Lab 2

Computer Networks Lab (CS302)

Team Members:

Alan Tony (191CS207)

Aashish Prateek (191CS225)

Question 1:

Code:

htdocs/index.html

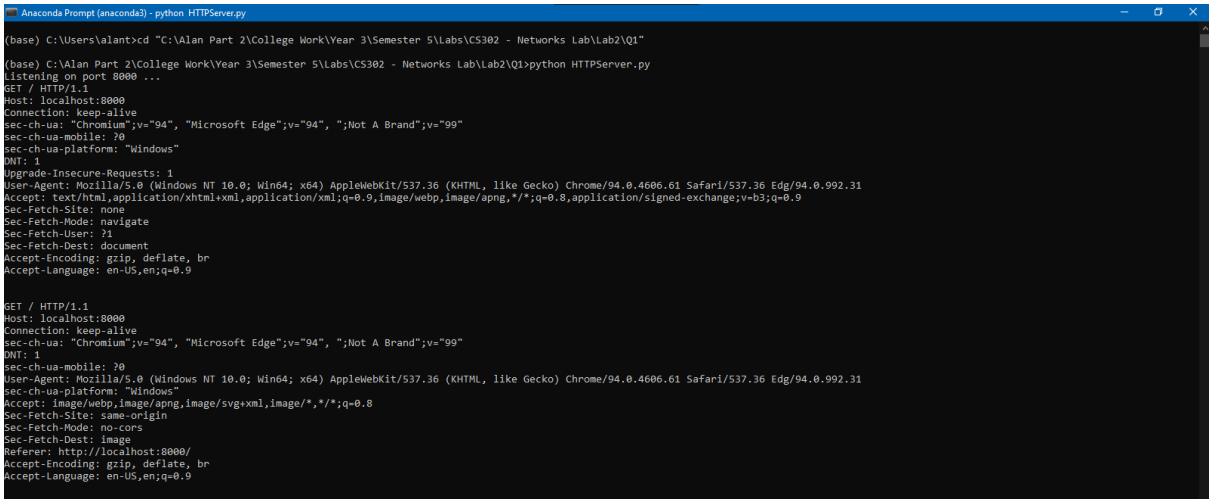
```
<html>
<head>
    <title>Hello World</title>
    <link rel="shortcut icon" href="#" />
</head>
<body>
    <h1>Hello World!</h1>
</body>
</html>
```

