

COMPUTER NETWORKS MINI PROJECT

**Project Title: Design a tool to identify the type of devices used in
communication (Mobile/Desktop)**

Submitted by

RHEA ADHIKARI (190905156)

R LEKHYA REDDY (190905125)

In partial fulfillment for the award of degree of

B.TECH

IN

COMPUTER SCIENCE ENGINEERING



**MANIPAL INSTITUTE
OF TECHNOLOGY**
MANIPAL
A Constituent Institution of Manipal University

Department of Computer Science & Engineering

NOVEMBER 2021

BONAFIDE CERTIFICATE

Certified that this project report is the bonafide work of “RHEA ADHIKARI (190905156) & R LEKHYA REDDY (190905125)” who carried out the mini project work under my supervision.

Dr. Ashalatha Nayak
HoD

Mr Krishanamoorthi Makkithaya
Assistant Professor, Senior Scale

Submitted to the Viva Voce Examination held on _____.

EXAMINER 1

EXAMINER 2

ABSTRACT

The motivation of the project is to develop a tool that can be used to identify the types of devices (such as mobile and desktop) and also tell which vendor has manufactured the device used in communication.

This tool has been built to analyze the raw traffic files from Wireshark by scrapping out the useful information such as User-Agent and MAC addresses of the communicating devices from the pcap files and mapping them to their corresponding vendors and device types.

ACKNOWLEDGEMENT

We express our deep sense of respect and our gratitude to our supervisor and professor Krishanamoorthi Makkithaya who created a healthy learning environment, and patiently guided us. It was really a great experience for us to work under him. We would also like to extend our thanks to our HoD, Dr. Ashalatha Nayak.

We would also like to extend our thanks to other faculty members and non-teaching staff of Manipal Institute of Technology for providing needed support for this project.

Place: Manipal

Date: 30/11/2021

TABLE OF CONTENTS

1. **CHAPTER ONE:** NIC, MAC Address and IP Address
 - 1.1 Relevance
 - 1.2 Methodology
2. **CHAPTER TWO:** Programs (in python)
 - 2.1 Identifying device type using Useragent from raw traffic files.
 - 2.1.1 Code
 - 2.1.2 Implementation / Screenshots
 - 2.1.3 Explanation
 - 2.2 Identifying MAC addresses of the communicating devices
 - 2.2.1 Code
 - 2.2.2 Implementation / Screenshots
 - 2.2.3 Explanation
3. **REFERENCES**

1. CHAPTER ONE

1.1 Relevance

The MAC addresses of all devices on the same network subnet are distinct. A network adapter is given a MAC address when it is created. It's hardwired into the NIC of your computer and is unique to it. An IP address is converted to a MAC address via the ARP (Address Resolution Protocol) that transmits data from an IP address to a piece of computer hardware. While IP addresses can change dynamically, MAC addresses do not, which makes it a reliable mode of identifying communicating devices.

1.2 Methodology

Using the background information and relevant research, we have come up with the following methodology to tackle the problem statement.

- From the pcap file, extract the following information:
 - User-Agent string
 - MAC address
- Map the MAC address to the vendor of the communicating device API

