FAST NUCES ,LAHORE

HAFIZ AHMAD

19L-1316

EE-5A2

DATA COMMUNICATIONS AND NETWORKS

LAB#9

Wireshark Lab- Analysis of Hyper Text Transfer Protocol (HTTP)

**INTRODUCTION:**

In this lab, we’ll 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.

**OBJECTIVE:**

Understand the Hyper Text Transfer Protocol using Wireshark

**PROCEDURE:**

**1. 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:

1. Start up your web browser.

2. Start up the WireShark packet sniffer, as described in the introductory lab (but don’t yet begin

packet capture).

3. Wait a bit more than one minute (we’ll see why shortly), and then begin Wireshark packet capture.

4. 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.

5. Stop Wireshark 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.

answers 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?

4. What is the status code returned from the server to your browser?

5. When was the HTML file that 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.

**2. 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 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.

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. Close the browser and open the link again. Capture this packet. 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?

**3. Retrieving Long Documents**

In our examples thus far, the documents retrieved have been simple and short HTML files. 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 response to your HTTP GET request. This multiple-packet response deserves a bit of explanation. 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 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 (see Figure 1.22 in the text). Each TCP segment is recorded as a separate packet by Wireshark, and the fact that the single HTTP response was fragmented across multiple TCP packets is indicated by the “Continuation” phrase displayed by Wireshark. We stress here that there is no “Continuation” message in HTTP!

12. How many HTTP GET request messages were sent by your browser?

13. How many data-containing TCP segments were needed to carry the single HTTP response?

**4. 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. As discussed in the textbook, your browser will have to retrieve these logos from the indicated web sites. Our publisher’s logo is retrieved from the [www.awl.com](http://www.awl.com/) web site. The image of our book’s cover is stored at the manic.cs.umass.edu server.

• Stop Wireshark packet capture, and enter “http” in the display-filter-specification window, so that

only captured HTTP messages will be displayed.

14. How many HTTP GET request messages were sent by your browser? To which Internet addresses

were these GET requests sent?

15. Can you tell whether your browser downloaded the two images serially, or whether they were

downloaded from the two web sites in parallel? Explain.

**ANSWERS ARE BELOW OF ALL QUESTIONS IN 4 DIFFERENTS TASKS OF LAB:**



Version 1.1

Text/html

Status code 404



Date: Fri, 19 Nov 2021 09:47:56 GMT\r\n

Byte size 

7:

No header

Task 2:

8:



9



10:



Task 3;:

12:

3



13:



Task 4:

14:

5 gets





15:

They were downloaded serially.

**ISSUES:**

No issues regarding this lab.

**APPLICATIONS:**

Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Wireshark reassembles all of the actual data packets containing a particular webpage and displays it within the packet labeled as the HTTP response. ... It is very similar to that of an HTTP request, except that it substitutes an HTTP response message for the URL and uses a different collection of headers.SSL/TLS is especially suited for HTTP, since it can provide some protection even if only one side of the communication is authenticated. This is the case with HTTP transactions over the Internet, where typically only the server is authenticated (by the client examining the server's certificate).

**CONCLUSION:**

In ths lab we learn Hyper Text Transfer Protocol using Wireshark .The Hyper Text Transport Protocol is a text-based request-response client-server protocol. The first protocol is a security protocol like SSL, TLS or PCT. ... The second protocol, which runs on top of this security protocol, is HTTP.

**POST LAB:**

**HTTP Authentication**

Let’s try visiting a web site that is password-protected and examine the sequence of HTTP message

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). Open this URL in your browser and capture packets.

What is the server’s response (status code and phrase) in response to the initial HTTP GET

message from your browser?

What is the server's response (status code and phrase) in response to the initial HTTP GET message from your browser? According to the diagram 9, the initial HTTP GET message should be packet 6 and the packet 9 is the response to the packet 6. Thus the server's response is 401 Authorization Required.



When your browser’s sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

When your browser's sends the HTTP GET message for the second time, what new field is included in the HTTP GET message? Answer: As seen in the screenshot the new field (highlighted) is Authorization.



Analyze the second HTTP GET packet and check if you can see the username and password?

This page is pasward protected.This header is called “If-Modified-Since”, i.e., it asks the server to send the content if it has been modified since a given time (see the header below).Wireshark reassembles all of the actual data packets containing a particular webpage and displays it within the packet labeled as the HTTP response. ... The actual data being carried by the HTTP protocol (the requested web page) is encapsulated within the data section of the HTTP packet.How many HTTP GET request messages did your browser send? Answer: there were three HTTP GET messages sent: packet 10 in the trace (to get the base file), packet 17 (to get the Pearson logo) and packet 20 (to get the 5th edition textbook cover).

Is your password on this site secured? If not, explain briefly what additional measures can be taken to secure a website. ?

Not secure on this site.Check the URL of your website and see if it says “HTTPS” at the start of the address (instead of “HTTP”). This means the website is secure with an SSL certificate. The SSL certificate is used to secure all data that is passed from the browser to the website's server.To ensure that the password for an important website is kept secure, these are few of the steps: ... To protect your website, passwords should be complex, containing uppercase letters, lowercase letters, numerals, and special characters. Passwords should be at least 10 characters long.

8 surefire ways to protect your online passwords

Don't pick a weak password. ...

Use multifactor authentication. ...

If biometrics is an option, take it. ...

Different accounts need different passwords. ...

Consider a password manager. ...

Don't share your password. ...

Don't fall for phishing. ...

Always update software.