

# LAB 2

## COURSE: COMPUTER NETWORK

---

### ANALYZING HTTP

**Student name:** Đặng Ngọc Thái Sơn, Hoàng Ân Thiên

**Student ID:** ITCSIU23033, ITCSIU23035

#### **Objective**

- Students can analyze the HTTP communication.

#### **Requirements**

- Students use the .pcap file from Lab 1.

## **1 Analyzing basic HTTP GET/response interaction**

Open the .pcap from lab 1, type “http” into display-filter window to get the HTTP message and answer the following questions:

1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

137	12.685387	192.168.137.19	192.168.137.1	HTTP	520	GET /ITCSIU23033.html	HTTP/1.1
-----	-----------	----------------	---------------	------	-----	-----------------------	----------

**The server is running HTTP version 1.1.**

2. What is the IP address of the server? Of the client?

- Sever IP: **192.168.137.1**
- Client IP: **192.168.137.19**

137	12.685387	192.168.137.19	192.168.137.1	HTTP	520	GET /ITCSIU23033.html	HTTP/1.1
146	13.124609	192.168.137.1	192.168.137.19	HTTP	503	HTTP/1.1	200 OK (text/html)

3. List the status codes and phrase returned from the server to your browser?

146	13.124609	192.168.137.1	192.168.137.19	HTTP	503	HTTP/1.1	200 OK (text/html)
149	13.254923	192.168.137.1	192.168.137.19	HTTP	1437	HTTP/1.1	404 Not Found (text/html)
155	16.882976	192.168.137.1	192.168.137.19	HTTP	196	HTTP/1.1	304 Not Modified

4. How many bytes of content are being returned to your browser?

```
146 13.124609 192.168.137.1 192.168.137.19 HTTP 503 HTTP/1.1 200 OK (text/html)
```

**503 bytes of content are being returned.**

5. 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?

```
Hypertext Transfer Protocol
  ▶ GET /ITCSIU23033.html HTTP/1.1\r\n
    Host: 192.168.137.1\r\n
    Connection: keep-alive\r\n
    DNT: 1\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36 Edg/123.0.0.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
  \r\n
  [Full request URL: http://192.168.137.1/ITCSIU23033.html]
  [HTTP request 1/7]
  [Response in frame: 146]
  [Next request in frame: 148]
```

No.

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

```
▼ Line-based text data: text/html (12 lines)
  <!DOCTYPE html>\r\n
  <html>\r\n
  <head>\r\n
  <title>Computer Network lab - 2</title>\r\n
  </head>\r\n
  <body>\r\n
  <h1>StudentID 1: FullName</h1><br>\r\n
  <h2>Class: </h2><br>\r\n
  <h1>StudentID 2: FullName</h1><br>\r\n
  <h2>Class: </h2><br>\r\n
  </body>\r\n
  </html>\r\n
```

Yes. The sever explicitly return the contents of the file.

7. 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?

Yes.

```
▼ Hypertext Transfer Protocol
  ▶ GET /ITCSIU23033.html HTTP/1.1\r\n
    Host: 192.168.137.1\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
    DNT: 1\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36 Edg/123.0.0.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    If-None-Match: "40f963cc87cdaf1:0"\r\n
    If-Modified-Since: Sat, 23 Mar 2024 02:16:00 GMT\r\n
  \r\n
  [Full request URI: http://192.168.137.1/ITCSIU23033.html]
  [HTTP request 3/7]
  [Prev request in frame: 148]
  [Response in frame: 155]
  [Next request in frame: 167]
```

8. 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.

**The code return is 304. The server does not return the contents of the file.**

```
▼ Hypertext Transfer Protocol
  ▶ GET /ITCSIU23033.html HTTP/1.1\r\n
    Host: 192.168.137.1\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
    DNT: 1\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36 Edg/123.0.0.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    If-None-Match: "40f963cc87cdaf1:0"\r\n
    If-Modified-Since: Sat, 23 Mar 2024 02:16:00 GMT\r\n
  \r\n
  [Full request URI: http://192.168.137.1/ITCSIU23033.html]
  [HTTP request 3/7]
  [Prev request in frame: 148]
  [Response in frame: 155]
  [Next request in frame: 167]
```

**Explain:** Code 304 means that the requested resource has not been modified since the last time being accessed. The code informs the client that there is no need to transmit the resource again, as the previous cache is still valid.

## 2. 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.

- (*Note:* If you are unable to run Wireshark on a live network connection, you can use the http-ethereal-trace-3 packet trace to answer the questions below; see footnote 1. This trace file was gathered while performing the steps above on one of the author’s computers.)

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 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 1.24 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 indicate 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!

Answer the following questions:

9. 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?