2. CHAPTER TWO

2.1 Identify Device Type using User-Agent

2.1.1: Code

```
import os
import sys

# Used to capture packets from pcap
import pyshark

# Getting device type from useragent
from user_agents import parse

# GUI
from tkinter import *
from tkinter import filedialog
import tkinter.font as tkFont

root = Tk()
fontStyle = tkFont.Font(family="Lucida Grande", size=15)
root.geometry("1500x600")

def isMobileDevice(useragent):
    user_agent = parse(useragent)
    if user_agent.is_mobile:
        return True
    else:
        return False
```

```
def isTabletDevice(useragent):  
    user_agent = parse(useragent)  
    if user_agent.is_tablet:  
        return True  
    else:  
        return False
```

```
def isPC(useragent):  
    user_agent = parse(useragent)  
    if user_agent.is_pc:  
        return True  
    else:  
        return False
```

```
def getUserAgent():  
    root.filename = filedialog.askopenfilename(  
        initialdir="/", title="Select file")  
  
    if root.filename == "":  
        print("No file selected")  
        sys.exit()  
    else:  
        useragents = []  
        cap = pyshark.FileCapture(  
            root.filename, display_filter='frame contains "GET"')  
        for packet in cap:  
            print(packet['http'].user_agent)
```



```

        useragents.append(packet['http'].user_agent)
i = 0
for useragent in useragents:
    i = i+1
    if isMobileDevice(useragent):
        myLabel = Label(
            root, text="Device Type : Mobile", font=fontStyle)
        myLabel.pack()
    elif isTabletDevice(useragent):
        myLabel = Label(
            root, text="Device Type : Tablet", font=fontStyle)
        myLabel.pack()
    elif isPC(useragent):
        myLabel = Label(
            root, text="Device Type : PC", font=fontStyle)
        myLabel.pack()
    else:
        myLabel = Label(
            root, text="Device Type : Unknown", font=fontStyle)
        myLabel.pack()

    myLabel = Label(root, text="Packet"+str(i) +
        ": " + useragent+"\n", font=fontStyle)

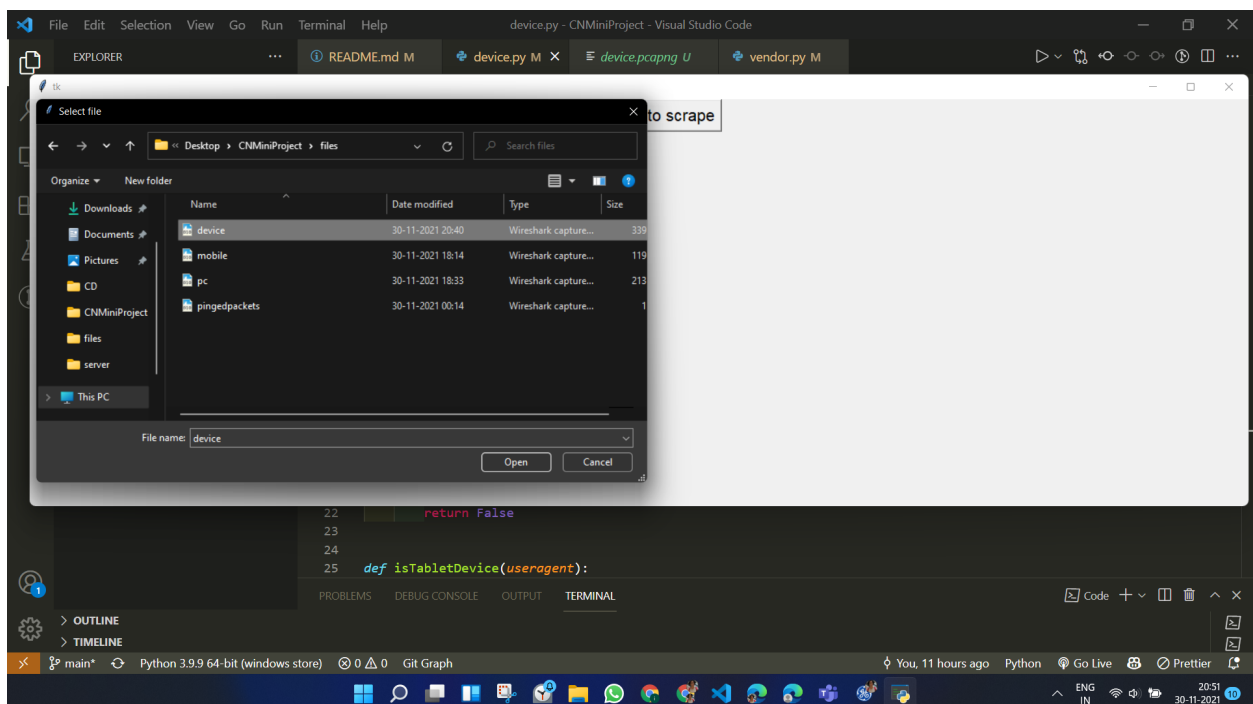
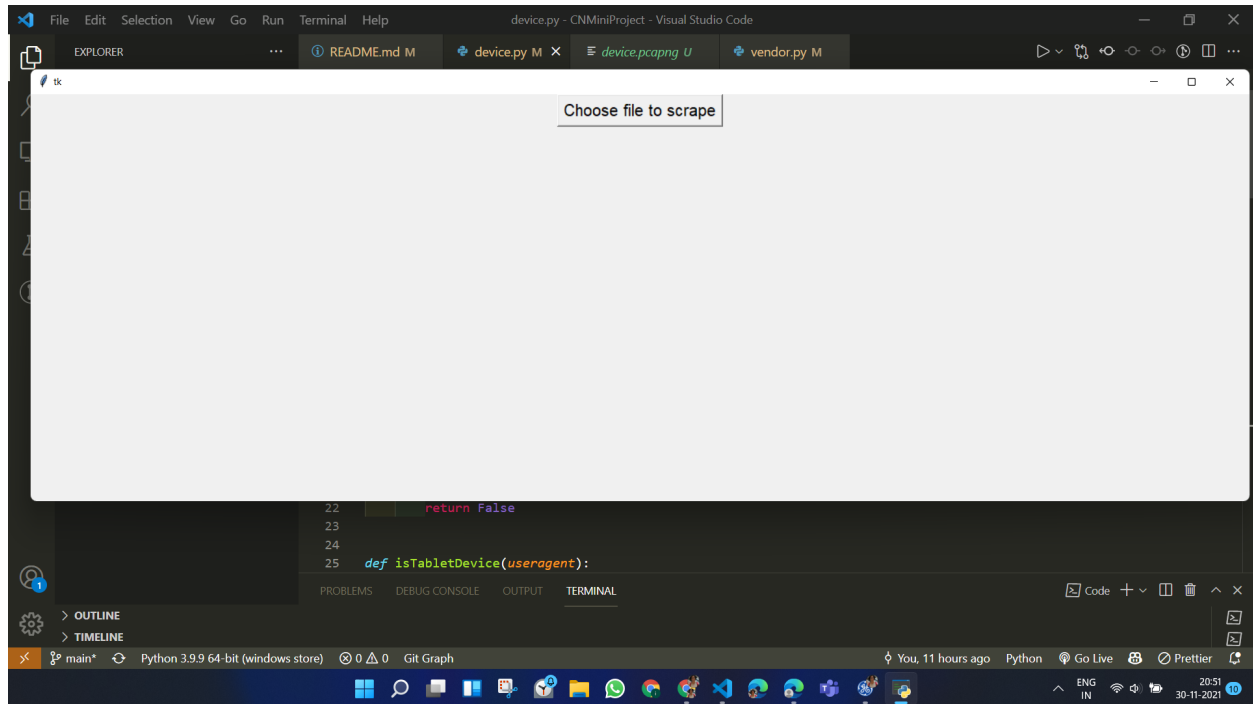
    myLabel.pack()

cap.close()
fileInputBtn = Button(root, text="Choose file to scrape", font=fontStyle,
    command=getUserAgent)
fileInputBtn.pack()
root.mainloop()

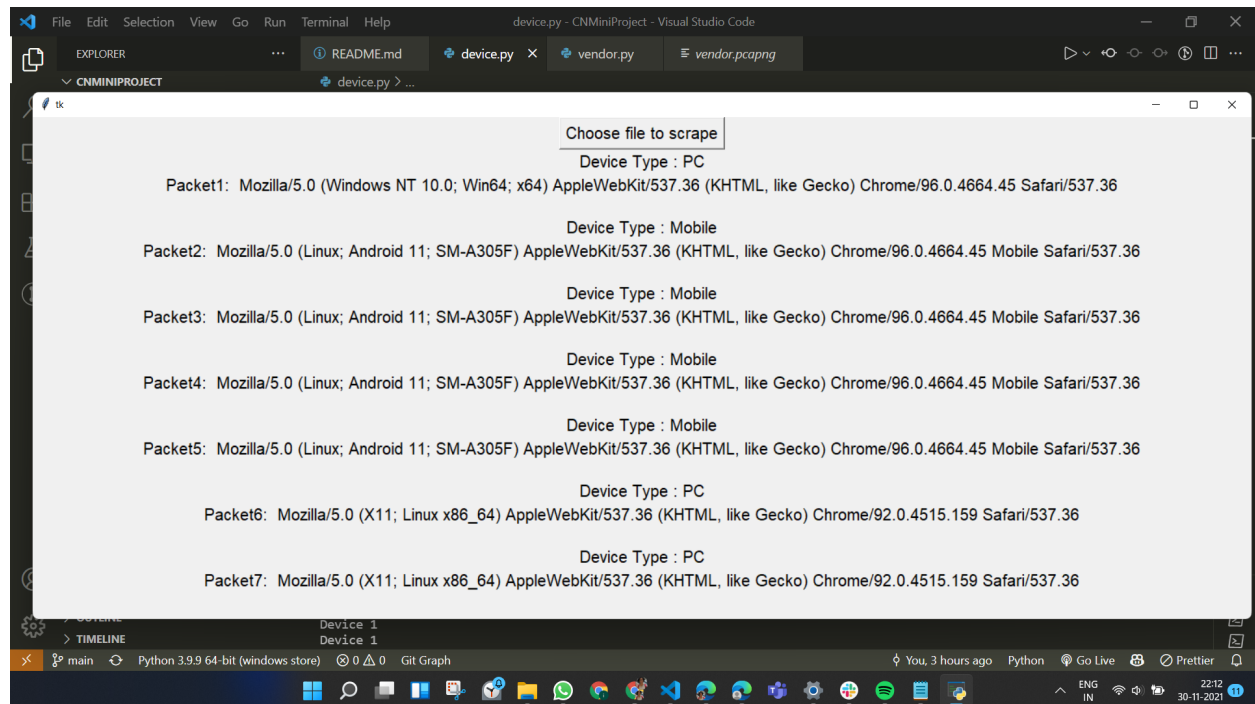
```

