TNE20003 – Internet and Cybersecurity for Engineering Applications

Portfolio Task – Lab 8 Credit Task

Aims:

- To develop a Python implementation of a simple protocol using TCP. Students will rewrite their UDP solution to TCP. As TCP is a stream-based protocol, students will be required to modify the program to allow both client and server to locate the beginning and end of each message in the Protocol

Preparation:

- View "Internet Enabled Programming" & "Network Protocol Implementation"

Due Date:

- All tasks in this lab are to be completed and demonstrated to your Lab instructor preferably during or at the end of the current lab, but if you do not complete the tasks you may demonstrate it at the beginning of your next lab class. To do this you must upload all documents up to Canvas to ensure that you complete and hand the task in on time. This submission is to be no later 9pm on the day of the next lab. For example if your lab is on the 18/9, then final submission is no later than 9pm on the 25/9.

Methodology:

You will be developing a new set of programs implementing a modified version of the protocol to function with TCP. It would be strongly advised to use the threaded echo server code from the tutorial as a base for building your TCP server.

The current implementation of the Protocol does not require delineation of the start and end of messages. As the Protocol is UDP based, all messages are delivered as distinct chunks. When you send a message, the remote end (client or server) is guaranteed to receive the entire message in a single read. They are also guaranteed that two messages will not be merged in a single read. This means that the program is simple in that you know you are parsing an entire message for validity and to construct a response.

TCP is more complex as it is a Stream-based Protocol. This means that all messages run into each other and a read from a remote system can contain multiple messages, part of a message, or the end part of one message and the start of a second message. Your task will require modifying the protocol to allow the receiving program to find the start and end of each message from the TCP stream. This means maintaining some persistence of the message between reads. There are multiple valid mechanisms for modifying the protocol, you are free to choose any mechanism you like to do so.

Task:

You will need to develop two Python programs to implement the TCP version of the protocol. The functionality otherwise needs to be exactly the same in terms of parsing and responding to packet contents

Assessment:

As a Credit task, not completing this task will result in the maximum achievable base grade for your Portfolio being restricted.

To pass this task, you must demonstrate the functioning program to your Lab Supervisor. Your supervisor will ask you some questions about how the code functions to validate that it is your work. Upon successful demonstration and answering questions, this task will be marked as complete