CA – 3

(Task Based)

Name – Yash Vashistha
Registration No. - 11901753
Roll No. – 40
Section – KE015
Course Code – INT 301
Assigned Question – Q5

Submitted to – Dr. Manjot Kaur

Date of Submission – 08/04/2023

Network Forensics

1. Introduction

Fig 1 (Network Forensics)

Network forensics is a branch of digital forensics that involves the collection, analysis, and preservation of network traffic and data in order to investigate network-based security incidents, such as cyberattacks, data breaches, and other malicious activities.

It involves capturing and examining network packets, system logs, and other digital artifacts to identify the source of an attack, reconstruct the sequence of events, and gather evidence that can be used in legal proceedings or to improve security measures. Network forensics techniques can be used to identify the root cause of a security incident, determine the extent of the damage, and prevent similar incidents from occurring in the future.

1.1 Objectives of the project

The primary objective of network forensics is to identify the source, nature, and scope of a security incident or attack that occurred on a network. This involves analyzing network traffic, logs, and other data to determine what happened, when it happened, and who was involved. Another key objective is to preserve the evidence related to the security incident or attack. This involves capturing and preserving data, such as network packets, system logs, and other artifacts, in a forensically sound manner, ensuring that the data is not altered or destroyed in the process. Once the evidence is preserved, the next objective is to analyze the data to determine the cause and impact of the security incident or attack. This involves examining the data to identify the methods used by the attacker, the vulnerabilities that were exploited, and the extent of the damage caused. Another objective of network forensics is to identify the individual or group responsible for the security incident or attack. This may involve tracing the attack back to its origin or identifying the attackers through other means, such as analyzing their tactics, techniques, and procedures. The final objective is to remediate the security incident or attack, by implementing measures to prevent similar incidents from occurring in the future. This may involve patching vulnerabilities, improving security controls, or implementing new security measures to prevent similar attacks from occurring in the future.

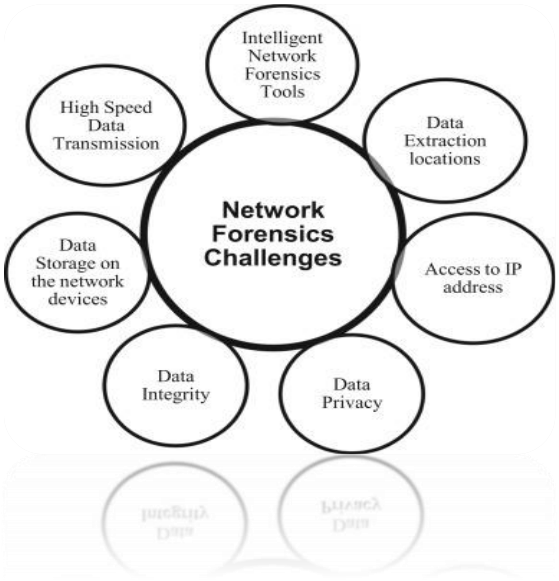


Fig 1.2 (Objectives)

1.2 Description of the project

Network forensics is the process of capturing, analyzing, and preserving network traffic and data to investigate security incidents, intrusions, or other unauthorized activities on a network. It involves the use of specialized tools and techniques to identify and analyze network traffic, logs, and other data sources to determine what happened, when it happened, and who was involved.

Network forensics can be used to investigate a variety of security incidents, such as malware infections, unauthorized access, data exfiltration, and other types of cyberattacks. By examining network traffic and other data sources, network forensics can help to identify the methods used by attackers, the vulnerabilities that were exploited, and the extent of the damage caused.

The process of network forensics typically involves several steps, including data capture, analysis, and reporting. The data capture phase involves collecting and preserving data, such as network packets, system logs, and other artifacts, in a forensically sound manner.

The analysis phase involves examining the data to identify the cause and impact of the security incident or attack, as well as any evidence of malicious activity. The reporting phase involves documenting the findings and presenting them in a clear and concise manner that can be understood by non-technical stakeholders.

Overall, network forensics is an important tool in the fight against cybercrime, allowing investigators to gather evidence and trace the actions of attackers to their source. It is a critical component of any incident response plan and is essential for organizations to maintain the security and integrity of their networks and data.

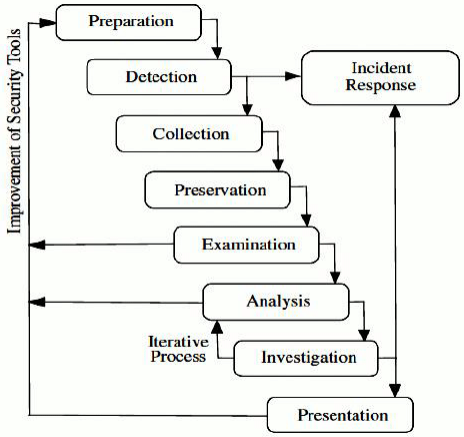


Fig 1.3 (Process of Network Forensics)

1.3 Scope of the project

The scope of network forensics is broad and encompasses various aspects of network security and incident response. Some of the key areas within the scope of network forensics include, Network forensics can help to identify vulnerabilities and weaknesses in network infrastructure, such as routers, switches, firewalls, and other network devices. It can also help to detect unauthorized access and potential threats to network security. Network forensics can be used to investigate and analyze malware infections, including the behavior of malware on the network, the types of systems that are targeted, and the methods used to propagate the malware. Network forensics is a critical component of incident response, allowing organizations to quickly detect and respond to security incidents, such as data breaches, network intrusions, and other types of cyberattacks. Network forensics can help to identify instances of data exfiltration or unauthorized access to sensitive data, allowing organizations to take steps to prevent further data loss or damage. Network forensics is also essential for organizations that must comply with regulations and standards related to data security, such as HIPAA, PCI-DSS, and GDPR. By conducting network forensics, organizations can ensure that they are meeting these requirements and taking appropriate measures to protect sensitive data. Overall, the scope of network forensics is vast and covers a wide range of areas related to network security and incident response. It is an essential component of any organization's security strategy and can help to minimize the impact of security incidents and protect critical assets.

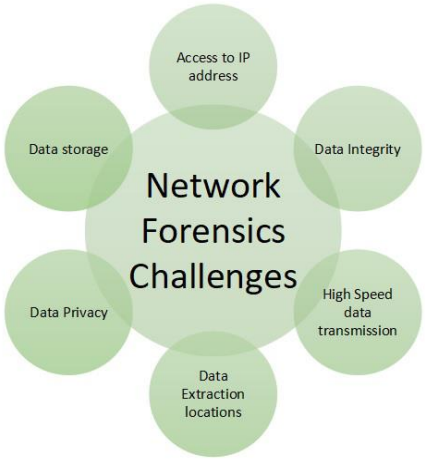


Fig 1.4 (Network Forensics Cycle)

2. System Description

2.1 Target System description

Wireshark is a popular network protocol analyzer software that allows users to capture, analyze, and troubleshoot network traffic in real-time. It runs on various operating systems, including Windows, macOS, and Linux, and is open source software that can be downloaded and used free of charge.
The target system for Wireshark is any device that can capture network traffic, including desktops, laptops, and servers. Wireshark can be used to capture network traffic on a wired or wireless network interface card (NIC), allowing users to analyze traffic on local networks or on the internet.
Once installed, Wireshark provides a graphical user interface (GUI) that allows users to capture and analyze network traffic. The GUI provides various tools and features, including the ability to filter and search captured packets, decode and display network protocols, and analyze network statistics.
In addition to the GUI, Wireshark also includes a command-line interface (CLI) that allows users to automate and script network analysis tasks. Wireshark also provides a set of APIs and libraries that allow developers to integrate the software with other tools and applications.
Overall, Wireshark is a versatile and powerful tool for network analysis and troubleshooting, and can be used on a wide range of target systems to capture and analyze network traffic.

