MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Institution of Manipal University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

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

A Mini Project Report
Submitted by

Akshay Gupta - 200905040
Shubham Srivastava - 200905152

In partial fulfillment for the award of the degree of
Bachelor of Technology
In
Computer Science and Engineering

Department of Computer Science and Engineering

BONAFIDE CERTIFICATE

Certified that this project report, **"Design a tool to identify the type of devices used in communication (Mobile/Desktop)"** is the bonafide work of

Akshay Gupta - 200905040
Shubham Srivastava - 200905152

who carried out the project under my supervision.

Dr. Ashalatha Nayak
Head of Department
Department of CSE
Manipal Institute of Technology
Manipal India

Ms. Tanuja Shailesh
Professor
Department of CSE
Manipal Institute of Technology
Manipal, India

Submitted to the Viva Voce exam held on

_____

Examiner 1

Examiner 2

ACKNOWLEDGEMENT

We would like to thank the Department of Computer Science and Engineering for giving us the opportunity to work on a project to help in our understanding of the coursework.

We would like to thank our teacher, Ms Tanuja Shailesh, for guiding us through this project and our coursework in Computer Networks. Her teachings have inspired us to take up this implementation. We would also like to express our sincere gratitude to Ms Sucharitha Shetty, who also guided us during our labs.

A particular acknowledgement goes to our labmates, who helped us by exchanging exciting ideas and information. Finally, we would like to thank our parents for their unwavering support and continuous encouragement in our educational journey.

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).

Device detection enables developers to identify device properties and characteristics in order to determine the best content, layout, mark-up or application to serve to a given device. These characteristics include screen size, browser type and version, media support, and the level of support for CSS, HTML and JavaScript.

This programme was created to examine raw traffic files from Wireshark by extracting relevant information such as the User-Agent of communicating devices from pcap files and mapping it to their appropriate device types.

KEYWORDS

Wireshark, User-Agent, Device Detection, PCAP Files, Tkinter, Python, Pyshark

TABLE OF CONTENTS

CHAPTER 1

NIC, MAC ADDRESS AND IP ADDRESS

1.1 INTRODUCTION

NIC: A network interface controller is a computer hardware component that connects a computer to a computer network

Mac Address: A media access control address is a unique identifier assigned to a network interface controller for use as a network address in communications within a network segment.

IP Address: An Internet Protocol address is a numerical label such as 192.0.2.1 that is connected to a computer network that uses the Internet Protocol for communication.

1.2 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.

1.3 METHODOLOGY

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

One of the fields on the HTTP tab revealed device browser information, which we discovered. We retrieved this data from all of the packets using **'pyshark'** and a third-party package called **'user agents'** in Python to determine if the network packet was from a mobile/tablet/PC or an Unknown Device.

1.4 WHAT IS HTTP?

The Hypertext Transfer Protocol (HTTP) is an application layer protocol in the Internet protocol suite model for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web, where hypertext documents include hyperlinks to other resources that the user can easily access, for example by a mouse click or by tapping the screen in a web browser.

1.5 WHAT IS AN HTTP REQUEST?

An HTTP request is the way internet communications platforms such as web browsers ask for the information they need to load a website.

Each HTTP request made across the Internet carries with it a series of encoded data that carries different types of information. A typical HTTP request contains:

1. HTTP version type
2. a URL
3. an HTTP method
4. HTTP request headers
5. Optional HTTP body.

```
GET /home.html HTTP/1.1
Host: developer.mozilla.org
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:50.0)
Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://developer.mozilla.org/testpage.html
Connection: keep-alive
Upgrade-Insecure-Requests: 1
If-Modified-Since: Mon, 18 Jul 2016 02:36:04 GMT
If-None-Match: "c561c68d0ba92bbeb8b0fff2a9199f722e3a621a"
Cache-Control: max-age=0
```

Fig 1 HTTP Request Header

CHAPTER 2

DEVICE DETECTION TOOL

2.1 INTRODUCTION TO WIRESHARK

Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and try to display that packet data as detailed as possible. Wireshark helps us capture network packets and display them at a granular level. Once these packets are broken down, we can use them for real-time or offline analysis. This tool lets us put your network traffic under a microscope, and then filter and drill down into it, zooming in on the root cause of problems, assisting with network analysis and ultimately network security.

