2021

Research Manual
Network Sniffer

CW258 CYBERCRIME & IT SECURITY

ERLANDAS BACAUSKAS

INSTITUTE OF TECHNOLOGY CARLOW | Supervisor: Paul J. Barry

Abstract

Dating back to 1980, computer security threat monitoring and surveillance and the concept of intrusion detection was born. It was introduced by James Anderson who written for a government organization to introduce the concept of auditing trails that would contain vital information for tracking and detecting misuse and understanding user behavior. James Anderson's work was a steppingstone for a host-based intrusion detection systems in general. The following are concepts proposed by James Anderson in 1988 - the Haystack project at Lawrence Livermore Labs released a version of IDS that was able to analyze audited data by comparing it with defined patterns. One of the employees of a former Haystack Project team criticized such a project by saying "Searching through this large amount of data for one specific misuse was equivalent to looking for a needle in a haystack" (Banerjee, et al., 2010). Following IDS concepts in 1990, UC Davis Todd Heberlein introduced the idea of network intrusion detection systems. Heberlein himself was the primary developer and author of Network Security Monitor (NSM) system. This system was first of its kind that was able to monitor network traffic. NSM gave grounds for the first system that would lead to commercial development of such technology. Commercial development of IDS systems began in early 1990 by Haystack labs that were also the first commercial vendors of such a system. Intrusion detection system popularity skyrocketed by gaining fame and generating revenues. In around 1997, Gerald Combs started to write a tool that would capture and analyze network packets. He managed to release the very first version of this tool in 1998 and since 2006 this toll was renamed to WireShark. (Banerjee, et al., 2010)

Table of Contents

Introduction

For my final year project, I have undertaken to develop a network analyzer application. I am planning to develop an application that will be capable to capture, inspect network packets and perform OSI layer 2 network attacks. I envision this application to be used by any students who might be interested to learn about this area but also network administrators, penetration testers and anyone else who might have interest in networking and wants to expand on their knowledge.

To begin on this project, I will be conducting extensive researching of various tools that have similar functionality to what I will be trying to achieve as well as additional research to implement additional functionalities that are not included in some of these tools such as WireShark and TCPdump. Additionally, I will be researching the front-end and back-end technologies which would be suitable to develop this application that I envision. I also plan to research network packets structure in detail, analyzing each of the fields that are included in UDP, TCP, IP and ICMP packets.

Additionally, I will be examining the benefits that these tools are offering and the use cases of these tools, to better understand the areas where my project could be improved beyond what is included in the other tools mentioned above.

Network Packet Analyzed

Main goal of the project which I aim to fulfill is to capture and analyze network packets that flows through device that is connected to a network. To accomplish this goal, I must analyze the main types of packets that are present within the network environment. These packets are: UDP, TCP, and IP.

The key aspect of networking is its capability to build data into packets, the packets then travel around the network from one destination to another. At some point, engineers or network administrators need to disassemble network packets while troubleshooting. They do this to take a closer look at the contents of packets to see what's happening on the network. The Layer 3 typically utilizes two major protocols, TCP and UDP, the packets range from 64 to 1,500 bytes in size. (Mullins, 2021)

However, to analyze network packets I must have a strong understanding of Open System Interconnection (OSI) model. Packets that I will be analyzing resides on OSI layer 3 and OSI layer 4 where is UDP and TCP are Layer 4 protocols and IP is Layer 3 protocol. In addition, to note TCP/IP model have resemblances to OSI model, layers of OSI that will be analyzed are on TCP/IP model. Below are tables outlining OSI and TCP/IP models. (Mullins, 2021)

OSI Model		
Layer	Function	Example
Applications (Layer 7)	Services that are used with end user application	SMTP
Presentation (Layer 6)	Formats the data so that it can be viewed by the user, responsible for encryption and decryption of data	JPG, GIF, HTTPS, SSL, TLS
Session (Layer 5)	Establishes and ends connections between two hosts	NetBIOS, PPTP
Transport (Layer 4)	Responsible for the transport protocol and error handling	TCP, UDP

Network (Layer 3)	Reads the IP address from the packet	IP, Routers, Layer 3 Switches
Data Link (Layer 2)	Reads the MAC address from the data packet	Switches
Physical (Layer 1)	Send data on to the physical wire	Hubs, network interface cards, cables

Table 1 - OSI Model (https://www.blackmoreops.com/wp-content/uploads/2016/05/OSI-Layer-Please-Do-Not-Tell-Secret-Passwords-Anytime-blackMORE-Ops-1.png)

TCP/IP		
Layer	Function	Example
Application	Presents data to the user, encoding and session control	SMTP, JPG, GIF, HTTPS, SSL, TLS, NetBIOS, PPTP
Transport	Support of communication between diverse devices and network	TCP, UDP
Network	Determines the path in the network	IP, Routers, Layer 3 Switches
Data Link	Reads the MAC address from the data packet	Switches
Physical	Send data on to the physical wire	Hubs, network interface cards, cables

Table 2 - TCP/IP Model (https://www.certbros.com/wp-content/uploads/2020/03/TCP-IP-Model-Updated-1024x483.jpg)

User Datagram Protocol (UDP)

UDP is the transport protocol that has a key aspect which is a connectionless protocol that does not have any reliability, flow control and error recovery functions. Because of its composition and lack of functionality UDP headers consists of fewer bytes in size and consumes significantly less network bandwidth in comparison with TCP protocol. Main use of UDP protocol is in situations where reliability is not needed or in some cases where higher level protocols might provide error handling and flow control. UDP protocol user which is similar to TCP service without overhead, unlike TCP, UDP packets can be discarded before reaching their destination. UDP is very useful when TCP would be too slow, unnecessary, or even too complex. Some examples of applications that use UDP protocol are media streaming, online gaming, voice over IP (VoIP), Trivial File Transfer Protocol (TFTP), Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), Simple Network Management Protocol (SNMP) and Routing Information Protocol (RIP). Below is an image of UDP packet. (Mullins, 2021)

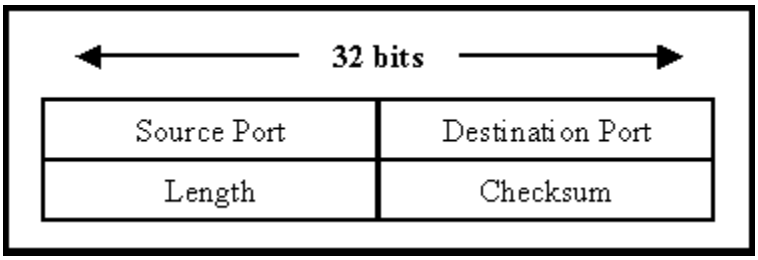


Figure 1 - UDP Packet (https://www.techrepublic.com/a/hub/i/2015/06/03/59570453-0987-11e5-940f-14feb5cc3d2a/r00220010702mul01_01.gif)

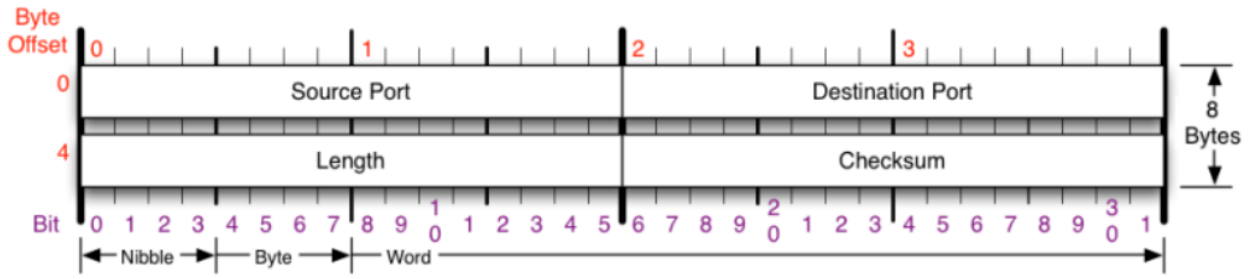


Figure 2 - UDP Packet Measurements

Composition of UDP packets consists of four fields:

- **Source Port** and **Destination Port -** length of 16 bits each. They identify the end points of the connection.
- **Length field** - that has 16 bits in size specifies the length of the header and data.
- **Checksum** - field that has 16 bits in size has a purpose to allow packet integrity checking, however this field is optional.

UDP works by taking the message received from the layers above it on the OSI model and then formats that message to UDP packets. Then the sending applications sends the packets to a peer application on to the receiving host. UDP does not require any notification of receipt and does not use three-way handshake as TCP protocol. (Mullins, 2021)

Transmission Control Protocol (TCP)

Same as UDP protocol, TCP is a transport layer protocol. The main difference to UDP that TCP is connection-oriented protocol that provides full duplex, acknowledgment, and flow-controlled service to upper layer protocols. TCP works by moving data in a continuous and unstructured byte stream. With the help of sequence numbers, it identifies within that stream. TCP also has the capability to support several synchronized upper layer communications.

TCP protocol shines in situations where reliability is required. It also provides stream-oriented connections. TCP nearly always operates in full duplex mode where two independent byte streams are travelling in opposite directions. TCP uses segments to identify if the receiving host is ready to receive the data.

Mechanisms that are incorporated in TCP packets are very complex, these mechanisms helps to ensure connection state and reliability of data packets. These mechanisms are:

- **Round trip time estimations** – TCP protocol constantly monitors the exchange of data packets. It can approximate the duration of how long it would take to receive acknowledgment and automatically retransmit if this time is exceeded.
- **Flow control** – TCP manages the rate at which data packets travel, meaning this buffer never overflow fast senders and will be stopped periodically to keep up with slower receivers.
- **Streams** – TCP data is structured as a stream of bytes.
- **Reliable delivery** – Sequence numbering helps to identify which data has been transmitted and received. TCP will make retransmission if it decides that any of the packets were missing.

- **Network adaptation** – TCP has the skill to learn the delay qualities of a network and adjust its operation to increase throughput.

To establish reliable connection TCP protocol uses a three-way handshake. Before establishing connection, TCP sends a segment called SYN to the peer with who it needs to establish a connection. The receiving TCP returns a segment called ACK. ACK acknowledges the successful acceptance of the segment. Then, the sending TCP client sends another ACK segment and then continues to send the data. Below is an illustration of a three-way handshake. (Mullins, 2021)

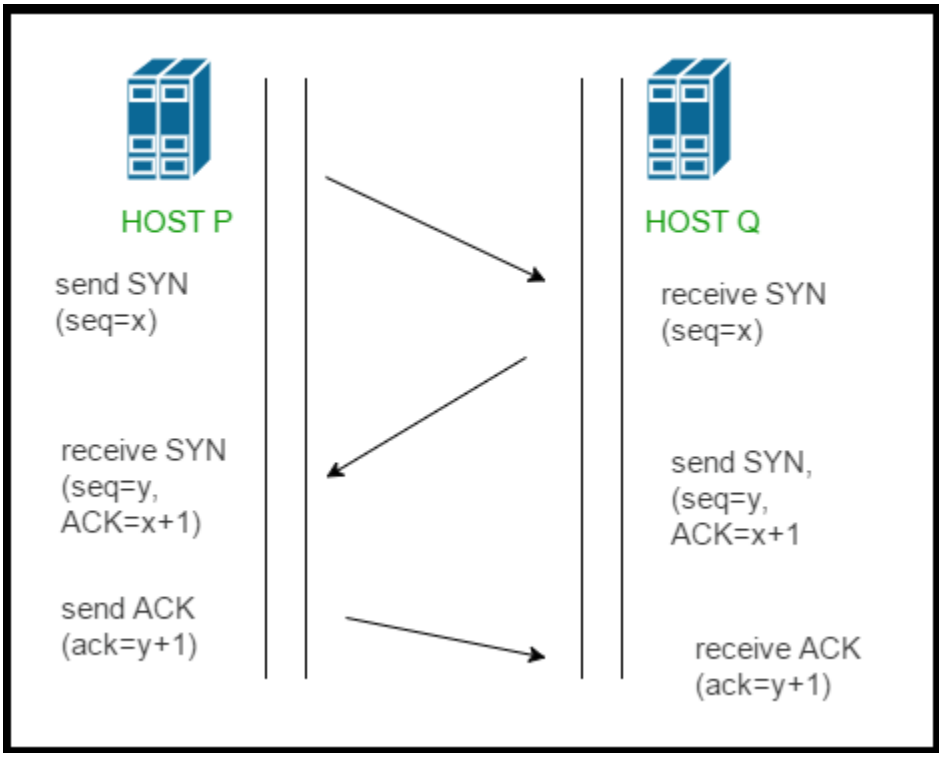


Figure 3 - 3-way handshake (https://media.geeksforgeeks.org/wp-content/uploads/TCP-connection-1.png)

