Lab 2

Team Members:

Alan Tony (191CS207)

Aashish Prateek (191CS225)

Computer Networks Lab (CS302)



**Question 1:**

Code:

*htdocs/index.html*

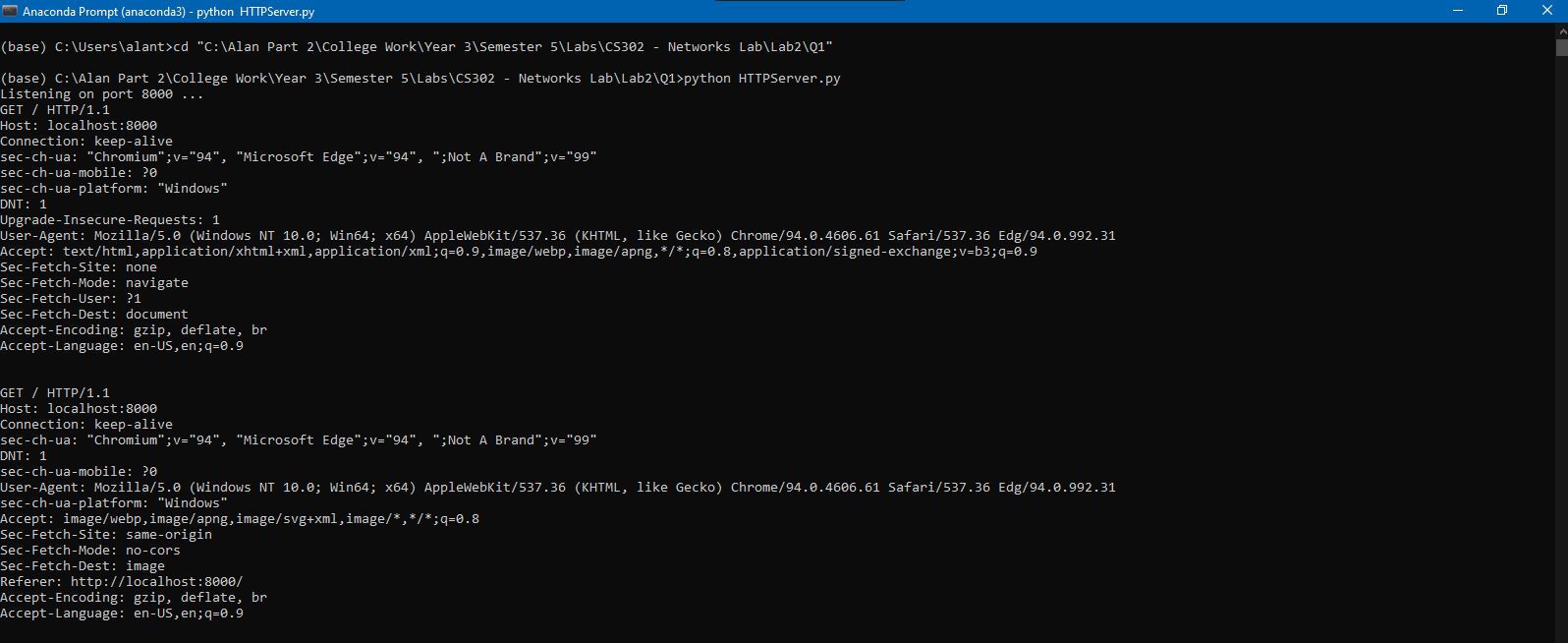
**

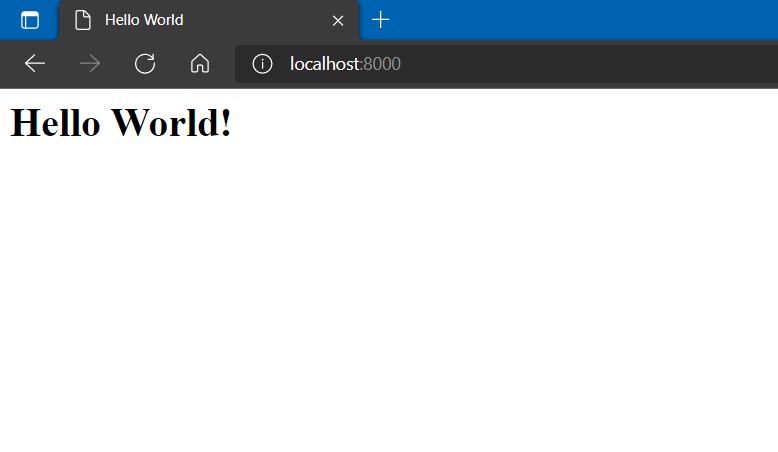
*P.T.O*

*HTTPServer.py*



Output:



****

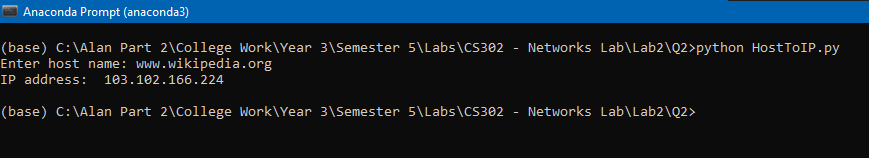
**Question 2:**

Code:

*HostToIP.py*



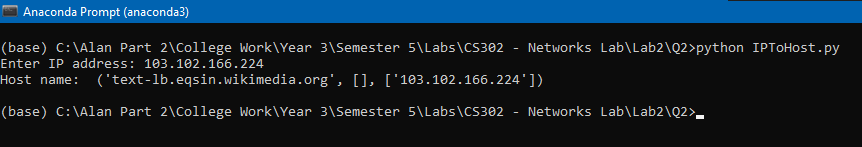
Output:



*IPToHost.py*



Output:

****

*P.T.O*

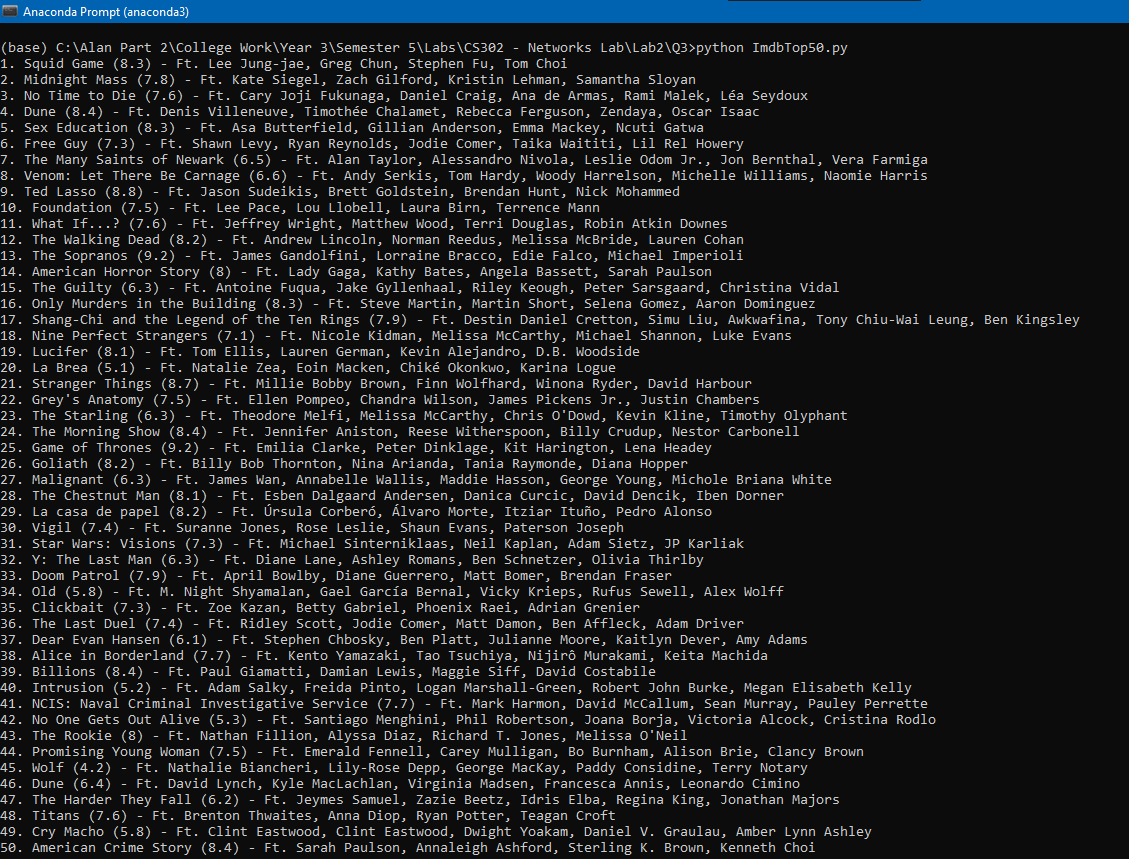
**Question 3:**

Code:

*ImdbTop50.py*

**

Output:

