSL routines:SSL_UNDEFINED_FUNCTION:called a function you should not call

I learned this was because U++ is using SSLv3_client_method() when creating a context. As this code is the server side, I had to change it to SSLv3_server_method()

Now I'm having new errors, but I believe those are related to certificates and keys.

Subject: Re: SSL handshake error
Posted by mirek on Tue, 04 Mar 2014 06:40:32 GMT
View Forum Message <> Reply to Message

Thanks. I have to admit server side was never much tested.

I have created RM task for this

http://www.ultimatepp.org/redmine/issues/706

while doing my own testing/fixing.

Mirek

Subject: Re: SSL handshake error
Posted by bryan.js00 on Tue, 04 Mar 2014 14:27:33 GMT
View Forum Message <> Reply to Message

Once I get SSL communication working, I will put together an example and although I won't make any promises, I will try to put together a patch for any changes I make to the SSL classes in the core.

I think that my issue and the request for choosing protocol version (see this thread) are very related and one patch could deal with both. Not only let the user choose the version, but also if it's client or server.

Subject: Re: SSL handshake error
Posted by mirek on Sat, 08 Mar 2014 19:29:46 GMT
View Forum Message <> Reply to Message

Well, after the proposed patch (SSLv3_server_method(), which admittedly is stupid overlook), I was able to get it working.

I have committed the reference example HTTPS (including certificates for 'localhost').

Mirek

Subject: Re: SSL handshake error
Posted by bryan.js00 on Mon, 10 Mar 2014 02:39:17 GMT
View Forum Message <> Reply to Message

I was just getting ready to create a patch for my changes, but as I was browsing the SVN, I saw the patch for the client vs server issue.

I took a more invasive approach that allows the user to pass a value (from an enum) which decides which method to use when creating the SSL context. It allows for the choice of any of the 15 methods that OpenSSL provides.

It looks something like this:

socket.StartSSL(SSLV3_SERVER_METHOD);

I like the simple approach of checking the socket connection mode to determine client or server.

If you would like to take a look at my patch, let me know.

Thanks for the patch!