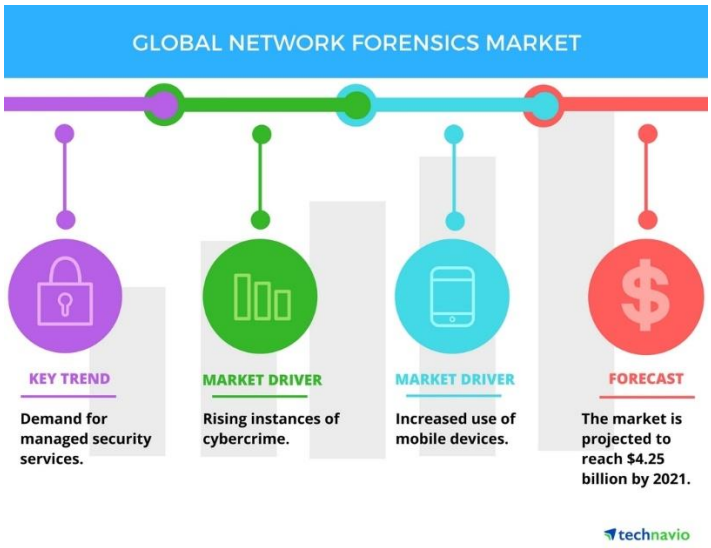


Fig 2.1(Market of Network Forensics)

2.2 Assumptions and dependencies

As a network protocol analyzer, Wireshark relies on several assumptions and dependencies to function effectively. Some of these include:

- Access to Network Traffic: Wireshark assumes that the user has access to the network traffic that they wish to analyze. This means that the user must have permission to capture traffic on the network, either by owning the network or having permission from the network owner.

- Network Interface Card (NIC) Drivers: Wireshark relies on the drivers for the network interface card (NIC) to capture network traffic. If the NIC drivers are outdated or incompatible with Wireshark, it may not be able to capture traffic effectively.

- System Resources: Wireshark requires a significant amount of system resources to capture and analyze network traffic. This includes CPU power, memory, and disk space. If the system resources are insufficient, Wireshark may not be able to capture or analyze network traffic effectively.

- Protocol Dissection: Wireshark depends on a library of protocol dissectors to decode and analyze network protocols. If the protocol dissectors are outdated or incomplete, Wireshark may not be able to analyze certain protocols or may provide inaccurate results.

- User Knowledge: Wireshark assumes that the user has a basic understanding of network protocols and networking concepts. Without this knowledge, the user may not be able to effectively analyze and interpret the network traffic captured by Wireshark.

2.3 Advantages of Wireshark

Wireshark is a powerful and versatile tool for network analysis and troubleshooting, and provides several advantages for users, including:

1. Comprehensive Network Analysis: Wireshark allows users to capture and analyze network traffic in real-time, providing a comprehensive view of the network and all its activities. This allows users to identify and troubleshoot network issues, including performance problems, security threats, and network configuration issues.
2. Protocol Analysis: Wireshark supports a wide range of network protocols and provides detailed protocol analysis, allowing users to view the contents of each packet, decode protocols, and analyze network behavior.
3. Easy to Use: Wireshark provides a user-friendly GUI that makes it easy for users to capture and analyze network traffic, even if they have limited experience with network analysis tools.
4. Platform Independence: Wireshark runs on various operating systems, including Windows, macOS, and Linux, providing platform independence and flexibility for users.
5. Open Source: Wireshark is an open-source software, which means that it is free to download and use, and its source code is available for modification and customization. This provides users with the ability to customize and extend the tool to meet their specific needs.
6. Wide Range of Features: Wireshark provides a wide range of features, including packet filtering, packet coloring, packet annotations, statistics, and more, allowing users to customize their analysis and gain deeper insights into network behavior.

   Overall, Wireshark provides a comprehensive, user-friendly, and extensible platform for network analysis and troubleshooting, making it an essential tool for network administrators, security analysts, and other IT professionals.

2.4 Disadvantages of Wireshark

Although Wireshark is a powerful and versatile tool for network analysis, it also has some disadvantages, including:

1. Steep Learning Curve: Wireshark is a complex and feature-rich application, which can make it difficult for novice users to get started with the tool. Learning to use all the features of Wireshark effectively can take a significant amount of time and effort.

2. Resource Intensive: Wireshark can be resource-intensive, especially when capturing and analyzing large amounts of network traffic. The tool may consume significant amounts of system memory and CPU resources, which can impact the performance of the underlying system.

3. Security Risks: Wireshark has the potential to expose sensitive information on a network, including usernames, passwords, and other confidential data. If Wireshark is not used securely, it can be exploited by attackers to steal sensitive information.

4. Overwhelming Amount of Data: Wireshark captures and analyzes packets at a very granular level, which can result in an overwhelming amount of data to analyze. Users may need to filter and focus on specific packets to analyze, which can be time-consuming and complex.

5. Legal Issues: In some jurisdictions, using Wireshark to capture network traffic without permission may be illegal. Users must ensure that they have the legal authority to capture and analyze network traffic before using Wireshark.

Overall, while Wireshark is a powerful and versatile tool for network analysis, it also has some disadvantages, including a steep learning curve, resource-intensive nature, potential security risks, overwhelming amounts of data, and legal issues.

2.3 Functional Dependencies of Wireshark

Wireshark is a complex software application that relies on several functional dependencies to perform its network analysis and packet capture functions. Some of the main functional dependencies of Wireshark which includes –

Network Interface Card (NIC) drivers: Wireshark depends on NIC drivers to capture packets on a network. The NIC drivers must be installed and configured correctly on the system where Wireshark is running for it to function properly.

Operating System (OS): Wireshark is dependent on the underlying operating system where it is installed. Different operating systems may provide different capabilities and APIs for capturing packets and analyzing network traffic, and Wireshark must be compatible with the underlying OS.

Libraries: Wireshark depends on several third-party libraries to perform its analysis functions. These libraries include libpcap, which provides packet capture capabilities, and several other libraries that provide protocol decoders and other analysis features.

Protocol Decoders: Wireshark relies on protocol decoders to analyze the packets captured on the network. These decoders are specific to different protocols and must be installed and configured correctly for Wireshark to be able to decode and analyze network traffic.

User Interface: Wireshark has a complex and feature-rich user interface that allows users to capture and analyze network traffic. The user interface depends on several underlying components, including the operating system's graphical user interface (GUI) libraries and the user's input devices.