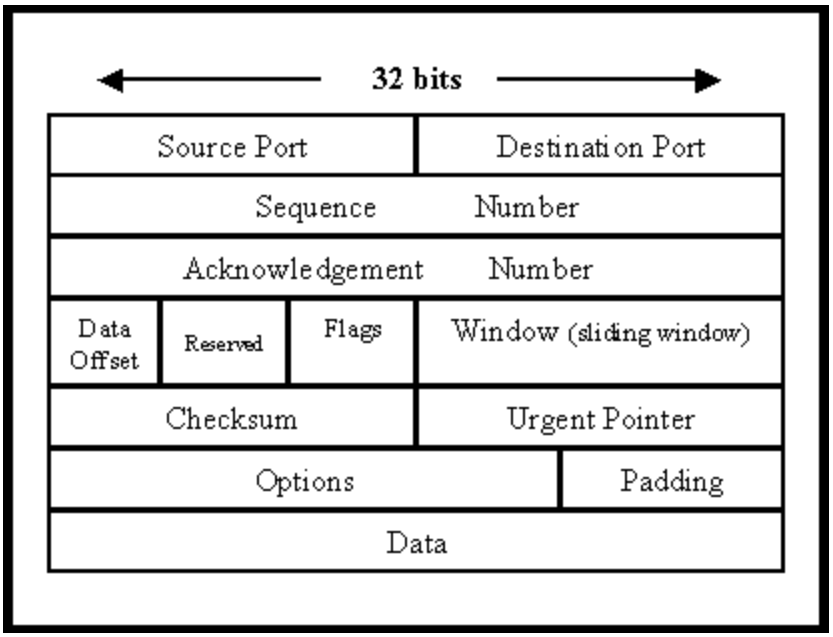


Figure 4 - TCP packet (https://www.techrepublic.com/a/hub/i/2015/06/03/596ecee7-0987-11e5-940f-14feb5cc3d2a/r00220010702mul01_02.gif)

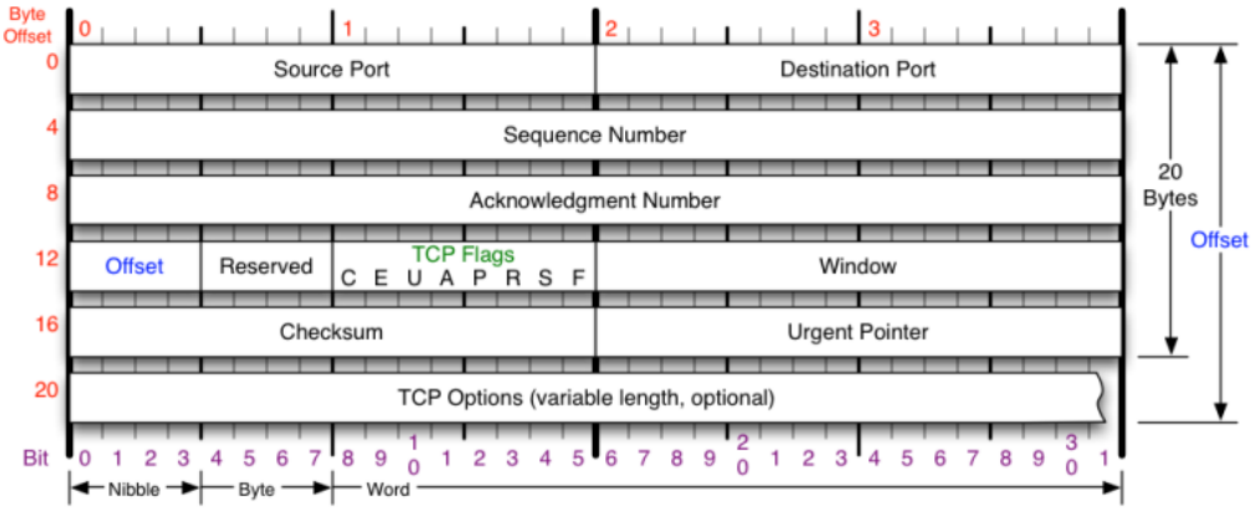


Figure 5 - TCP Packet Measurements

Composition of TCP packets consists of the following fields:

- **Source port and Destination port** – it is 16 bits in size, and it is identifying the end points of the connection.
- **Sequence number field** – it is 32 bits in size and its purpose is to specify the number assigned to the first byte of data in the current message.
- **Acknowledgment number field** – 32 bits in size. It contains a value of the next sequence number that the sender of the segment is expecting to receive.
- **Data offset** – this field is variable length. It tells how many 32-bit words are present in TCP header.
- **Reserved field** – it is 6 bits in length, now it must be set to zero as it is reserved for future use.

- **Flags field** – it is 6 bits in size, and it does contain various flags:
    - **URG** – indicates that data is urgent.
    - **ACK** – Indicates that acknowledgment number is valid.
    - **PSH** - Indicates that data received should be passed to application as soon as possible.
    - **RST** – Resets the connection.
    - **SYN** – Synchronizes the sequences numbers to start connection.
    - **FIN** – Means that the sender of the flag has finished sending data.
- **Window field** – it is 16 bits in size and specifies the size of the senders receive window, this is a buffer space available for incoming data.
- **Checksum field** – 16 bits in size indicates if the header was damaged in transit.
- **Urgent pointer field** – 16 bits in size, it is a pointer to the first urgent data byte in the packet.
- **Options field** – variable length meant for various TCP options.
- **Data field** – variable size contains information for upper layers.

Internet Protocol (IP)

Internet Protocol (IP) stands in layer 3 of OSI model. IP protocol purpose is to provide fragmentation and resemblance of datagrams and error reporting. In combination with TCP protocol, IP represents the core of the internet protocol suite. IP protocol works by attaching an IP header to the packets header in addition to the information added by TCP or UDP protocols. Information that is included within IP header is IP addresses of the sender and receiver, datagram length and sequence number of datagram. This information is provided in case the datagram exceeds the allowable byte size for network packets and must be segmented. (Mullins, 2021)

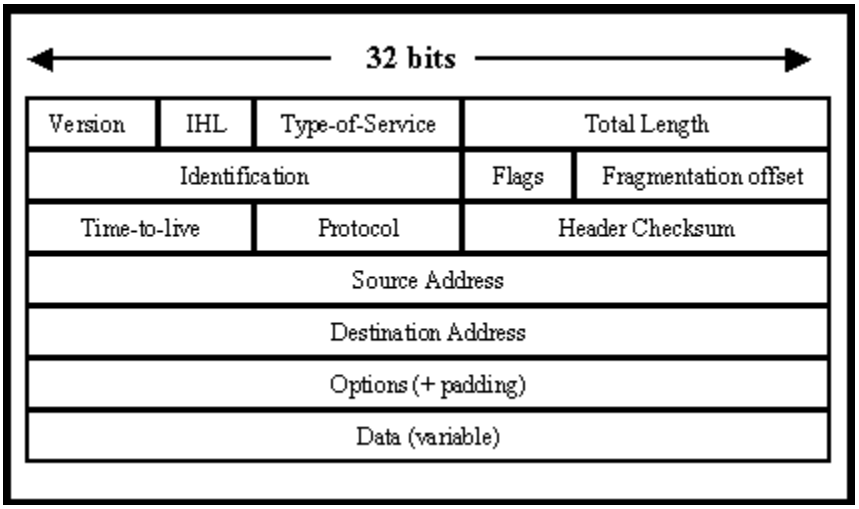


Figure 6 - IP Packet (https://www.techrepublic.com/a/hub/i/2015/06/03/59843f0d-0987-11e5-940f-14feb5cc3d2a/r00220010702mul01_03.gif)

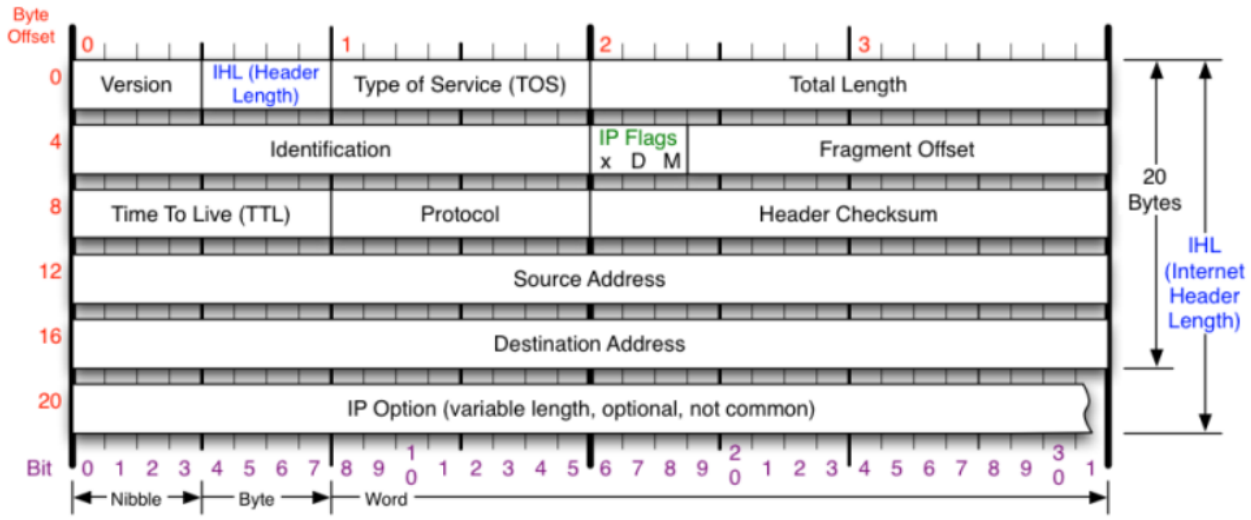


Figure 7- IP Packet Measurements

Composition IP packet consists of the following fields:

- **Version field** – 4 bits in length, contains the information of IP currently used.
- **IP Header Length (IHL)** – 4 bits in length has an information of how many 32-bit words are in the IP header.
- **Type of service field** – 8 bits in length specifies of how upper layer protocol wants data to be handled. Some datagrams can have various levels of importance assigned to it within this field.
- **Total length field** – 16 bits in length. Its purpose is to specify total length of IP packet, including data and header in bytes.
- **Identification field** – it is 16 bits in size and it contains a number that distinguishes the current datagram. This field has a usage of help to reconstruct fragments.
- **Flags field** – 4 bits in length controls if routers are allowed to fragment a packet and denotes the parts of a packet to receiver.
- **Time-to-live (TTL) field** – 8 bits in length, it is a counter field that is decremented gradually to zero. Datagrams are discarded when zero value is reached, prevents of endless loops within network.
- **Protocol field** – 8 bits in length, indicates which upper layer protocol receives incoming packets after IP processing is completed.
- **Header checksum field** – 16 bits in length, helps to ensure integrity.
- **Source address field** – 32 bits in length, specifies sending node.
- **Destination address field** – 32 bits in length, specifies receiving node.
- **Options field** – 32 bits in length, allows to support various options, such as security.
- **Data field** – 32 bits in length, contains information for upper layers.

Internet Control Message Protocol (ICMP)

Internet Control Message Protocol (ICMP) is an error reporting protocol that network devices such as routers use to generate error messages to the source IP address when network problems prevent delivery of IP packets. ICMP has the capability to create and send messages to senders IP address indicating that a gateway to the internet cannot be reached for packet delivery. All the IP network equipment's has the capability to send, process or receive ICMP messages. ICMP is also used in network diagnostics, specifically ping, MTR, and traceroute terminal applications. (Lutkevich, 2021)

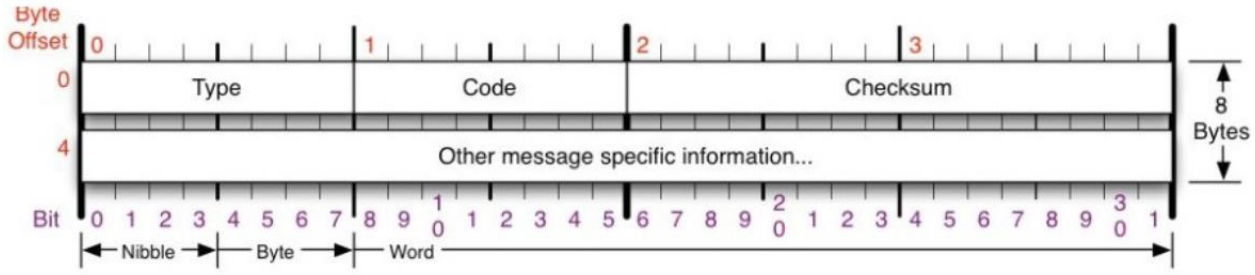


Figure 8 - ICMP Packet Measurements

Composition of ICMP packet below:

- **Type** – 8 bits in length, some common messages types are:
  - **Type 0** – echo reply
  - **Type 3** – destination unreachable
  - **Type 8** – echo
  - **Type 5** – redirect
- **Code** – 8 bits in length, contains message type code thus providing additional information about the error type.
- **Checksum** – 16 bits in length, provides integrity.