2.1.2: Implementation

Users are allowed to choose any file of the format .pcap/ .pcapng from the file explorer via the graphical user interface.



We can clearly see the output required has been achieved in the following screenshot :



2.1.3 : Explanation

To get the network packets on Wireshark we had to make GET requests to certain websites using the HTTP protocol.

We noticed that one of the fields inside the HTTP tab contained device browser information. We extracted this field from all the packets using **‘pyshark’** and used a third party library **‘user_agents’** in python to allow us to identify whether the network packet that showed up was from a mobile/ tablet/ PC or a Unknown Device.

2.2 Identify the vendor who created the device using the MAC address of users in the same network.

2.2.1: Code

```
import os
import sys

# Used to capture packets from pcap
import pyshark

# Getting device type from useragent
from user_agents import parse

# GUI
from tkinter import *
from tkinter import filedialog
import tkinter.font as tkFont

root = Tk()
fontStyle = tkFont.Font(family="Lucida Grande", size=15)
root.geometry("1500x600")

def isMobileDevice(useragent):
    user_agent = parse(useragent)
    if user_agent.is_mobile:
```

```
        return True
    else:
        return False
```

```
def isTabletDevice(useragent):
    user_agent = parse(useragent)
    if user_agent.is_tablet:
        return True
    else:
        return False
```

```
def isPC(useragent):
    user_agent = parse(useragent)
    if user_agent.is_pc:
        return True
    else:
        return False
```

```
def getUserAgent():
    root.filename = filedialog.askopenfilename(
        initialdir="/", title="Select file")

    if root.filename == "":
        print("No file selected")
        sys.exit()
    else:
        useragents = []
        cap = pyshark.FileCapture(
```

```

        root.filename, display_filter='frame contains "GET"')
for packet in cap:
    print(packet['http'].user_agent)
    useragents.append(packet['http'].user_agent)
i = 0
for useragent in useragents:
    i = i+1
    if isMobileDevice(useragent):
        myLabel = Label(
            root, text="Device Type : Mobile", font=fontStyle)
        myLabel.pack()
    elif isTabletDevice(useragent):
        myLabel = Label(
            root, text="Device Type : Tablet", font=fontStyle)
        myLabel.pack()
    elif isPC(useragent):
        myLabel = Label(
            root, text="Device Type : PC", font=fontStyle)
        myLabel.pack()
    else:
        myLabel = Label(
            root, text="Device Type : Unknown", font=fontStyle)
        myLabel.pack()

    myLabel = Label(root, text="Packet"+str(i) +
        ": " + useragent+"\n", font=fontStyle)

    myLabel.pack()
cap.close()
fileInputBtn = Button(root, text="Choose file to scrape", font=fontStyle,
    command=getUserAgent)

