Name- Suraj Kumar Giri

Roll – RKE011B50

INT301 | OPEN SOURCE TECHNOLOGIES

To Use a wireshark tool to analyse your network at the microscopic level and investigate at least 10 protocols, read the live data from Bluetooth and USB.

Submitted to – Dr. Manjot Kaur

Wireshark?

Wireshark is an open-source packet analyser, which is used for education, analysis, software development, communication protocol development, and network troubleshooting.

It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyzer, and network analyzer. It is also used by network security engineers to examine security problems.

Wireshark is a free to use application which is used to apprehend the data back and forth. It is often called as a free packet sniffer computer application. It puts the network card into an unselective mode, i.e., to accept all the packets which it receives.

Here are some different protocols we could analyze using Wireshark:

1. HTTP
2. DNS
3. TCP
4. UDP
5. ARP
6. SMTP

7. POP3
8. IMAP
9. FTP
10. Telnet

1) INTRODUCTION

1.1 Objective:

The objective of this project is to gain a deep understanding of the network traffic on a given network. By analyzing packets captured by Wireshark, we will be able to identify various protocols and their behaviors, detect anomalies and suspicious activities, and ultimately improve the security of the network.

1.2 Description:

We will use Wireshark to capture and analyze traffic on a network of our choosing. We will examine at least 10 different protocols and investigate their functions, as well as any potential security risks associated with them. We will also look for patterns in the traffic, such as frequent requests or unusual traffic spikes, and attempt to determine their causes. Additionally, we will explore the use of filters and other features of Wireshark to streamline our analysis and identify key findings.

1.3 Scope of the Project:

The scope of this project is limited to analyzing the traffic on a single network using Wireshark. We will focus on a range of protocols commonly used in networks, including HTTP, DNS, FTP, SMTP, SSH, TCP, UDP, ICMP, ARP, and SNMP. While we will strive to identify any security risks or anomalous activities on the network, this project is not intended to be a comprehensive security audit or penetration test. Rather, our goal is to gain a deeper understanding of network traffic and identify potential areas for further investigation.

2) System Description

2.1 Target System Description: The target system for this project is a network that is accessible through a computer running the Wireshark network analysis tool.

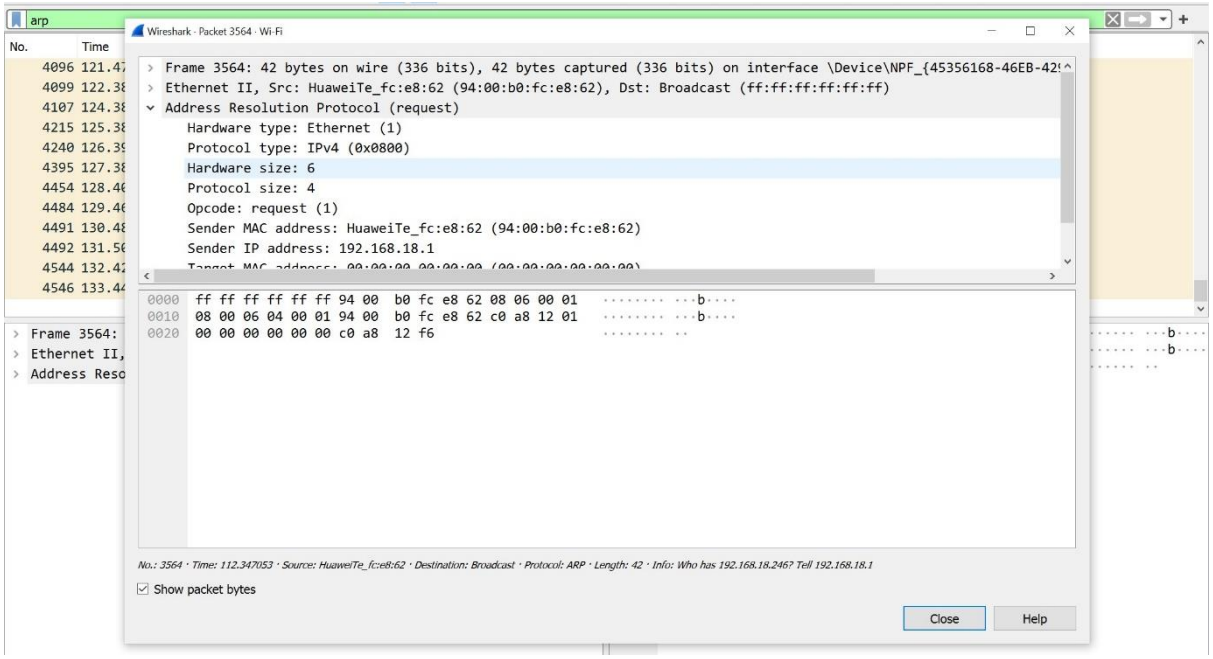
The network can be a local area network (LAN), wide area network (WAN), or any other network that can be accessed by Wireshark.