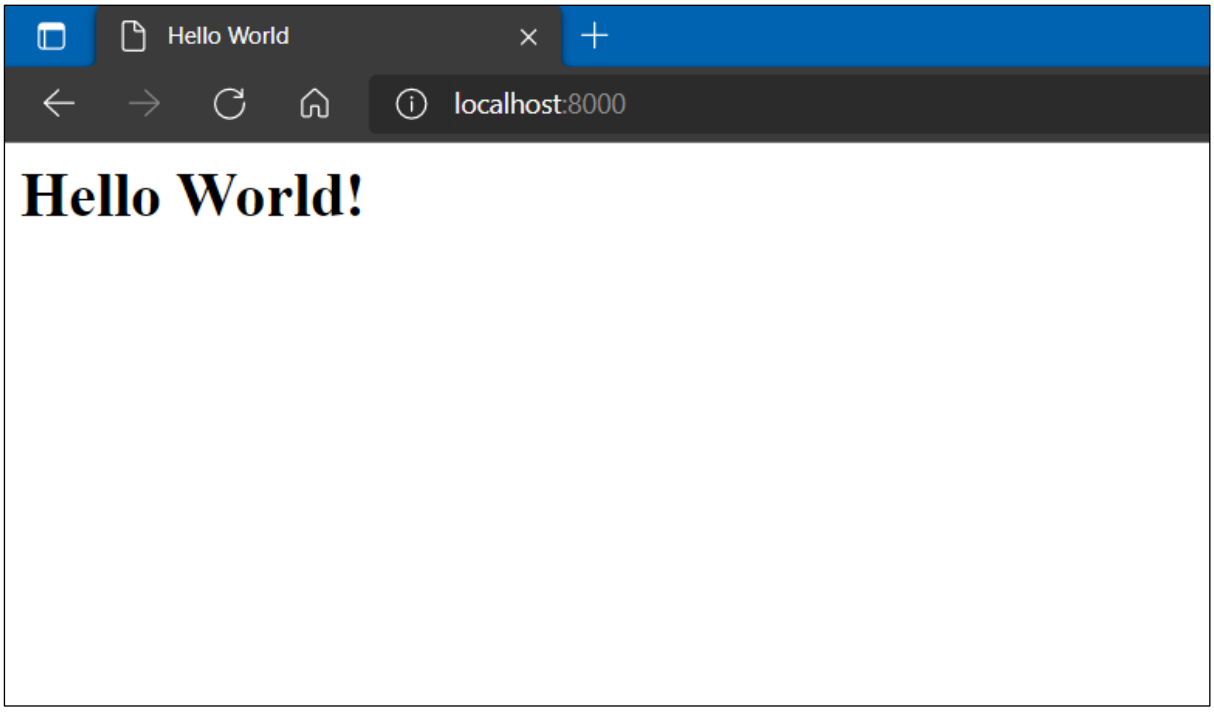
P.T.O

HTTPServer.py

```
import socket


# Define socket host and port
SERVER_HOST = '0.0.0.0'
SERVER_PORT = 8000

# Create socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server_socket.bind((SERVER_HOST, SERVER_PORT))
server_socket.listen(1)
print('Listening on port %s ...' % SERVER_PORT)

while True:
    # Wait for client connections
    client_connection, client_address = server_socket.accept()

    # Get the client request
    request = client_connection.recv(1024).decode()
    print(request)

    try:
        # Get the content of webpage
        file_name = request.split()[1]
        if file_name == "/":
            file_name = "/index.html"
        fin = open('htdocs' + file_name)
        content = fin.read()
        fin.close()
        response = 'HTTP/1.0 200 OK\n\n' + content
    except:
        content= "<h1>Content not found</h1>"
        response = 'HTTP/1.0 404 Not Found\n\n' + content

    # Send HTTP response
    client_connection.sendall(response.encode())
    client_connection.close()
```

Output:





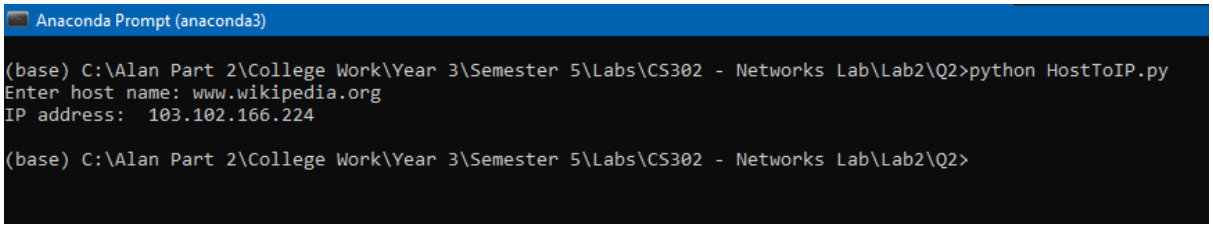
Question 2:

Code:

HostToIP.py

```
import socket

hostname = input("Enter host name: ")
ip = socket.gethostbyname(hostname)
print("IP address: ", ip)
```

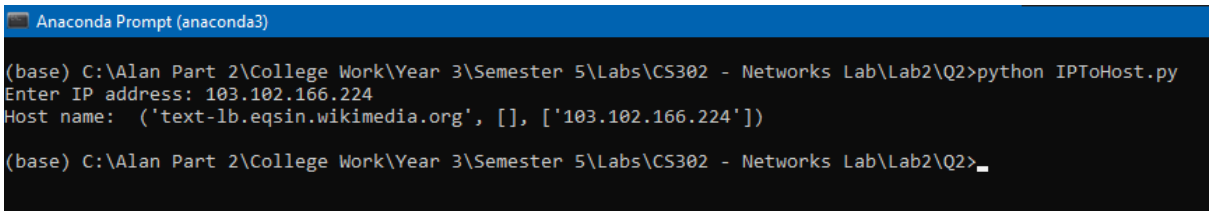
Output:



IPToHost.py

```
import socket

IP = input("Enter IP address: ")
hostname = socket.gethostbyaddr(IP)
print("Host name: ", hostname)
```