****

*P.T.O*

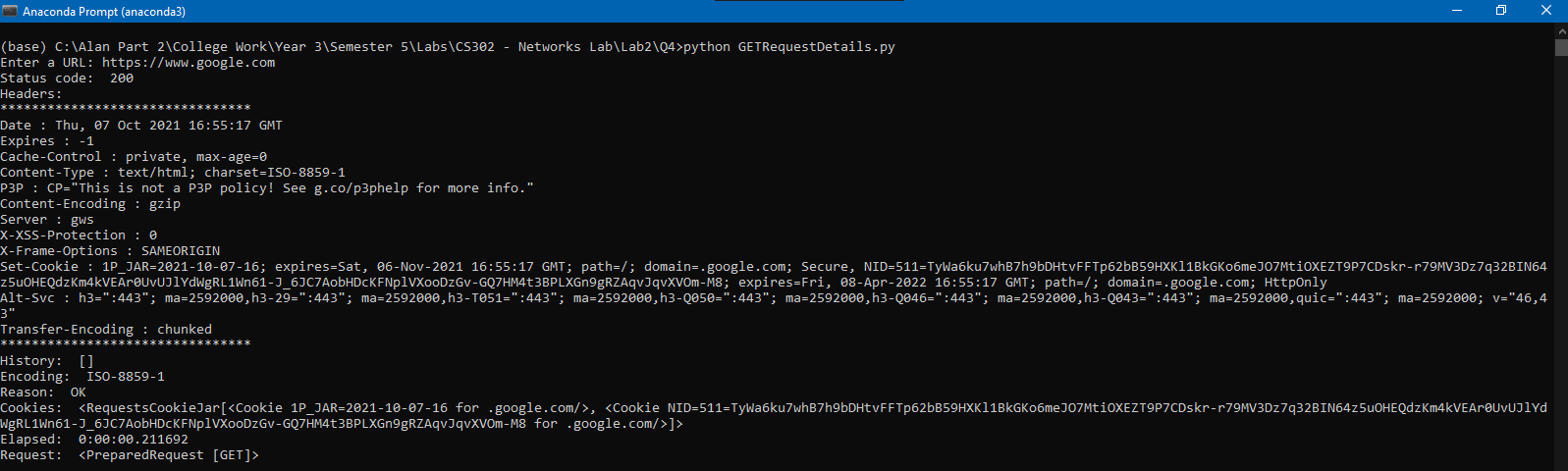
**Question 4:**

Code:

*GETRequestDetails.py*

**

Output:

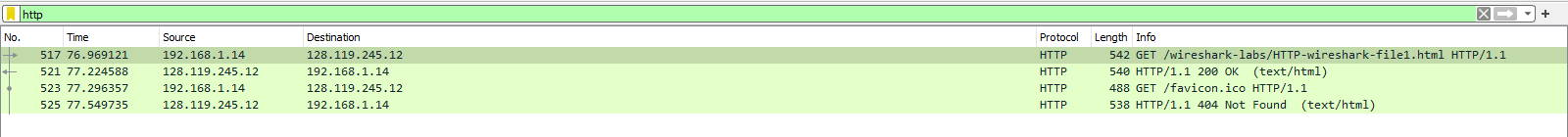


**Question 5:**

**Analyzing HTTP messages:**

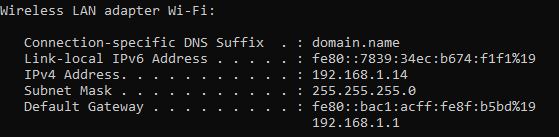
1. Webpage URL: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html

After requesting for the webpage on a private window (where no cookies are stored), this is the output I get:

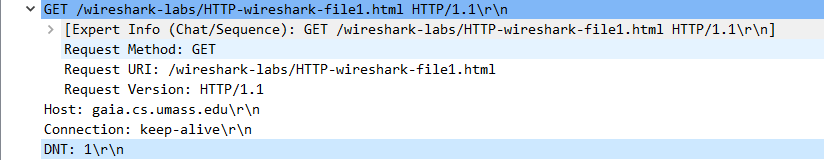


Observations:

* 1st GET request is sent to the server with IP 128.119.245.12 from our local private IP 192.168.1.14. We can confirm our local IP with the **ipconfig** command. Screenshot:



The following was the HTTP header:



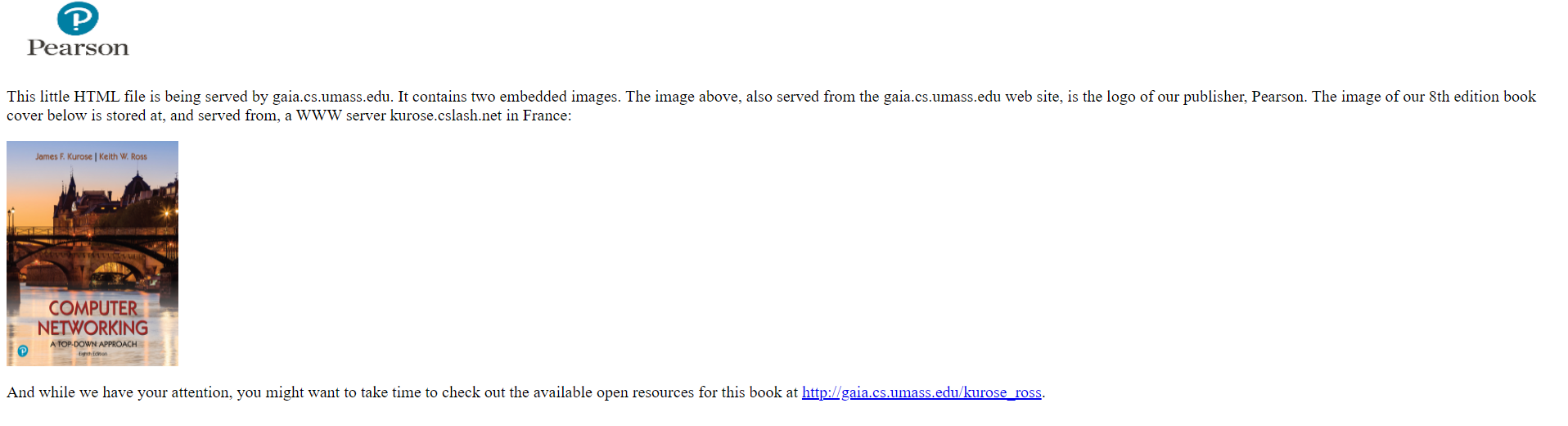
* The response to the request was **200 OK,** which means “request has succeeded.”
* The **connection** header-line is set to **keep alive** to request a persistent connection.
* Upon reloading the page, our browser sent a GET request. But the server returned an **HTTP 304 Not Modified** response because the content on the server has not been modified, and the website contents are cached on our local machine.