2.2 Assumptions and Dependencies (if applicable): The assumption for this project is that we have the necessary permissions and access to the target network to capture and analyze traffic using Wireshark. Additionally, we assume that the target network is operating normally and that there are no significant disruptions or outages that would impact our ability to capture traffic. Dependencies include having a computer with Wireshark installed and a network interface that can be used for capturing traffic.

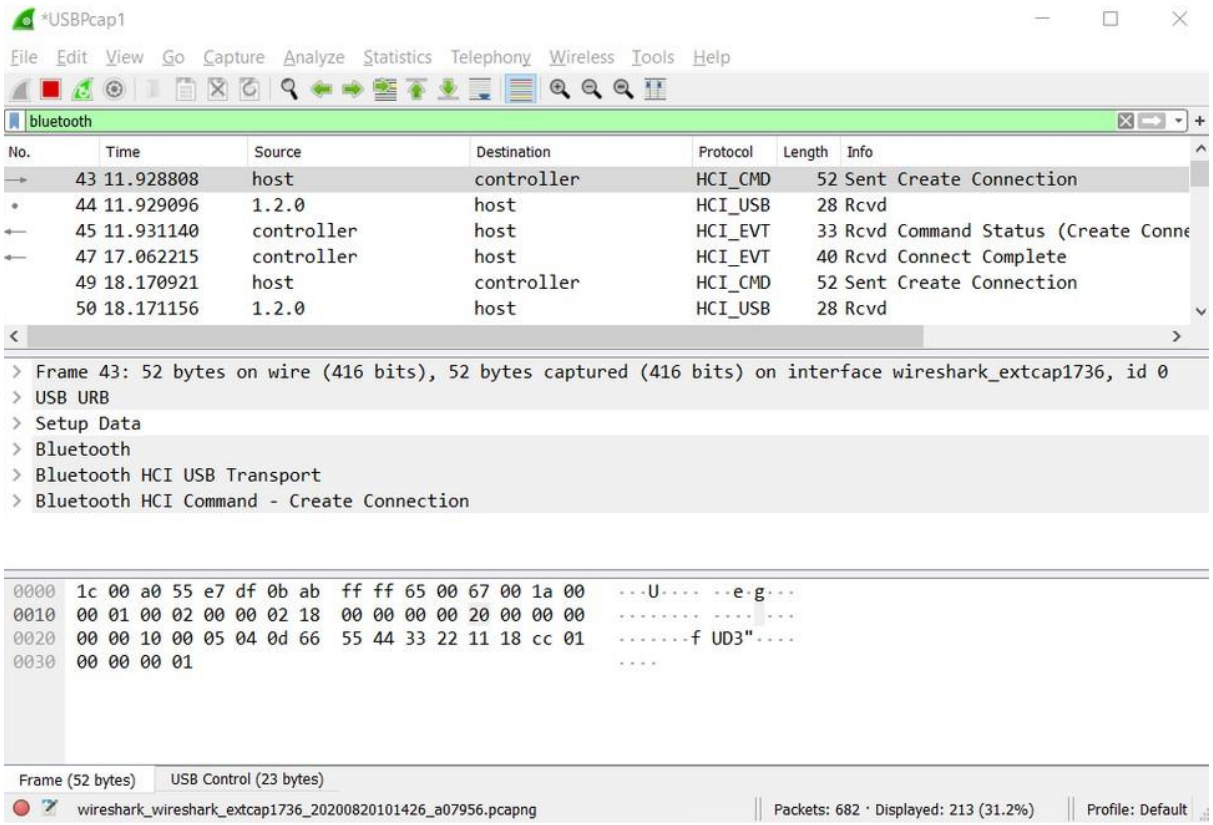
2.3 Functional and Non-Functional Dependencies (if any): The functional dependencies of this project include the ability to capture network traffic using Wireshark and the ability to analyze that traffic to identify protocols and potential security risks. The non-functional dependencies include the speed and capacity of the computer and network interface used for capturing traffic, which may impact the amount of traffic that can be captured and the speed at which it can be analyzed.

