

LAB 2 – Application Layer

Name – Sati, Ankit

Date – 9/30/2021

Section - 001

Total in points (Maximum 100 points)–

Professors Comments –

Affirmation of Independent Effort – Ankit Sati

Answers.

Problem 1

- a. Below are the properties of my system to match the URL's opened in this Lab activity. I have hit the NYU browser and I have mentioned the link of the same below. The ASCII value for that urn hit are mentioned below as well.

Local System Properties

IPv4 : 10.0.0.220

IPv6 : 2601:87:8302:f830

- URL Capture on ASCII

00b0 00 0e 00 00 0b 77 77 77 2e 6e 79 75 2e 65 64 75www.nyu.edu

00c0 00 17 00 00 ff 01 00 01 00 00 0a 00 0a 00 08 aa

00d0 aa 00 1d 00 17 00 18 00 0b 00 02 01 00 00 23 00#.

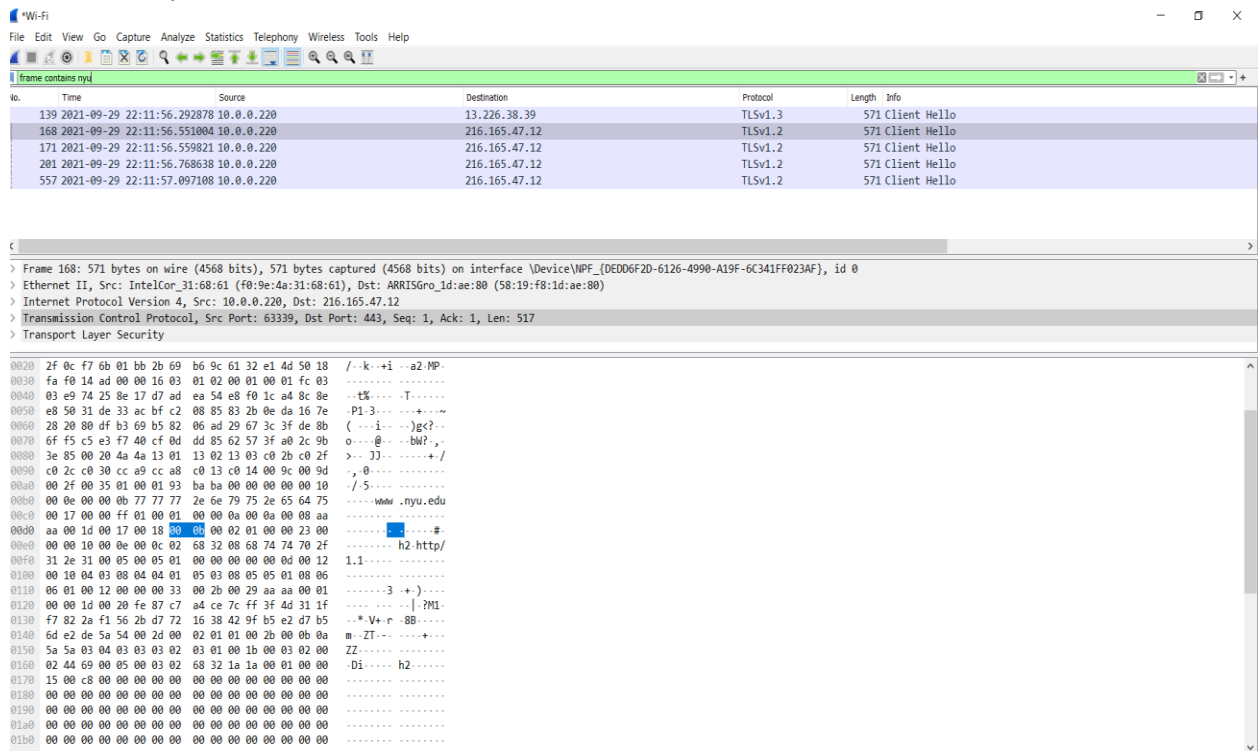
- Protocol : HTTPS or TLS1.2
- It looked Persistence , there is continuous data exchange back between server and local laptop without keep alive messages.

