Network Programming 2024
Assignment 2

Submission Instruction:

==================

- Due date: April 30, 2024
- File format: Compress/zip the following files in a zip format

(1) Solution/code: server.py and client.py
(2) Report (including Execution screenshots and explanations): report.pdf
(3) Certificate file: cert.crt
(4) Key file: key.pem

- zip file name: {student ID}.zip (e.g., 123456789.zip)

Example File tree

```
     - 123456789.zip
          - server.py
          - client.py
          - report.pdf
          - cert.crt
          - key.pem
```

- There will be a deduction of points (up to 50%) if you do not follow the above instructions.

Evaluation Criteria:

==================

- Your codes will be cross-checked with others to detect copying, and you will get zero points under such circumstances.
- Your codes will be evaluated based on the following:
  1) Programming style (organization, comments, readability),
  2) Program functionality (completeness, exception handling),

3) Completeness of documentation(report). (execution screenshots, method/function explanation, etc.)

Inquiries:

==================

- Send inquiries to the TA: Ankit Kumar Singh (ankit@soongsil.ac.kr).

Some Suggestions to solve the problem:

- Read the submission instructions carefully (Incorrect submission format may lead to the deduction of points).
- Read textbook chapters 5 and 6.
- Read problem statements 2-3 times to understand.
- To further solve your doubt or get clarity, please inquire with the TA.

Assignment #2 [Total: 10 points]

Problem Statement:

Assignment #2 is the extension of Assignment #1, in which you will implement a secure, robust TCP client-server communication system for the "Guess the Number" game using Python's socket and SSL libraries. This implementation should ensure that all communications are encrypted using SSL/TLS and handle data transmission and reception reliably.

Requirements:

1. Server-Client Communication:
    a. Implement a synchronous server that can handle one client at a time.
    b. The server chooses a random number between 1 and 10, which the client must guess.
    c. The client will request to start the game by sending the first request as "start."
    d. The client will have only five attempts to guess the correct number
    e. The client will only win if he guesses the correct number within five attempts.
2. Data Handling:
    a. Communications between the server and the client should be serialized using JSON.
    b. Implement error handling to manage incomplete or improperly formatted JSON data.
3. Security Features:
    a. Use SSL/TLS to secure all client and server communications.
    b. Generate your own SSL certificates for testing.
4. Game History:
    a. The server and the client should maintain a history of all messages exchanged during the game.
    b. This history should be compressed and saved at the end of each game session using Python's *pickle* and *zlib* modules.

c. Ensure the chat history can be loaded and displayed upon restarting the server or client.

5. Logging:
   a. Implement logging to capture important events such as connections, disconnections, and errors.
6. Error Handling
   a. The program should handle the proper termination of the client and server.
   b. The application should be able to detect and handle network errors such as Connection, Socket, Timeout, invalid certificates, invalid client requests, etc.

The exchange of messages between the server and client during the game will follow the following conditions and message text based on the difference between the actual number with the server and the guessed number by the client:

x = randomly chosen number by server
guess = number guessed by a client
attempts = number of attempts by the client

Game Conditions and messages:
(x = guess)   → "Congratulations, you did it."
(x > guess)   → "Hint: You guessed too small!"
(x < guess)   → "Hint: You Guessed too high!"
(attempts > 5) → "Sorry, you've used all your attempts!"