Packet Analyzer

Packet Analyzer Overview

Main objective of packet analyzers is to observe network assets and to identify abnormal behavior and misuse. Network packet sniffing is a process of capturing and monitoring data packets that are passed through the network. Network analyzers could be in two forms, as a software or a hardware, and it is used to analyze network traffic by capturing packets. Network analyzers or sniffers works by examining network streams of data packets that flows between computers on a network as well as between networked computers and the internet. Network packets are addressed to specific devices but if packet sniffer is used in monitoring mode, it allows for examination of any packets regardless the destination. Network sniffers could be configured in two ways. One includes unfiltered traffic where all packets are captured and stored for later investigation, and the second mode will enable to set filtered condition on which traffic will be captured i.e., it could be destination IP, protocol, source IP and many more conditions. (Kumar, 2021)

Packet sniffer could be used in both types of networks, in a wired or wireless. Their effectiveness depends on the network security protocols on how much they are able to gather. As an example, on wired network, packet sniffers would be able to gather packets on every connected device or it might be limited by the

placement of network switches were on wireless network, most analyzers can only scan one channel at a time, but ability to use multiple wireless interfaces can expand this capability. (Kaspersky, 2021)

Benefits of Packet Analyzer

Packet sniffing has some benefits related to it. Just to mention the top benefits, these include - detecting root cause of a network issue. This benefit would help network administrators to continuously monitor and track down bottlenecks in a network environment, thus helping to remove the root cause of a problem. The next benefit is troubleshooting network issues. This could help by performing analysis on a network to measure response times or latency issues in a network. It could help in determining the amount of time packets need to travel from destination to source. This analysis could help to identify congested networks, detect applications that produces unreasonable amount of traffic and take remedy measures to resolve the issue. Another benefit is traffic analysis that helps to determine peeks in network demand over time. Advanced packet sniffers can categorize the data based on multiple criteria. The next benefit is bandwidth management. Analyzing traffic, using packet sniffers, can help by preventing business from slow network operations. Taking this approach network administrators can identify critical applications that requires more bandwidth and even restrict certain applications. Network security and compliance is one of the bigger benefits related to packet sniffers. Thus, bad actors can infiltrate and use business resources for their gains, so packet sniffers can help identify any suspicious behavior on the network. Some more advanced futures of packets sniffers related to security are that it could help to identify surge in traffic, attempts at network intrusion and enable deeper evaluation and mitigation of security threats. (Anon., 2020)

Sniffing Around the Hub and in a Switched Network

The most common network used today is a switched network. Thus, the visibility window for an a packet sniffer on a network is limited to the port on which sniffer is plugged in.

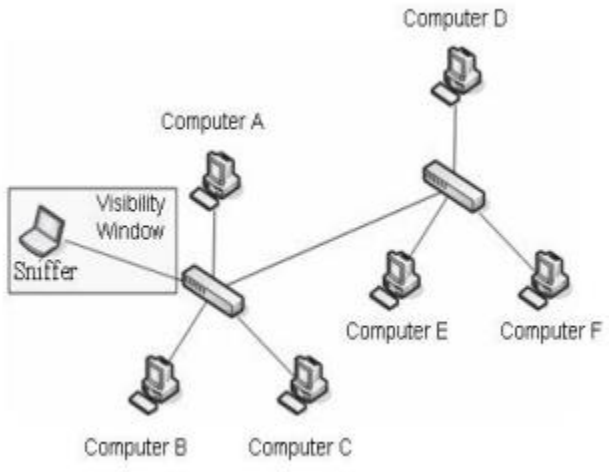


Figure 9 - Visibility of a Sniffer on a Switched Network

There is a technique related to switched network to expand the visibility window. This technique is called port mirroring. It works by copying traffic from one port to another thus expanding the visibility window of packet sniffer.

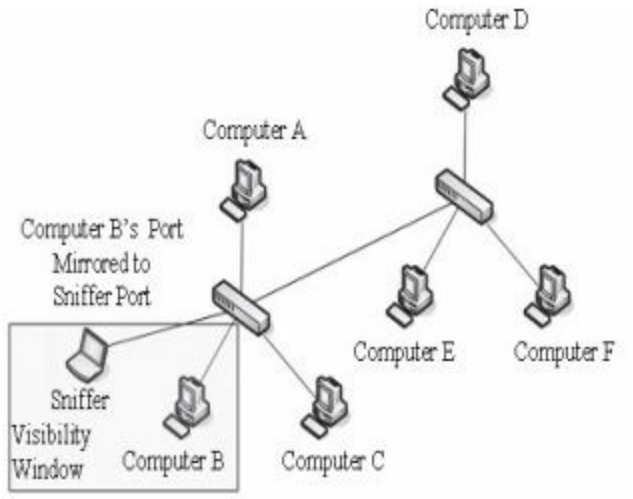


Figure 10 - Port Mirroring on a Switched Network

Sniffing around a network that has hubs installed expands visibility window to all networks. As traffic that is sent from devices flows freely around all networks it allows packet sniffer to capture all traffic and analyze it. Sniffing on a hub-based network provides limitless visibility window. (Patel, et al., 2009)

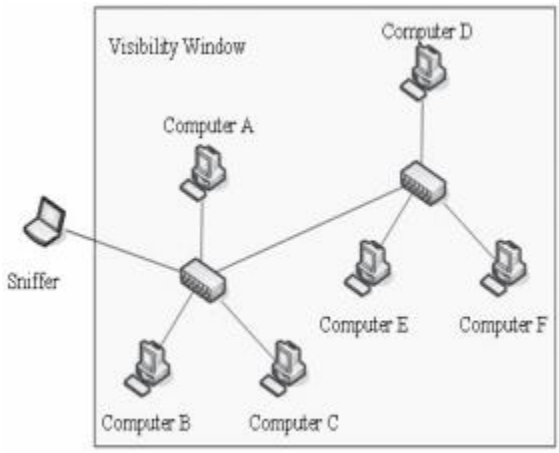


Figure 11 - Visibility Window in a Hub Network

Methods of Packet Sniffing

There are three types of packet sniffing methods. Some methods have the capability to work in a non-switched networks while other methods will work in switched networks. The following bullet points outline methods of packet sniffing:

- **IP Based sniffing** – this way is an original way of packet sniffing. It works by putting network interface card into promiscuous mode and gathering all packets matching the IP address in filter. By default, the IP address filter it is not set so it will capture all the packets. This method only works in non-switched networks.
- **MAC based sniffing** – Works the same way as IP based sniffing just in a filter MAC address is specified.

- **ARP based sniffing** – Any traffic that is meant for a particular IP address would be mistakenly sent to an attacker, then the attacker can choose to forward the traffic to the actual default gateway or modify the data before forwarding, thus performing man in the middle attack. There is also the possibility to launch denial of service attack by not forwarding pockets at all. (Patel, et al., 2009)

Use Cases of a Packet Sniffer

When we hear the phrase 'packet sniffer' we usually think about black hat hackers. However, packet sniffers have plenty of legitimate uses. The following are some typical legitimate uses for packet sniffers:

- **Asset Discovery/Passive Reconnaissance** – Packets include their source and destination IP addresses; thus packet capture can be used to discover active endpoints on a given network. With decent amount of data, it is possible to fingerprint the endpoints. When this use case is performed for legitimate reasons, this is called inventory or discovery. Thus, this technique is heavily used by bad actors for gathering the information needed for follow up stages of an attack.
- **Troubleshooting** – Inspecting network traffic might be the most effective troubleshooting step that could help to narrow down the root cause of a problem. Thus, packet sniffers allow network engineers and administrators to view the contents of packets traversing the network. This capability is highly used when troubleshooting foundational network protocols. While troubleshooting, sniffing traffic can verify that network packets are taking the correct path across the network. A broken or congested network is easy to spot using packet sniffers because only one side of a two-sided communication will be present. Thus, connections with large amount of retries or dropped packets often indicate broken network devices.
- **Intrusion Detection** – Network traffic that is suspicious can be fed into an intrusion detection system for further analysis. Some aspects that might be spotted with packet sniffers include known malicious IP addresses and telltale payloads. Even somewhat like a DNS request repeat if repeated at a regular interval.
- **Incident response and Forensics** – Packet sniffers have an ability to spot signs of an incident that most other tools miss. Anything that was sent across the network can't be unsend, thus allowing packet sniffers to identify endpoints from where traffic was sent. This capability can aid in investigations allowing to trace from where packets originated exposing bad actors at the other end. (Grimmick, 2021)

Targeted Audience

Packet sniffers are used by network managers, network administrators, network engineers, penetration testers and bad actors. This list can go on longer, as students in college use packet sniffers to learn more about networks and protocols related to networking. The targeted audience is very broad for such a tool, as it could be used in many places from troubleshooting network issues to aid for intrusion detection systems. It can also be used for asset discovery, incident response and forensics. Broad range of IT professionals could use packet analyzers to help with upcoming tasks such as inspecting network traffic or simply identifying assets on a network. Therefore, it could be used for asset discovery bad actors can use this tool for reconnaissance identifying assets on a network thus planning further attacks.

Existing Network Analysis Tools

WireShark

WireShark is the worlds most popular tool that analyzes network protocols. It has many futures that I will write about in detail, and it is able to run on most computing platforms that includes Windows, MacOS, Linux and UNIX. WireShark is heavily used by network professionals, security experts, developers and educators. This tool is freely available as an open-source tool and it is released under General Public License (GNU) version 2. WireShark was developed by a global team of protocol experts who look after it and maintain this tool. Is also an example of a disruptive technology. Formerly, WireShark was called Ethereal. It got its name know now as WireShark in 2006. It was renamed from Ethereal due to trademark issues. (Banerjee, et al., 2010)

Wireshark's main use is for capturing, viewing and analysis of network packets. It also has advanced tools to help administrators to troubleshoot wireless networks combined with appropriate drivers and wireless network card. It could be set to promiscuous mode to capture packets from the air and decode it into a format that helps network engineers to track down issues that are causing poor performance, connectivity issues and other common problems. (Banerjee, et al., 2010)

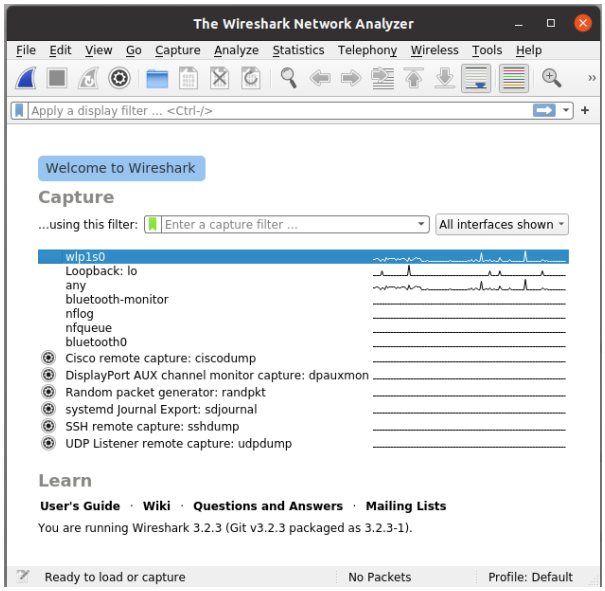
Sniffing tools provided by WireShark are very simple to setup. When we are starting packet capture in WireShark network, the interface card is configured in promiscuous mode and waits until the needed amount of traffic has been captured. Wireshark's capability of network sniffing can expand through wired and wireless networks covering many topologies. It provides capability to capture all packets travelling through all networks through a desired interface. Capture tool of WireShark is considered one of the primary tools. (Banerjee, et al., 2010)

Logging tools provided by WireShark have an amazing flexibility over other intrusion detection system devices related to log maintenance. Log files can be captured at an hourly or weekly rate based on requirement of the network. It gives opportunity to catch logs over fast nodes and later on to transfer them to a slower database. WireShark also allows captured log files to export to various formats including plain text, postscript, CSV and many more. (Banerjee, et al., 2010)

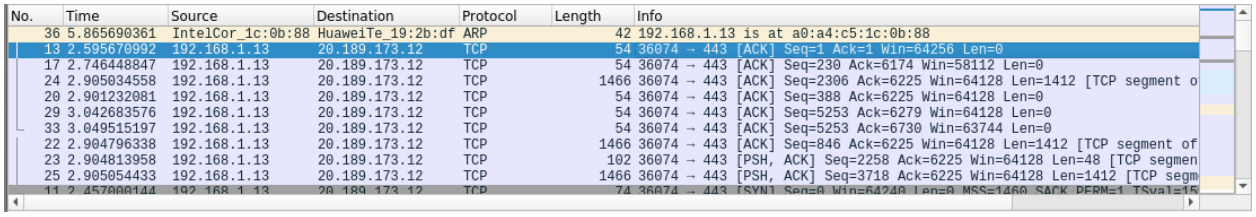
Filtering while capturing capability of WireShark allows to capture packets based on condition that may include source address, destination address, protocol used and values of fields. It also provides capability to filter already captured packets based on the same conditions. Thus, it can help a lot in analysis of packets were also WireShark has capability to decode packets into various modes such as the ASCII. (Banerjee, et al., 2010)

Below are the example usage and functionality of Wireshark. To note, on Linux platform I had to use elevated privileges to be able to capture any traffic:

- Allows choice of interfaces to be sniffed on



- When sniffing mode is active, at the top frame Wireshark displays basic information about the packets that includes number of a packet, time, source IP, destination IP, protocol used, length of a packet and information.



