CS2107 Self-Exploration Activity 7: TLS/SSL

Notes:

In this Activity 7 about **TLS/SSL**, you will perform the following:

1. To inspect and use **`openssl s_client`** sub-command, which establishes a connection to a remote server speaking TLS/SSL;
2. To observe a **target server's** employed TLS configuration, protocol details, and certificate(s) involved;
3. To access a free **online service from SSL Labs** to test a TLS/SSL **server** and **client** (e.g. your web browser), as well as observe the outputted reports.

Task 1: Inspecting and Using openssl s_client Sub-Command

You also can use `openssl`, by invoking its **`s_client` sub-command** (i.e. TLS/SSL client program), to establish **a connection** to a remote server speaking TLS/SSL. Note, however, that this sub-command is intended for *testing purposes* only, as it provides only rudimentary interface functionality. Nonetheless, it internally uses mostly all functionality of the OpenSSL `ssl` library. Hence, you can use it as a **very useful diagnostic tool** for TLS/SSL servers.

To know more how you can use the sub-command, first run:

```
$ man s_client
```

To **connect** to a TLS/SSL server, run:

```
$ openssl s_client -connect <server_name>:443
```

You can replace <*server_name*> in the command above with, for example, www.google.com. If the connection succeeds, then an HTTP command can be given such as "`GET /`" to retrieve a web page.

Task 2: Observing a Target Server's TLS Configuration and Protocol Details

From the output of the `openssl s_client` command above, you can inspect various pieces of information about the server including, among others, its:

- Certificate chain;
- Server's certificate, whose portion in the output starts with
  "`-----BEGIN CERTIFICATE-----`"
  and ends with "`-----END CERTIFICATE-----`";
- TLS/SSL configuration.

As such, you can thus check the following:

- Any **CA(s) involved;**
- The TLS/SSL server's ***certificate content***:
  First, you copy the outputted server-certificate portion, starting from the above-mentioned *certificate beginning marker* until its *ending marker* (inclusive), and paste it into <*server_cert*>`.crt` file. Then, you can run the following, which was previously explained in Self-Exploration Activity 6:

  ```
  openssl x509 -text -in <server_cert>.crt -noout
  ```

- The TLS/SSL **protocol version** used;
- The **Cipher** used by TLS;
- The **Session-ID** used;
- The **Master Secret** generated.

Task 3: Accessing SSL Labs' Online Service to Test a TLS/SSL Server and Client

Additionally, you can access a free online service from **SSL Labs** to test a TLS/SSL server and client (e.g. our web browser).

To **test a TLS/SSL server**, visit https://www.ssllabs.com/ssltest/ and specify the **server's hostname**. You can try testing several servers, including those under the https://badssl.com/, which have certificate problems as discussed in our previous Self-Exploration Activity 6.

The online service performs an analysis of the configuration of the server. By observing the outputted report, you can thus inspect the following:

- **Certificates involved**;
- The server's TLS **configuration**: information about protocols, cipher suites, and handshake simulation;
- The server's **protocol details**, including whether the server is *vulnerable* to several known server issues, e.g. Heartbleed, and OpenSSL Padding Oracle.

To test a **TLS/SSL client**, such as **your browser**, you can visit https://clienttest.ssllabs.com:8443/ssltest/viewMyClient.html.
The online service performs an analysis of the configuration of the client.
By observing the output, you can inspect the following:

- Your browser's **TLS/SSL capabilities**;
- Whether your browser is **vulnerable** to several known client issues, such as CurveBall, Logjam, FREAK, and POODLE vulnerabilities.