2.1.1 HTTP IN WIRESHARK

HTTP traffic shows up as a light green in Wireshark and can be filtered using http. However, since HTTP runs over TCP and http only shows packets using the HTTP protocol, this can miss many of the packets associated with the session because they are TCP packets (SYN, ACK and so on). Wireshark reassembles all of the actual data packets containing a particular webpage and displays it within the packet labeled as the HTTP response.

Fig 2 HyperText Transfer Protocol in Wireshark

2.2 IDENTIFY DEVICE TYPE USING USER-AGENT

The User-Agent (UA) string in the HTTP headers is intended to identify devices requesting online content. The User-Agent tells the server what the visiting device is (among many other things), and this information can be used to determine what content to return.
There are millions of User-Agent combinations, given that UAs change with the software and hardware.

The main components we can identify with a User-Agent are:

- Physical Device - The model of the device including chips, price, age, screen dimensions and supported mobile network technologies.
- Operating System - Whether it's running on Android, iOS, Windows, or another OS, including the version.
- Browser - Whether it's Chrome or Firefox, or another browser.

Device detection solutions use User-Agents in the following way:

1. When a user accesses your site, the website or CMS receives the request.
2. Your website returns a page optimized for the precise screen and device type, ensuring maximum performance and the best user experience every time.
3. Other revenue enhancing customizations can be applied based on other factors such as whether the device supports telephone calls, SMS or has a built-in physical qwerty pad.

For example, a Chrome browser on an iPhone 6 will introduce itself using a different UA than a Safari browser on the same phone.
Every device type, including phones, tablets, and desktops, may come with its own UA that makes it possible to detect this device for any purpose.

2.2.1 CODE SNIPPETS

1. Used Tkinter GUI Toolkit for the Graphical User Interface

```
heading = 'Device Type Detection'
root = Tk(className=heading.title())
fontStyle = tkFont.Font(family="Lucida Grande", size=10)
fontStyle1 = tkFont.Font(family="Lucida Grande", size=25, weight="bold")
root.geometry("2000x800")
root.title(heading)


# create background image bg.jpg
image = Image.open("bg.jpg")
photo = ImageTk.PhotoImage(image)
label = Label(root, image=photo)
label.place(x=1, y=1, relheight=3.5, relwidth=3.5)
```

2. getUserAgent() for extracting useful information through pyshark python library

```
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)
```

3. Various methods for checking the device type

```
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
```

4. Mapping of the information returned from user_agent to various devices

```
    for useragent in useragents:
        i = i+1
        if isMobileDevice(useragent):
            myLabel = Label(
                root, text="Device Type : Mobile 📱", font=fontStyle1, fg="white", bg="#042592")
            myLabel.pack()
        elif isTabletDevice(useragent):
            myLabel = Label(
                root, text="Device Type : Tablet 📱", font=fontStyle1, fg="white", bg="#042592")
            myLabel.pack()
        elif isPC(useragent):
            myLabel = Label(
                root, text="Device Type : Desktop 🖥️", font=fontStyle1, fg="white", bg="#042592")
            myLabel.pack()
        else:
            myLabel = Label(
                root, text="Device Type : Unknown❓", font=fontStyle1, fg="white", bg="#042592")
            myLabel.pack()

        myLabel = Label(root, text="Packet"+str(i) +
                        ": " + useragent+"\n", font=fontStyle, fg="white", bg="#042592")
```

2.2.2 Implementation

Users can utilize the graphical user interface to choose any file of the format .pcap/.pcapng from the file explorer.

