Assignment 2

March 15th , 2017

Deadline : 11:55 PM, 30th March

Part 1 : Wireshark HTTP

A. The Basic HTTP GET/response interaction

Download a very simple HTML file - one that is very short, and contains no embedded objects. Do the following:

- Start up your web browser.
- Start up the Wireshark packet sniffer (but don't yet begin packet capture). Enter "http" (just the letters, not the quotation marks) in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window. (We're only interested in the HTTP protocol here, and don't want to see the clutter of all captured packets).
- Wait a bit more than one minute (you'll see why shortly), and then begin Wireshark packet capture.
- Enter the following to your browser

http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html
Your browser should display the very simple, one-line HTML file.

- Stop Wireshark packet capture.

If you are unable to run Wireshark on a live network connection, you can download a packet trace that was created when the steps above were followed.

By looking at the information in the HTTP GET and response messages, answer the following questions. When answering the following questions, you should print out the GET and response messages and indicate where in the message you've found the information that answers the following questions.

1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running? What languages (if any) does your browser indicate that it can accept to the server?
2. What is the IP address of your computer? Of the gaia.cs.umass.edu server?
3. What is the status code returned from the server to your browser?
4. When was the HTML file that you are retrieving last modified at the server?
5. How many bytes of content are being returned to your browser?
6. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

B. The HTTP CONDITIONAL GET/response interaction

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. (To do this under Firefox, select Tools->Clear Private Data, or for Internet Explorer, select Tools->Internet Options->Delete File; these actions will remove cached files from your browser's cache.)

Now do the following:

- Start up your web browser, and make sure your browser's cache is cleared, as discussed above.
- Start up the Wireshark packet sniffer
- Enter the following URL into your browser http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html

Your browser should display a very simple five-line HTML file.

- Quickly enter the same URL into your browser again (or simply select the refresh button on your browser)
- 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.

(Note: If you are unable to run Wireshark on a live network connection, you can use the http-ethereal-trace-2 packet trace to answer the questions below)

Answer the following questions:

7. Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE” line in the HTTP GET?
8. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?
9. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header?
10. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

Part 2 : Wireshark DNS

A. Tracing DNS with Wireshark

First you need to capture the DNS packets that are generated by ordinary Web-surfing activity.

- Use ipconfig(in Windows)/ifconfig to empty the DNS cache in your host.

- Open your browser and empty your browser cache. (With Internet Explorer, go to Tools menu and select Internet Options; then in the General tab select Delete Files.)
- Open Wireshark and enter “ip.addr == your_IP_address” into the filter, where you obtain your_IP_address (the IP address for the computer on which you are running Wireshark) with ipconfig. This filter removes all packets that neither originate nor are destined to your host.
- Start packet capture in Wireshark.
- With your browser, visit the Web page: http://www.ietf.org
- Stop packet capture.

Answer the following questions:

1. Locate the DNS query and response messages. Are they sent over UDP or TCP?
2. What is the destination port for the DNS query message? What is the source port of DNS response message?
3. To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same?
4. Examine the DNS response message. How many “answers” are provided? What does each of these answers contain?
5. Consider the subsequent TCP SYN packet sent by your host. Does the destination IP address of the SYN packet correspond to any of the IP addresses provided in the DNS response message?
6. This web page contains images. Before retrieving each image, does your host issue new DNS queries?

Now do the following.

- Start packet capture.
- Do an nslookup on www.mit.edu.
- Stop packet capture.

Answer the following questions:

7. What is the destination port for the DNS query message? What is the source port of DNS response message?
8. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
9. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
10. Examine the DNS response message. How many “answers” are provided? What does each of these answers contain?

Part 3 : Wireshark TCP

A. Capturing a bulk TCP transfer from your computer to a remote server. Do 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.
- 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->Options) and then press OK on the Wireshark Packet Capture Options screen (we’ll 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.
- Filter the packets displayed in the Wireshark window by entering “tcp” (lowercase, no quotes, and don’t forget to press return after entering!) into the display filter specification window towards the top of the Wireshark window.

What you should see is series of TCP and HTTP messages between your computer and gaia.cs.umass.edu. You should see the initial three-way handshake containing a SYN message. You should see an HTTP POST message and a series of “HTTP Continuation” messages being sent from your computer to gaia.cs.umass.edu. You should also see TCP ACK segments being returned from gaia.cs.umass.edu to your computer.

Answer the following questions, by opening the Wireshark captured packet file tcp-ethereal-trace-1 in http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip (that is download the trace and open that trace in Wireshark). Whenever possible, when answering a question you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. To print a packet, use File->Print, choose Selected packet only, choose Packet summary line, and select the minimum amount of packet detail that you need to answer the question.

1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details of the selected packet header window”.
2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

If you have been able to create your own trace, answer the following question:

3. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

Change Wireshark’s “listing of captured packets” window so that it shows information about the TCP segments containing the HTTP messages, rather than about the HTTP messages. To have Wireshark do this, select Analyze->Enabled Protocols. Then uncheck the HTTP box and select OK. Use the packet trace that you have captured (and/or the packet trace tcp-ethereal-trace-1 in http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip) to study TCP behavior.

Answer the following questions:

4. 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 the segment as a SYN segment?
5. 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.

Part 4 : Wireshark UDP

Start capturing packets in Wireshark and then do something that will cause your host to send and receive several UDP packets. (One way to do this would be to use the nslookup command, as we saw in the DNS Wireshark lab.) After stopping packet capture, set your packet filter so that Wireshark only displays the UDP packets sent and received at your host. Pick one of these UDP packets and expand the UDP fields in the details window. Whenever possible, when answering a question you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printout to explain your answer. To print a packet, use File->Print, choose Selected packet only, choose Packet summary line, and select the minimum amount of packet detail that you need to answer the question.

1. Select one packet. From this packet, determine how many fields there are in the UDP header. Name these fields.
2. The value in the Length field is the length of what? Verify your claim with your captured UDP packet.
3. What is the maximum number of bytes that can be included in a UDP payload? What is the largest possible source port number?
4. What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. (To answer this question, you'll need to look into the IP header.)
5. Search “UDP” in Google and determine the fields over which the UDP checksum is calculated.

General Instructions:

- Include screen shots of the wireshark display wherever needed.
- Use Moodle for general doubts and discussion and resolving queries. This assignment has to be done individually.
- There will be manual evaluation for this assignment.
- Plagiarism in any form shall not be tolerated and a straight ‘F’ grade for the course will be given.

Submission Format:

- Your submission will be a single PDF file ‘<roll number>.pdf’ with the answers.