Output:



P.T.O

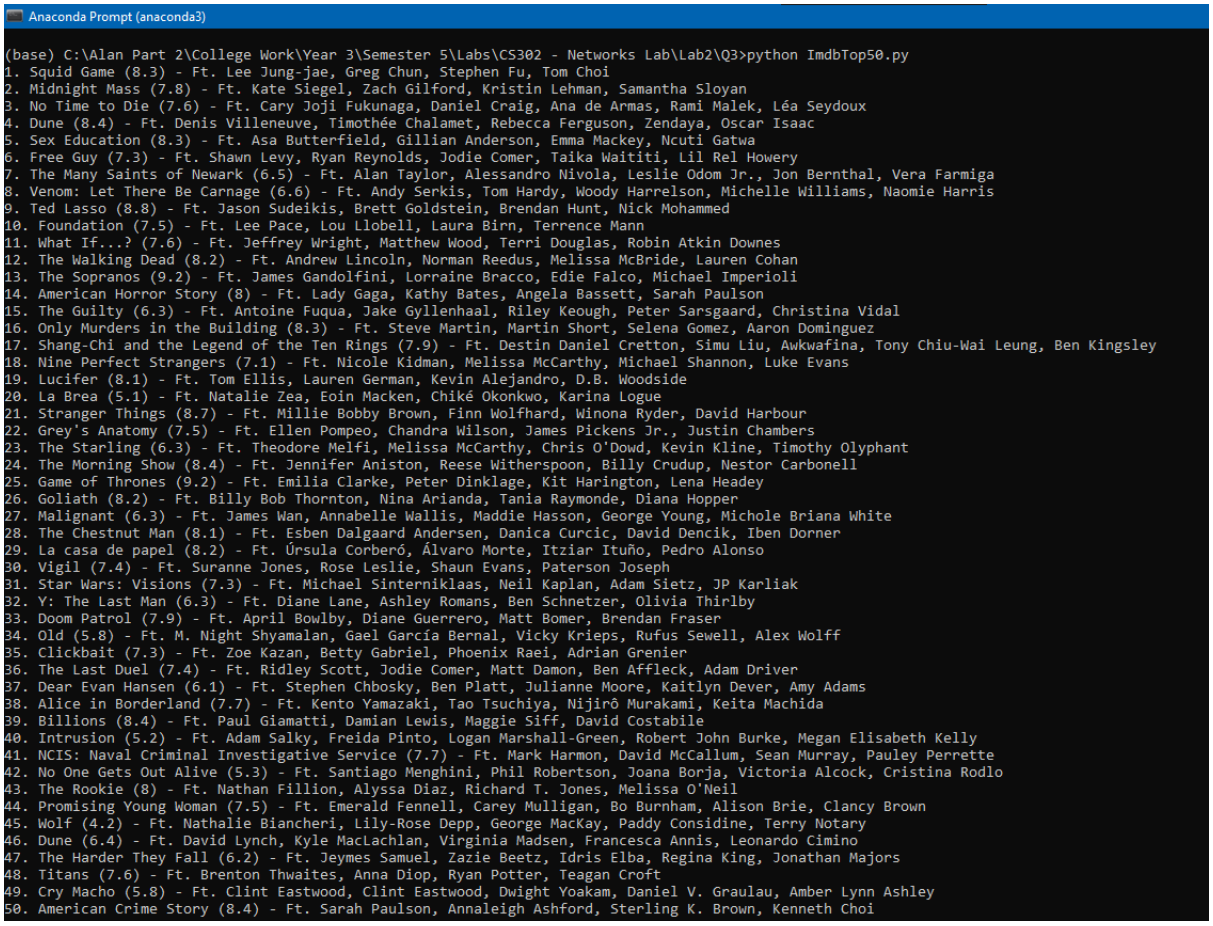
Question 3:

Code:

ImdbTop50.py

```
import requests
import bs4
from datetime import datetime

curr_date = str(datetime.now()).split()[0]

#The release date is a parameter in the HTTP request with the following format
:
#release_date=from_date,till_date
#Commas are required to separate multiple value for a single parameter
#Since I want movies of all time, from_date is left blank, till_date is put as
 current date
URL="http://www.imdb.com/search/title?release_date=," + curr_date

r = requests.get(URL)

soup = bs4.BeautifulSoup(r.content, 'html5lib')

table = soup.find_all('div', attrs = {'class' : 'lister-item-content'})
for i, content in enumerate(table):

    movie_name = content.h3.a.text
    rating = content.find('span', attrs = {'class' : {'rating-
rating'}}).span.text

    actors = content.find('p', attrs = {'class' : ''})
    actors = [name.text for name in actors.find_all('a')]

    print("{}. {} ({}) - Ft. ".format(i+1, movie_name, rating), end='')
    print(*actors, sep=', ')
```

Output:


Anaconda Prompt (anaconda3)
(base) C:\Alan Part 2\College Work\Year 3\Semester 5\Labs\CS302 - Networks Lab\Lab2\Q3>python ImdbTop50.py
1. Squid Game (8.3) - Ft. Lee Jung-jae, Greg Chun, Stephen Fu, Tom Choi
2. Midnight Mass (7.8) - Ft. Kate Siegel, Zach Gilford, Kristin Lehman, Samantha Sloyan
3. No Time to Die (7.6) - Ft. Cary Joji Fukunaga, Daniel Craig, Ana de Armas, Rami Malek, Léa Seydoux
4. Dune (8.4) - Ft. Denis Villeneuve, Timothée Chalamet, Rebecca Ferguson, Zendaya, Oscar Isaac
5. Sex Education (8.3) - Ft. Asa Butterfield, Gillian Anderson, Emma Mackey, Ncuti Gatwa
6. Free Guy (7.3) - Ft. Shawn Levy, Ryan Reynolds, Jodie Comer, Taika Waititi, Lil Rel Howery
7. The Many Saints of Newark (6.5) - Ft. Alan Taylor, Alessandro Nivola, Leslie Odom Jr., Jon Bernthal, Vera Farmiga
8. Venom: Let There Be Carnage (6.6) - Ft. Andy Serkis, Tom Hardy, Woody Harrelson, Michelle Williams, Naomie Harris
9. Ted Lasso (8.8) - Ft. Jason Sudeikis, Brett Goldstein, Brendan Hunt, Nick Mohammed
10. Foundation (7.5) - Ft. Lee Pace, Lou Llobell, Laura Birn, Terrence Mann
11. What If...? (7.6) - Ft. Jeffrey Wright, Matthew Wood, Terri Douglas, Robin Atkin Downes
12. The Walking Dead (8.2) - Ft. Andrew Lincoln, Norman Reedus, Melissa McBride, Lauren Cohan
13. The Sopranos (9.2) - Ft. James Gandolfini, Lorraine Bracco, Edie Falco, Michael Imperioli
14. American Horror Story (8) - Ft. Lady Gaga, Kathy Bates, Angela Bassett, Sarah Paulson
15. The Guilty (6.3) - Ft. Antoine Fuqua, Jake Gyllenhaal, Riley Keough, Peter Sarsgaard, Christina Vidal
16. Only Murders in the Building (8.3) - Ft. Steve Martin, Martin Short, Selena Gomez, Aaron Dominguez
17. Shang-Chi and the Legend of the Ten Rings (7.9) - Ft. Destin Daniel Cretton, Simu Liu, Awkwafina, Tony Chiu-Wai Leung, Ben Kingsley
18. Nine Perfect Strangers (7.1) - Ft. Nicole Kidman, Melissa McCarthy, Michael Shannon, Luke Evans
19. Lucifer (8.1) - Ft. Tom Ellis, Lauren German, Kevin Alejandro, D.B. Woodside