- Middle frame displays more detailed information from OSI model layers two, three and four. Information included relat6es to frame, ethernet, network layer and transport layer.

```
Frame 13: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface wlp1s0, id 0
    Interface id: 0 (wlp1s0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Nov  2, 2021 10:02:55.056803982 GMT
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1635847375.056803982 seconds
    [Time delta from previous captured frame: 0.000064572 seconds]
    [Time delta from previous displayed frame: 0.000064572 seconds]
    [Time since reference or first frame: 2.595670992 seconds]
    Frame Number: 13
    Frame Length: 54 bytes (432 bits)
    Capture Length: 54 bytes (432 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp]
    [Coloring Rule Name: TCP]
    [Coloring Rule String: tcp]
Ethernet II, Src: IntelCor_1c:0b:88 (a0:a4:c5:1c:0b:88), Dst: HuaweiTe_19:2b:df (e4:fb:5d:19:2b:df)
    Destination: HuaweiTe_19:2b:df (e4:fb:5d:19:2b:df)
    Source: IntelCor_1c:0b:88 (a0:a4:c5:1c:0b:88)
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.1.13, Dst: 20.189.173.12
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0x72a8 (29352)
    Flags: 0x4000, Don't fragment
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0x44a9 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.13
    Destination: 20.189.173.12
Transmission Control Protocol, Src Port: 36074, Dst Port: 443, Seq: 1, Ack: 1, Len: 0
    Source Port: 36074
    Destination Port: 443
    [Stream index: 3]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    Sequence number (raw): 3118722451
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Acknowledgment number (raw): 361536131
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x010 (ACK)
    Window size value: 502
    [Calculated window size: 64256]
    [Window size scaling factor: 128]
    Checksum: 0x4433 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
    [SEQ/ACK analysis]
    [Timestamps]
```

- Bottom frame displays information in hexadecimal and ASCII formats.

```
0000  e4 fb 5d 19 2b df a0 a4  c5 1c 0b 88 08 00 45 00   ··]·+··· ·····E·
0010  00 28 72 b3 40 00 40 06  44 9e c0 a8 01 0d 14 bd   ·(r·@·@· D·····
0020  ad 0c 8c ea 01 bb b9 e4  02 17 15 8c b3 09 50 10   ········ ·····P·
0030  01 f5 17 2a 00 00                                  ···*··
```

- Various filtering techniques can be used within Wireshark tool. Below are few examples using filtering in Wireshark.

arp

No.	Time	Source	Destination	Protocol	Length	Info
36	5.865690361	IntelCor_1c:0b:88	HuaweiTe_19:2b:df	ARP	42	192.168.1.13 is at a0:a4:c5:1c:0b:88
37	6.449390292	Espressi_e0:13:8c	Broadcast	ARP	42	ARP Announcement for 192.168.1.10
53	8.395046235	Espressi_e7:9c:e5	Broadcast	ARP	42	ARP Announcement for 192.168.1.6
48	6.657976829	Espressi_e0:11:9e	Broadcast	ARP	42	ARP Announcement for 192.168.1.7
38	6.451574739	HuaweiTe_19:2b:df	Broadcast	ARP	60	Who has 192.168.1.10? Tell 192.168.1.1
35	5.865664476	HuaweiTe_19:2b:df	IntelCor_1c:0b:88	ARP	42	Who has 192.168.1.13? Tell 192.168.1.1
54	8.397372120	HuaweiTe_19:2b:df	Broadcast	ARP	60	Who has 192.168.1.6? Tell 192.168.1.1
49	6.659608040	HuaweiTe_19:2b:df	Broadcast	ARP	60	Who has 192.168.1.7? Tell 192.168.1.1

ip.src == 20.189.173.12

No.	Time	Source	Destination	Protocol	Length	Info
27	3.042248208	20.189.173.12	192.168.1.13	TCP	60	443 → 36074 [ACK] Seq=6225 Ack=2306 Win=525056 Len=0
30	3.042778096	20.189.173.12	192.168.1.13	TCP	60	443 → 36074 [ACK] Seq=6279 Ack=3718 Win=525056 Len=0
31	3.043530044	20.189.173.12	192.168.1.13	TCP	60	443 → 36074 [ACK] Seq=6279 Ack=5253 Win=525056 Len=0
12	2.595606420	20.189.173.12	192.168.1.13	TCP	66	443 → 36074 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MS
28	3.042641786	20.189.173.12	192.168.1.13	TLSv1.2	108	Application Data
32	3.049490678	20.189.173.12	192.168.1.13	TLSv1.2	505	Application Data
19	2.901186120	20.189.173.12	192.168.1.13	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
16	2.746389235	20.189.173.12	192.168.1.13	TLSv1.2	6227	Server Hello, Certificate, Certificate Status, Server

frame.len == 102

No.	Time	Source	Destination	Protocol	Length	Info
23	2.904813958	192.168.1.13	20.189.173.12	TCP	102	36074 → 443 [PSH, ACK] Seq=2258 Ack=6225 Win=64128 Len=48 [TCP segment c

- Ability to save captures to .pcap file and open with other tools for further analysis. Wireshark also has ability to open .pcap files that were captured with other tools

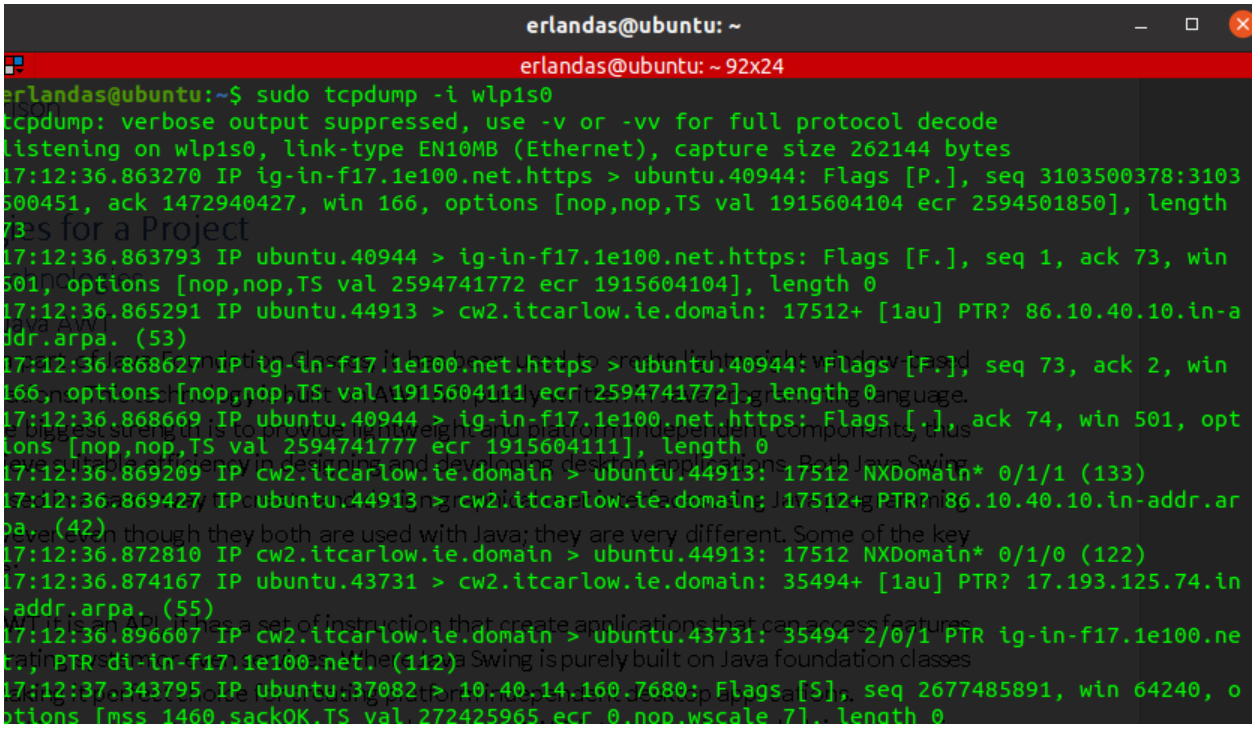
```
erlandas@ubuntu:~/Desktop$ tcpdump -nn -r capture.pcap
reading from file capture.pcap, link-type EN10MB (Ethernet)
10:02:52.461132 IP 192.168.1.6.49154 > 255.255.255.255.6667: UDP, length 188
10:02:52.517338 IP 192.168.1.13.49840 > 52.114.92.111.443: Flags [P.], seq 3417415642:3417415699, ack 326594530, win 501, length 57
10:02:52.535061 IP 52.114.92.111.443 > 192.168.1.13.49840: Flags [P.], seq 1:47, ack 57, win 2050, length 46
10:02:52.535102 IP 192.168.1.13.49840 > 52.114.92.111.443: Flags [.], ack 47, win 501, length 0
10:02:53.310017 IP 192.168.1.13.41381 > 52.114.92.93.443: Flags [P.], seq 647478372:647478429, ack 3694203088, win 501, length 57
10:02:53.329062 IP 52.114.92.93.443 > 192.168.1.13.41381: Flags [P.], seq 1:47, ack 57, win 2049, length 46
10:02:53.329134 IP 192.168.1.13.41381 > 52.114.92.93.443: Flags [.], ack 47, win 501, length 0
10:02:53.335336 IP 52.108.196.24.443 > 192.168.1.13.55138: Flags [P.], seq 2342594227:2342594260, ack 1015681343, win 16413, length
33
```

Tcpdump

Tcpdump is a command line tool. It primarily displays and intercepts TCP/IP and other packets that are transmitted or received over a network to which device it is connected. TCP dump works on most UNIX-like operating systems. There is a port of Tcpdump on Windows called Windump. Tcpdump works primarily by analyzing network behavior, performance and applications that generates or receive network traffic. Tcpdump can also be used to analyze the network infrastructure by determining whether all necessary routing is occurring properly, allowing network engineers to further isolate problems within network. It is also possible to use Tcpdump for intercepting and displaying the communication of another user or device. Tcpdump is limited to report on only what it finds. Further, if any data is forged within packet such as source address Tcpdump can't identify that. Also, it has steep learning curve as not everyone is able to use command line tools. (Asrodia & Patel, 2012)

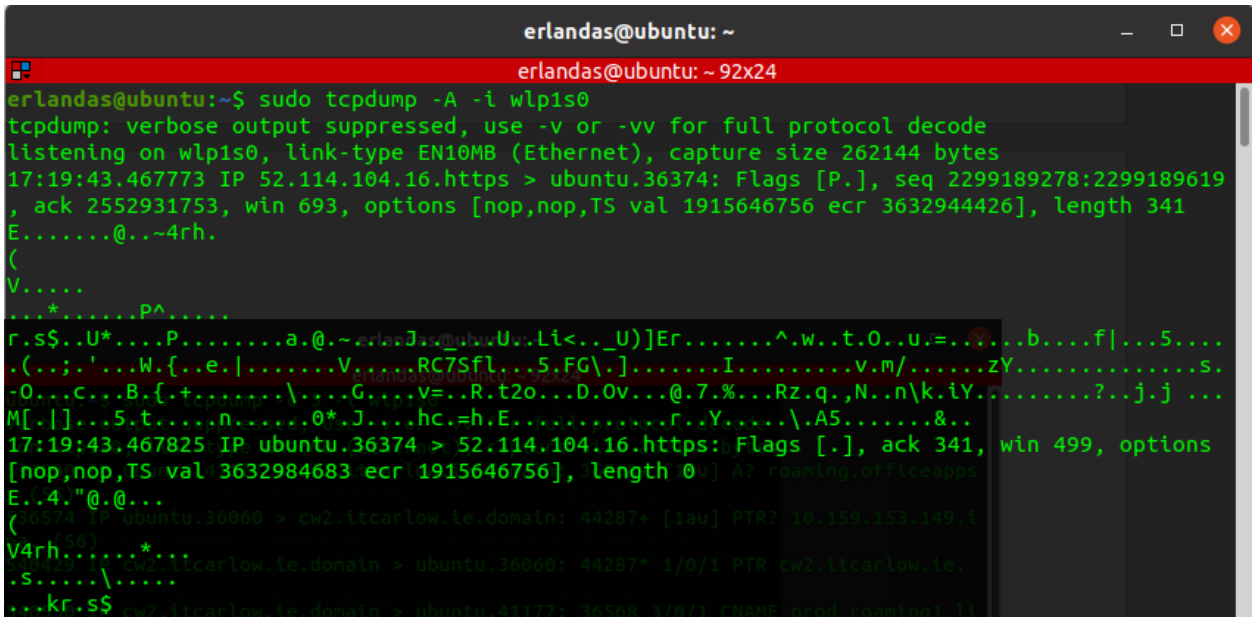
Examples usage of Tcpdump below with a description. Just to note privileges must be escalated using "sudo" in front of some commands:

- Capture packets from specific interface, command used "**tcpdump -i wlp1s0**"

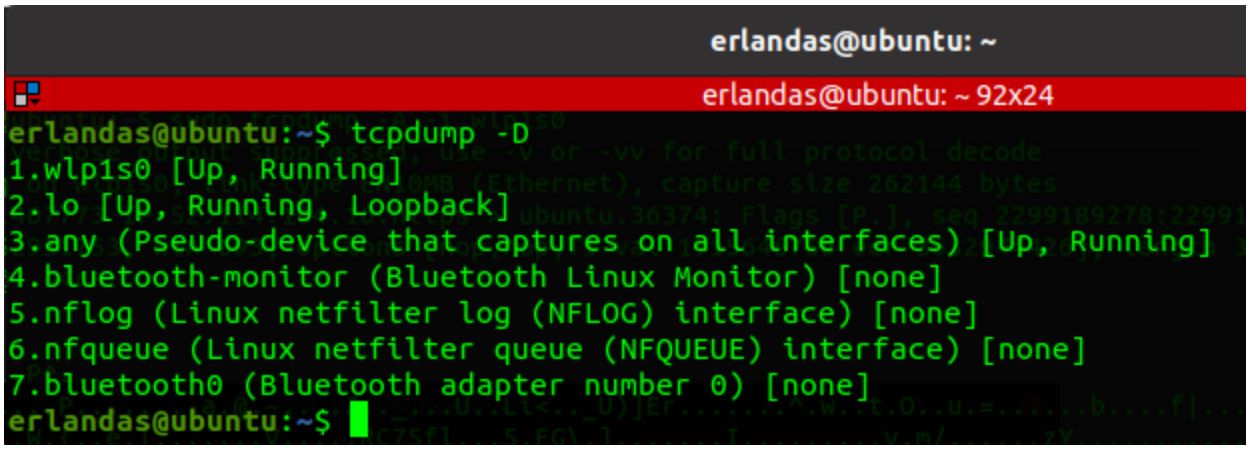


- Capture only N number of packets, command used “tcpdump -c 5 -i wlp1s0”

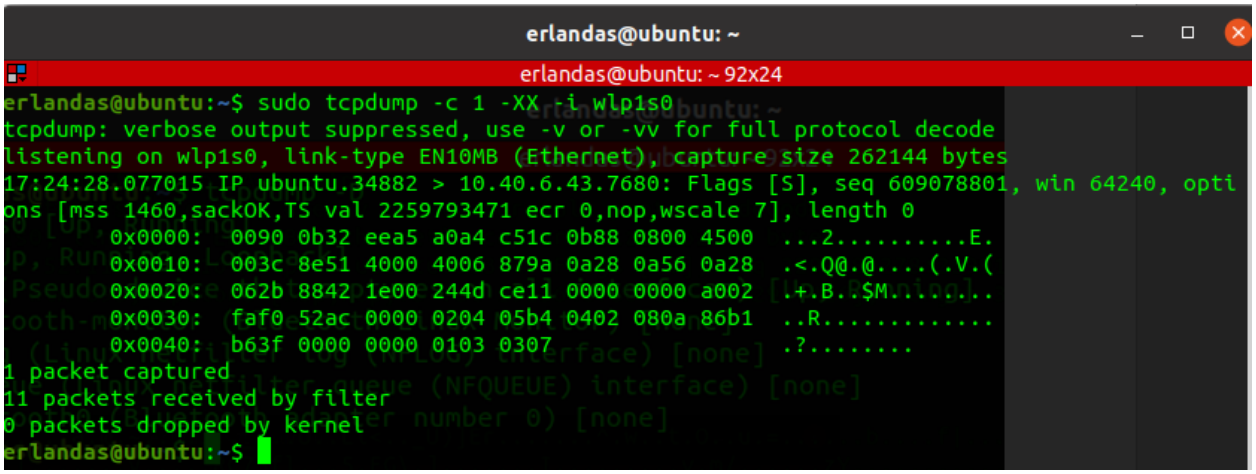
- Print captured packets in ASCII, command used “tcpdump -A -i wlp1s0”



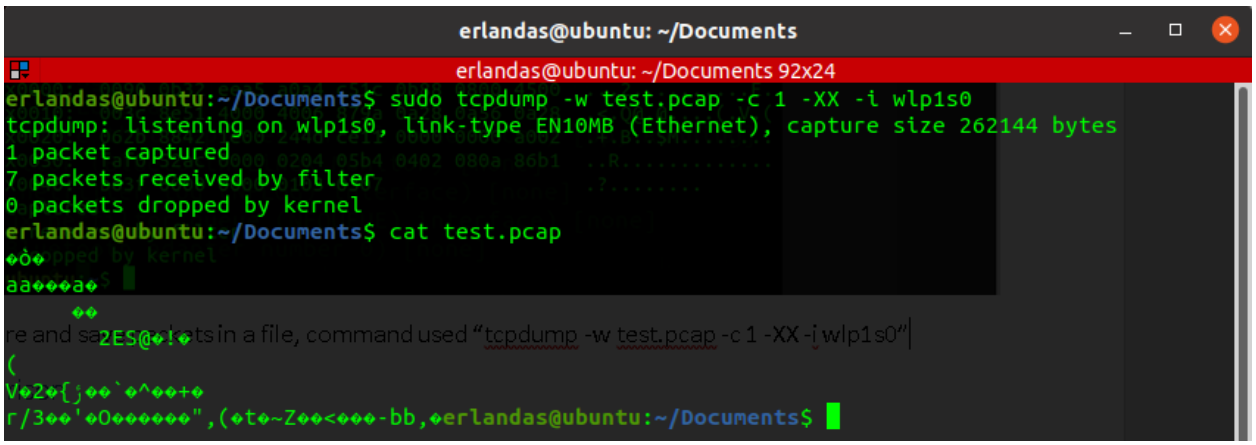
- Display available interfaces, command used “tcpdump -D”



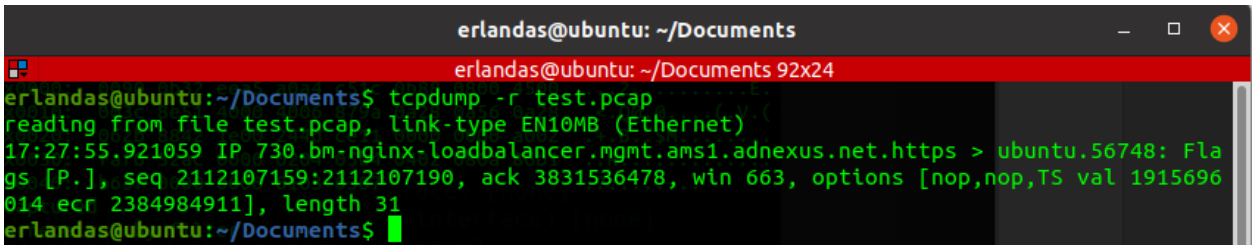
- Display captured packets in HEX and ASCII, command used “tcpdump -c 1 -XX -i wlp1s0”



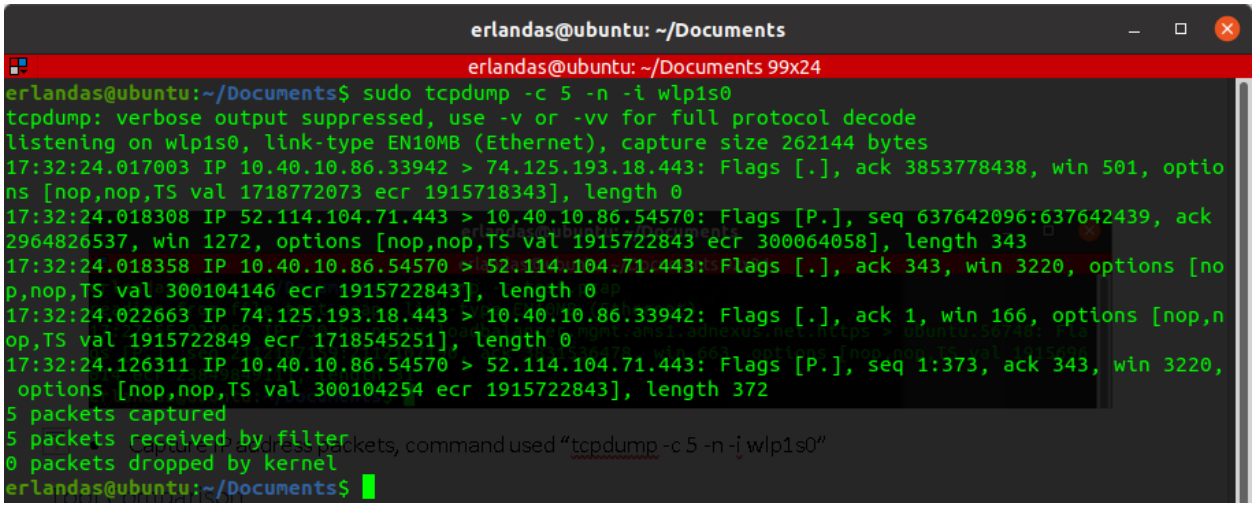
- Capture and save packets in a file, command used “tcpdump -w test.pcap -c 1 -XX -i wlp1s0”



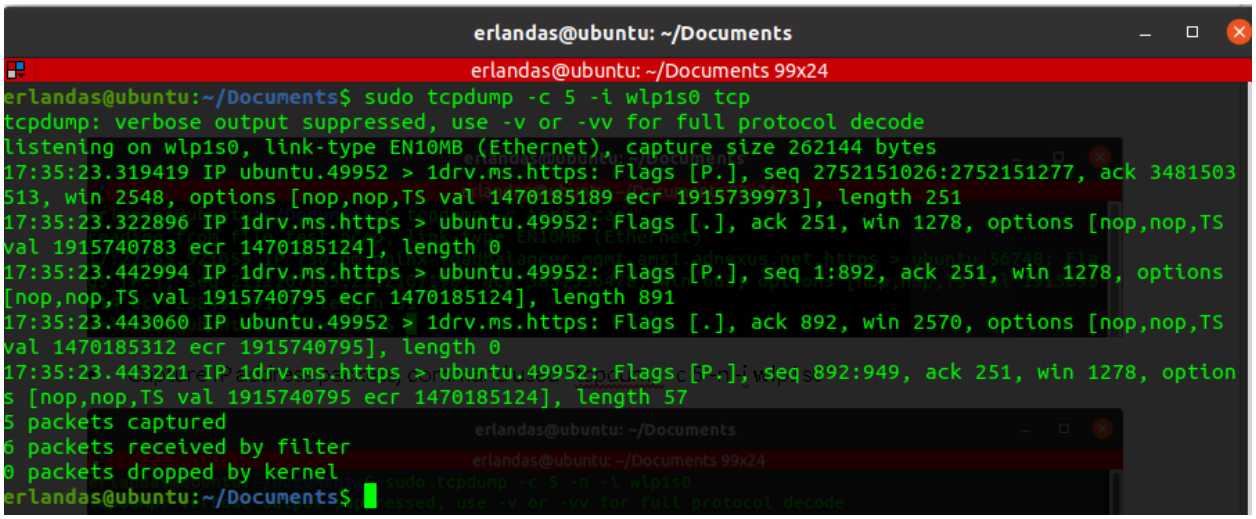
- Read captured packet file, command used “tcpdump -r test.pcap”



