CS 467 Computer Networks

Spring 2012

Name: C2C Gavin Delphia

Questions (20 points)

(4 pts) In this assignment, your instructor mandated the use of threads. Explain the rationale

behind this mandate, with special emphasis on the specific conditions which makes

multithreading useful in creating a networked application.

The point of using threads for this PEX is so that the TCP client can constantly wait for messages from the server while at the same time send messages to the server. If it had not been multithreaded, when the blocking wait for message was called the entire application would have frozen. The conditions that make multi-threading useful for networked applications are when you want to send and receive data to and from a server at the same time.

(4 pts) Suppose, after reexamining the communications protocol, your instructor believes that the

overhead incurred by using TCP is preventing the server from scaling up to thousands of

concurrent users. As a result, he is considering switching transmission protocols from TCP to

UDP.

a. (2 Pts) Can such a switch be accomplished without changing your client application?

Explain your answer.

Other than changing the actual protocol implementation in JAVA my client should still work the same using UDP. This is because the protocol for UDP and TCP basically receive and send packets in the same coding way, it is just the implementations inside their classes that are different.

b. (2 Pts) Describe some of the drawbacks of switching Smallworld from a TCP to UDP

application?

The drawbacks of switching to UDP is that the messages are no longer known or not if they reached their destination.

(4 pts) Your instructor is considering revising the format of the ATTACK command. In addition

to specifying the target, as is currently required, the new format would include a second

parameter which determines how much damage the player inflicts on his/her target. Is this

approach a good idea from a client/server design perspective? Explain your answer.

This is not a good idea. First off you can never trust the client. This means that I could put any number I wanted for the damage amount regardless if it is correct. Also it would defeat any of the servers logic that adds a randomness to the calculation of damage.

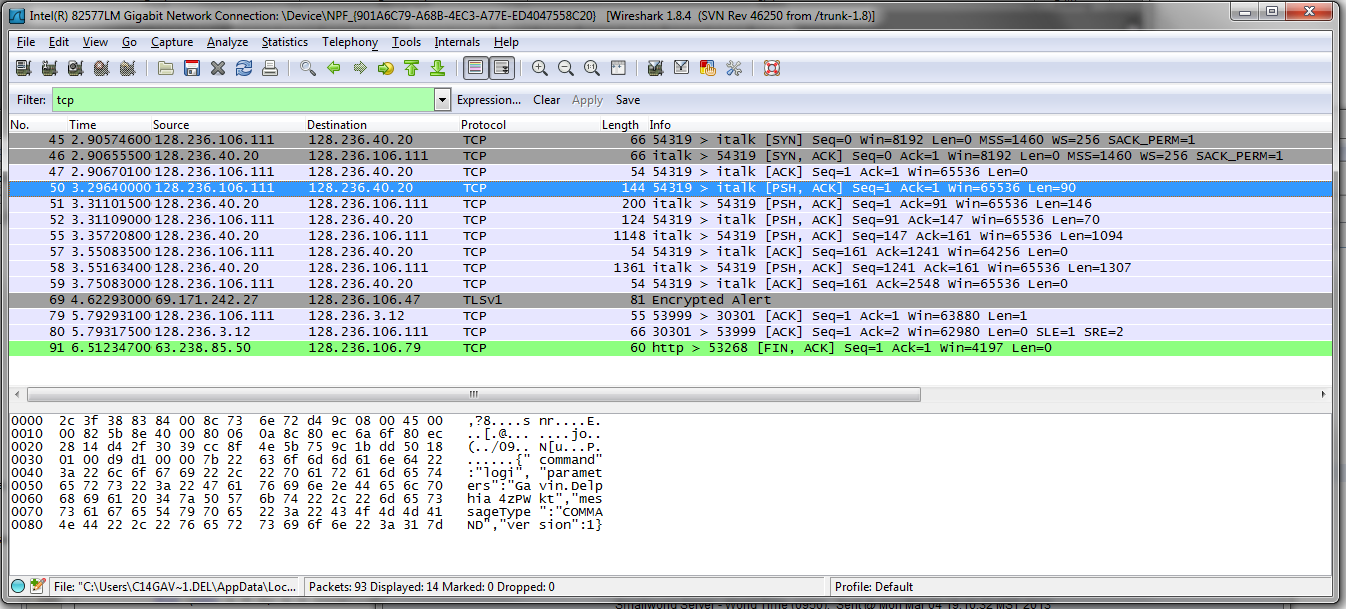
(4 pts) Perform a Wireshark capture of your application’s network traffic when it is trying to log

into the server (attach a screenshot that shows the data in the packets). What do you notice about

the login command message that is troubling from a security perspective? How would you

address this issue?

The login command sends the username and password in plaintext. This means that anyone can steal the login information if they were watching the network traffic. One way to fix this would be to encrypt the data on the clients end but this would require the server to use the same encryption scheme.



(4 pts) The Smallworld server transmits field of view data to the client through the use of easily

readable String descriptors. This works well when the amount of data being sent to the client is

small, but can result in lengthy transmissions when a large number of entities are within the

player’s field of view. Propose a modification to the current communications protocol which

will reduce the size of each transmission. What (if any) are the potential drawbacks of your

proposal?

If every entity in the world had a unique type identifier you could cut down on the data transmission amount. This would allow you to send a small string of integers to each client connected rather than a huge list of entities. Then you also provide each client with their own copy of a lookup table to convert the integer identifier data into a useable entity that can then be processed. One drawback is that every client needs the lookup table present on their machine. This could cause difficulties when updating the server with new content.