```

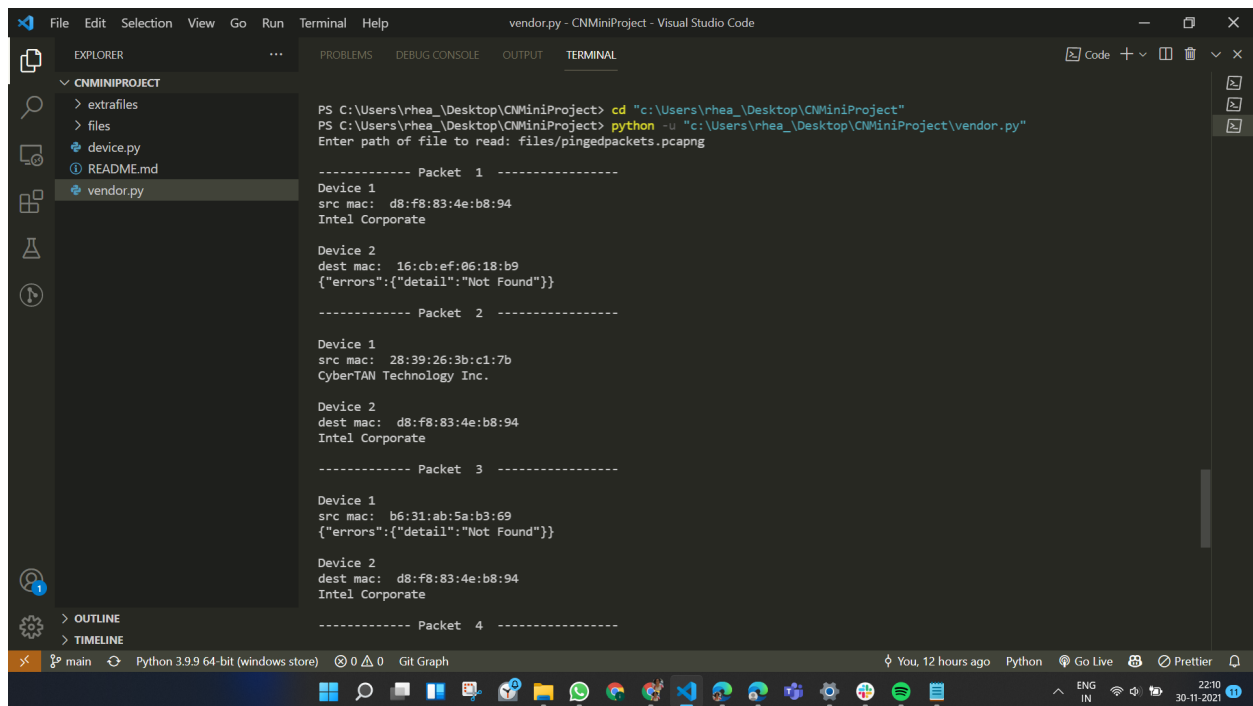
```
fileInputBtn.pack()
```

```
root.mainloop()
```

2.2.2 : Implementation

User needs to type in a path of a .pcap or .pcapng file inorder to show the vendors of the respective manufactured devices.

The errors here suggest that the API used wasnt able to figure out who the vendor was and is unknown to us.



```
PS C:\Users\rhea\Desktop\CNMiniProject> cd "c:\Users\rhea\Desktop\CNMiniProject"
PS C:\Users\rhea\Desktop\CNMiniProject> python -u "c:\Users\rhea\Desktop\CNMiniProject\vendor.py"
Enter path of file to read: files/pingedpackets.pcapng

----- Packet 1 -----
Device 1
src mac: d8:f8:83:4e:b8:94
Intel Corporate

Device 2
dest mac: 16:cb:ef:06:18:b9
{"errors":{"detail":"Not Found"}}

----- Packet 2 -----

Device 1
src mac: 28:39:26:3b:c1:7b
CyberTAN Technology Inc.

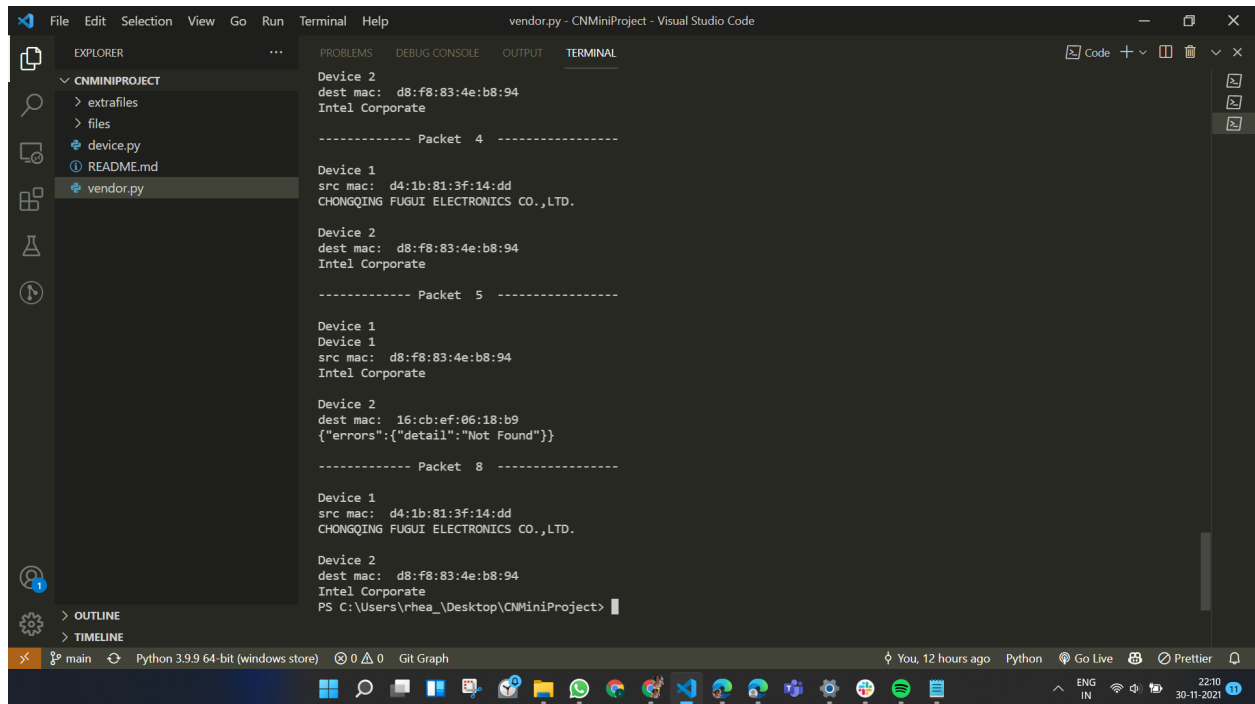
Device 2
dest mac: d8:f8:83:4e:b8:94
Intel Corporate

----- Packet 3 -----

Device 1
src mac: b6:31:ab:5a:b3:69
{"errors":{"detail":"Not Found"}}

Device 2
dest mac: d8:f8:83:4e:b8:94
Intel Corporate

----- Packet 4 -----
```