20. La Brea (5.1) - Ft. Natalie Zea, Eoin Macken, Chiké Okonkwo, Karina Logue
21. Stranger Things (8.7) - Ft. Millie Bobby Brown, Finn Wolfhard, Winona Ryder, David Harbour
22. Grey's Anatomy (7.5) - Ft. Ellen Pompeo, Chandra Wilson, James Pickens Jr., Justin Chambers
23. The Starling (6.3) - Ft. Theodore Melfi, Melissa McCarthy, Chris O'Dowd, Kevin Kline, Timothy Olyphant
24. The Morning Show (8.4) - Ft. Jennifer Aniston, Reese Witherspoon, Billy Crudup, Nestor Carbonell
25. Game of Thrones (9.2) - Ft. Emilia Clarke, Peter Dinklage, Kit Harington, Lena Headey
26. Goliath (8.2) - Ft. Billy Bob Thornton, Nina Arianda, Tania Raymonde, Diana Hopper
27. Malignant (6.3) - Ft. James Wan, Annabelle Wallis, Maddie Hasson, George Young, Michole Briana White
28. The Chestnut Man (8.1) - Ft. Esben Dalgaard Andersen, Danica Curcic, David Dencik, Iben Dorner
29. La casa de papel (8.2) - Ft. Úrsula Corberó, Álvaro Morte, Itziar Ituño, Pedro Alonso
30. Vigil (7.4) - Ft. Suranne Jones, Rose Leslie, Shaun Evans, Paterson Joseph
31. Star Wars: Visions (7.3) - Ft. Michael Sinterniklaas, Neil Kaplan, Adam Sietz, JP Karliak
32. Y: The Last Man (6.3) - Ft. Diane Lane, Ashley Romans, Ben Schnetzer, Olivia Thirlby
33. Doom Patrol (7.9) - Ft. April Bowlby, Diane Guerrero, Matt Bomer, Brendan Fraser
34. Old (5.8) - Ft. M. Night Shyamalan, Gael García Bernal, Vicky Krieps, Rufus Sewell, Alex Wolff
35. Clickbait (7.3) - Ft. Zoe Kazan, Betty Gabriel, Phoenix Raei, Adrian Grenier
36. The Last Duel (7.4) - Ft. Ridley Scott, Jodie Comer, Matt Damon, Ben Affleck, Adam Driver
37. Dear Evan Hansen (6.1) - Ft. Stephen Chbosky, Ben Platt, Julianne Moore, Kaitlyn Dever, Amy Adams
38. Alice in Borderland (7.7) - Ft. Kento Yamazaki, Tao Tsuchiya, Nijirô Murakami, Keita Machida
39. Billions (8.4) - Ft. Paul Giamatti, Damian Lewis, Maggie Siff, David Costabile
40. Intrusion (5.2) - Ft. Adam Salky, Freida Pinto, Logan Marshall-Green, Robert John Burke, Megan Elisabeth Kelly
41. NCIS: Naval Criminal Investigative Service (7.7) - Ft. Mark Harmon, David McCallum, Sean Murray, Pauley Perrette
42. No One Gets Out Alive (5.3) - Ft. Santiago Menghini, Phil Robertson, Joana Borja, Victoria Alcock, Cristina Rodlo
43. The Rookie (8) - Ft. Nathan Fillion, Alyssa Diaz, Richard T. Jones, Melissa O'Neil
44. Promising Young Woman (7.5) - Ft. Emerald Fennell, Carey Mulligan, Bo Burnham, Alison Brie, Clancy Brown
45. Wolf (4.2) - Ft. Nathalie Biancheri, Lily-Rose Depp, George MacKay, Paddy Considine, Terry Notary
46. Dune (6.4) - Ft. David Lynch, Kyle MacLachlan, Virginia Madsen, Francesca Annis, Leonardo Cimino
47. The Harder They Fall (6.2) - Ft. Jeymes Samuel, Zazie Beetz, Idris Elba, Regina King, Jonathan Majors
48. Titans (7.6) - Ft. Brenton Thwaites, Anna Diop, Ryan Potter, Teagan Croft
49. Cry Macho (5.8) - Ft. Clint Eastwood, Clint Eastwood, Dwight Yoakam, Daniel V. Graulau, Amber Lynn Ashley
50. American Crime Story (8.4) - Ft. Sarah Paulson, Annaleigh Ashford, Sterling K. Brown, Kenneth Choi

