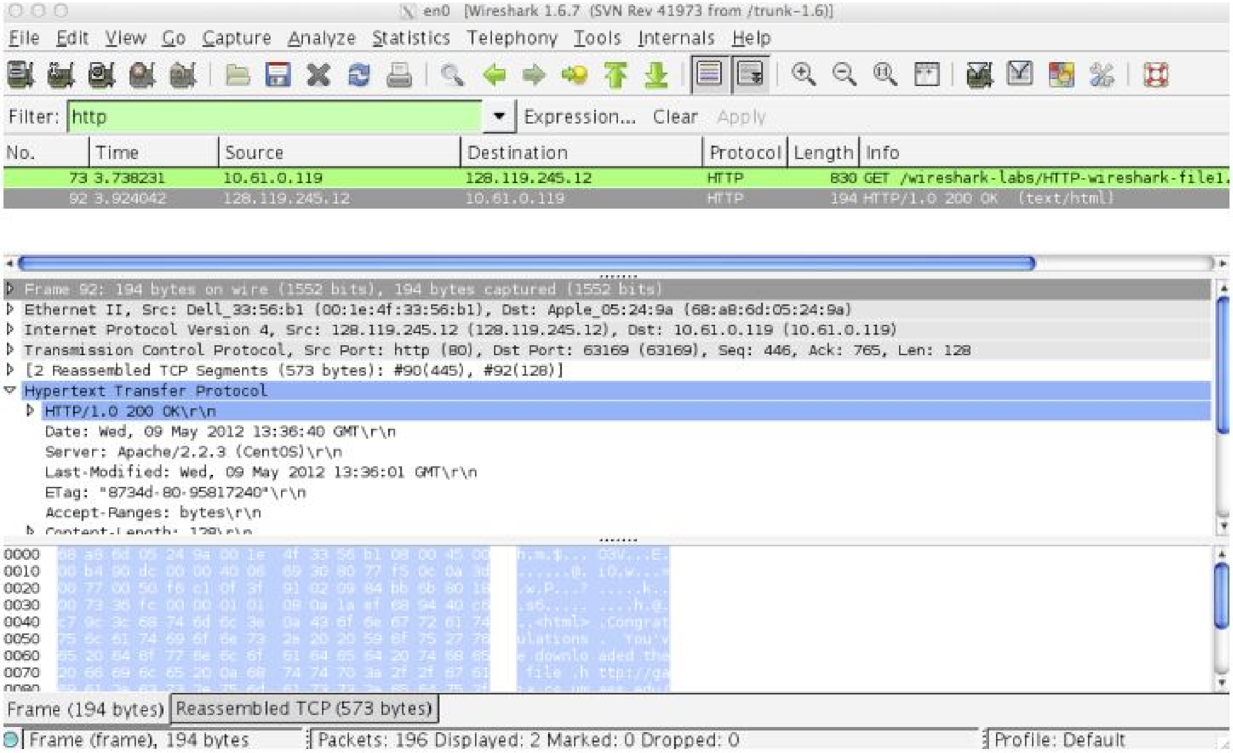
Computer Networks (CS212)
Assignment 5
Assignment given on: 15-02-2021
Due Date: 05-03-2021, 11.59 pm on Moodle

Q1. The Basic HTTP GET/response interaction

Let's 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 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 (we'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.

Your Wireshark window should look similar to the window shown in Figure 1.


en0 [Wireshark 1.6.7 (SVN Rev 41973 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply
Frame (194 bytes)
Reassembled TCP (573 bytes)
Profile: Default