The following header-line implements this:

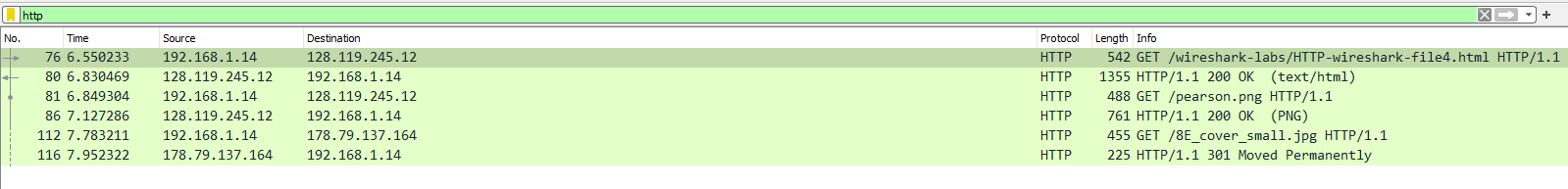
If-Modified-Since: Thu, 07 Oct 2021 05:59:01 GMT\r\n

1. Webpage URL: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html

Screenshot of the webpage:



The HTTP messages:



Exchanges:

1. The first exchange was for the HTML file of the webpage. There are 2 images embedded in the HTML, which were delivered subsequently
2. The 2nd exchange was for a .jpg file from the same host.
3. The 3rd exchange was for a .png image file delivered from another host. So server returns a **301 Moved Permanently** message, which indicates that the requested content is present in another location:

Location: <https://kurose.cslash.net/8E_cover_small.jpg\r\n>

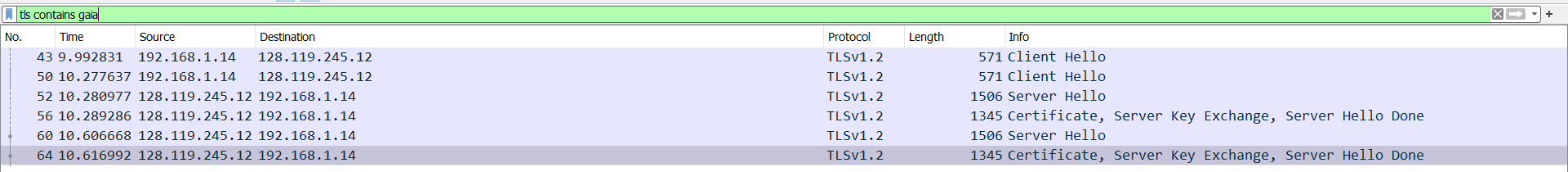
**Analyzing HTTPS messages:**

HTTPS is a more secure form of HTTP. TLS is used to encrypt requests and responses from a host. TLS uses a method called public-key encryption, where the host and the servers exchange public keys. Each host uses the public key to encrypt the outgoing data. Each host uses their private key to decrypt traffic from the other host. TLS is also used to verify the identity of the server. Once the identity is confirmed, the keys are generated and exchanged. This process is called the TLS handshake.

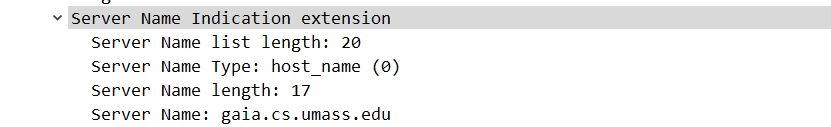
For this reason, Wireshark needs the session keys generated by the TLS handshake to decrypt the HTTP requests and responses made using the HTTP protocol. This is information can be logged by the web browser on setting an environment variable **SSLKEYLOGFILE**. We should also set the path of this file in Wireshark preferences.

Once the TLS handshake is decrypted, additional HTTP and TCP packets are visible, containing the decrypted exchange between the client and the server.