Properties

SSID:	Ravi 5
Protocol:	Wi-Fi 5 (802.11ac)
Security type:	WPA2-Personal
Network band:	5 GHz
Network channel:	153
Link speed (Receive/Transmit):	650/585 (Mbps)
IPv6 address:	2601:87:8302:f830::291b 2601:87:8302:f830:852b:4afa:3ded:dd67
Link-local IPv6 address:	fe80::852b:4afa:3ded:dd67%17
IPv6 DNS servers:	2001:558:feed::1 2001:558:feed::2
IPv4 address:	10.0.0.220
IPv4 DNS servers:	75.75.75.75 75.75.76.76
Manufacturer:	Intel Corporation
Description:	Intel(R) Wi-Fi 6 AX201 160MHz
Driver version:	22.60.0.6
Physical address (MAC):	F0-9E-4A-31-68-61

- URL Opened on Browser [NYU](https://www.nyu.edu)

- Wireshark Capture



- To check the HTTP part in this section, I pinged a HTTP website used in the previous lab.
 - URL Opened on Browser <http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html>
 - Version on HTTP ACK - Response Version: HTTP/1.1
 - Version of request (GET) - Request Version: HTTP/1.1
 - OS of User - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61 Safari/537.36\r\n
 - OS of Server - Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.24 mod_perl/2.0.11 Perl/v5.16.3\r\n
 - Wireshark Capture

- HTTP has a persistent connection function that allows the channel to remain open rather than be closed after a requested exchange of data.(p.s generally it is non persistant as well but over here it was persistant)

For example in the question above, the connection is kept alive.

Hypertext Transfer Protocol

HTTP/1.1 304 Not Modified\r\n

Date: Thu, 30 Sep 2021 03:10:02 GMT\r\n

Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.24 mod_perl/2.0.11

Perl/v5.16.3\r\n

Connection: Keep-Alive\r\n

Keep-Alive: timeout=5, max=100\r\n

- d. The browser that sent the request is basically seen in the GET file when we are trying to reach the web server. In this example the information for the GET is shared below.

Hypertext Transfer Protocol

GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n

Host: gaia.cs.umass.edu\r\n

Why is this necessary? - This basically tell the system which port, handshake, gateway etc needs to be used in order to reply to this request. If the system is aware about this request only in then it can go ahead and do what is needed and send an acknowledgement. This information is generally stored under the header GET.

- e. Yes, the request was successful. Post this there was an acknowledgement shared which is stored in the file with Header OK.

HTTP/1.1 200 OK\r\n

[Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]

[HTTP/1.1 200 OK\r\n]

[Severity level: Chat]

[Group: Sequence]

- f. Opened 5 different web servers.

- NYU (HTTPS)
- Traceroute (HTTPS)
- YouTube (HTTPS)
- GIAS (HTTP)
- GSuites and GitHub (HTTP)

Operations performed

- Checked the GET and OK files were received for all
- Compared the ASCII values to the URL that was hit while opening the sites.
- Took the IPv4 IP address of the websites.
- Check the other protocols such as TLS while surfing the internet.
- Check the protocols in use while surfing a video.
- Checked the different ports that were used on the local system while surfing these web pages.

Question 2

Answer 2

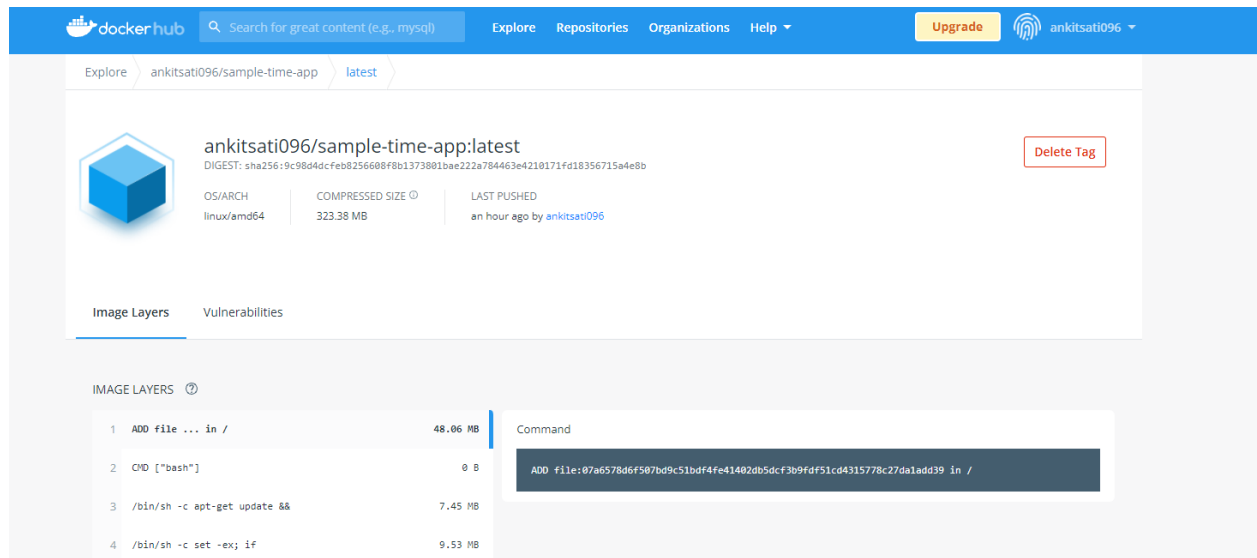
Pre req – Installed the Docker with WSL2.

