Lab Exercise 1
Examining HTTP with Wireshark[1]

OBJECTIVE:

The aim of this lab is to get familiar with Wireshark and use it to examine HTTP. Wireshark is an open source software tool that allows you to examine packets captured by any network interface on your machine. Note that Wireshark is the new version of Ethereal.

BACKGROUND:

The basic tool for observing the messages exchanged between executing protocol entities is called a ***packet sniffer***. As the name suggests, a packet sniffer captures ("sniffs") messages being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured messages. A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself. Similarly, received packets are never explicitly addressed to the packet sniffer. Instead, a packet sniffer receives a copy of packets that are sent/received from/by application and protocols executing on your machine.

LAB ACTIVITIES:

Please work through each of the tasks discussed below. A number of the questions will touch on the fundamental networking concepts that we have not yet fully covered in class. For this senior course we expect you will consult the textbook, the Web and other courses to guide you. Please note, as always, citations must be provided with your answers if you consult any external source for information.

1. To get started please read the attached basic introduction on Wireshark.

2. Get your first capture with Wireshark:

    a. Start Wireshark **as an administrator** and launch a packet capture. Select the Capture pull down menu and select Options. Select the network interface (i.e., the physical connection) that your home computer has. Click "Start".

    b. Before performing the steps below, make sure your browser's cache is empty. Start up your favourite Web browser and display the webpage http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html. In order to display this page, your browser will contact the HTTP server and exchange HTTP messages with the server in order to download this page. The Ethernet or Ethernet II frames containing these HTTP messages will be captured by Wireshark.

    c. After your browser has displayed the webpage, stop Wireshark packet capture by selecting stop in the Wireshark capture window. This will cause the Wireshark capture window to disappear and the main Wireshark window to display all packets captured since you began packet capture. Please go to "File" and select save file (e.g., call it a ***trace***).

3. Examine the trace which contains live packet data exchanged between your machine and the Web server (the host providing the webpage to your machine).

    a. Even though the only action you took was to download a webpage, there were evidently many other protocols running on your computer that are unseen by the user. We'll learn much more about these protocols as we progress through the text! In the trace you can see many protocols

[1] Adapted from "Wireshark Labs for Computer Networking: A Top-Down Approach, 8th Edition" by Kurose & Ross.

listed. Choose two or three of these protocols from your list and use Google or go to the IETF site to find out what they are being used to support.

b. The protocol used to exchange messages between a Web browser and Web server is HTTP. Type in "http" (without the quotes, and in lower case - all protocol names are in lower case in Wireshark) into the display filter specification window at the top of the main Wireshark window. Then apply the filter (to the right of where you entered "http"). This will cause only HTTP message to be displayed in the packet-listing window.

4. Based on the above trace, answer the following questions:

   a. Select the first HTTP message shown in the packet-listing window. This should be the HTTP GET message that was sent from your computer to the HTTP server. When you select the HTTP GET message, the Ethernet or Ethernet II frame, IPv4 datagram, TCP segment, and HTTP message header information will be displayed in the packet-header window. How long did it take from when the HTTP GET message was sent until the first HTTP response was received? (By default, the value of the Time column in the packet listing window is the amount of time, in seconds, since Wireshark tracing began.)

   b. In the trace you will find IPv4 addresses within the packets. Find an example packet in the trace where the IPv4 address associated with your machine is present. Compare this address with the IPv4 address you can find by using the following command: *ipconfig* on Windows, or *ifconfig* on MacOS or Linux. Running this command also gives a 48-bit physical address for the network interference. This is the MAC (medium access control) address (not discussed in class yet) of your machine. Can you find the MAC address of the packet in the trace? Is it the same as the physical address returned by the above command?

   To protect the information, you can hide your MAC address and IPv4 address in your answers.

5. Next, let's have a new Wireshark capture. As we discussed in the lectures, most Web browsers perform object caching and thus perform a conditional GET when retrieving an HTTP object. Before performing the steps below, make sure your browser's cache is empty. Now do the following:

   a. Start a new capture with Wireshark;

   b. Enter http://cs.unb.ca/~wsong/lab1.html into your browser and your browser should display a very simple five-line HTML file[2];

   c. Click the reload button to refresh the webpage within **the same tab** of your browser;

   d. Stop Wireshark packet capture, and enter "http" in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window.

6. Examine your new trace captured in item 5 and answer the following questions:

   a. Inspect the contents of the first HTTP GET request **for the Webpage "lab1.html"** from your browser to the server[3]. Do you see an "If-Modified-Since" line in the HTTP GET?

[2] Note: This is the exact URL you should use. Make sure that you are not visiting the secured UNB website. Otherwise, Wireshark may not be able to capture the encrypted packets from a secured website. In case that the above URL does not work, you can visit the following webpage: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html.

[3] If you use the webpage: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html in step 5, inspect the HTTP GET request for "HTTP-wireshark-file3.html".

b. Inspect the contents of the server response to the first HTTP GET request. Did the server explicitly return the contents of the file? How can you tell?

c. Now inspect the contents of the next HTTP GET request **for the Webpage "lab1.html"** from your browser to the server[4]. Do you see an "If-Modified-Since:" line in the HTTP GET? If so, what information follows the "If-Modified-Since:" header?

d. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET for "lab1.html"? Did the server explicitly return the contents of the file? Explain.

LAB REPORT:

Go through the above lab activities and write a report **answering the questions listed in item 4 and item 6**.

- **Please follow the lab report template in Microsoft Word posted on D2L.** The template includes one example answer for a question listed in item 4. Once you are done, convert your Word document into **a single PDF file** and submit it to the corresponding dropbox on D2L by the due time.
- In your report, you need to attach a screenshot of the packet trace to indicate where you find the answer to each question. You can capture the display in Wireshark using a screen capture software and then annotate the output packet trace with color markups to show the information related to your answers.

[4] If you use the webpage: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html in step 5, inspect the HTTP GET request for "HTTP-wireshark-file3.html".