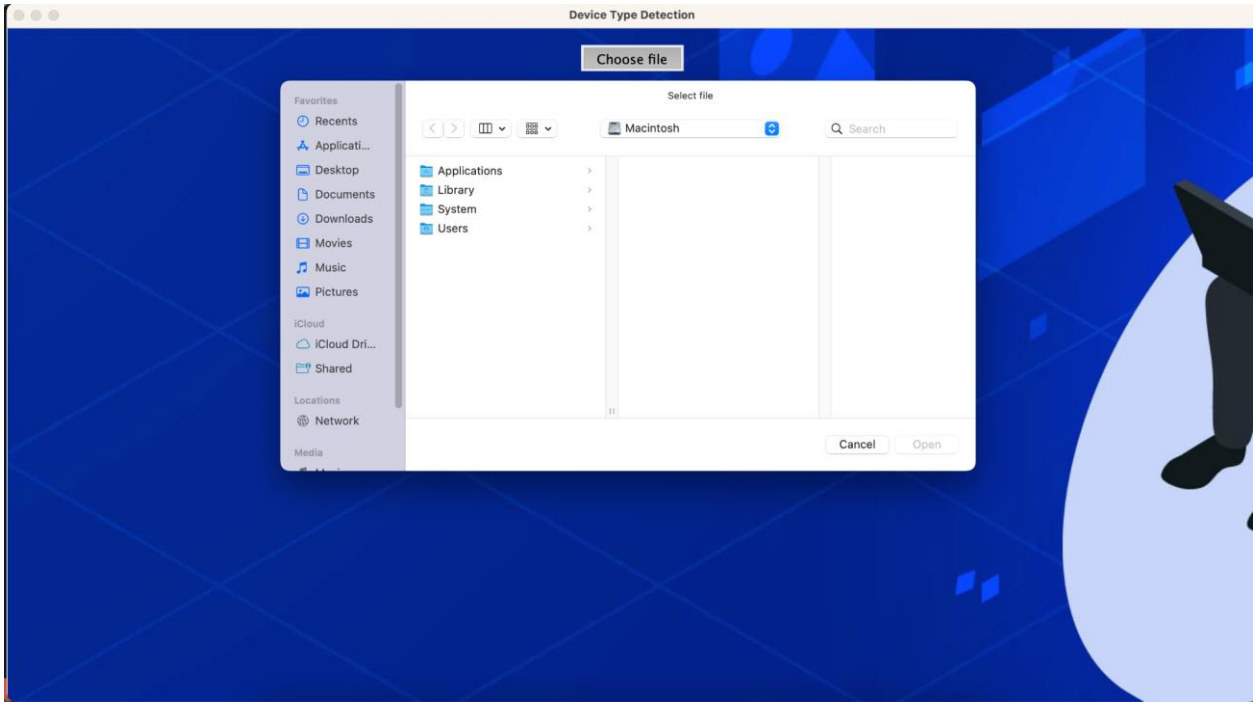

Fig 3 Dialog Box for selecting PCAP file

Fig 4 Device Type along with the user agent field

2.2.3 Explanation

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

Every time you visit a website, your device transmits a User-Agent to that website. A User-Agent is a string of characters which contains information about your device.

We noticed that one of the fields inside the HTTP tab contained a User-Agent Field. 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 an Unknown Device.

Here is an example User-Agent:

Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36

CONCLUSION

It is usually a terrible idea to serve different Web pages or services to various browsers. The Web is intended to be available to everyone, regardless of browser or device. There are methods for developing your website so that it gradually improves based on the availability of functionality rather than by targeting specific browsers.

However, browsers and standards are not flawless, and there are still some edge circumstances where the browser must be detected. Developers can use device detection to identify device traits and characteristics in order to select the optimum content, layout, markup, or application to offer to a given device. These features include screen size, browser type and version, media support, and the level of CSS, HTML, and JavaScript support.

REFERENCES

1. Jasraj. “HTTP Headers: User-Agent.” GeeksforGeeks, 11 Oct. 2019, https://www.geeksforgeeks.org/http-headers-user-agent/

2. TechGeekShan. Find the Manufacturer Using MAC Address - Youtube. https://www.youtube.com/watch?v=2Rah5Pi1PTY

3. 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/

4. 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-inspectpackets/

5. “User-Agent - Http: MDN.” HTTP | MDN, https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent.

6. 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/