Filter: http

```
No.   Time       Source          Destination     Protocol Length Info
   73 3.738231   10.61.0.119     128.119.245.12  HTTP     830 GET /wireshark-labs/HTTP-wireshark-file1.
   92 3.924042   128.119.245.12  10.61.0.119     HTTP     194 HTTP/1.0 200 OK  (text/html)
```

```
▷ Frame 92: 194 bytes on wire (1552 bits), 194 bytes captured (1552 bits)
▷ Ethernet II, Src: Dell_33:56:b1 (00:1e:4f:33:56:b1), Dst: Apple_05:24:9a (68:a8:6d:05:24:9a)
▷ Internet Protocol Version 4, Src: 128.119.245.12 (128.119.245.12), Dst: 10.61.0.119 (10.61.0.119)
▷ Transmission Control Protocol, Src Port: http (80), Dst Port: 63169 (63169), Seq: 446, Ack: 765, Len: 128
▷ [2 Reassembled TCP Segments (573 bytes): #90(445), #92(128)]
▽ Hypertext Transfer Protocol
  ▷ HTTP/1.0 200 OK\r\n
    Date: Wed, 09 May 2012 13:36:40 GMT\r\n
    Server: Apache/2.2.3 (CentOS)\r\n
    Last-Modified: Wed, 09 May 2012 13:36:01 GMT\r\n
    ETag: "8734d-80-95817240"\r\n
    Accept-Ranges: bytes\r\n
  ▷ Content-Length: 128\r\n
```

```
0000  68 a8 6d 05 24 9a 00 1e  4f 33 56 b1 08 00 45 00   h.m.$... O3V...E.
0010  00 b4 90 dc 00 00 40 06  69 30 80 77 f5 0c 0a 3d   ......@. i0.w...=
0020  00 77 00 50 f6 c1 0f 3f  91 02 09 84 bb 6b 80 18   .w.P...? .....k..
0030  00 73 36 fc 00 00 01 01  08 0a 1a ef 68 94 40 cc   .s6..... ....h.@.
0040  c7 9c 3c 68 74 6d 6c 3e  0a 43 6f 6e 67 72 61 74   ..<html> .Congrat
0050  75 6c 61 74 69 6f 6e 73  2e 20 20 59 6f 75 27 76   ulations .  You'v
0060  65 20 64 6f 77 6e 6c 6f  61 64 65 64 20 74 68 65   e downlo aded the
0070  20 66 69 6c 65 20 0a 68  74 74 70 3a 2f 2f 67 61    file .h ttp://ga
```

Frame (frame), 194 bytes   Packets: 196 Displayed: 2 Marked: 0 Dropped: 0

Figure 1: Wireshark Display after http://gaia.cs.umass.edu/wireshark-labs/ HTTP-wireshark-file1.html has been retrieved by your browser

The example in Figure 1 shows the packet-listing window that two HTTP messages were captured: the GET message (from your browser to the gaia.cs.umass.edu web server) and the response message from the server to your browser. The packet-contents window shows details of the selected message.By looking at the information in the HTTP GET and response messages, answer the following questions.

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 gaia.cs.umass.edu 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.

In your answer to question 5 above, you might have been surprised to find that the document you just retrieved was last modified within a minute before you downloaded the document. That's because (for this particular file), the gaia.cs.umass.edu server is setting the file's last-modified time to be the current time, and is doing so once per minute. Thus, if you wait a minute between accesses, the file will appear to have been recently modified, and hence your browser will download a "new" copy of the document.

Q2. 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 Recent History and check the Cache box, 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.

Answer the following questions:

8. 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?

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

10. 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?

11. 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.

Q3. Retrieving Long Documents

Let's next see what happens when we download a long HTML file.
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-file3.html
  Your browser should display the rather lengthy US Bill of Rights.
- 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. 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 at 4500 bytes 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 . 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.
Answer the following questions:

12. How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill or Rights?

13. Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?

14. What is the status code and phrase in the response?

15. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?

Q4. HTML Documents with Embedded Objects

Now that we've 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 embedded objects, i.e., a file that includes other objects (in the example below, image files) that are stored on another server(s).

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-file4.html
  Your browser should display a short HTML file with two images. These two images are referenced in the base HTML file. That is, the images themselves are not contained in the HTML; instead the URLs for the images are contained in the downloaded HTML file. Your browser will have to retrieve these logos from the indicated web sites. The publisher's logo is retrieved from the gaia.cs.umass.edu web site. The image of the cover for our 5th edition (one of our favorite covers) is stored at the caite.cs.umass.edu server. (These are two different web servers inside cs.umass.edu).
- 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:

16. How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?

17. Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two websites in parallel? Explain.