P.T.O

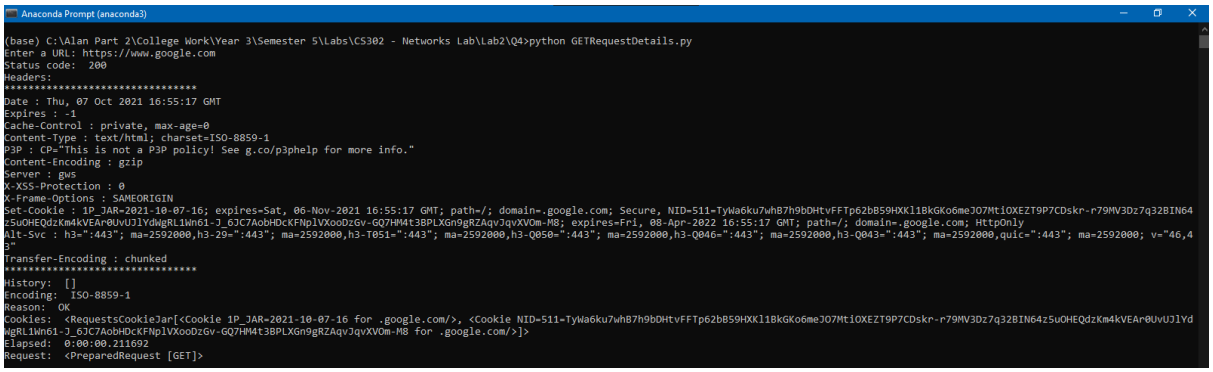
Question 4:

Code:

GETRequestDetails.py

```
import requests
from requests.api import head

url = input('Enter a URL: ')
response = requests.get(url)

print("Status code: ", response.status_code)    #Returns a number that indicat
es the status (200 is OK, 404 is Not Found)

print("Headers: ")    #Returns a dictionary of response headers
print("*******************************")
headers = response.headers
for key, value in headers.items():
    print(f"{key} : {value}")
print("*******************************")

print("History: ", response.history)    #Returns a list of response objects ho
lding the history of request (url)
print("Encoding: ", response.encoding)  #Returns the encoding used to decode r
esponse.text
print("Reason: ", response.reason)  #Returns a text corresponding to the statu
s code
print("Cookies: ", response.cookies)    #Returns a CookieJar object with the c
ookies sent back from the server
print("Elapsed: ", response.elapsed)    #Returns a timedelta object with the t
ime elapsed from sending the request to the arrival of the response
print("Request: ", response.request)    #Returns the request object that reque
sted this response
```

Output:

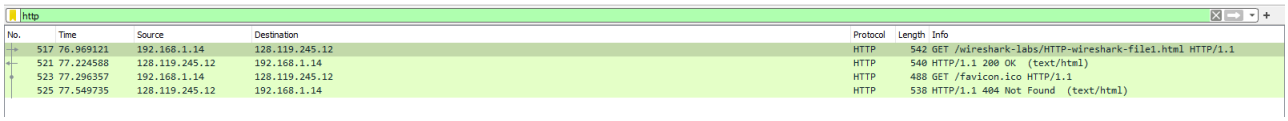


Question 5:

Analyzing HTTP messages:

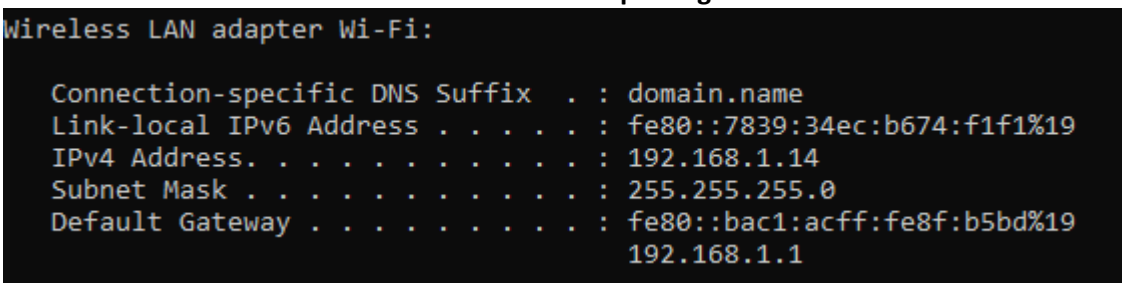
1. Webpage URL: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html