2.4 Dataset Used in Support of Project (if any): For this project, we will use a variety of datasets to support our analysis. These datasets will include packets captured by Wireshark during our network analysis, as well as any external datasets that we may use to supplement our findings. We will also use a range of tools and techniques, including filters and visualizations, to help us analyze and interpret these datasets. Additionally, we may use publicly available datasets, such as the Network Forensic Challenge Dataset, to test and validate our findings.

3 Analysis Report (Snapshot)

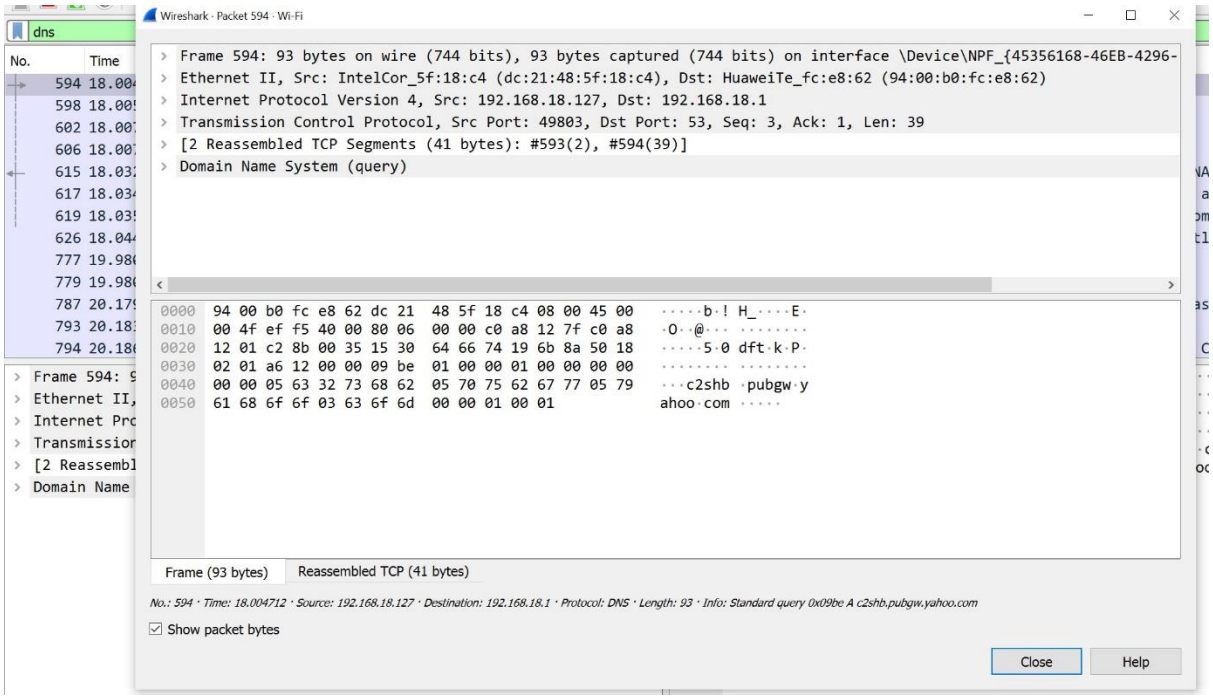


ARP (Address Resolution Protocol): ARP is used to map a network address (such as an IP address) to a physical address (such as a MAC address). ARP analysis can help identify potential network attacks such as ARP spoofing.