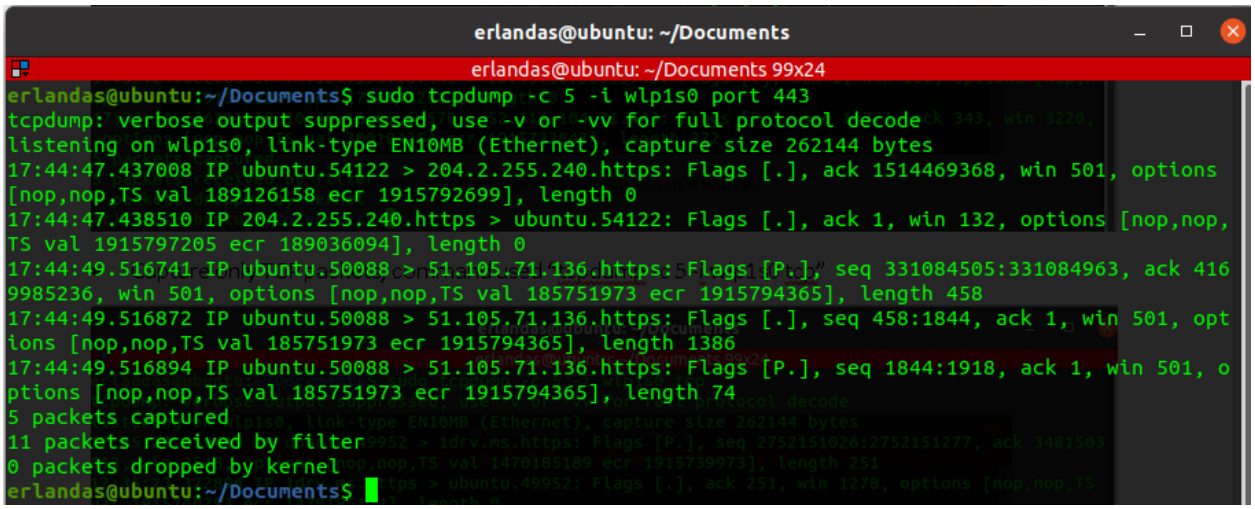
- Capture IP address packets, command used “tcpdump -c 5 -n -i wlp1s0”



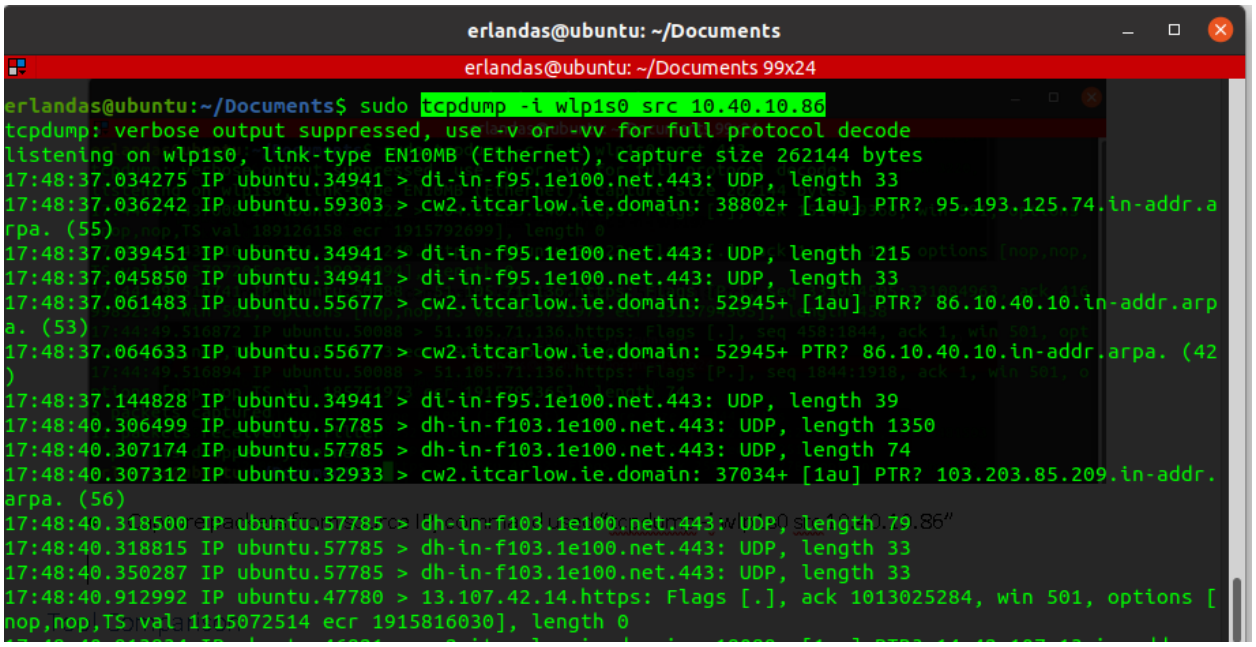
- Capture only TCP packets, command used “tcpdump -c 5 -i wlp1s0 tcp”



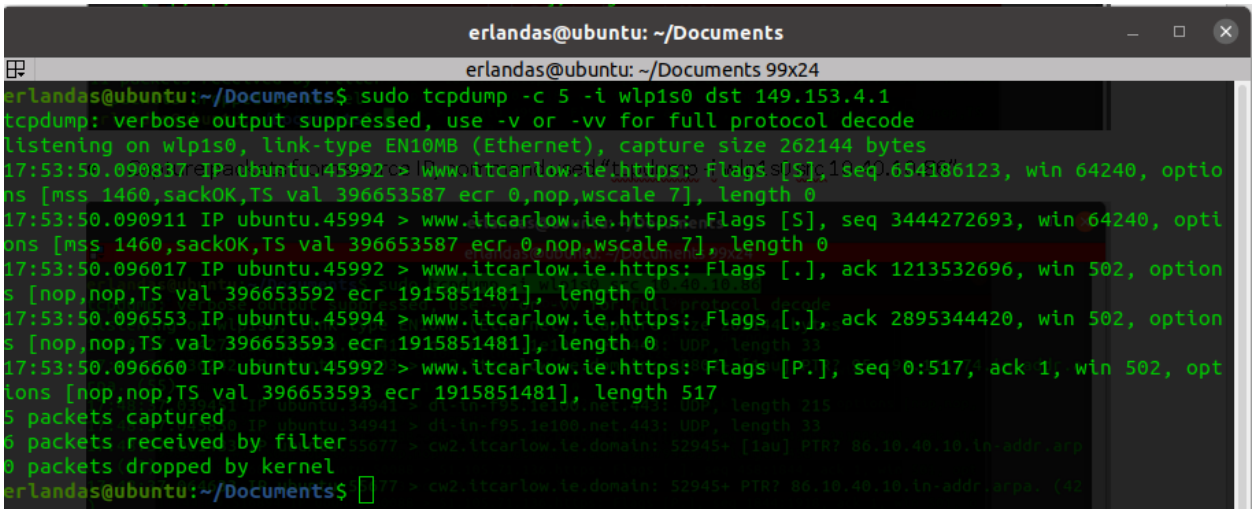
- Capture packets from specific port, command used “tcpdump -c 5 -i wlp1s0 port 443”



- Capture packets from source IP, command used “tcpdump -i wlp1s0 src 10.40.10.86”



- Capture packets from destination IP, command used “tcpdump -c 5 -i wlp1s0 dst 149.153.4.1”



Technologies for a Project

Front End Technologies

Java Swing & Java AWT

Java Swing is a part of Java Foundation Classes, it has been used to create lightweight window-based desktop applications. This technology is built on AWT API and purely written in Java programming language. Java swing has the biggest strength to provide lightweight and platform independent components, thus helping to achieve suitable efficiency in designing and developing desktop applications. Both Java Swing and AWT are used in the same way to create and design graphical user interfaces using Java programming language. However even though they both are used with Java; they are very different. Some of the key differences are:

- Java AWT - it is an API. It has a set of instructions that create applications, that can access features of an operating system or even services. Where Java Swing is purely built on Java foundation classes, thus making it a perfect choice for creating platform independent desktop applications.
- Java AWT components are very resource heavy as AWT interacts with underlying operating system (OS), to compare with Java Swing. Swing components are very lightweight as swing uses inbuilt java classes. Also, Swing is more powerful compared to AWT and has more functionality.
- Applications that are created using Java Swing are much faster compared to AWT counterparts. AWT has longer execution time compared to Swing.
- While Java Swing is platform independent, AWT relies on underlying operating system, making it a less portable choice for developers as AWT is an API framework which makes it platform dependent.
- Java Swing compared to Java AWT is very easy to learn. (Joseph, 2021)

To conclude the comparison related to Java AWT and Java Swing. Even though both technologies seem to look alike, as they are made on the grounds of Java programming language. They are very different and after looking at both, Java Swing is superior compared to Java AWT, not only because of portability but

also in execution speed. Some benefit related to Java Swing is that it has less of a learning curve as this technology is a little easier to get the hang of it.

Qt

QT is a C++ framework that has bindings to other programming languages such as: Java, Python, Go, PHP, Ruby and many more. It is mainly used for developing graphical user interfaces and cross platform applications for desktops and embedded systems. The Qt framework can function on different types of software and hardware. At the beginning, Qt was created as a class library but over time, the license underwent many changes. There is choice for users for a license - a commercial paid license and a free open-source license that I have interest in. Qt offers special software development language called QML which is highly implemented in projects that focuses on user interfaces. Qt has its own IDE, and it is known as Qt Creator. Using Qt developers always adds extra functionality through add-ons that are distributed through Qt creator's library. However, the free version doesn't come with add-on support.

Qt framework is formed of Qt vital modules, that are listed below:

Qt Essentials Modules	Features
Qt Core	These non-graphic core classes were not required by all other modules related to Qt framework
Qt GUI	Basic classes used in graphical user interface design, also includes OpenGL
Qt Multimedia	A set of QML types and C++ classes to process multimedia
Qt Multimedia Widgets	Classes that are based on widgets that implements multimedia features
Qt Network	Application programming interface for applications that rely on TCP/IP networks
Qt QML	Framework and types of Qt QML markup language
Qt Quick	Framework that enables of creation of dynamic applications with a custom QML user interface
Qt Quick Controls 2	Efficient QML types that simplifies the interface formulation
Qt Quick Dialogs	Types that enable interaction with system dialogs
Qt Quick Layouts	QML types that helps to set layout of objects that are present in the development interface
Qt Quick Test	Test framework for QML applications
Qt SQL	Classes that enable SQL database integration
Qt Test	Classes for detailed tests of Qt applications and libraries
Qt Widgets	Collection of user interface elements for classic user interface design

Table 3 - QT Modules (https://www.sam-solutions.com/blog/qt-framework/)

Qt framework is the number one choice for many developers. Some reasons why developers are choosing Qt over other frameworks include:

- Qt projects are written in a nice way making it easy to read
- Applications created with Qt frameworks has a very attractive user interface
- Cost friendliness of Qt attracts many starting developers that have constrains related to funds. Cost friendliness comes from Qt's simplicity making applications in quick manner.
- Supports application programming interfaces for easier development
- Has support for many libraries and enable high level of controls

- Qt is a portable framework allowing possibility to work on different operating systems and even works on embedded systems
- Qt framework has needed tools to create 3D graphics, it could be used in games development
- Applications created with Qt framework inherits a look from its host operating system making it native looking applications
- It has a huge choice of modules available to it, thus allowing rich functionality of created applications
- Usage of QML within Qt applications can build high performance applications in visual characteristics and animations. Thus, allowing to create dynamic content using Qt with QML

Qt framework runs on multiple platforms allowing deployment for desktop, mobile and embedded operating systems. There is no need to write different code for 32-bit or 64-bit operating systems. All what developers need, is the correct compiler and renderer. Below are lists of supported platforms and operating systems related to platforms mentioned:

- Desktop platforms: Linux, MacOS, Windows
- Mobile platforms: Android, iOS/tvOS/watchOS, Universal Windows Platform (UWP)
- Embedded platforms: Linux, QNX, INTEGRITY, VxWorks

Qt framework is a popular choice for projects that are focused on creating on embedded devices and Internet-of-Things (IoT) software. It has been also used in development of desktop and mobile applications. Great example where Qt was used is KDE Plasma desktop environment for Linux operating systems that I think it's one of the best looking desktop environments for Linux. Qt is adopted and in used by such companies as LG, Tesla, Microsoft, Samsung, BMW, Siemens, HP, Philips and many more. Some examples of Qt software are Adobe Photoshop, VLC, Telegram, BitTorrent client, Adobe Muse, Teamspeak.