Scripting Languages: Wireshark provides scripting capabilities that allow users to automate analysis tasks and extend the functionality of the application. Wireshark supports several scripting languages, including Lua and Python, and relies on the underlying scripting engine to run user scripts.

Implementation of Wireshark

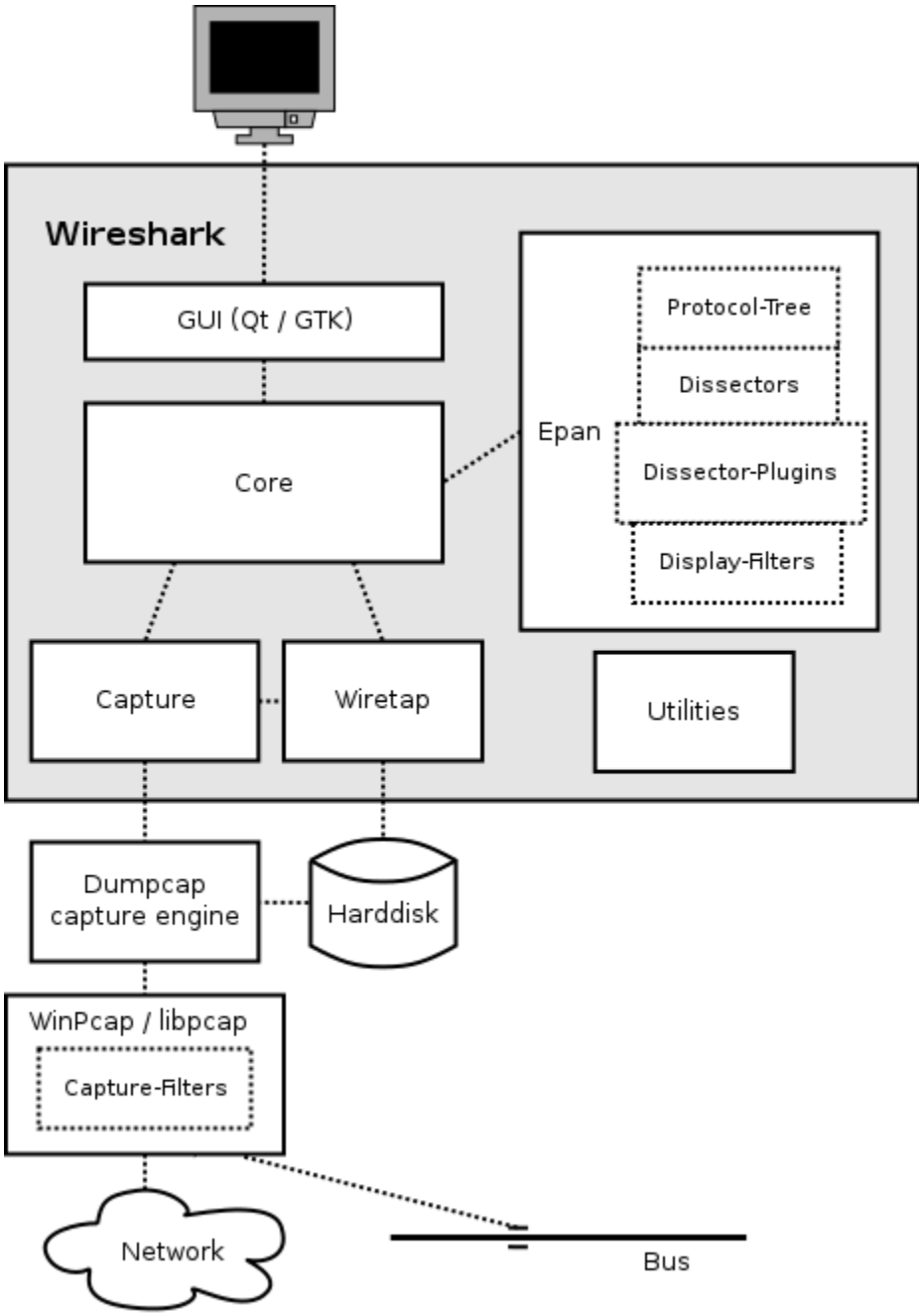


Fig 2.4 (Implementation of Wireshark)

IP address using Wireshark

To capture IP addresses with Wireshark, you can follow these steps:

1. Start Wireshark and select the network interface to capture packets from. You can do this by clicking on the interface name in the Wireshark main window.

2. Click on the "Capture Options" button to configure the capture settings. In the "Capture Filter" field, enter a filter expression to capture packets based on specific IP addresses. For example, if you want to capture packets sent to or from the IP address 192.168.1.10, you can use the filter expression "host 192.168.1.10". You can also capture packets based on IP address ranges or subnets using filter expressions such as "net 192.168.1.0/24".

3. Click on the "Start" button to begin capturing packets.

4. Once packets are captured, you can view the IP addresses in the Packet List pane or the Packet Details view. The Packet List pane shows a summary of all the captured packets, including the source and destination IP addresses. The Packet Details view provides a more detailed view of each packet, including the IP header information.

5. You can use display filters to isolate packets based on specific IP addresses or IP address ranges. To do this, enter a filter expression such as "ip.addr == 192.168.1.10" in the filter field at the top of the Wireshark window.

By following these steps, you can use Wireshark to capture and analyze network traffic based on specific IP addresses.

Mac Address using Wireshark

In Wireshark, you can view the MAC address of a network device by capturing network traffic and analyzing the packets.

To do this, follow these steps:

1. Open Wireshark and select the network interface you want to capture traffic on.

2. Start capturing packets by clicking the "Start" button in the toolbar.

3. Wait for some network traffic to be captured, then stop the capture by clicking the "Stop" button.

4. Look at the captured packets in the packet list pane. You should see a column labeled "Source" and "Destination" MAC address.

5. Find the packet of interest and click on it to expand the details in the packet details pane.

6. Look for the Ethernet section, and you will see the source and destination MAC addresses.

Note that MAC addresses are unique to each network device and are used for identifying devices on a network. MAC addresses are often used in network forensics to trace the source of network traffic or identify network devices involved in an attack.

Port Details using Wireshark

In Wireshark, you can view the port details of network traffic by analyzing the captured packets. Here are the steps to view port details in Wireshark:

Open Wireshark and select the network interface you want to capture traffic on.

Start capturing packets by clicking the "Start" button in the toolbar.

Wait for some network traffic to be captured, then stop the capture by clicking the "Stop" button.

Look at the captured packets in the packet list pane. You should see a column labeled "Source Port" and "Destination Port".

Find the packet of interest and click on it to expand the details in the packet details pane.

Look for the "Transmission Control Protocol" (TCP) section or "User Datagram Protocol" (UDP) section, depending on the type of traffic you are analyzing.

In the TCP or UDP section, you will see the source and destination ports used for the communication.

Note that ports are used to identify specific services or applications running on a device. Analyzing port details can help in understanding the type of traffic, identifying the type of application or service, and detecting unusual network activity that may be associated with a security incident.

Encryption Details using Wireshark

In Wireshark, you can view the encryption details of network traffic by analyzing the captured packets. Here are the steps to view encryption details in Wireshark:

Open Wireshark and select the network interface you want to capture traffic on.

