1) Catch a SSL communication (for example from your favorite commerce site) with a sniffer (for example www.wireshark.org).

- Identify all the SSL - PDU's
- Which port does SSL use?

4) Try to read the X.509 certificate received from a SSL server

5) Have a close look on the cipher suite negotiation.

6) Configure your browser to run only SSL v2, SSL v3 and then TLS (Explorer: Tool - Internet Options - Advanced - Security). Catch the cipher suite negotiations for the different versions with Ethereal, and then have a close look on the negotiations. – The Mozilla (not Firefox) browser is very configurable with regards to ciphers and protocols and available for both Windows and Linux. – The newer SeaMonkey (a new version of the Mozilla suite) only allows selection between SSL v 3.0 and TLS

7) Explore the links specified for chapter 7 at the web site for the textbook http://WilliamStallings.com/NetSec/NetSec3e.html

8 and 9 below need some experience in compiling and running Java programs. Your are going to use the support for security described at
http://java.sun.com/j2se/1.4/docs/guide/security/index.html
especially the secure socket extension described at
http://java.sun.com/j2se/1.4.2/docs/guide/security/jsse/JSSERefGuide.html
The Keytool utility included in a Java installation (the /bin directory) is described at http://java.sun.com/j2se/1.4.2/docs/tooldocs/windows/keytool.html

8) Pick the file WebServer.java from fronter/exercises folder.
This is a very simple web server that supports SSL/TLS (HTTPS). When running it could, for example, be accessed by using a browser and the URL `https://localhost:9090/text.txt`
But the server needs a certificate (and a private key). This can be produced by using the Keytool utility. When running the server, Java environment variables must tell where certificate (and private key) is located and the password accessing the private key. It could be something like this:

```
-Djavax.net.ssl.keyStore=c:\keystore
-Djavax.net.ssl.keyStorePassword=xyzv
```

Try also to use

```
-Djavax.net.debug=SSL:handshake:verbose
```

for getting some debug information output on the SSL handshake.

9) Pick the file WebBrowser.java from fronter/exercies. This is a very simple browser that supports HTTPS.

In order to accept server certificates it need to have a local store/file with copies of trusted certificates. Hence, you must make sure that a copy of the server certificates is put in the file with this name:

`....\j2sdk1.4.....\jre\lib\security\cacerts`