In conclusion Qt framework is a very powerful tool that has a good record between industry giants as Siemens, Microsoft, LG and many more. It is simple to use, and it gets work done well. It is also in use by many companies that concentrates on IoT devices, embedded systems and desktop applications. Qt is lightweight, has bindings to many popular programming languages and it is portable tool. It is also a time saving solution for graphical user interface development. (Shapel, 2021)

Tkinter

Tkinter is a way to create graphical user interfaces using Python. It is included in all Python standard distributions. Tkinter provides interface to the Tk toolkit, and it works as object-oriented layer of Tk. It is a multi-platform collection of graphical elements called widgets. Tkinter provides its user with a simple way to create graphical elements using widgets implemented within Tk toolkit. To create applications using Tkinter we have to import the module itself, create the main window that includes the desired number of widgets and lastly applying event triggers to desired widgets. (Active State, 2021)

How to start with Tkinter is illustrated in the below as an example and explanations are listed below:

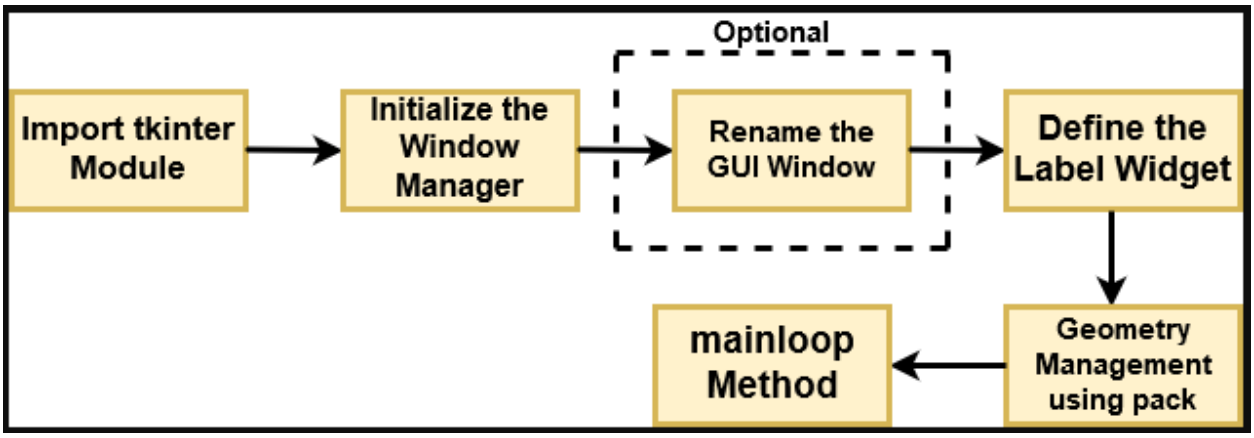


Figure 12 - Tkinter flow chart (https://res.cloudinary.com/dyd911kmh/image/upload/f_auto,q_auto:best/v1575996243/gui3_lwuckb.png)

- The very first step would be to import Tkinter module. To note, it is included with Python 3 by default so no installation required.
- Following the previous step, we have to initialize the window manager using "Tkinter.Tk()" method and assign it to a variable. Tk method is used to create an empty window that contains minimize button, maximize button, and close button.
- As an optional step we could rename title bar for the window. We can do this by using the following method "window.title(name of new title)".
- The next step while following flow chart would be to add a label widget that is used to insert some text.
- Then using geometry management with a pack method that displays the widget created in required size.
- The very last step "mainloop()" method must be called, it is responsible to run application within infinity loop while some condition is met to exit the application. (Sharma, 2019)

In conclusion, Tkinter is used for small projects. Most of the information I have researched I found only how to use it for small projects that would include simple text boxes to display, calculators, button clicks etc. However, Tkinter looks like a very attractive option with its simplicity and portability through different platforms. It removes overhead of installing it, as it is included within Python 3 distribution. However, I won't be too keen to use it as there are other tools that are used within industry and there is no better way to get exposure to these tools while working on a project.

Xamarin

Xamarin is an open-source platform that enables developers to build applications for android, iOS and Windows platforms using .NET. Xamarin works as an intermediary layer that manages communication of shared code with a platform. Xamarin has included memory management and garbage collection thus developers don't have to implement these themselves. This technology is highly used within enterprise environment. It is used by over 15,000 companies in fields such as healthcare, energy, transport, etc. Applications using Xamarin can be developed using Windows computer or MacOS computer, thus eliminating Linux environment to work as it is owned by Microsoft since 2016. It supports only one programming language C#. So, to be proficient using Xamarin platform developers must know strongly C# language. Developers must understand mobile development that includes how different platforms work,

how to use controllers, how to store and retrieve files from the device, and how to use all of the features that are present in a smartphone. Examples of applications that are created using Xamarin is a: Storyo, Skulls of the Shogun, APX, The World Bank, and SuperGiant Games.

Benefits of using Xamarin for development include:

- Xamarin is a single technology stack that is created with Visual Studio. Applications are developed using a single programming language. It utilizes C# and shared codebase. Using Xamarin and allowing it to handle all cross-platform implementation will save on development time.
- Developers using Xamarin have access to a cross platform user interface toolkit therefore helping to create interfaces that would work on multiple devices.
- Xamarin enables access of every native API, hence allowing of usage of native user interface and components. Applications created with Xamarin won't have any differences from native applications on a device.
- As Xamarin uses shared code base consequently eliminating time that would be spent translating applications to different platforms. This benefit speeds up time to market.
- There is less maintenance needed as Xamarin allows developers to commit updates of an application directly.
- Applications can be created simultaneously using Xamarin for different platforms, if let's say you have great desktop application and you want to create mobile port of it and vice versa.

Downfalls of Xamarin is that it is very expensive for enterprises to use it. It has a complication layer of using open-source libraries attached to it, so it doesn't support third party libraries for Android and iOS without extras needed. As a technology it is not suitable for applications that requires a lot of graphics and also applications that are heavier in size, as Xamarin tends to add up to five megabytes of extra weight to it including close to twenty megabytes of debug builds.

Xamarin has a built-in different emulators that enables development for various mobile platforms. It offers a couple of different debugging options with plenty of freedom for developers. Key features why developers tend to choose Xamarin platform to work with is performance related to applications as it offers 14% faster loading time of such applications created with Xamarin. Using this platform, development speed boost is gained due to its ability of using native elements and standard interfaces. Sharable code base allows to develop on one platform and run on the other making this tool quick and efficient if developing for multiple platforms. (Madeshvaran, 2019)

In conclusion, Xamarin is a very powerful tool. It has many great features that developers could get very much attracted to. However, it has some limitations related to platforms it can develop for, thus including Windows, Android and iOS. This downfall eliminates the possibility to create applications using Linux platform. Therefore it won't be my choice of technology to work with on this project as I will be creating a desktop application that is able to run on Windows and Linux.

Back End Technologies

Python

In summary Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, blended with dynamic typing and dynamic binding, making it a very attractive choice for quick application development. It is also highly used as scripting or glue language that unites existing components together. Python proposes simple and easy to learn syntax that emphasizes on readability and consequently reduces the cost of program maintenance. Python also highly supports packages and modules which promotes program modularity and code reusability. The Python interpreter is available in a source or binary form without any charge for all the platforms and it is freely distributed. Many of the programmers fall in love with Pythons due to its increased productivity. Even debugging in Python is made simple, as a small bug or bad input would never cause a segmentation fault. Instead of crashing, a program exception is raised. If by any chance the program doesn't raise exception stack, a trace is printed to console allowing developer to track down the issue. (Anon., 2021)

Currently Python is one of the most popular programming languages in the world. It is used in everything from machine learning to building websites and software testing. Python has a small learning curve and can be used by developers and not developers alike. Python is commonly used for developing websites, task automation, data analysis and data visualization. As Python is relatively easy to learn, it has been adopted by many non-programmers such as accountants and scientists for many everyday tasks needed in their field like organizing finances.

As Python is used in many areas, below is a more detailed look at these areas where Python is used for:

Data analysis and machine learning (ML) – Python has the ability of allowing data scientists and other professionals to use this language to perform complicated statistical calculations, create machine learning algorithms, manipulate and analyze data, create data visualizations, and to complete other data related tasks. Python is able to create broad range of various data visualizations such as line or bar graphs, pie charts, histograms, and 3D plots. Python also enables developers to implement already existing libraries for machine learning and data analysis, making coding time shorter.

Web development – Very often Python is used to develop the back end of web applications. It has roles in web development that includes sending data to or from servers, processing data and communicating with databases, URL routing and guaranteeing security. Python has many frameworks that helps with web development, two of most popular ones is a Django and Flask.

Automation or scripts – if there are any tasks that are done on a daily basis, Python allows of creation of scripts that can perform repetitive tasks. As an example, Downloads/ directory gets filled in quickly, we can create Python script that organizes files by type and date and stores in folders dependent on type and time downloaded.

Software testing and prototyping – In a development environment, Python can help in tasks like build controls, bug tracking, and testing. With Python software developers can automate testing tasks for new products. Few products that are used for software testing is Green and Requestion.

Everyday tasks – Python could also be used to perform everyday tasks such as: keeping track of stock markets, send yourself text reminders, update shopping lists, renaming large batches of files, converting file formats and many more tasks that can be implemented using Python.

As python is general purpose language, it is used between many of fields and industries. Below are just a couple of job titles listed that may use Python:

- Developer
- Data analyst
- Data scientist
- Ethical hacker/penetration tester
- Software engineer
- Data journalist
- Cloud architect
- QA engineer
- … n

In conclusion there are many reasons why Python became such a popular language between developers and non-developers alike. As this language is very versatile and it can be used for many different tasks from web development to machine learning, it has a simple syntax that mimics natural language so it is easier to read and understand. It is also very a beginner friendly language making it a perfect choice for entry level roles. As Python is an open source it is free and everyone can get started with it. Python offers large collection of modules and libraries that expands Pythons capabilities. Also, this language has a large and active community. And there are many opportunities for employment after having Python on your resume, as Python is adapted by many worldwide organizations, and it is sought after language with a very bright feature.

Java

Java is a multi-platform, general purpose, class based, object-oriented programming language designed for having lesser implementation dependencies. Java is a computing platform for application development, it is a fast, secure, and reliable choice. It has many uses but to highlight main uses is developing for mobile platforms, scientific supercomputers, game consoles, data centers and java applications for computers. Java platform consists of several programs that helps developers to develop and execute Java applications efficiently. It includes an execution engine, a compiler, and a set of libraries in it. In other words, Java is a set of computer software and specifications. Latest version to date is Java SE 15 that appeared on the 15th of September 2020.

The developer who is responsible for Java is James Gosling. He developed Java programming language for Sun Microsystems and Oracle Corporation later acquired it. The original name of Java was OAK. Its main purpose was to handle portable devices and set-top boxes, however OAK was a massive failure at the time. In year 1995, Sun changed its name to Java and heavily modified the language. Since year 2009 Oracle owns Java programming language.

Below are some important Java applications:

- Java is used for server-side technologies like Apache, JBoss, GlassFish, etc.
- Java programming of hardware devices.
- Used for big data analytics.
- Scientific computing applications.
- Wide range of mobile Java applications.

- It has ability to create enterprise software.
- It is used for developing Android applications.

Important features related to java programming language is that it is one of the easier to use programming languages and easy to learn. You can say Java is coded once and can run on many platforms. Java is platform independent language, so it doesn't really matter if an application is created on a UNIX environment, it will be run on other OS as easy as it did on UNIX system. Java by design is an object-oriented programming language aiding in code reusability. It has memory management built in taking overhead from developers. It was created mainly for distributed environment of the internet. Also, it facilitates distributed computing as it is network centric language.

Java development kit is needed as it is a software development environment that is used for making applets and Java applications. JDK can be used on the main operating systems, it helps them to code and run Java programs. Some reasons to use JDK is that it contains all the required tools to write Java applications and also has a Java Runtime Environment (JRE) to execute them.

With java language also comes Java Virtual Machine (JVM). It is an engine that provides runtime environment to run the Java applications. JVM has a main task to convert Java code into byte code thus allowing to run applications on different processors. Some important reasons to use JVm is that it provides a platform independent way of executing java code, it has many libraries, tools and frameworks. JVM comes with just-in-time (JIT) compiler that converts Java source code into low level machine language, helping to run applications faster.

Another component of Java programming language is Java Runtime Environment (JRE). It contains class libraries, loader class, and JVM. Main reasons for using JRE is that it contains class libraries, JVM, and other supporting files that would aid in running Java applications. It uses many important classes and runtime libraries. (Hartman, 2021)

In conclusion, Java has many good benefits including where it is used. It has platform independent benefit meaning applications developed will be platform independent and once created it would be multiplatform applications. By statistic times it is recorded as the second top programming language. This language seems very appealing to use and master it, however I didn't find any use of networking components related to Java as it doesn't have much use in networking programming as needed for the project that I am working on at the moment.