Start capturing packets by clicking the "Start" button in the toolbar.

Wait for some network traffic to be captured, then stop the capture by clicking the "Stop" button.

Look at the captured packets in the packet list pane. You should see a column labeled "Protocol".

Find the packet of interest and click on it to expand the details in the packet details pane.

Look for the "Secure Sockets Layer" (SSL) section or "Transport Layer Security" (TLS) section, depending on the type of encryption used.

In the SSL or TLS section, you will see the encryption details, including the type of encryption algorithm used and the SSL/TLS version.

Note that encryption is used to protect the confidentiality and integrity of data transmitted over a network. Analyzing encryption details can help in understanding the level of security provided by the encryption and detecting any vulnerabilities or weaknesses in the encryption used.

Banner Information using Wireshark

In Wireshark, you can view banner information of network traffic by analyzing the captured packets. A banner is a message or information sent by a service or application running on a network device. Here are the steps to view banner information in Wireshark:

Open Wireshark and select the network interface you want to capture traffic on.

Start capturing packets by clicking the "Start" button in the toolbar.

Wait for some network traffic to be captured, then stop the capture by clicking the "Stop" button.

Look at the captured packets in the packet list pane. You should see a column labeled "Protocol".

Find the packet of interest and click on it to expand the details in the packet details pane.

Look for the "Protocol" section, and you will see the protocol used for the communication.

If the protocol is a well-known service, such as HTTP, FTP, or Telnet, you can usually find the banner information in the packet details pane under the corresponding protocol section.

Look for any messages or information sent by the service or application in the packet details pane.

Note that banner information can provide useful information about the version, vendor, and configuration of a service or application running on a network device. Analyzing banner information can help in understanding the types of services and applications running on a network, detecting vulnerabilities, and identifying potential security risks.

RDP using Wireshark

In Wireshark, you can analyze Remote Desktop Protocol (RDP) traffic to troubleshoot RDP issues or investigate security incidents. Here are the steps to analyze RDP traffic using Wireshark:

Open Wireshark and select the network interface you want to capture traffic on.

Start capturing packets by clicking the "Start" button in the toolbar.

Launch the RDP session to the target device.

Wait for some RDP traffic to be captured, then stop the capture by clicking the "Stop" button.

Look at the captured packets in the packet list pane. You should see a column labeled "Protocol".

Filter the captured packets to display only RDP traffic. To do this, type "rdp" in the display filter field.

Find the packet of interest and click on it to expand the details in the packet details pane.

Look for the "Remote Desktop Protocol" section in the packet details pane.

Analyze the RDP packets to troubleshoot issues or investigate security incidents. You can look at the RDP handshake, RDP channel setup, RDP data transfer, and RDP disconnection to identify any issues or anomalies.

Note that RDP is a protocol used to remotely access and control Windows-based computers. Analyzing RDP traffic can help in troubleshooting RDP issues, identifying RDP security risks, and detecting any malicious RDP activity.

FTP using Wireshark

In Wireshark, you can analyze File Transfer Protocol (FTP) traffic to troubleshoot FTP issues or investigate security incidents. Here are the steps to analyze FTP traffic using Wireshark:

Open Wireshark and select the network interface you want to capture traffic on.

Start capturing packets by clicking the "Start" button in the toolbar.

Launch the FTP session to the target device.

Wait for some FTP traffic to be captured, then stop the capture by clicking the "Stop" button.

Look at the captured packets in the packet list pane. You should see a column labeled "Protocol".

Filter the captured packets to display only FTP traffic. To do this, type "ftp" in the display filter field.

Find the packet of interest and click on it to expand the details in the packet details pane.

Look for the "File Transfer Protocol" section in the packet details pane.

Analyze the FTP packets to troubleshoot issues or investigate security incidents. You can look at the FTP commands and responses, FTP data transfer, and FTP control channel to identify any issues or anomalies.

Note that FTP is a protocol used to transfer files between networked computers. Analyzing FTP traffic can help in troubleshooting FTP issues, identifying FTP security risks, and detecting any malicious FTP activity. It's worth noting that FTP sends all its data in clear text, including passwords, so it's recommended to use Secure FTP (SFTP) or FTPS (FTP over SSL/TLS) for secure file transfer.

SMTP using Wireshark

In Wireshark, you can analyze Simple Mail Transfer Protocol (SMTP) traffic to troubleshoot email delivery issues or investigate security incidents. Here are the steps to analyze SMTP traffic using Wireshark:

Open Wireshark and select the network interface you want to capture traffic on.

Start capturing packets by clicking the "Start" button in the toolbar.

Launch your email client and send an email.

Wait for some SMTP traffic to be captured, then stop the capture by clicking the "Stop" button.

Look at the captured packets in the packet list pane. You should see a column labeled "Protocol".

Filter the captured packets to display only SMTP traffic. To do this, type "smtp" in the display filter field.

Find the packet of interest and click on it to expand the details in the packet details pane.

Look for the "Simple Mail Transfer Protocol" section in the packet details pane.

Analyze the SMTP packets to troubleshoot issues or investigate security incidents. You can look at the SMTP commands and responses, email header information, email content, and SMTP communication between the sender and recipient mail servers.

Note that SMTP is a protocol used to send email messages between mail servers. Analyzing SMTP traffic can help in troubleshooting email delivery issues, identifying email security risks, and detecting any malicious email activity. It's worth noting that SMTP sends all its data in clear text, including email contents and attachments, so it's recommended to use Secure SMTP (SMTPS) or Transport Layer Security (TLS) for secure email transmission.

Netbios using Wireshark

In Wireshark, you can analyze Network Basic Input/Output System (NetBIOS) traffic to troubleshoot issues related to network browsing, file sharing, and printer sharing on Windows-based computers. Here are the steps to analyze NetBIOS traffic using Wireshark:

Open Wireshark and select the network interface you want to capture traffic on.

Start capturing packets by clicking the "Start" button in the toolbar.

Launch any NetBIOS-based application, such as Windows File Explorer or Command Prompt.

Wait for some NetBIOS traffic to be captured, then stop the capture by clicking the "Stop" button.