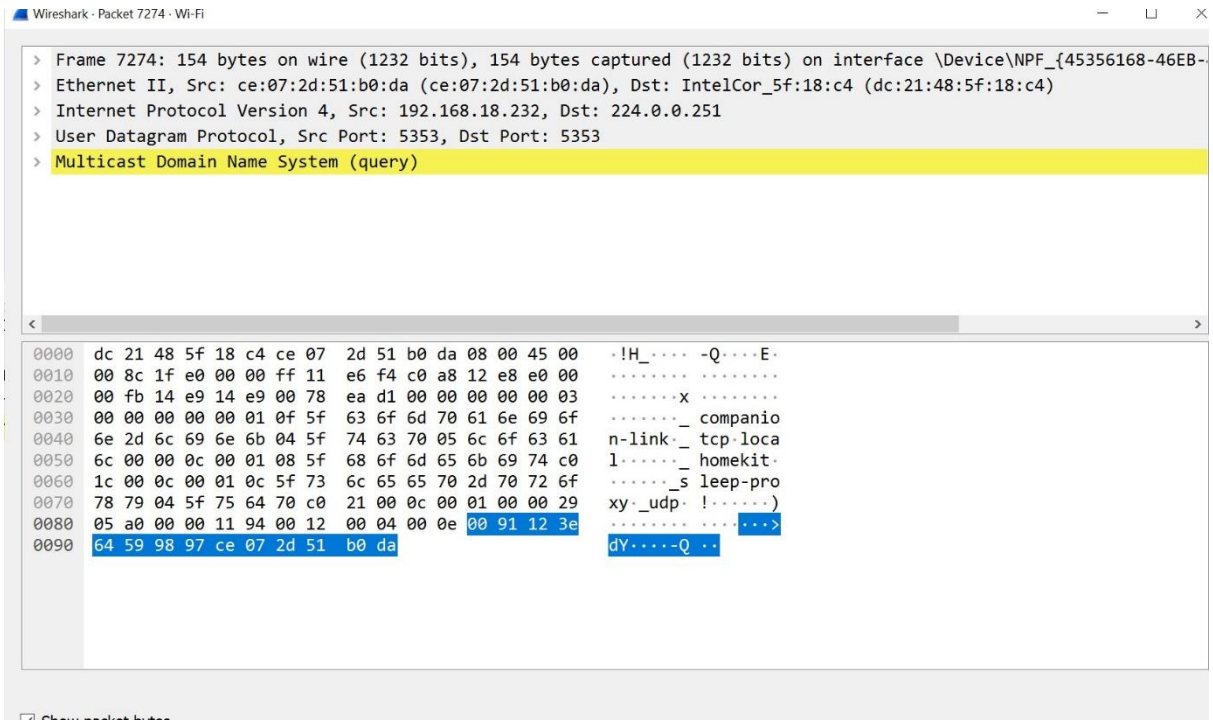


Bluetooth: Bluetooth is a wireless technology used for short-range communication between devices, typically for audio streaming, file transfer, and device control. Bluetooth analysis can help identify

