CS458 Information Security Programming Assignment 2

Instructor: Kevin Jin TA: Xin Liu

Due date: 10/29/2015

1. Introduction

SSL/TLS are crucial to the success of e-business. In this programming assignment, you will learn how to establish secure communication using SSL/TLS.

You first need to establish a secure connection between a client and a server, using the OpenSSL library. To make your task easier, the template files **sclient.cc** and **sserver.cc** are provided for you. Also, a tutorial of SSL programming is attached for your reference.

You will need to use options "-lssl -lcrypto" for compiling your code. The OpenSSL library and C code compilation are very similar to the ones in MP1.

Once an SSL connection has been established, the client and the server proceed as follows:

[1] Client --> Server: Prompt to ask the user to input a number X
[2] Server --> Client: X + 1.
[3] Client --> Check whether the answer from the server is correct, and output the result. The client repeats step 1.

Please modify the provided C files to do the tasks above. During evaluation, we will replace your client program with ours to test your server implementation, and also replace your server program with ours to test your client implementation.

The server needs the certificate file and the key file to start with. You can do that with the following command:

```
"openssl req –x509 –newkey rsa:2048 –keyout key.pem –out cert.pem
–days XXX",
```

where the options are explained as follows:

req
PKCS#10 certificate request and certificate generating utility. (More details about PKCS#10: http://en.wikipedia.org/wiki/PKCS)

-x509
This option outputs a self-signed certificate instead of a certificate request. This is typically used to generate a test certificate or a self-signed root CA.

-newkey arg
This option creates a new certificate request and a new private key. The argument takes one of several forms. rsa:nbits, where nbits is the number of bits, generates an RSA key nbits in size.

-keyout filename
This option gives the output filename of the newly created private key.

-out filename
This option specifies the output filename. The default one is the standard output.

-days n
When the -x509 option is in use, this option specifies the number of days that the certificate is valid for. The default one is 30 days.

2. Instructions

Download the assignment package from Blackboard or Course website or Piazza. The package includes:

- This assignment sheet
- Two template files: sclient.cc and sserver.cc
- A tutorial: openssl_tutorial.pdf

You MUST **use C/C++ in Linux environment** to develop your code. You can use native Linux or setup a Virtual Machine (e.g. virtual box).

Only a softcopy submission is required. The deliverables include:

- The two modified files: sclient.cc and sserver.cc. (You can slightly modify the existing code I put there, as long as the task is accomplished.)
- A readme file, describing how to compile and run your code.

Please zip all the files and submit it through Blackboard, with the name
"Prog_2_Lastname_Firstname_A#.zip"

- Please first put all your deliverables in a folder and then zip it. (instead of compress three separate files directly)
- Please **compress your files to a ".zip" file**, not other format, such as ".rar" or ".7z".

3. Grading criteria:

Weight of this programming assignment: 7.5% of the total score of this course.

We will grade your work based on:

- **On time submission**
  - Late submission policy: half mark deduction if less than one day, and zero marks otherwise
- Code (e.g., using the required methods, readability)
- Successful compilation and execution
- Correct functionality tested with our testing code

Note that we will use our own sserver.cc and sclient.cc to test your code.