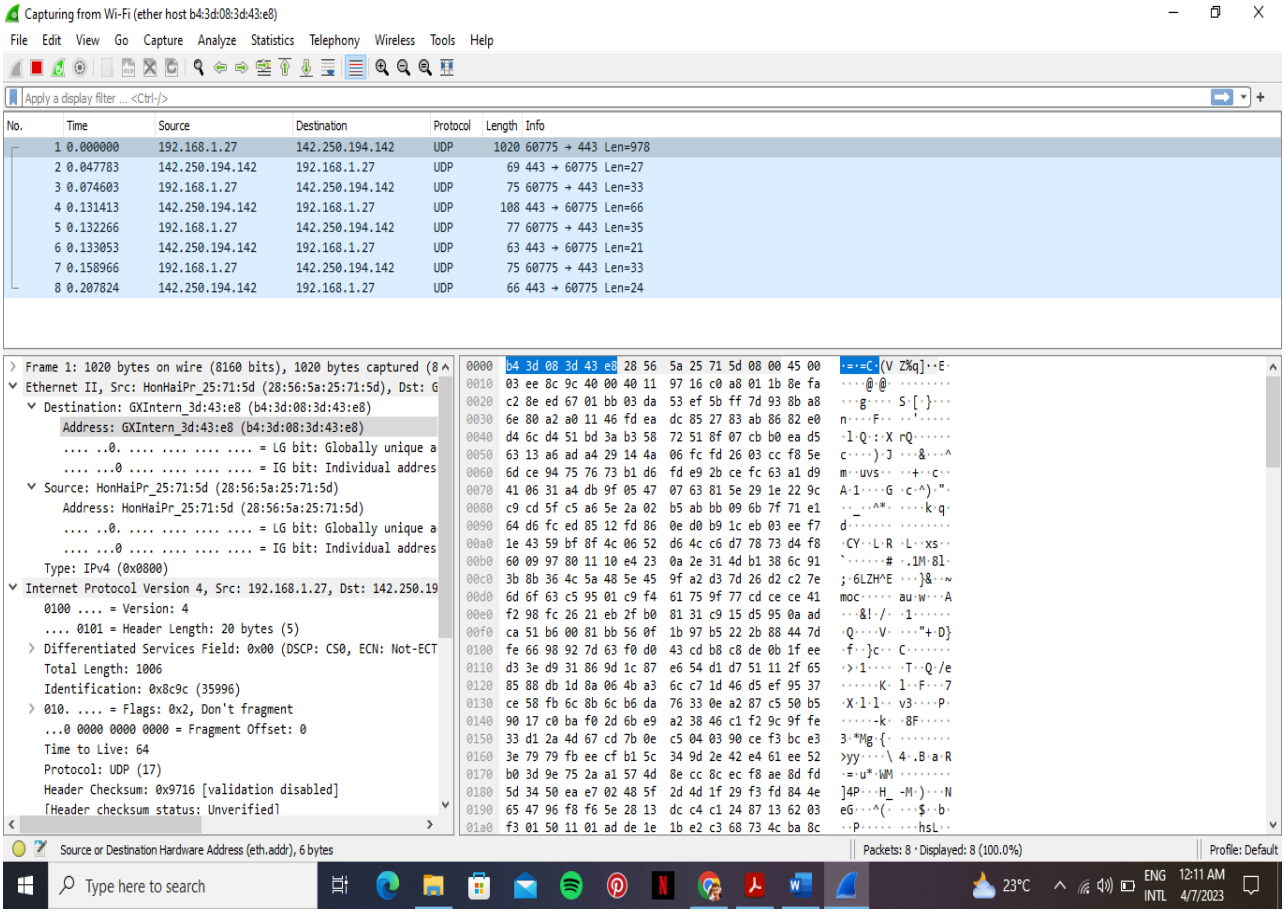
Look at the captured packets in the packet list pane. You should see a column labeled "Protocol".

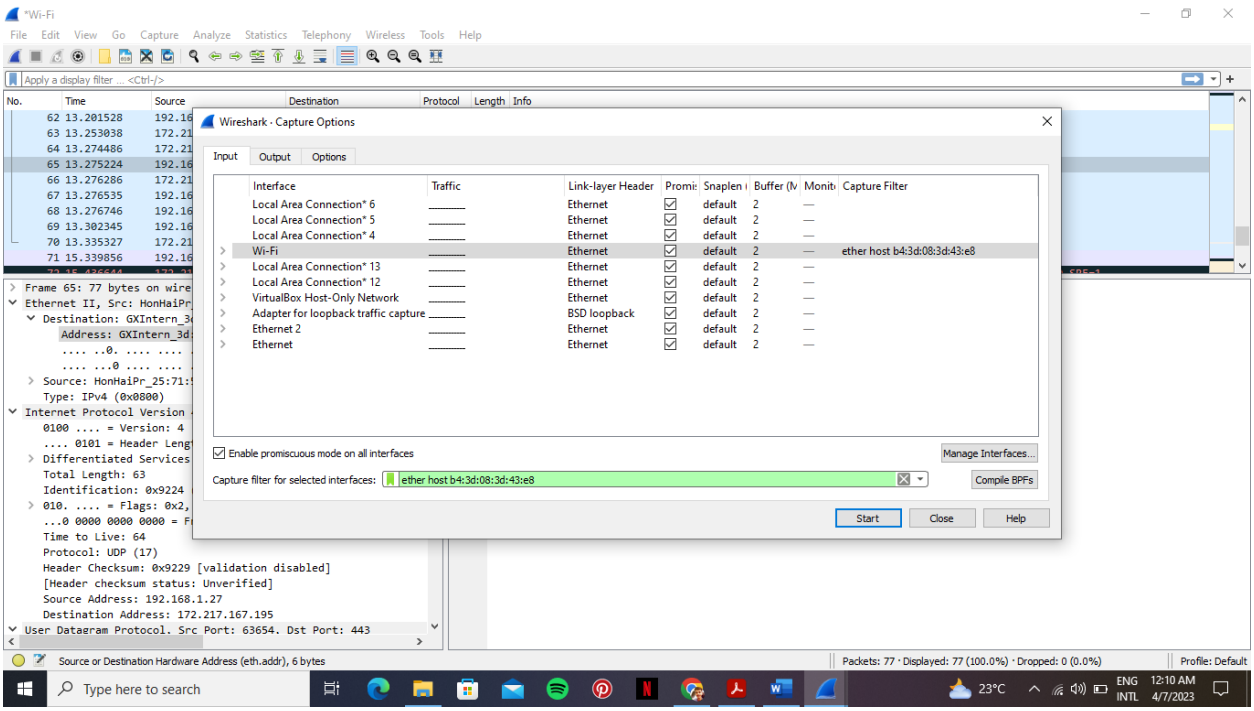
Filter the captured packets to display only NetBIOS traffic. To do this, type "netbios" in the display filter field.
Find the packet of interest and click on it to expand the details in the packet details pane.

