

COMP/ELEC 416 : Computer Networks

Project #2

Due: Mar 29, 2020, 11.59pm (Late submissions will not be accepted.)

This is a group (3-student) project.

We suggest a fair task distribution among group members at the end of this project description.

Transport Protocols and Secure Sockets Layer Analysis with WireShark

This project is about **transport layer** of the network protocol stack.

The focus is on the **SSL, TCP and UDP protocols**. For this purpose, you are asked to modify the provided SSL client/server codes as specified below, experiment with TCP and UDP features and use the **WireShark network protocol analyzer** tool to answer transport layer related questions.

WireShark is the world's foremost network protocol analyzer, and is the **de facto standard** across many industries and educational institutions. It can be downloaded freely at <http://www.wireshark.org/download.html>. WireShark allows users to trace the network activity by capturing all the packets that hit to your network interface. It tags the information of each layer by parsing the given byte stream according to the corresponding protocol.

You should read this project document carefully before starting your implementation and tasks. Every part assumes the communication between server and multiple clients.

Part 1 - SSL Implementation and Experiments:

Figure 1 illustrates an overview of SSL protocol. Recall that, you are provided an SSL client/server code that performs echoe on top of an SSL socket. The corresponding SSL PS codes and slides are available through the course web site.

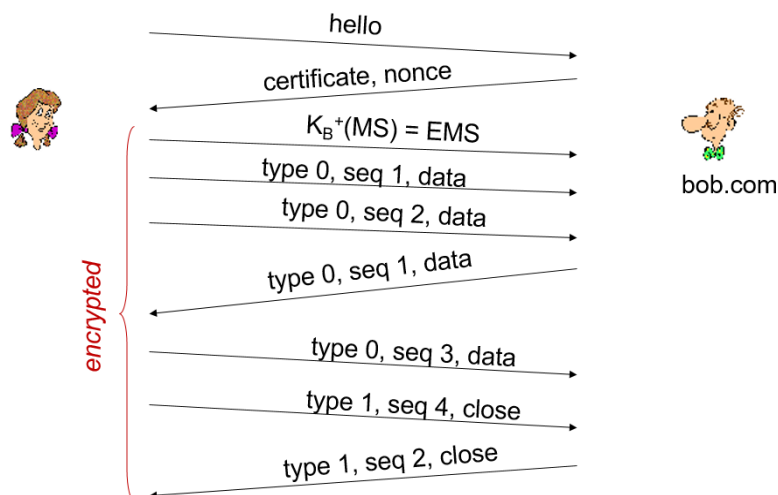


Figure 1. SSL Protocol Overview

As presented in the figure, the certificate is sent by the server to the user in start of the session. User adds this certificate to the local key store and uses it for authentication. The code is provided to add the certificate to the local key store, but the part where the server sends the certificate to the user is missing. In this part, you are asked to modify the provided code as follows.

- Set up a TCP connection on which the certificate will be transferred to the client. TCP connection can listen to any port. The server should ask for client verification before the certificate is transferred to the client. You can keep an already known users file on the server side and use them for login.
- Use the certificate to connect to the server through SSL. You should keep the certificate in the right directory.
- SSL connection at server side should listen the port with SUM(KUSIS ID's of group members) number (look at the first question). Handle the case where the number becomes larger than the available ports and explain the answer in the report.
- After SSL connection is established, your client should receive full email addresses of all the group members character by character in separate messages (to have multiple messages to examine) with a non-persistent manner and should print the character it receives for every new connection. For example, if there are two emails abc@ku.edu.tr and xyz@ku.edu.tr then the individual messages would be 'ax', 'by' 'rr'. Then, the email addresses should be printed at the client side.

Important Notes:

- Your modified SSL codes must be submitted along with your project report. In your project report, you should explain your answers and provide your WireShark outputs for each question, in order to get credit.
- On Windows operating system, you might not be able to capture Loopback interface, which is the traffic inside your operating system. When you run your server and client software in single operating system, in order to capture incoming and outgoing packets, you need to capture Loopback interface. To solve this problem, you can run one of the application (client or server) in a separate machine. For instance, you can use a computer in the Computer labs.

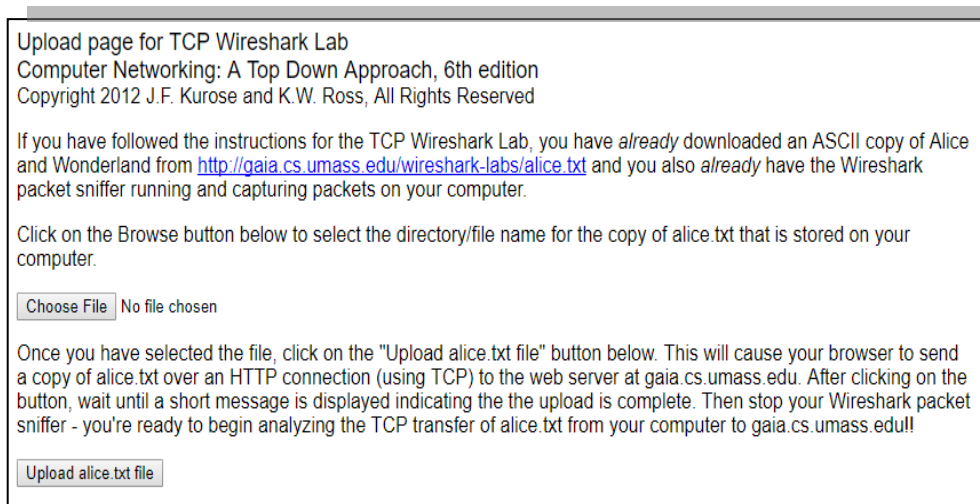
After running your code, answer the following questions:

