

Internet and Intranet Applications and Protocols

Assignment 3: Performance of SSL

Prof. Arthur P. Goldberg

Spring, 2004

Version: March 9, 2004

Due: April 6

This assignment is at

<http://www.cs.nyu.edu/artg/internet/Spring2004/assignments/SSL/instructions.pdf>

Changes

The specification of this assignment may change somewhat while you're working on it. Be prepared to accommodate such 'specification creep'.

Design

The Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols were designed to help protect the privacy and integrity of data while it is transferred across a network. Build a client and server that communicate over SSL using the Java Secure Socket Extension (JSSE). Use `SSLSockets` and `SSLServerSockets`.

The port number should be a command-line argument for the client. The server should find a free port, and then print its value. It is okay to provide defaults if no options are given.

Build a system that will authenticate both the client and server. Do not establish a connection if the other endpoint cannot be authenticated or if an encrypted connection cannot be established. In these cases produce appropriate error messages.

Run the client and server on different machines.

Measurements

Then use your system to measure the performance of SSL and compare it with the performance of raw TCP. Fill in performance figures for the Xs in this table:

		TCP		SSL
Roundtrip time to open a connection (at the client)		X	New session	X
			Reuse session key	X
	Using <code>OutputStream</code> writes of this many bytes			
Bi-directional Bandwidth (client and server both writing—do not worry about other network traffic)	10	X		X
	100	X		X
	1000	X		X
	10000	X		X

For the 'Roundtrip time to open a connection', when SSL is used, the client should run at least twice. The first run will create a new SSL session, and the second will reuse the previous session's key.

To measure ‘Bi-directional bandwidth’, make the server be an echo server. Have the client repeatedly send ‘messages’ of the indicated size. A message can be an OutputStream write of that many bytes, terminated in CRLF.

Run the bandwidth test for long enough to reach stability. At the end of the ‘run’ calculate bandwidth by dividing ‘data transmitted’ by ‘elapsed time’.

Build the server and client so they would use finite memory even if the bandwidth test ran forever. The client needs two threads or NIO, so it can both read the responses and send more messages.

Certificates

I’m not sure what we’ll do about certificates. Either we’ll create a signing authority and generate two certificates for you to use, or we’ll let you do it.

References

- SSL discussion on weeks 9 and 10
- Chapter 7 and Section 7.8.2 in Kurose & Ross
- *Java™ Secure Socket Extension (JSSE) Reference Guide* for the Java™ 2 SDK, Standard Edition, v 1.4, at <http://java.sun.com/j2se/1.4.2/docs/guide/security/jsse/JSSERefGuide.html>
- Java™ Secure Socket Extension (JSSE) 1.0.3 API User's Guide (quite similar to the reference guide) at http://java.sun.com/products/jsse/doc/guide/API_users_guide.html
- Java docs for Package javax.net.ssl at <http://java.sun.com/j2se/1.4.2/docs/api/javax/net/ssl/package-summary.html>
- The SSL Protocol version 3.0 Internet Draft - <http://home.netscape.com/eng/ssl3/ssl-toc.html>
- The TLS Protocol version 1.0 Internet Draft - <http://www.ietf.org/rfc/rfc2246.txt>

Libraries

Your code must be written in Java. The networking code must be written using javax.net.ssl.

Grading

Your SSL code

Your server should follow the code quality guidelines we discussed. We will evaluate this by reading your code. You should write unit tests of your code.

Your grade will be allocated as follows:

	%
Functionality	80
Code quality guidelines	15
Unit tests	5

Handing in your SSL Code

When you’re satisfied with your SSL code please hand in the following:

- Your source code (with your name in all java files) in a jar or zip file named with your name, “Firstname Lastname”.

Please name the file with your name, “Firstname Lastname”, and email it to me, and hand-in a hardcopy in class.