CSP334: Computer Networks
Homework Assignment No 4
Chap 3, Application Layer: http Wireshark Assignment

Date of submission : 23:59 hrs, 23rd September 2018

Autumn Semester 2018-19

Instructions:

- The exercise questions here have been adapted from the same available with the Text book supplement.
- If you are not able to submit by the specified time, whtsoever may be the reason, without any late policy, the assignment grading will be zero.
- The total points for this exercise are 100.

In this lab, well explore several aspects of the HTTP protocol: the basic GET/response interaction, HTTP message formats, retrieving large HTML files, retrieving HTML files with embedded objects, and HTTP authentication and security. Before beginning these labs, you might want to review Section 2.2 of the text.

SET 1: The Basic HTTP GET/response interaction :

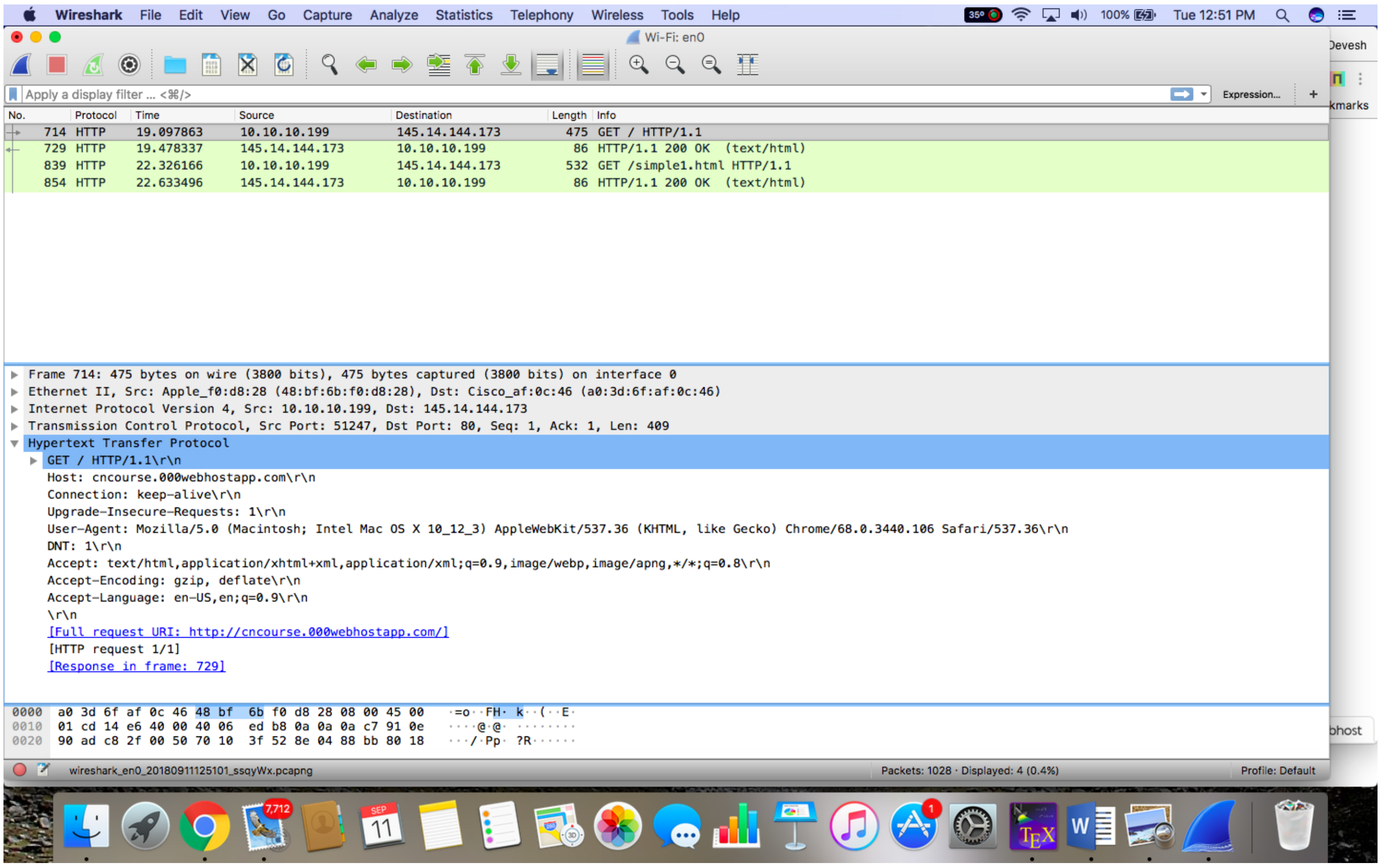
Let us begin our exploration of HTTP by downloading a very simple HTML file - one that is very short, and contains no embedded objects. Do the following:

- Start up your web browser.
- Start up the Wireshark packet sniffer, as described in the Introductory lab (but dont 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. (Were only interested in the HTTP protocol here, and dont want to see the clutter of all captured packets).
- Wait a bit more than one minute (well see why shortly), and then begin Wireshark packet capture.
- Enter the following to your browser `https://cncourse.000webhostapp.com/simple1.html` Your browser should display the very simple, one-line HTML file. Stop Wireshark packet capture. Your Wireshark window should look similar to the window shown in Figure 1. The example in Figure 1 shows in the packet-listing window that two HTTP messages were captured: the GET message (from your browser to the *cncourse* web server) and the response message from the server to your browser.

  Recall that the HTTP message was carried inside a TCP segment, which was carried inside an IP datagram, which was carried within an Ethernet frame. Wireshark displays the Frame, Ethernet, IP, and TCP packet information as well. We want to minimize the amount of non-HTTP data displayed (were interested in HTTP here, and will be investigating these other protocols is later labs). so, as you learnt in the first exercise, make sure the boxes at the far left of the Frame, Ethernet, IP and TCP information have a plus sign or a right-pointing triangle (which means there is hidden, undisplayed information), and the HTTP line has a minus sign or a down-pointing triangle (which means that all information about the HTTP message is displayed).

  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 youve found the information that answers the following questions.

  When you hand in your assignment, annotate the output so that its clear where in the output youre getting the information for your answer (e.g., annotate electronic copies with text in a colored font).


Wireshark File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
100% Tue 12:51 PM
Wi-Fi: en0
Apply a display filter ... <⌘/>
Expression...

```
No.  Protocol  Time       Source          Destination     Length  Info
714  HTTP      19.097863  10.10.10.199    145.14.144.173     475  GET / HTTP/1.1
729  HTTP      19.478337  145.14.144.173  10.10.10.199        86  HTTP/1.1 200 OK  (text/html)
839  HTTP      22.326166  10.10.10.199    145.14.144.173     532  GET /simple1.html HTTP/1.1
854  HTTP      22.633496  145.14.144.173  10.10.10.199        86  HTTP/1.1 200 OK  (text/html)
```

```
▶ Frame 714: 475 bytes on wire (3800 bits), 475 bytes captured (3800 bits) on interface 0
▶ Ethernet II, Src: Apple_f0:d8:28 (48:bf:6b:f0:d8:28), Dst: Cisco_af:0c:46 (a0:3d:6f:af:0c:46)
▶ Internet Protocol Version 4, Src: 10.10.10.199, Dst: 145.14.144.173
▶ Transmission Control Protocol, Src Port: 51247, Dst Port: 80, Seq: 1, Ack: 1, Len: 409
▼ Hypertext Transfer Protocol
  ▶ GET / HTTP/1.1\r\n
    Host: cncourse.000webhostapp.com\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36\r\n
    DNT: 1\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    \r\n
    [Full request URI: http://cncourse.000webhostapp.com/]
    [HTTP request 1/1]
    [Response in frame: 729]
```

```
0000  a0 3d 6f af 0c 46 48 bf  6b f0 d8 28 08 00 45 00   ·=o··FH· k··(··E·
0010  01 cd 14 e6 40 00 40 06  ed b8 0a 0a 0a c7 91 0e   ····@·@· ········
0020  90 ad c8 2f 00 50 70 10  3f 52 8e 04 88 bb 80 18   ···/·Pp· ?R······
```

```
wireshark_en0_20180911125101_ssqyWx.pcapng      Packets: 1028 · Displayed: 4 (0.4%)      Profile: Default
```

Figure 1: WiresharkScreenshot

1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?
2. What languages (if any) does your browser indicate that it can accept to the server?
3. What is the IP address of your computer? Of the *cncourse* web server?
4. What is the status code returned from the server to your browser?
5. When was the HTML file that you are retrieving last modified at the server?
6. How many bytes of content are being returned to your browser?
7. 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.
8. Click the following command on Wireshark menu: Statistics → http → Requests. Find out how many requests were sent by your browser and how many did the server respond, with ?
9. Find out how to obtain the http traffic flow graph showing the packet exchanges between the client and the server and take a dump of the flow graph for http packets and paste in your answer sheet after all the questions above are answered.
10. Now, enter the following URL to your browser, and answer the questions 5,6, 8 and 9 for the following file : `http://cncourse.000webhostapp.com/simple2.html`
11. Now, enter the following URL to your browser, and answer the questions 5,6, 8 and 9 for the following file : `http://cncourse.000webhostapp.com/simple3.html`
12. Now, enter the following URL to your browser, and answer the questions 5,6, 8 and 9 for the following file : `http://cncourse.000webhostapp.com/simple4.html`. Can you tell whether your browser downloaded the ten images serially, or whether they were downloaded from the two web sites in parallel? Explain.