1. What is the maximum value of a port number? Why is this the case? Locate the SSL Server IP address and port number, client IP address and port number that through these agents are communicating by using Wireshark.
2. Locate the data containing TCP segments. What is written in the data field? Compare it with the data you exchanged between the client and server. Why do you think is this the case?
3. How many TCP segments transmitted in total while your email address is exchanged one by one with non-persistent connections?
4. What difference did you see in the payload of SSL and TCP? Can you locate the login name and password entered by the user? Can you locate the email information?

Part 2.a - TCP Experiments:

Before beginning the exploration of TCP, you need to use Wireshark to obtain a packet trace of the TCP transfer of a file from your computer to a remote server. You are asked to do so by accessing a Web page that will allow you to enter the name of a file stored on your computer (which contains the ASCII text of *Alice in Wonderland*), and then transfer the file to a Web server using the HTTP POST method. We are using the POST method rather than the GET method as we would like to transfer a large amount of data *from* your computer to another computer. Of course, you need to run Wireshark during this time to obtain the trace of the TCP segments sent and received from your computer. Perform the following:

- Start up your web browser. Go the <http://gaia.cs.umass.edu/wireshark-labs/alice.txt> and retrieve an ASCII copy of *Alice in Wonderland*. Store this file somewhere on your computer.
- Next go to <http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html>
- You should see a screen that looks like:



The screenshot shows a web page with the following content:

Upload page for TCP Wireshark Lab
Computer Networking: A Top Down Approach, 6th edition
Copyright 2012 J.F. Kurose and K.W. Ross, All Rights Reserved

If you have followed the instructions for the TCP Wireshark Lab, you have *already* downloaded an ASCII copy of Alice and Wonderland from <http://gaia.cs.umass.edu/wireshark-labs/alice.txt> and you also *already* have the Wireshark packet sniffer running and capturing packets on your computer.

Click on the Browse button below to select the directory/file name for the copy of alice.txt that is stored on your computer.

No file chosen

Once you have selected the file, click on the "Upload alice.txt file" button below. This will cause your browser to send a copy of alice.txt over an HTTP connection (using TCP) to the web server at gaia.cs.umass.edu. After clicking on the button, wait until a short message is displayed indicating the the upload is complete. Then stop your Wireshark packet sniffer - you're ready to begin analyzing the TCP transfer of alice.txt from your computer to gaia.cs.umass.edu!!

- Use the *Browse button* in this form to enter the name of the file (full path name) on your computer containing *Alice in Wonderland* (or do so manually). Don't yet press the "*Upload alice.txt file*" button.
- Now start up Wireshark and begin packet capture (*Capture->Start*) and then press *OK* on the Wireshark Packet Capture Options screen (we will not need to select any options here).
- Returning to your browser, press the "*Upload alice.txt file*" button to upload the file to the `gaia.cs.umass.edu` server. Once the file has been uploaded, a short congratulations message will be displayed in your browser window.
- Stop Wireshark packet capture.

Answer the following questions for the TCP segments:

5. How do you extract the IP address and port numbers from the captured segment? Explain with screenshots.
6. How do you extract the payload from the captured segment? Explain with screenshots.
7. What are the sequence numbers (which appear in the Wireshark program) of the segments used for the 3-way handshake protocol that initiates the first TCP connection? What are the sequence numbers of those segments and the port numbers used on client and server sides?
8. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and `gaia.cs.umass.edu`? What is it in the segment that identifies it as a SYN segment?
9. What is the sequence number of the SYNACK segment sent by `gaia.cs.umass.edu` to the client computer in reply to the SYN?
10. What is the value of the Acknowledgement field in the SYNACK segment? How did `gaia.cs.umass.edu` determine that value? What is it in the segment that identifies the segment as a SYNACK segment?
11. What is the sequence number of the TCP segment containing the HTTP POST command? Note that, in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.
12. Consider the TCP segments containing the HTTP POST as the last segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection? At what time was each segment sent? When was the ACK for each segment received? Given the difference between when each TCP segment was sent, and when its acknowledgement was

received, what is the RTT value for each of the six segments? What is the **EstimatedRTT** value (see Section 3.5.3 in the textbook) after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation (Section 3.5.3 in the textbook) for all subsequent segments.

13. How many TCP segments are sent from client to server? How many IP packets are received? Is there any difference in between the two answers? If yes, why?
14. Did you observe any re-transmission by TCP protocol in your captures? Which part of the Wireshark did you check for this purpose?
15. Can you identify the packets that are lost before reaching the receiver?
16. Did you see the same initial sequence number for TCP over the different executions of your client-server connections? Explain your answer in detail.

Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the “listing of captured packets” window that is being sent from the client to the gaia.cs.umass.edu server. Then select: *Statistics->TCP Stream Graph- >Round Trip Time Graph*.

Part 2.b - Comparison (TCP vs SSL) Experiments:

17. How do you retrieve the exchanged application layer’s message from the captured segments?
18. What are the differences between the SSL and TCP parts?
19. Which part of the information is not retrievable in SSL? Do you have any solution to retrieve those parts in SSL? You may need to do some research to answer these questions.
20. How many checksums does an SSL/TCP segment have in checksum field? Why?

Part 3 - UDP Experiments:

In this part, you are assigned a **unique** URL to work with. The list is provided in file Project2_URL_List.pdf, and you must use the URL assigned you. You should provide the appropriate screenshots and work on the correct domain, in order to get credit.

nslookup command works as an IP address resolver. When you provide a domain name as argument, it will return the IP address of that domain. Now, take the following steps provided and answer the questions accordingly.

- Start your Wireshark software and start capturing packets from the appropriate interface.
- Use nslookup command in order to resolve the IP address of the URL that is assigned you.
- Stop packet capturing in the Wireshark.
- Apply an appropriate **display filter**.

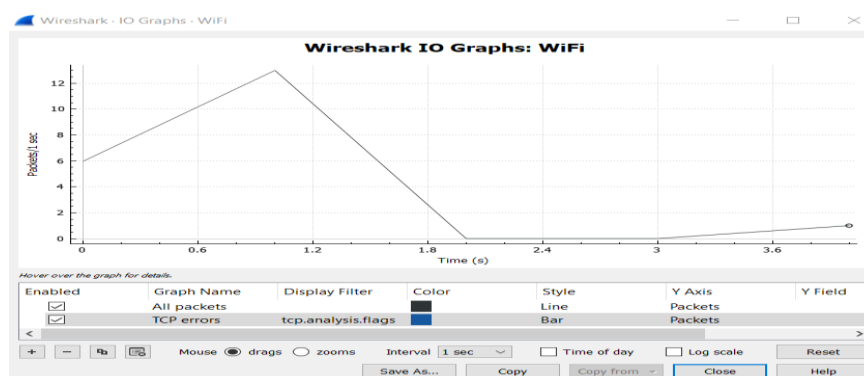
21. What display filter did you apply in order to see appropriate packets?
22. Which application layer and transport layer protocol do nslookup work on? What is the reason that transport layer protocol is chosen?
23. What is the IP address of the website you that you sent a query?
24. Can you derive the local DNS server you connected work in iterative or recursive manner? If you can or cannot please provide a detailed explanation. Please also briefly explain the advantages and disadvantages iterative and recursive approach over each other.
25. What are the header lengths of application layer protocol and transport layer protocol that nslookup works on?
26. How many checksums does an UDP segment has in checksum field? Why?

Part 4 – Experiments for Two Competing Streams:

In this section, you are asked to use Wireshark's IO graph tool to see the effect of running multiple streams on the network i.e. how they react to contention of the resources. For this, you should do the following tests:

1. UDP vs UDP: Send two lookup requests for DNS simultaneously.
2. TCP vs TCP: Perform Part 2.a setup using two connections simultaneously.
3. UDP vs TCP: Send DNS lookup and URL query simultaneously.

Provide the graphs for each case in your report. Based on graphs, explain how the streams adjust themselves. How the flow control and congestion window is adjusted in TCP? Sample graph is.



Note: To access IO graph tool select: *Statistics->IO graph.*

Project Deliverables:

Important Note: You are expected to submit a project report, in PDF format, that documents and explains all the steps you have performed in order to achieve the assigned tasks of the project. A full grade report is one that clearly details and illustrates the execution of the project. Anyone who follows your report should be able to reproduce your performed tasks without effort. Use screenshots to illustrate the steps and provide clear and precise textual descriptions as well. All reports would be analyzed for plagiarism. Please be aware of the KU Statement on Academic Honesty.

The name of your project .zip file must be <Group Number>-<surnames>.zip

You should submit a single .zip file including:

- Source Codes: A .zip or .rar file that contains your codes in a single Eclipse or IntelliJ IDEA. If you aim to implement your project in any IDE other than the mentioned ones, you should first consult with TA and get confirmation.
- Report: Answers and the corresponding Wireshark screenshots. The **report** is an **important part of your project**, which should be submitted as both a .pdf and Word file. The **report acts as a proof of work to assert your contributions to this project**. Figures in your report should be scaled to be visible and clear enough. All figures should have captions, should be numbered according to their order of appearance in the report, and should be referenced and described clearly in your text. All pages should be numbered, and have headers same as your file naming criteria.
- Saved capture file from the Wireshark.

If you use any (online) resources in this project, you must reference them in your report.

There is no page limit for your report, and no specific requirements on the design.

Suggested Task Distribution:

Student 1: Part 1 and Part 2.b

Student 2: Part 2.a

Student 3: Part 3 and Part 4

All: Wireshark packet captures, codes, and the report.

Good Luck!