Q5. HTTP Authentication

Finally, let's try visiting a web site that is password-protected and examine the sequence of HTTP messages exchanged for such a site. The URL http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html is password protected. The username is **"wireshark-students"** (without the quotes), and the password is **"network"** (again, without the quotes). So let's access this "secure" password-protected site. Do the following:

- Make sure your browser's cache is cleared, as discussed above, and close down your browser. Then, start up your browser .
- Start up the Wireshark packet sniffer.
- Enter the following URL into your browser http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wiresharkfile5.html
  Type the requested username and password into the pop up box.

- 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.

Answer the following questions:
18. What is the server’s response (status code and phrase) in response to the initial HTTP GET message from your browser?
19. When your browser sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

The username (wireshark-students) and password (network) that you entered are encoded in the string of characters (d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=) following the “Authorization: Basic” header in the client’s HTTP GET message. While it may appear that your username and password are encrypted, they are simply encoded in a format known as Base64 format. The username and password are not encrypted! To see this, go to http://www.motobit.com/util/base64-decoder-encoder.asp and enter the base64-encoded string d2lyZXNoYXJrLXN0dWRlbnRz and decode. Voila! You have translated from Base64 encoding to ASCII encoding, and thus should see your username! To view the password, enter the remainder of the string Om5ldHdvcms= and press decode. Since anyone can download a tool like Wireshark and sniff packets (not just their own) passing by their network adaptor, and anyone can translate from Base64 to ASCII (you just did it!), it should be clear to you that simple passwords on WWW sites are not secure unless additional measures are taken.

Submission Format

1. Mention your roll number along with your machine ip address at the beginning of the report in bold.Otherwise marks will be reduced.
2. Solve the questions individually. Submit a soft copy of the report, preferably PDF, on all these experiments. If report size exceeds more than 25MB, reduce the image size. (online website: http://www.simpleimageresizer.com/).
3. Include two clear screenshots (HTTP GET and response) for every question (Q1 to Q5) with a packet listing window displaying all the fields of the packet.
4. Include the wireshark trace files for each of the following features after performing the given experiment - **The Basic HTTP GET/response interaction,The HTTP CONDITIONAL GET/response interaction,Retrieving Long Documents, HTML Documents with Embedded Objects,HTTP Authentication.**
5. Finally make a .zip file and upload on moodle.

Grading

Q1- 30% (1 - 4 mark, 2 - 5 marks , 3 - 5 marks , 4 - 5 marks , 5 - 3 marks , 6 - 3 marks , 7 - 5 marks)
Q2 - 20% (8 - 5 marks , 9 - 5 marks , 10 - 5 marks , 11 - 5 marks)
Q3 - 20% (12 - 5 marks , 13 - 5 marks , 14 - 3 marks , 15 - 7 marks)
Q4 - 20% (16 - 10 marks , 17 - 10 marks)
Q5 - 10% (18 - 5 marks , 19 - 5 marks)

Note:
a. 10% penalty for 1 day late submission, 25% penalty for more than 1 day.
b. Found copied, plagiarism fails, 0 marks

Submission Details:
1. Please read the questions carefully and complete it.
2. The zip file name should start with <Your_Roll_Number> and submit it on moodle.
3. Submit only through moodle and well in advance. Any hiccups in the moodle/Internet at the last minute is never acceptable as an excuse for late submission. Submissions through email will be ignored.
4. Submit to moodle on or before 05-03-2021 (23:55 PM)
5. Any queries related to the assignment send an email to Tas well in advance(chandrashekar.s@ittdh.ac.in, priyash@iitdh.ac.in)

Doubts

- Roll number 18001001 to 180010026 can send the doubts to email id - **priyash@iitdh.ac.in**

- Roll number 180010027 to remaining students can send the doubts to email id - **chandrashekar.s@iitdh.ac.in**