Docker – Installed and account created. (UserID – ankitsati096)

1. Making changes to the Docker file. – Completed.

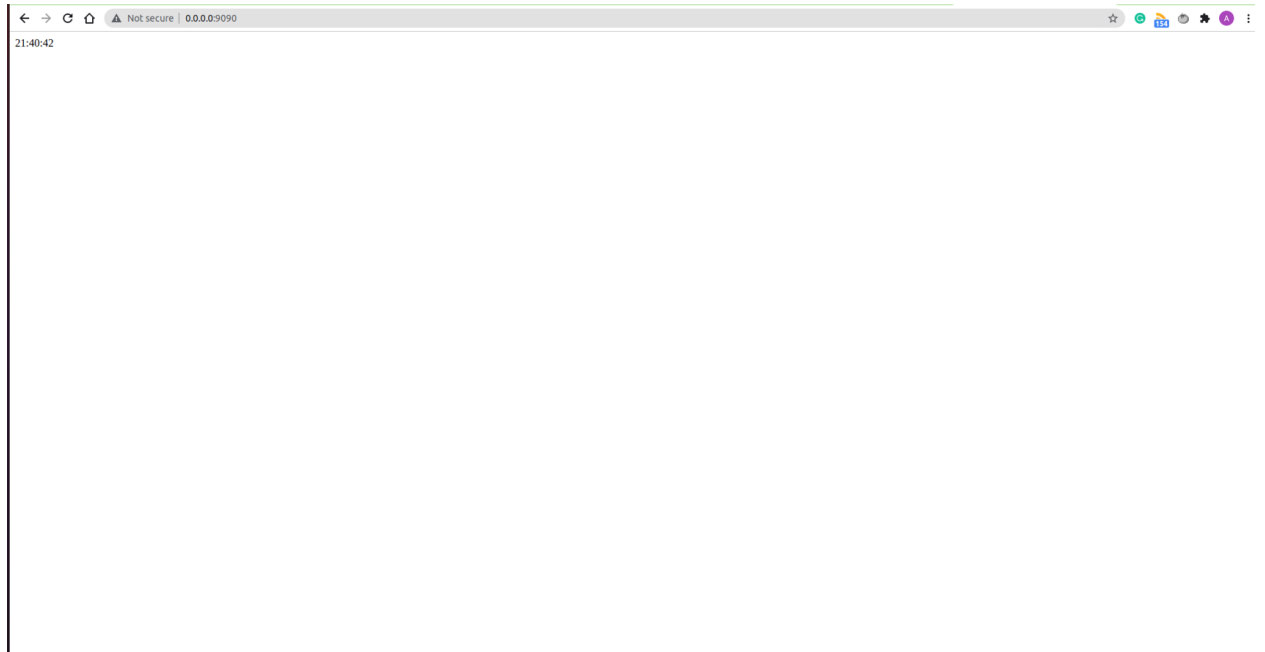
```
2. from flask import Flask
3. from datetime import datetime
4. app = Flask(__name__)
5.
6. @app.route('/')
7. def return_time():
8.     return datetime.now().strftime("%H:%M:%S")
9.
10. app.run(host='0.0.0.0',
11.          port=8080,
12.          debug=True)
```

2. File has been pushed to docker. (UserID – ankitsati096) - **ankitsati096/sample-time-app**



The screenshot displays the Docker Hub interface for the repository `ankitsati096/sample-time-app:latest`. The page includes a search bar, navigation links (Explore, Repositories, Organizations, Help), and a user profile for `ankitsati096`. The main content area shows the repository details, including the image digest, OS/ARCH (linux/amd64), compressed size (323.38 MB), and last pushed time (an hour ago). Below the image details, there is a section for 'IMAGE LAYERS' showing a list of layers with their commands and sizes. The first layer is 'ADD file ... in /' with a size of 48.06 MB. The second layer is 'CMD [\"bash\"]' with a size of 0 B. The third layer is '/bin/sh -c apt-get update &&' with a size of 7.45 MB. The fourth layer is '/bin/sh -c set -ex; if' with a size of 9.53 MB. A 'Delete Tag' button is visible in the top right corner.

3. File Deployed on K8 and the screenshot of the local host.
The App is up and running.



4. Finally the code is pushed to GIT.

<https://github.com/Satiankit96/sample-time-app.git>