13. Now, enter the following URL to your browser: `https://www.iitsystem.ac.in/sites/default/files/councilminutes/minutes/51/Revised_Minutes_51st_Meeting_of_the_IITCouncil_28.4.2017.PDF` Find the time required to access this file.

14. Note the time required to access the files viz. simple1.html, simple2.html, simple3.html also and prepare a table that shows the time required to access each of the files simple1.html, simple2.html, simple3.html, the pdf files in Q13 and the html file about counties in US, in question in section 3 that follows.

    Note that for each of the file accesses in Q10, Q11, Q12, Q13 you must restart the wireshark everytime, clear the browser cache and then access the page to prevent the previous traces from coming in your output.

SET 2: The HTTP CONDITIONAL GET/response interaction :

1. 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?

2. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

3. 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?

4. 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.

SET 3: Retrieving Long Documents :

In our examples thus far, the documents retrieved have been simple and short HTML files. Lets next see what happens when we download a long file. Do the following:

- Start up your web browser, and make sure your browsers cache is cleared, as discussed above.

- Start up the Wireshark packet sniffer

- Enter the following URL into your browser `https://en.wikipedia.org/wiki/List_of_United_States_counties_and_county_equivalents`. Your browser should display a rather long file. Stop Wireshark packet capture, and enter http in the display-filter-specification window, so that only captured HTTP messages will be displayed.

  In the packet-listing window, you should see your HTTP GET message, followed by a multiple-packet TCP response to your HTTP GET request. This multiple-packet response deserves a bit of explanation. Recall from Section 2.2 (see Figure 2.9 in the text) that the HTTP response message consists of a status line, followed by header lines, followed by a blank line, followed by the entity body. In the case of our HTTP GET, the entity body in the response is the entire requested HTML file. In our case here, the HTML file is rather long, and is too large to fit in one TCP packet. The single HTTP response message is thus broken into several pieces by TCP, with each piece being contained within a separate TCP segment (see Figure 1.24 in the text). In recent versions of Wireshark, Wireshark indicates each TCP segment as a separate packet, and the fact that the single HTTP response was fragmented across multiple TCP packets is indicated by the TCP segment of a reassembled PDU in the Info column of the Wireshark display. Earlier versions of Wireshark used the Continuation phrase to indicated that the entire content of an HTTP message was broken across multiple TCP segments.. We stress here that there is no Continuation message in HTTP!

- Now enter the following URL into your browser `https://www.thehindu.com/news/national/article24880700.ece/binary/Sec377judgment.pdf` Stop Wireshark packet capture, and enter http in the display-filter-specification window, so that only captured HTTP messages will be displayed.

- Answer the following questions for both the .html file as well as the .pdf file.

1. How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message ?
2. Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?
3. What is the status code and phrase in the response?
4. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the file ?

SET 4 - HTML Documents with CGI Script :

Now that weve seen how Wireshark displays the captured packet traffic for large HTML files, we can look at what happens when your browser downloads a file with a CGI script.

Do the following:

- Start up your web browser, and make sure your browsers cache is cleared, as discussed above.
- Start up the Wireshark packet sniffer
- Enter the following URL into your browser `http://http://cncourse.000webhostapp.com/simple5.php`. Fill in some data and submit the form. Stop Wireshark packet capture, and enter http in the display-filter-specification window, so that only captured HTTP messages will be displayed.

1. What is the method used in your HTTP message ? How many HTTP request messages did your browser send? To which Internet addresses were these requests sent?

SET 5 - HTTP Authentication :

Finally, lets try visiting a web site that is password-protected and examine the sequence of HTTP message exchanged for such a site. Try the URL http://mail.svnit.ac.in. Give the login name dcja@coed.svnit.ac.in. Give the password: dcj123. See that you are able to login successfully. If not, try at some other time, because its likely that one of you has already logged in and you are not allowed to login. PLEASE DO NOT CHANGE THE PASSWORD. Now, do the following:

- Make sure your browsers cache is cleared, as discussed above, and close down your browser. Then, start up your browser
- Start up the Wireshark packet sniffer
- Enter the same URL as above into your browser, typein the same login name and password. 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.

1. What is the servers response (status code and phrase) in response to the initial HTTP GET message from your browser?
2. When your browsers sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?
3. Highlight the content of that packet in Wireshark, where the password field is actually displayed in clear. It must be the same password as you entered while logging in.

*****
***
*