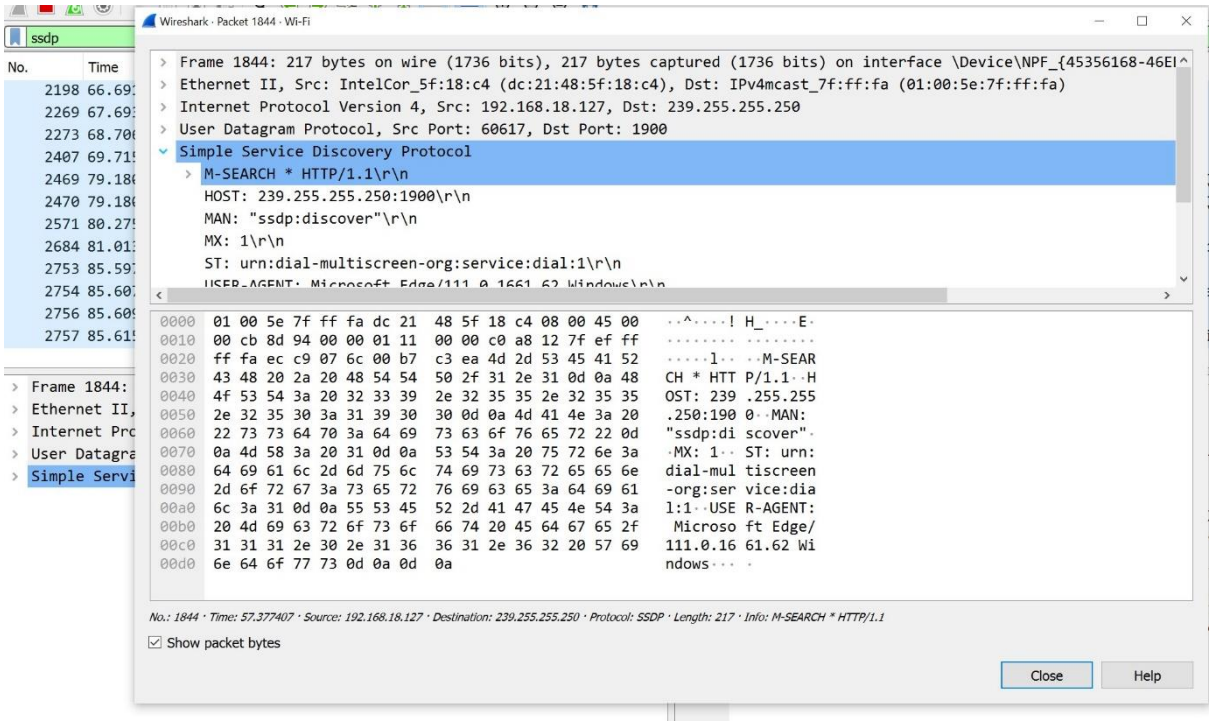
potential security risks associated with Bluetooth communication, such as unauthorized access or data interception.



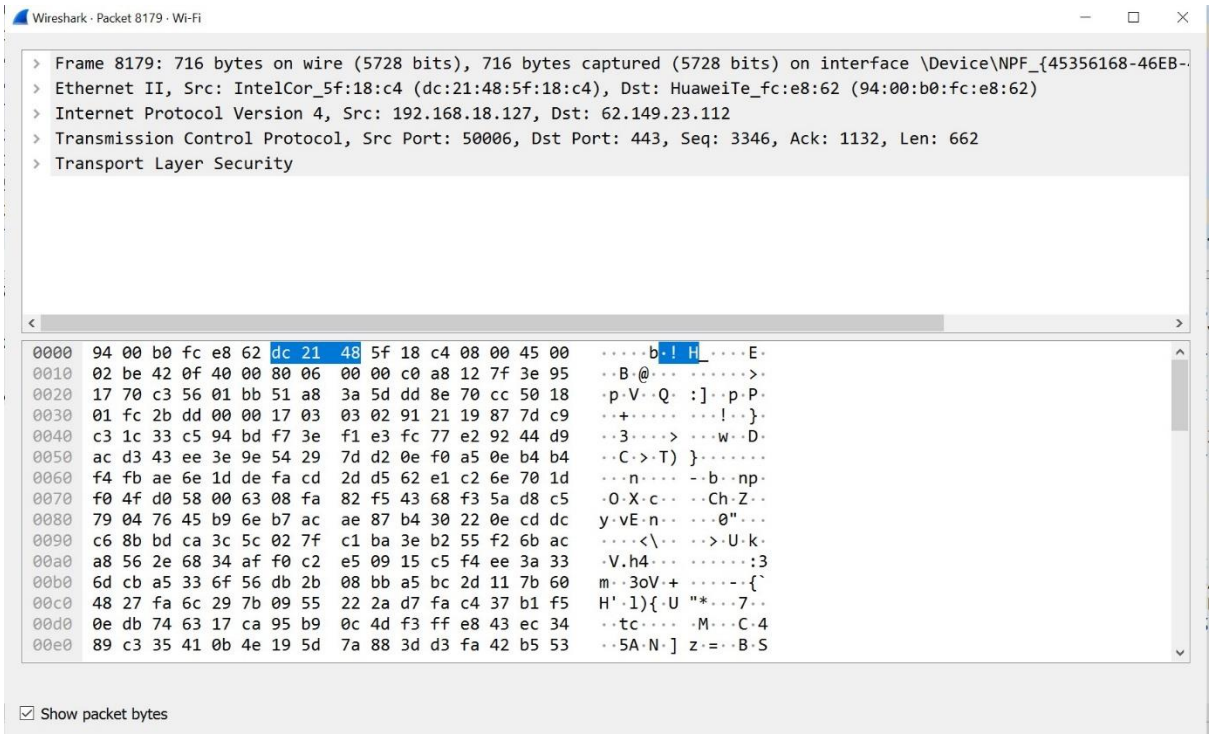
DNS (Domain Name System): DNS is used to translate human-readable domain names (such as www.google.com) into IP addresses. DNS analysis can help identify potential network attacks such as DNS spoofing or cache poisoning.



