CYEN 406/CS 444
Sangam Mulmi

Assignment 3: SSL/TLS Certificate

Total Points: 40

Due: May 10, 2017 (Wednesday) at 11:55 PM

Hello from the galaxy far, far away! Today, you are going to write a program in **Python 2.7** to understand the SSL/TLS Certificate by fetching and displaying some of its content. This is mostly an investigative assignment (and barely has anything to do with cryptography.) The goal of this assignment is to help you understand the role of a trusted third-party certificate authority in guarantying authenticity. Remember, the underlying mechanism behind SSL/TLS greatly depends on several advanced cryptographic principles.

Instructions:

- Take **two** arguments from the command-line:
  1. Hostname: the hostname of the server. *Example*: latech.edu.
  2. Port: the HTTPS port number. *Common port*: 443.
- Import the default `ssl` and `socket` libraries in Python.
- Establish a socket connection to the passed `hostname` and `port`.
- Get the certificate.
- Parse out the following information from the certificate:
  - List of domain names certified by the certificate.
  - Certificate Issue Date and Time
  - Certificate Expiry Date and Time
  - Certificate Issuer's Common Name
  - CA Issuers URL
  - OCSP (Online Certificate Status Protocol) URL
- Get the type of cipher being used. It is a three-value tuple:
  - the name of cipher,
  - the version of SSL protocol, and
  - the number of secret bits.
- Output all the extracted information on the console.

Bonus:

If your program performs a TLS handshake with the server, gets its public key, and prints it, then you will receive **five** bonus points in this assignment. `PyOpenSSL` library may come in handy for this.

Sample Execution Command and Output:

```
$  python  latech.edu  443
```

Output:

```
Domain Names:  *.latech.edu, latech.edu
Issue Date:    Oct 20 16:20:03 2014 GMT
Expiry Date:   Oct 23 15:42:27 2017 GMT
Issuer's CN:   Go Daddy Secure Certificate Authority - G2
CA Issuers:    http://certificates.godaddy.com/repository/gdig2.crt
OCSP:          http://ocsp.godaddy.com/
Cipher:        DHE-RSA-AES256-SHA
SSL Version:   TLSv1/SSLv3
Secret Bits:   256
```

Submission Guidelines:

1. Write comments on your source code file (*ssl_certificate_reader.py*). Include *author's name*, *date*, *description*, *list of resources used*, and so on.
2. Within the description of your program, specify whether you have successfully attempted the bonus problem.
3. Upload the source code file to **Moodle** by the deadline.