C++

C++ is an object-oriented programming language that was created by a computer scientist Bjorne Stroustrop as part of continuing of C language. C++ seen the day light in 1980's at the Bell Labs. Originally it was developed as a multi-platform improvement of C language. C++ was intent for developers to provide higher level of controls over memory management and system resources. Over the years C++ is still a very useful language not only as programming language but also as teaching tool of object-oriented principle for new developers and students. C++ shines not only as object-oriented language but also as procedural and functional language. It is highly flexible and scalable. It has been used in development of various applications including, computer software, operating systems, graphical user interfaces, games, and browsers. As C++ language is very close to C language, it has the ability to be used for low level programming due to its close relation to machine code.

Some disadvantages of C++ as that it has very hard to gasp concepts related to C programming languages. To be more specific, pointers, is a hard concept to gasp and wrong usage of it can lead to systems crashing. There is no garbage collection implemented by default in C++ thus making it hard to filter out unnecessary data. Another heavy flaw with C++ is the presence of security issues that relates to availability of pointers, friend functions, and global variables. (Buttice, 2021)

C++ language is most used for building large software and applications that run on limited resources. As C++ can directly manipulate hardware that it is running on, so programmers can tune their code to run more efficiently in any environment. Thus, making C++ programming language that can run applications quickly and reliably on multitude on devices. For these reasons C++ is a great choice if building large software that must be fast, can run on limited resources, and reliable in performing critical tasks.

Some examples where C++ is used:

- **Operating systems** – C++ was used for developing operating systems such as MacOS, Windows, Linux and mobile operating systems. At the bottom-line operating systems need to be fast and efficient at managing system resources. C++ fulfills these requirements as it is close to machine code, so developers can adjust even the smallest details to make operating systems fast and efficient.
- **Game development** – C++ is the most used language in game development. It was used to create many games for many different platforms such as personal computers, Nintendo consoles, XBOX, PlayStation, and mobile devices. As gaming demands lots of resources, C++ is highly efficient language, so it helps to optimize the use of resources, helping developers adjust how memory is handled. Even one of the most popular game engine "Unreal" is written in C++.
- **IoT devices** – It includes development for embedded systems that very often rely on C++. These devices are operating with limited resources thus making C++ a popular choice for developers to use for IoT devices. Some examples of development is smart televisions, home appliances, cars, and medical devices.
- **Databases** – This language is used in many of the popular databases including MariaDB and MySQL. These databases are heavily used between many applications. C++ is very helpful for developing databases making them to support efficient storage.
- **Web browsers** – C++ was used for web browser development such as Google Chrome, Mozilla Firefox, and Opera. Main use of C++ for web browsers is back-end technology helping to retrieve data from databases and render code into interactive web pages. As C++ is a very efficient and fast programming languages it helps for web browsers to complete these tasks with ease and making internet browsing quick and efficient.
- **Machine learning tools** – C++ has a huge collection of libraries for calculations that aids in machine learning algorithms and helps them to learn more efficiently. It is used as back-end technology for ML tools. As an example, TensFlow relies on C++ back-end technology.
- **AR/VR applications** – Most of virtual reality and augmented reality applications run using Unreal engine which is developed using C++. To create these applications, we need fast and reliable language thus C++ making perfect candidate for AR/VR technologies.
- **Scientific research** – C++ is used for analyzing scientific data. NASA uses it for building self-driving technologies for Mars Rover to navigate unknown terrain. Most of the space stations are build using C++.

- **Financial tools** – Many of the financial software's are created using C++, it has many applications for banking, trading, and financial modeling. Speed of C++ helps to process huge amounts of financial data such as transactions and trading.
- **Flight software** – This language is used for commercial aircrafts also as for military ones. It is used to build a lot of safety features of airplanes, thus making sure all of the crucial equipment of aircraft operates as expected during flights.
- **Google search engine** – One of the most popular search engines is build using C++, it is not only responsible to gather search queries and display quickly. It also uses machine learning algorithms to present best possible result.
- **Medical technologies** – It is used in many of medical technologies, due to C++ speed and reliability making it a perfect choice for such technologies.
- **Telecommunications** – It is used for building telecommunications infrastructure. It is able to handle many resources simultaneously with speed, efficiency and reliability.
- **Movie production** – Most of graphics and special effects in movies are generated using C++. It is able to handle high quality video files and make calculations needed for special effects. (Xiao, 2021)

In conclusion, C++ is a very powerful language. It is mainly used for crucial software, games, scientific, and large software. However, this language requires a steep learning curve and could be tricky to master. However, if mastered it is rewarding due to its efficiency, reliability and speed.

C#

C sharp (C#) is a general-purpose, object-oriented programming language. It was developed by Microsoft, and it was approved by European Computer Manufacturers Association and International Standards Organization. C# by design is a language that can be used with other languages. Mainly it was developed for .NET applications. As C# was developed by Microsoft it is mainly targeted for development on Windows platform, however there is IDE's that runs C# code on Linux platforms. IDE called for Linux that runs C# is "Monodevelop". As C# has many similarities to other programming languages, especially Java, it is not that hard to code in, as code looks human readable. Some benefits related to C# are:

- **Easy to start** – It is a high level programming language so it has human readable like syntax, it has many similarities to C, C++ and Java. Making it an easy to learn language for existing developers.
- **Widely used for developing desktop and web applications** – C# has been one of the most popular programming languages for developing for desktop applications. If development is for windows platform C# is the best choice.
- **Community** – C# has a large community of developers who adore this language. Therefore, when searching for some solutions it can be as easy as googling the problem and the solution is usually readily available. Also, this language is very well documented by Microsoft.
- **Game development** – Game engine developed for C# integration is called "Unity". C# is a popular choice for game developers as this language is easier to gasp than C++. It has functions such as garbage collection, interfaces, object-oriented, making it one of the most languages for game development.

Some advantages related to C# is that this language is very efficient in managing systems. All the garbage is automatically collected therefore preventing memory leaks. It has smaller costs to maintain C# code

due to its simplicity compared to C++. C# code is compiled using intermediary language, which is standard language, making it independent from running operating system and architecture.

Some disadvantages related to C# is that C# is not as flexible as other languages as it is highly dependent from .NET framework. Also, as C# compared to other languages is a slow language, as code must be compiled each time after changes are made. (Anon., 2019)

In conclusion, C# is a very popular choice for developers having some benefits to it as efficient system management, garbage collection. Prevention of memory leaks is one of the better benefits of using C# as a language of choice. However, applications using C# are mainly developed on Windows platforms. This downfall removes flexibility for me when creating applications using Linux platform. As there is IDE to create and execute C# applications using Linux, I have already tried it and it is buggy, making C# not the best choice for this project.

SQLite vs MySQL

The biggest difference between both database engines is that SQLite is serverless where MySQL requires a server to run, thus applications that use SQLite engine database runs as a part of the application. MySQL supports a lot of data types where SQLite has support only for few data types such as: blob, integer, Null, Text and Real. The SQLite weights just few kilobytes where MySQL requires at least 600 megabytes of storage to operate. Even SQLite has better portability making it perfect for applications that will stay on a device, it is not as scalable as MySQL. In terms of if database expends in size SQLite will have a hard time to deal with the huge amount of data making it a slower option if many users are needed. MySQL has better security features built into it while other database is not as security orientated, SQLite does not require a lot of configurations.

The use of SQLite is recommended for smaller projects and standalone applications if there is a need to read and write directly from the disk. MySQL is recommended for applications that has multiple user access if strong security is needed. Distributed systems would use it if application requires large database. Web based application will use MySQL. (S, 2021)

In conclusion, for the project I am working on SQLite looks like the perfect candidate. The project I am working on will be an application based on local devices making SQLite perfect for such a project. In addition, my plan is to develop an application for windows and Linux platforms, therefore by using SQLite I will reduce the overhead for the users so that they don’t have to install MySQL database as SQLite is created on demand.

Summary

In this research paper I have extensively researched the various tools that had similar functionality to what my project was going to be based on. The tools that I have researched included WireShark and Tcpdump. My research was primarily based on usage of these tools, their popularity and why they are deemed popular, as well as what are the common functionalities of these tools.

I was also able to research in depth the front-end and back-end technologies. Front end technologies included Java Swing, Qt framework, Tkinter and Xamarin. I was able to gain a good understanding of these technologies' areas where they would be used most commonly. I have also researched back-end technologies, such as Phyton, Java, C++, C#, SQLite and MySQL. Some of these technologies I had already known about from previous studies such as Java and Phyton. However, doing additional extensive research gave me great insight into these technologies, including where they are best used, and which technologies are mostly widely used and most popular.

Glossary

AR – Augmented Reality

VR – Virtual Reality

AWT – Abstract Windowing Toolkit

API – Application Programming Interface

QML – Qt modeling Language

IDE – Integrated Development Environment

UWP – Universal Windows Platform

ML – Machine Learning

OSI – Open System Interconnection

UDP – User Datagram Protocol

TCP – Transmission Control Protocol

IP – Internet Protocol

VoIP – Voice over IP

DHCP – Dynamic Host Configuration Protocol

DNS – Domain Name System

RIP – Routing Information Protocol

TFTP – Trivial File Transfer Protocol

ICMP – Internet Control Message Protocol

IoT – Internet of Things

ML – Machine Learning

Tables

Figures

References

Active State, 2021. *Active State.* [Online]
Available at: https://www.activestate.com/resources/quick-reads/what-is-tkinter-used-for-and-how-to-install-it/
[Accessed 1 Nov 2021].

Anon., 2019. *Geeks for geeks.* [Online]
Available at: https://www.geeksforgeeks.org/introduction-to-c-sharp/
[Accessed 31 Oct 2021].

Anon., 2020. *Tek-tools.* [Online]
Available at: https://www.tek-tools.com/network/all-about-packet-sniffers
[Accessed 16 Oct 2021].

Anon., 2021. *Python.* [Online]
Available at: https://www.python.org/doc/essays/blurb/
[Accessed 20 Oct 2021].

Asrodia, P. & Patel, H., 2012. Analysis of Various Packet Sniffing Tools for Network Monitoring and Analysis. *International Journal of Electrical, Electronics and Computer Engineering,* 2277(2626), p. 56.

Banerjee, U., Vashishtha, A. & Saxena, M., 2010. Evaluation of the Capabilities of WireShark as a tool for Intrusion Detection. *International Journal of Computer Applications (0975 – 8887) ,* 6(7), pp. 1, 2, 3, 4.

Buttice, C., 2021. *Techopedia.* [Online]
Available at: https://www.techopedia.com/definition/26184/c-plus-plus-programming-language
[Accessed 31 Oct 2021].

Grimmick, R., 2021. *Varonis.* [Online]
Available at: https://www.varonis.com/blog/packet-capture/#cases
[Accessed 29 Oct 2021].

Hartman, J., 2021. *guru99.* [Online]
Available at: https://www.guru99.com/java-platform.html
[Accessed 30 Oct 2021].

Joseph, A., 2021. *Section.io.* [Online]
Available at: https://www.section.io/engineering-education/introduction-to-java-swing/
[Accessed 19 Oct 2021].

Kaspersky, 2021. *What is a Packet Sniffer?.* [Online]
Available at: https://www.kaspersky.com/resource-center/definitions/what-is-a-packet-sniffer

Kumar, R., 2021. *Sniffing using bettercap in Linux.* [Online]
Available at: https://www.geeksforgeeks.org/sniffing-using-bettercap-in-linux/

Lutkevich, B., 2021. *Tech target.* [Online]
Available at: https://www.techtarget.com/searchnetworking/definition/ICMP
[Accessed 28 Oct 2021].

Madeshvaran, S., 2019. *Medium.* [Online]
Available at: https://medium.com/a-developer-in-making/everything-you-need-to-know-before-starting-xamarin-development-2019-edition-49744616196e
[Accessed 1 Nov 2021].

Mullins, M., 2021. *Tech republic.* [Online]
Available at: https://www.techrepublic.com/article/exploring-the-anatomy-of-a-data-packet/
[Accessed 27 Oct 2021].

Patel, N. P., Patel, R. G. & Patel, D. R., 2009. *Packet Sniffing: Network Wiretapping,* Patiala, India: Computer Engineering Department, Sardar Vallabhbhai National Institute of Technology.

S, E., 2021. *Hostinger.* [Online]
Available at: https://www.hostinger.com/tutorials/sqlite-vs-mysql-whats-the-difference/
[Accessed 1 Nov 2021].

Shapel, M., 2021. *Sam Solutions.* [Online]
Available at: https://www.sam-solutions.com/blog/qt-framework/
[Accessed 19 Oct 2021].

Sharma, A., 2019. *Data camp.* [Online]
Available at: https://www.datacamp.com/community/tutorials/gui-tkinter-python
[Accessed 1 Nov 2021].

Xiao, L., 2021. *codeacademy.* [Online]
Available at: https://www.codecademy.com/resources/blog/what-is-c-plus-plus-used-for/
[Accessed 31 Oct 2021].