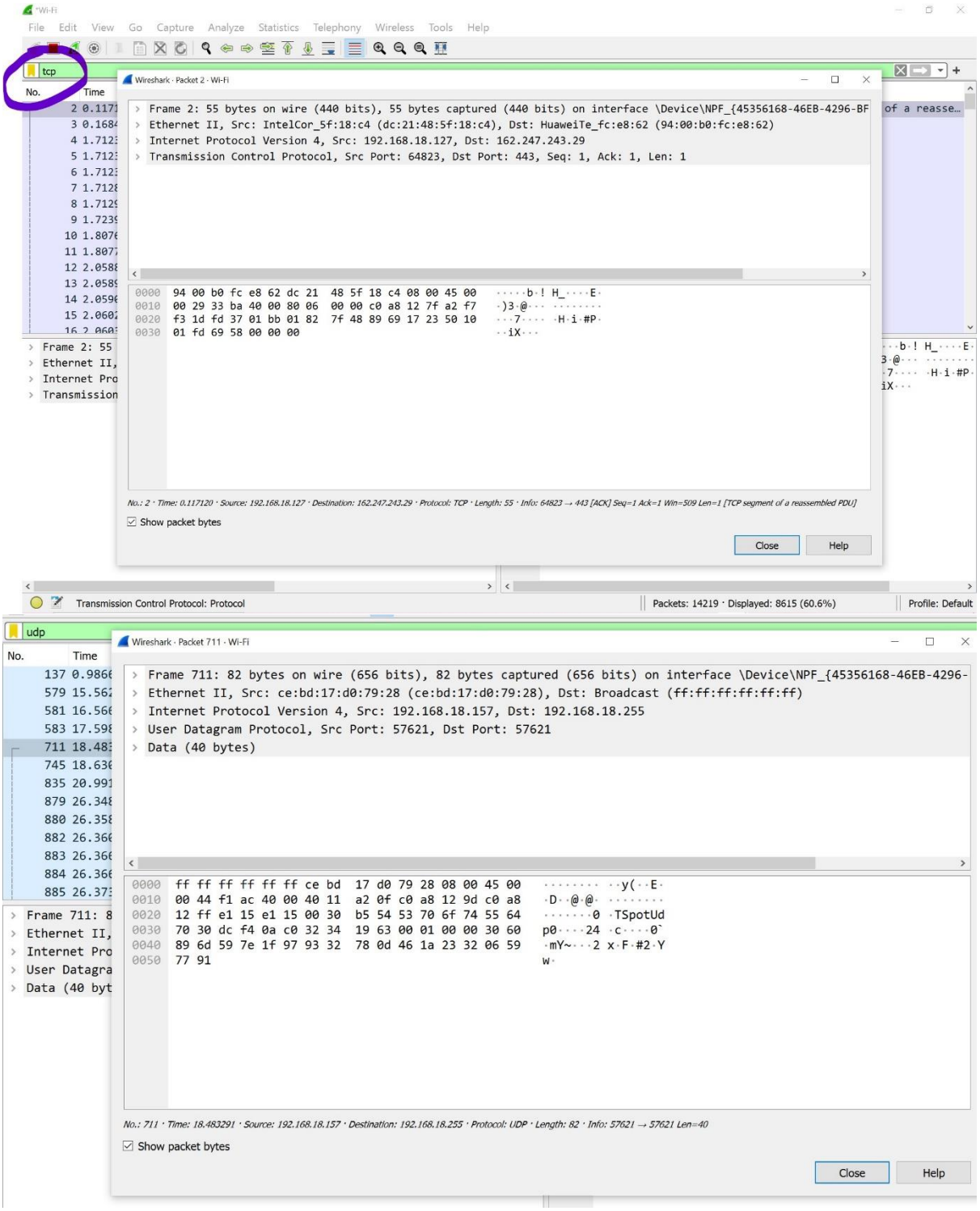
UDP (User Datagram Protocol): UDP is a connectionless protocol used for low-latency, unreliable data transmission. UDP analysis can help identify potential network performance or reliability issues.