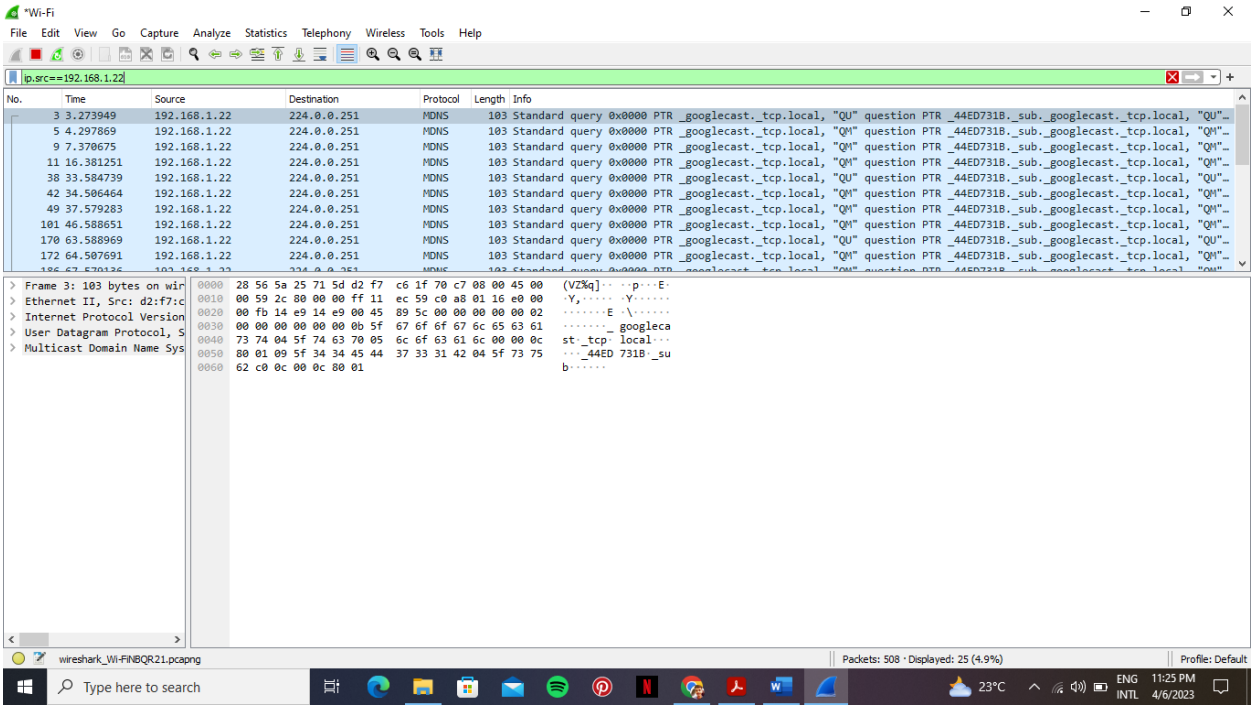
Look for the "NetBIOS" section in the packet details pane.

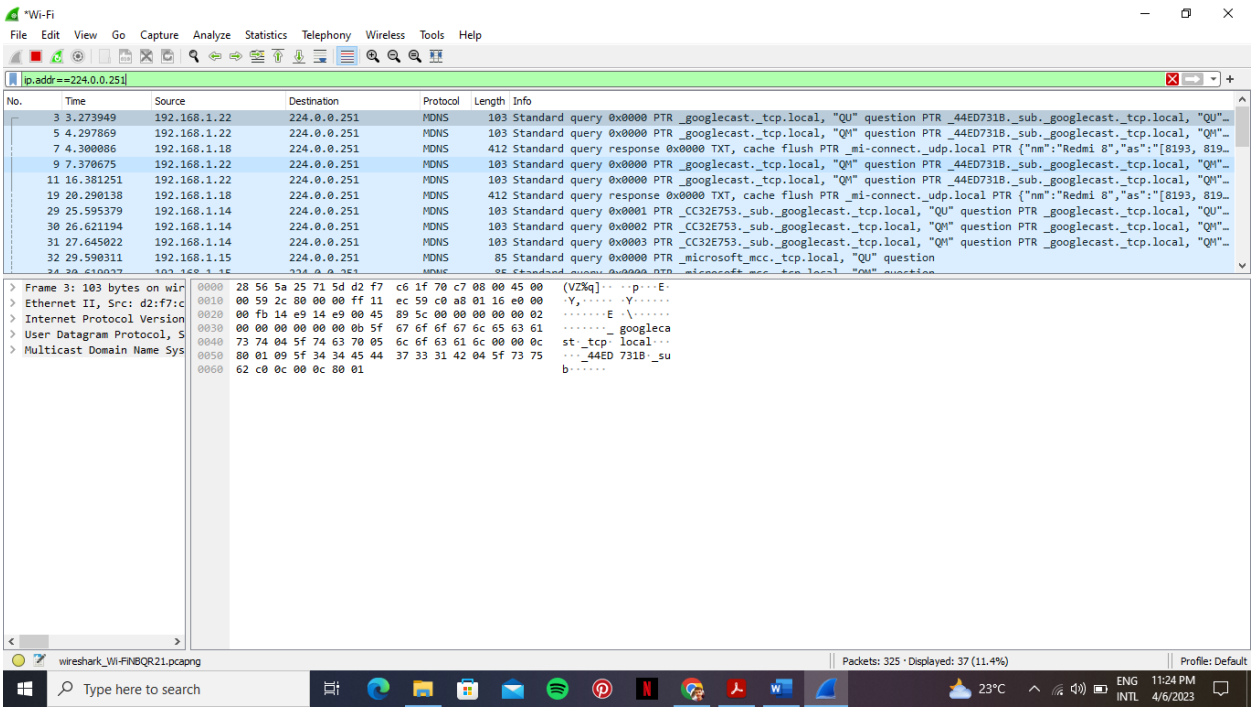
Analyze the NetBIOS packets to troubleshoot issues related to network browsing, file sharing, and printer sharing. You can look at the NetBIOS session establishment, NetBIOS name resolution, NetBIOS datagram service, and NetBIOS mailslot service to identify any issues or anomalies.
Note that NetBIOS is a protocol used to enable communication between computers on a local area network (LAN). Analyzing NetBIOS traffic can help in troubleshooting issues related to network browsing, file sharing, and printer sharing. However, NetBIOS sends all its data in clear text, including authentication credentials, so it's recommended to use Secure NetBIOS (SMB over IPsec) Server Message Block (SMB) protocol with encryption for secure file and printer sharing