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

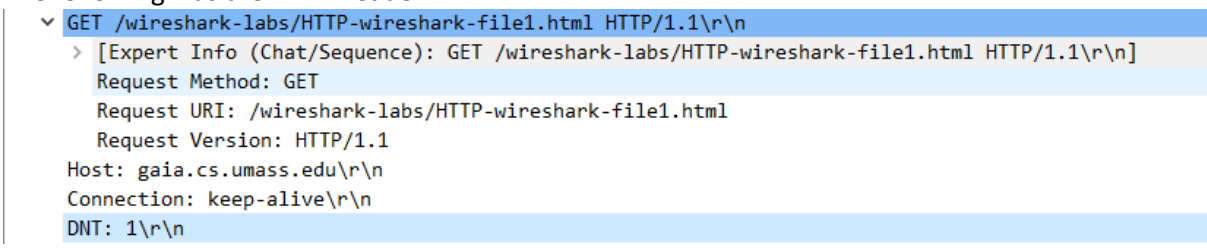


   Observations:

   - 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:

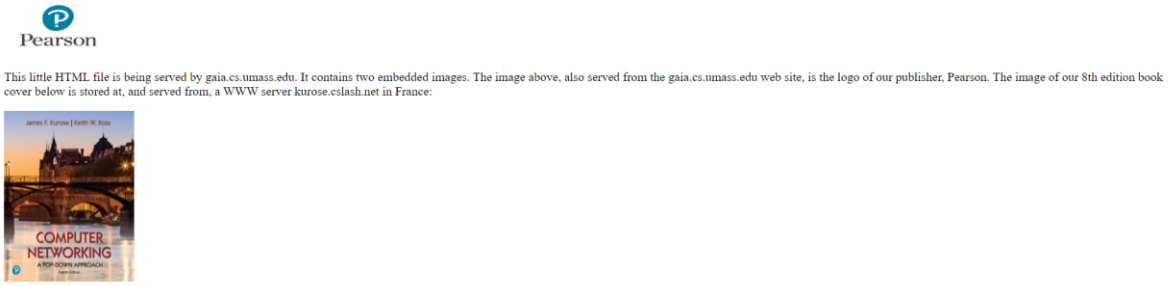


     The following was the HTTP header:



   - 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

2. Webpage URL: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html
   Screenshot of the webpage:



The HTTP messages:

Exchanges:

i. The first exchange was for the HTML file of the webpage. There are 2 images embedded in the HTML, which were delivered subsequently

ii. The 2nd exchange was for a .jpg file from the same host.

iii. 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

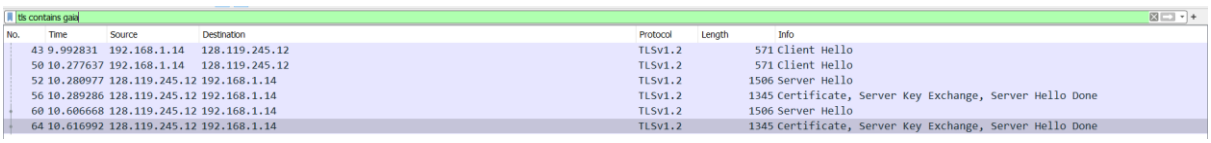
Analyzing HTTPS messages:

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.

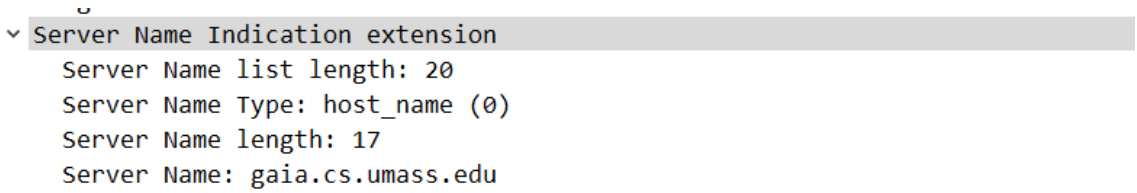
For this reason, Wireshark needs the session keys generated by the TLS handshake to decrypt the HTTP requests and responses made using the HTTP protocol. This is information can be logged by the web browser on setting an environment variable **SSLKEYLOGFILE**. We should also set the path of this file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

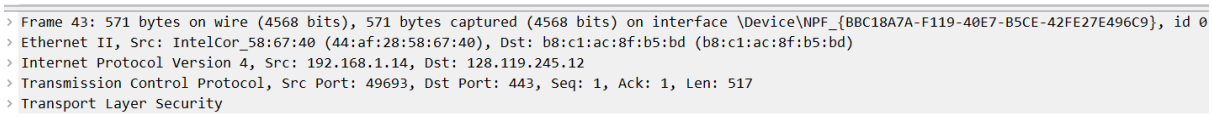
We can now filter the TLS messages with the keyword “gaia” since it is part of the domain name of the server:



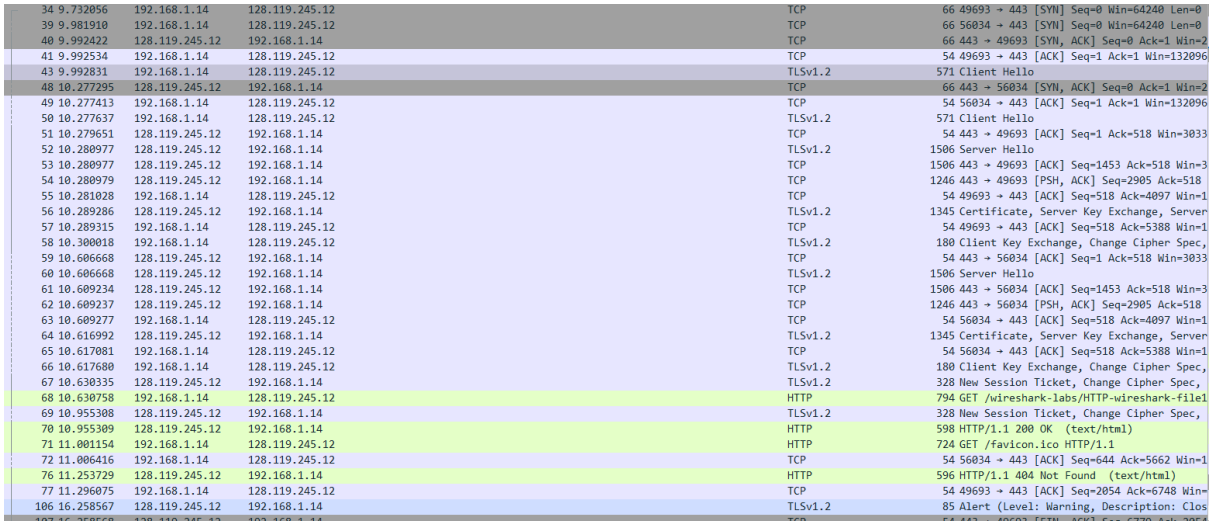
On expanding the details section of the first Client Hello TLS message, we see an extension named “server-name (len=22). This should contain the server’s name:



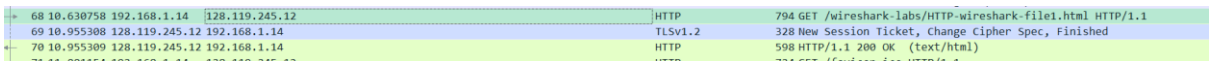
This is Client Hello frame is a part of the TLS handshake. It doesn’t contain the HTTP layer above it:



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:



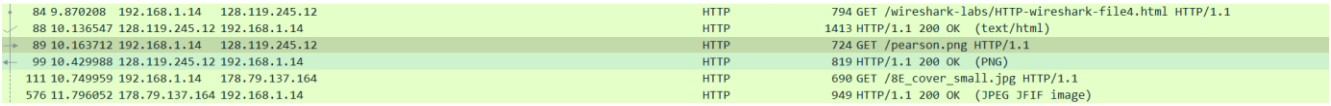
These messages contain an extra layer called HTTP, which are now decrypted:

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

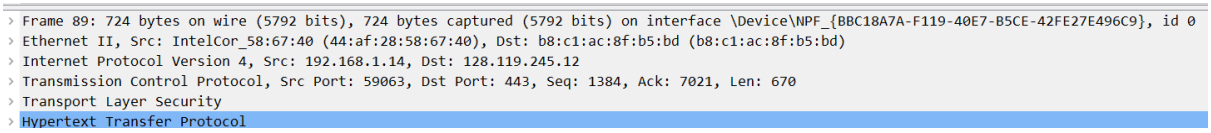
```
Line-based text data: text/html (4 lines)
    <html>\n
    Congratulations.  You've downloaded the file \n
    http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html!\n
    </html>\n
```

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:



*****