SSDP (Simple Service Discovery Protocol): SSDP is used to discover and interact with devices on a network, typically for media streaming or other multimedia services. SSDP analysis can help identify potential network misconfigurations or security risks associated with exposed services.



TCP (Transmission Control Protocol): TCP is a connection-oriented protocol used for reliable data transmission. TCP analysis can help identify potential network congestion or reliability issues.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
No. Time
Wireshark · Packet 2 · Wi-Fi
Frame 2: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{45356168-46EB-4296-BF
Ethernet II, Src: IntelCor_5f:18:c4 (dc:21:48:5f:18:c4), Dst: HuaweiTe_fc:e8:62 (94:00:b0:fc:e8:62)
Internet Protocol Version 4, Src: 192.168.18.127, Dst: 162.247.243.29
Transmission Control Protocol, Src Port: 64823, Dst Port: 443, Seq: 1, Ack: 1, Len: 1
0000 94 00 b0 fc e8 62 dc 21 48 5f 18 c4 08 00 45 00
0010 00 29 33 ba 40 00 80 06 00 00 c0 a8 12 7f a2 f7
0020 f3 1d fd 37 01 bb 01 82 7f 48 89 69 17 23 50 10
0030 01 fd 69 58 00 00 00
No.: 2 · Time: 0.117120 · Source: 192.168.18.127 · Destination: 162.247.243.29 · Protocol: TCP · Length: 55 · Info: 64823 → 443 [ACK] Seq=1 Ack=1 Win=509 Len=1 [TCP segment of a reassembled PDU]
Show packet bytes
Close
Help
Transmission Control Protocol: Protocol
Packets: 14219 · Displayed: 8615 (60.6%)
Profile: Default
udp
No. Time
Wireshark · Packet 711 · Wi-Fi
Frame 711: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface \Device\NPF_{45356168-46EB-4296-
Ethernet II, Src: ce:bd:17:d0:79:28 (ce:bd:17:d0:79:28), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 192.168.18.157, Dst: 192.168.18.255
User Datagram Protocol, Src Port: 57621, Dst Port: 57621
Data (40 bytes)
0000 ff ff ff ff ff ff ce bd 17 d0 79 28 08 00 45 00
0010 00 44 f1 ac 40 00 40 11 a2 0f c0 a8 12 9d c0 a8
0020 12 ff e1 15 e1 15 00 30 b5 54 53 70 6f 74 55 64
0030 70 30 dc f4 0a c0 32 34 19 63 00 01 00 00 30 60
0040 89 6d 59 7e 1f 97 93 32 78 0d 46 1a 23 32 06 59
0050 77 91
No.: 711 · Time: 18.483291 · Source: 192.168.18.157 · Destination: 192.168.18.255 · Protocol: UDP · Length: 82 · Info: 57621 → 57621 Len=40
Show packet bytes
Close
Help

4 Bibliography

Bibliography:

- Varonis. (2021). How to Use Wireshark: Essential Tutorial for Network Analysis. Retrieved from https://www.varonis.com/blog/how-to-use-wireshark
- Wireshark. (2021). Chapter 1. Introduction. Retrieved from https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html
- JavaTpoint. (n.d.). Wireshark Tutorial. Retrieved from https://www.javatpoint.com/wireshark

GitHub link

https://github.com/suraj-giri/int301

These resources were used to learn more about the Wireshark network analysis tool, including its capabilities, how to use it, and best practices for network analysis. They were used to guide the project and provide additional context and information on network analysis using Wireshark.