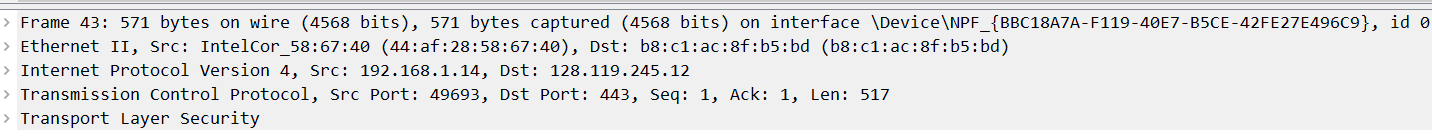
We can now filter the TLS messages with the keyword “gaia” since it is part of the domain name of the server:



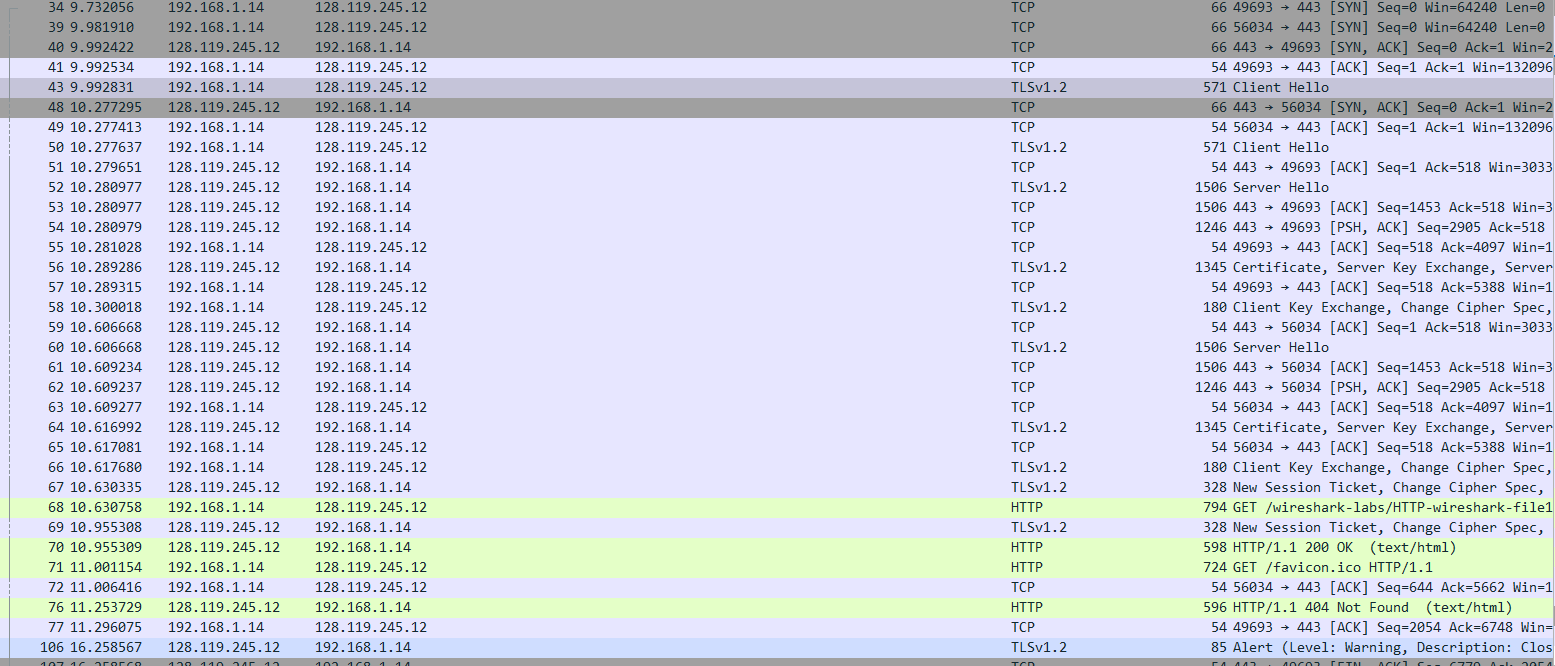
On expanding the details section of the first Client Hello TLS message, we see an extension named “server-name (len=22). This should contain the server’s name:



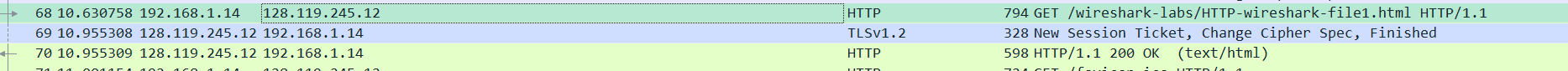
This is Client Hello frame is a part of the TLS handshake. It doesn’t contain the HTTP layer above it:



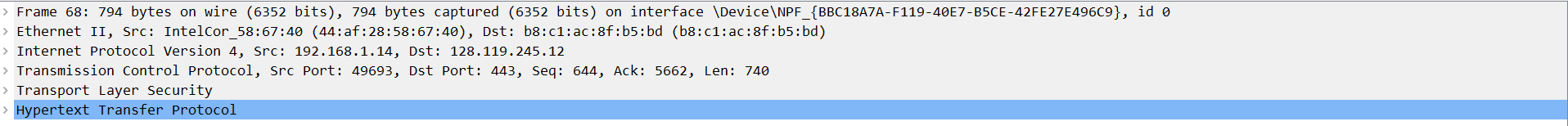
To access the HTTP queries, we can filter out the frames exchanged which involves the public IP of the server (128.119.245.12), using ip.addr == 128.119.245.12. Now we can see all the other frames:



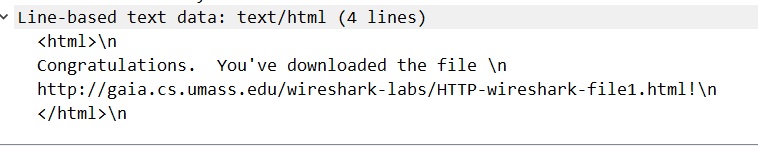
These are our main HTTP GET requests and responses:



These messages contain an extra layer called HTTP, which are now decrypted:

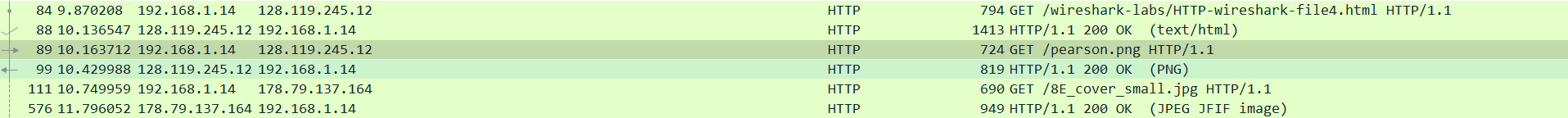


The GET response contains the HTML data which is rendered by our browser:

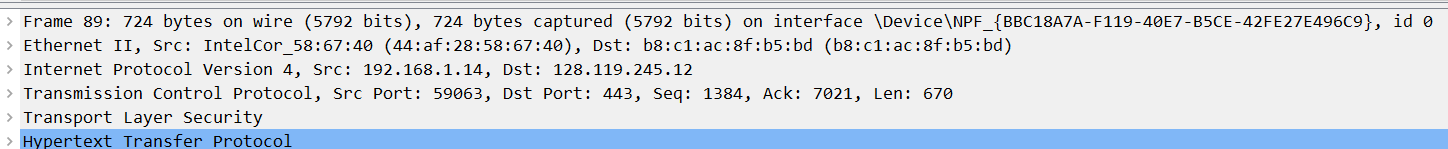


Webpage 2: <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html>

For the second webpage, the HTTP response was pretty much the same, but we got an **HTTP 200 OK** message instead of the **HTTP 304 Permanently Moved** message for the last image:



The difference between the HTTP protocol as experimented in the first section was that there is a TLS protocol layer beneath HTTP:



\*\*\*\*\*