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays network packet data, including source and destination MAC addresses and vendor information. The Explorer panel on the left shows a project named 'CNMINIPROJECT' with files like 'device.py', 'README.md', and 'vendor.py'. The Terminal panel on the right shows the output of a script, including packet details for Device 1 and Device 2.

```
Device 2
dest mac: d8:f8:83:4e:b8:94
Intel Corporate

----- Packet 4 -----

Device 1
src mac: d4:1b:81:3f:14:dd
CHONGQING FUGUI ELECTRONICS CO.,LTD.

Device 2
dest mac: d8:f8:83:4e:b8:94
Intel Corporate

----- Packet 5 -----

Device 1
Device 1
src mac: d8:f8:83:4e:b8:94
Intel Corporate

Device 2
dest mac: 16:cb:ef:06:18:b9
{"errors":{"detail":"Not Found"}}

----- Packet 8 -----

Device 1
src mac: d4:1b:81:3f:14:dd
CHONGQING FUGUI ELECTRONICS CO.,LTD.

Device 2
dest mac: d8:f8:83:4e:b8:94
Intel Corporate
PS C:\Users\rhea\Desktop\CNMiniProject>
```

2.2.3 : Explanation

We connected several devices on a single network and made each device ping my computer after which we extracted the packets containing 'arp' protocol. We made use of the fact that the MAC address for source destination and target destination were available inside the packet. We use a third party site to map the MAC address to its vendor and finally display the output.

3. REFERENCES

- Jasraj. "HTTP Headers: User-Agent." *GeeksforGeeks*, 11 Oct. 2019, <https://www.geeksforgeeks.org/http-headers-user-agent/>
- TechGeekShan. *Find the Manufacturer Using MAC Address - Youtube*. <https://www.youtube.com/watch?v=2Rah5Pi1PTY>
- Sieling, Gary. "How to Filter out a MAC Address in Wireshark." *Gary Sieling*, 11 Mar. 2016, <https://www.garysieling.com/blog/filter-mac-address-wireshark/>
- Hoffman, Chris. "How to Use Wireshark to Capture, Filter and Inspect Packets." *How, How-To Geek*, 14 June 2017, <https://www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packets/>
- "User-Agent - Http: MDN." *HTTP | MDN*, <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent>.
- Oberheide, Jon. "Dpkt Tutorial #2: Parsing a PCAP File." *Dpkt Tutorial #2: Parsing a PCAP File | Jon Oberheide*, <https://jon.oberheide.org/blog/2008/10/15/dpkt-tutorial-2-parsing-a-pcap-file/>