Snapshots of the above working project using Wireshark

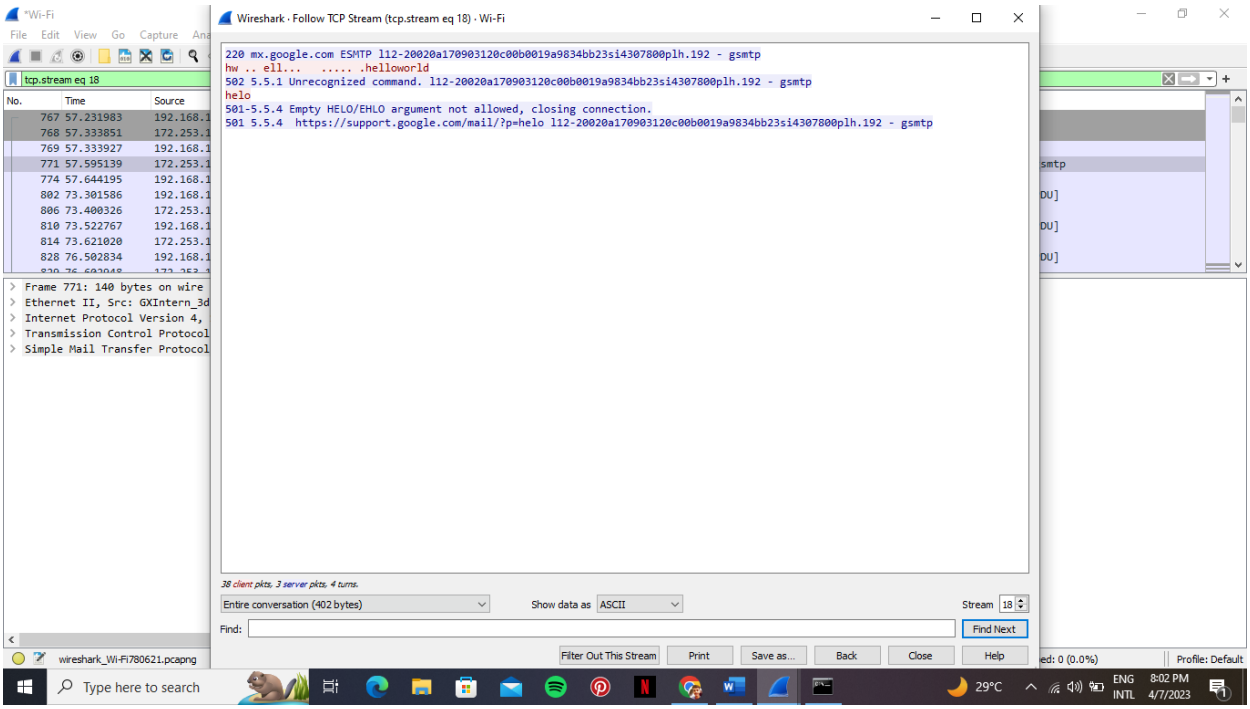



*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
Wireshark · Capture Options
Input Output Options
Interface Traffic Link-layer Header Promi: Snaplen Buffer (M Monit Capture Filter
Local Area Connection* 6 Ethernet default 2
Local Area Connection* 5 Ethernet default 2
Local Area Connection* 4 Ethernet default 2
Wi-Fi Ethernet default 2 ether host b4:3d:08:3d:43:e8
Local Area Connection* 13 Ethernet default 2
Local Area Connection* 12 Ethernet default 2
VirtualBox Host-Only Network Ethernet default 2
Adapter for loopback traffic capture BSD loopback default 2
Ethernet 2 Ethernet default 2
Ethernet Ethernet default 2
Enable promiscuous mode on all interfaces
Manage Interfaces...
Capture filter for selected interfaces: ether host b4:3d:08:3d:43:e8
Compile BPFs
Start Close Help
Source or Destination Hardware Address (eth.addr), 6 bytes
Packets: 77 · Displayed: 77 (100.0%) · Dropped: 0 (0.0%)
Profile: Default
Type here to search
23°C
ENG INTL 12:10 AM 4/7/2023

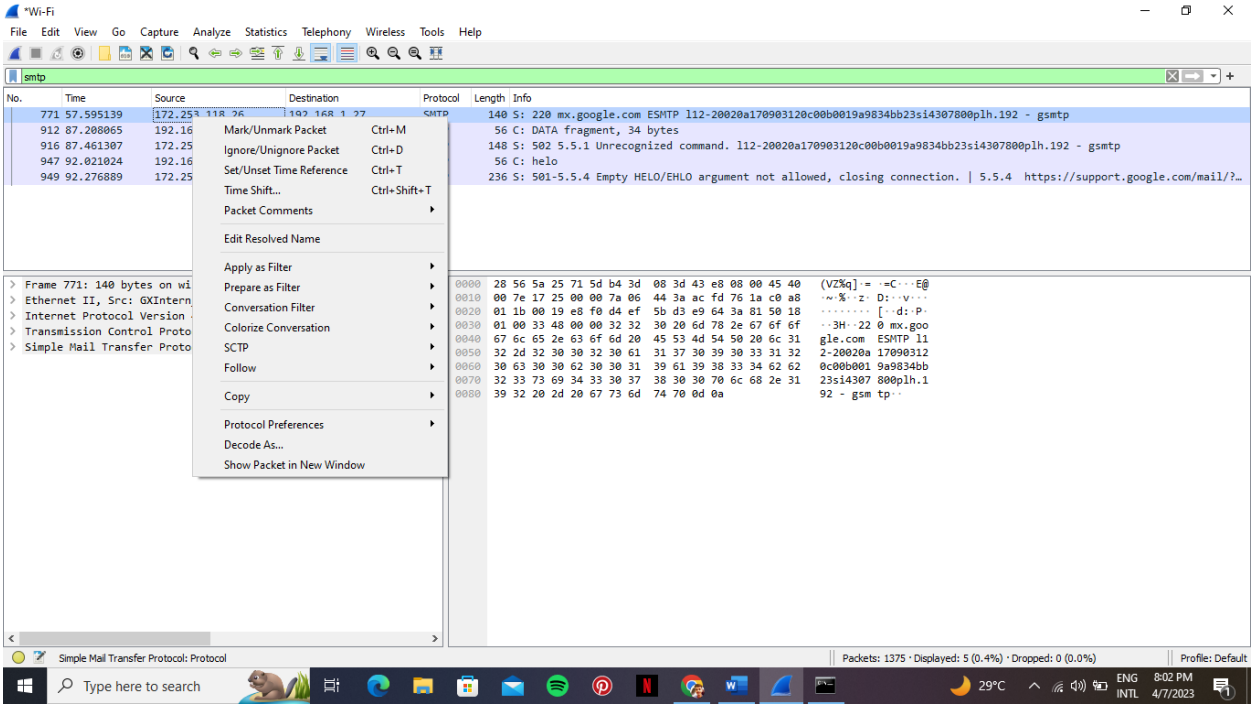

*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.src==192.168.1.22
No. Time Source Destination Protocol Length Info
3 3.273949 192.168.1.22 224.0.0.251 MDNS 103 Standard query 0x0000 PTR _googlecast._tcp.local, "QU" question PTR _44ED731B._sub._googlecast._tcp.local, "QU"
5 4.297869 192.168.1.22 224.0.0.251 MDNS 103 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question PTR _44ED731B._sub._googlecast._tcp.local, "QM"
9 7.370675 192.168.1.22 224.0.0.251 MDNS 103 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question PTR _44ED731B._sub._googlecast._tcp.local, "QM"
11 16.381251 192.168.1.22 224.0.0.251 MDNS 103 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question PTR _44ED731B._sub._googlecast._tcp.local, "QM"
38 33.584739 192.168.1.22 224.0.0.251 MDNS 103 Standard query 0x0000 PTR _googlecast._tcp.local, "QU" question PTR _44ED731B._sub._googlecast._tcp.local, "QU"
42 34.506464 192.168.1.22 224.0.0.251 MDNS 103 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question PTR _44ED731B._sub._googlecast._tcp.local, "QM"
49 37.579283 192.168.1.22 224.0.0.251 MDNS 103 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question PTR _44ED731B._sub._googlecast._tcp.local, "QM"
101 46.588651 192.168.1.22 224.0.0.251 MDNS 103 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question PTR _44ED731B._sub._googlecast._tcp.local, "QM"
170 63.588969 192.168.1.22 224.0.0.251 MDNS 103 Standard query 0x0000 PTR _googlecast._tcp.local, "QU" question PTR _44ED731B._sub._googlecast._tcp.local, "QU"
172 64.507691 192.168.1.22 224.0.0.251 MDNS 103 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question PTR _44ED731B._sub._googlecast._tcp.local, "QM"
Frame 3: 103 bytes on wir
Ethernet II, Src: d2:f7:c
Internet Protocol Version
User Datagram Protocol, S
Multicast Domain Name Sys
0000 28 56 5a 25 71 5d d2 f7 c6 1f 70 c7 08 00 45 00 (VZ%q]···· ··p···E·
0010 00 59 2c 80 00 00 ff 11 ec 59 c0 a8 01 16 e0 00 ·Y,····· ·Y····
0020 00 fb 14 e9 14 e9 00 45 89 5c 00 00 00 00 00 02 ·····E ·\····
0030 00 00 00 00 00 00 0b 5f 67 6f 6f 67 6c 65 63 61 ·····_ googleca
0040 73 74 04 5f 74 63 70 05 6c 6f 63 61 6c 00 00 0c st·_tcp· local···
0050 80 01 09 5f 34 34 45 44 37 33 31 42 04 5f 73 75 ···_44ED 731B·_su
0060 62 c0 0c 00 0c 80 01 b·····
wireshark_Wi-FINBQR21.pcapng
Packets: 508 · Displayed: 25 (4.9%)
Profile: Default
Type here to search
23°C
ENG INTL 11:25 PM 4/6/2023