1545	21.814569	10.238.33.98	128.119.245.12	HTTP	472	GET /favicon.ico	HTTP/1.1
------	-----------	--------------	----------------	------	-----	------------------	----------

- **There are 2 HTTP GET request messages**

1363	21.402915	10.238.33.98	128.119.245.12	HTTP	526	GET /wireshark-labs/HTTP-wireshark-file3.html	HTTP/1.1
------	-----------	--------------	----------------	------	-----	---	----------

- **The packet number in the trace contains the GET message for the Bill or Rights is 1363.**

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

1522	21.681996	128.119.245.12	10.238.33.98	HTTP	535	HTTP/1.1 200 OK	(text/html)
------	-----------	----------------	--------------	------	-----	-----------------	-------------

1569	22.102585	128.119.245.12	10.238.33.98	HTTP	538	HTTP/1.1 404 Not Found	(text/html)
------	-----------	----------------	--------------	------	-----	------------------------	-------------

### 3 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 gaia.cs.umass.edu web site. The image of the cover for our 5<sup>th</sup> 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.
- (*Note:* If you are unable to run Wireshark on a live network connection, you can use the http-ethereal-trace-4 packet trace to answer the questions below; see footnote 1. This trace file was gathered while performing the steps above on one of the author's computers.)

Answer the following questions:

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

**The browser had sent three GET requests to the server.**

630 4.332116 10.238.14.52 hatter.cslash.net HTTP	461 GET /8E_cover_small.jpg HTTP/1.1
587 3.969335 10.238.14.52 gaia.cs.umass.edu HTTP	494 GET /pearson.png HTTP/1.1
543 3.375973 10.238.14.52 gaia.cs.umass.edu HTTP	548 GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1

- **Gaia.cs.umass.edu**
- **Hatter.cslash.net**

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

**Two images were sent serially because the server had response two times, both respectively respond to each GET requests.**

587 3.969335 10.238.14.52 gaia.cs.umass.edu HTTP	494 GET /pearson.png HTTP/1.1
627 4.296930 gaia.cs.umass... 10.238.14.52 HTTP	745 HTTP/1.1 200 OK (PNG)
630 4.332116 10.238.14.52 hatter.cslash.net HTTP	461 GET /8E_cover_small.jpg HTTP/1.1
675 4.592952 hatter.cslash... 10.238.14.52 HTTP	225 HTTP/1.1 301 Moved Permanently

## 4 HTTP Authentication

Finally, 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](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-wireshark-file5.html](http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html)  
Type the requested user name 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:

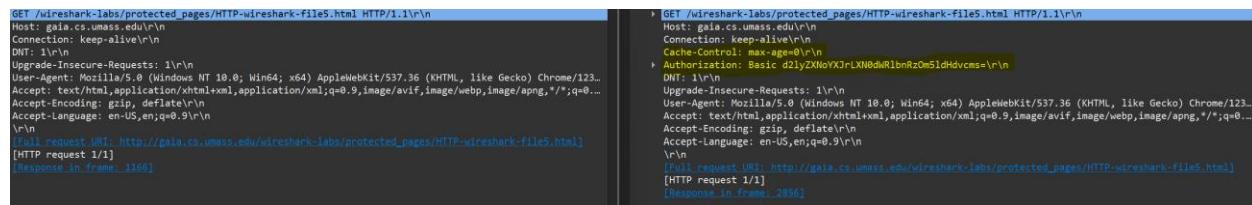
11. What is the server's response (status code and phrase) in response to the initial HTTP GET message from your browser?

- **401 Unauthorized**
- **200 OK**

1166 4.413356 gaia.cs.umass... 10.238.14.52	HTTP	771 HTTP/1.1 401 Unauthorized (text/html)
2856 13.492406 gaia.cs.umass... 10.238.14.52	HTTP	544 HTTP/1.1 200 OK (text/html)

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

### Cache-Control and Authorization



```
GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1\r\nHost: gaia.cs.umass.edu\r\nConnection: keep-alive\r\nDNT: 1\r\n\r\nUpgrade-Insecure-Requests: 1\r\nUser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123...Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8...Accept-Encoding: gzip, deflate\r\nAccept-Language: en-US,en;q=0.9\r\n\r\n[Full request URL: http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html]\n[HTTP request 1/1]\n[Response in frame: 1166]\n\n> [GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1\r\nHost: gaia.cs.umass.edu\r\nConnection: keep-alive\r\nCache-Control: max-age=0\r\n\r\nAuthorization: Basic d2lyZXNoYXJrLXN0dWRlbzOm5ldHdvcms=\r\n\r\nUpgrade-Insecure-Requests: 1\r\nUser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123...Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8...Accept-Encoding: gzip, deflate\r\nAccept-Language: en-US,en;q=0.9\r\n\r\n[Full request URL: http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html]\n[HTTP request 1/1]\n[Response in frame: 2856]
```

The username (wireshark-students) and password (network) that you entered are encoded in the string of characters (d2lyZXNoYXJrLXN0dWRlbzOm5ldHdvcms=) 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 Om5ldHdvcmss= 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.