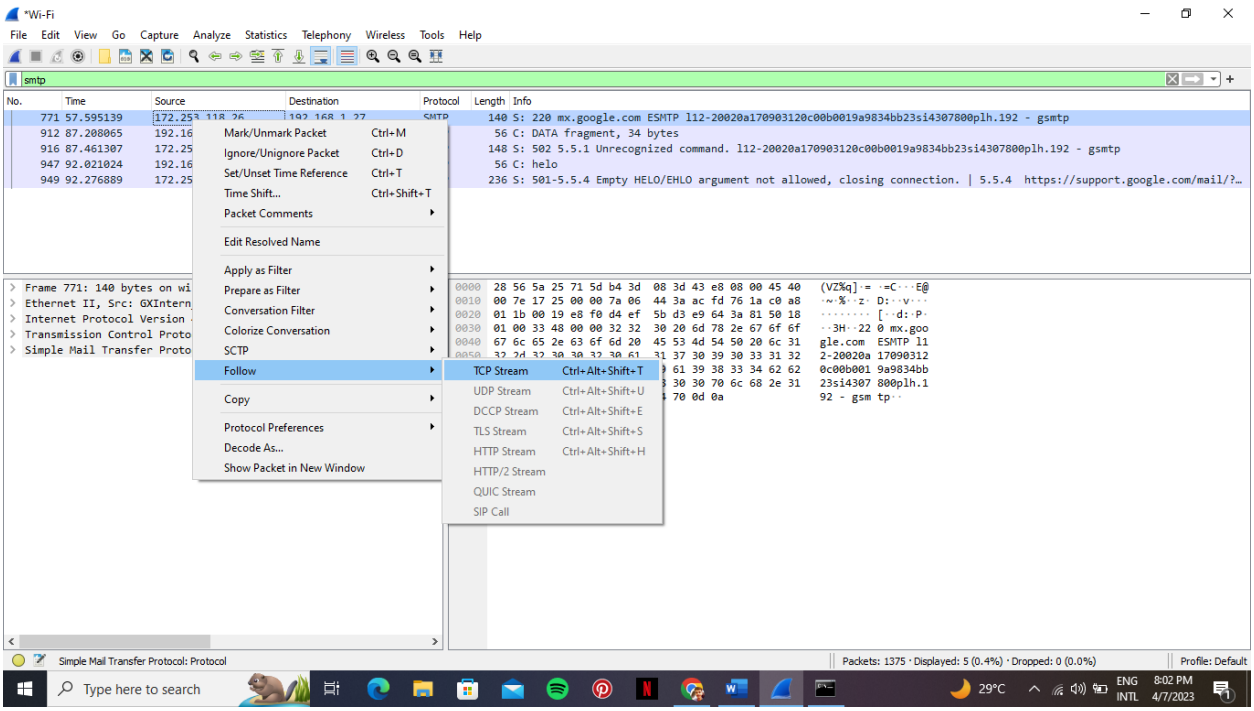

*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr==224.0.0.251
No. Time Source Destination Protocol Length Info
3 3.273949 192.168.1.22 224.0.0.251 MDNS 103 Standard query 0x0000 PTR _googlecast._tcp.local, "QU" question PTR _44ED731B._sub._googlecast._tcp.local, "QU"…
5 4.297869 192.168.1.22 224.0.0.251 MDNS 103 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question PTR _44ED731B._sub._googlecast._tcp.local, "QM"…
7 4.300086 192.168.1.18 224.0.0.251 MDNS 412 Standard query response 0x0000 TXT, cache flush PTR _mi-connect._udp.local PTR {"nm":"Redmi 8","as":"[8193, 819…
9 7.370675 192.168.1.22 224.0.0.251 MDNS 103 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question PTR _44ED731B._sub._googlecast._tcp.local, "QM"…
11 16.381251 192.168.1.22 224.0.0.251 MDNS 103 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question PTR _44ED731B._sub._googlecast._tcp.local, "QM"…
19 20.290138 192.168.1.18 224.0.0.251 MDNS 412 Standard query response 0x0000 TXT, cache flush PTR _mi-connect._udp.local PTR {"nm":"Redmi 8","as":"[8193, 819…
29 25.595379 192.168.1.14 224.0.0.251 MDNS 103 Standard query 0x0001 PTR _CC32E753._sub._googlecast._tcp.local, "QU" question PTR _googlecast._tcp.local, "QU"…
30 26.621194 192.168.1.14 224.0.0.251 MDNS 103 Standard query 0x0002 PTR _CC32E753._sub._googlecast._tcp.local, "QM" question PTR _googlecast._tcp.local, "QM"…
31 27.645022 192.168.1.14 224.0.0.251 MDNS 103 Standard query 0x0003 PTR _CC32E753._sub._googlecast._tcp.local, "QM" question PTR _googlecast._tcp.local, "QM"…
32 29.590311 192.168.1.15 224.0.0.251 MDNS 85 Standard query 0x0000 PTR _microsoft_mcc._tcp.local, "QU" question
Frame 3: 103 bytes on wir
Ethernet II, Src: d2:f7:c
Internet Protocol Version
User Datagram Protocol, S
Multicast Domain Name Sys
0000 28 56 5a 25 71 5d d2 f7 c6 1f 70 c7 08 00 45 00 (VZ%q]·· ··p···E·
0010 00 59 2c 80 00 00 ff 11 ec 59 c0 a8 01 16 e0 00 ·Y,····· ·Y····
0020 00 fb 14 e9 14 e9 00 45 89 5c 00 00 00 00 00 02 ·····E ·\····
0030 00 00 00 00 00 00 0b 5f 67 6f 6f 67 6c 65 63 61 ·····_ googleca
0040 73 74 04 5f 74 63 70 05 6c 6f 63 61 6c 00 00 0c st·_tcp· local···
0050 80 01 09 5f 34 34 45 44 37 33 31 42 04 5f 73 75 ···_44ED 731B·_su
0060 62 c0 0c 00 0c 80 01 b·····
wireshark_Wi-FiNBQR21.pcapng
Packets: 325 · Displayed: 37 (11.4%)
Profile: Default
Type here to search
23°C
ENG INTL 11:24 PM 4/6/2023

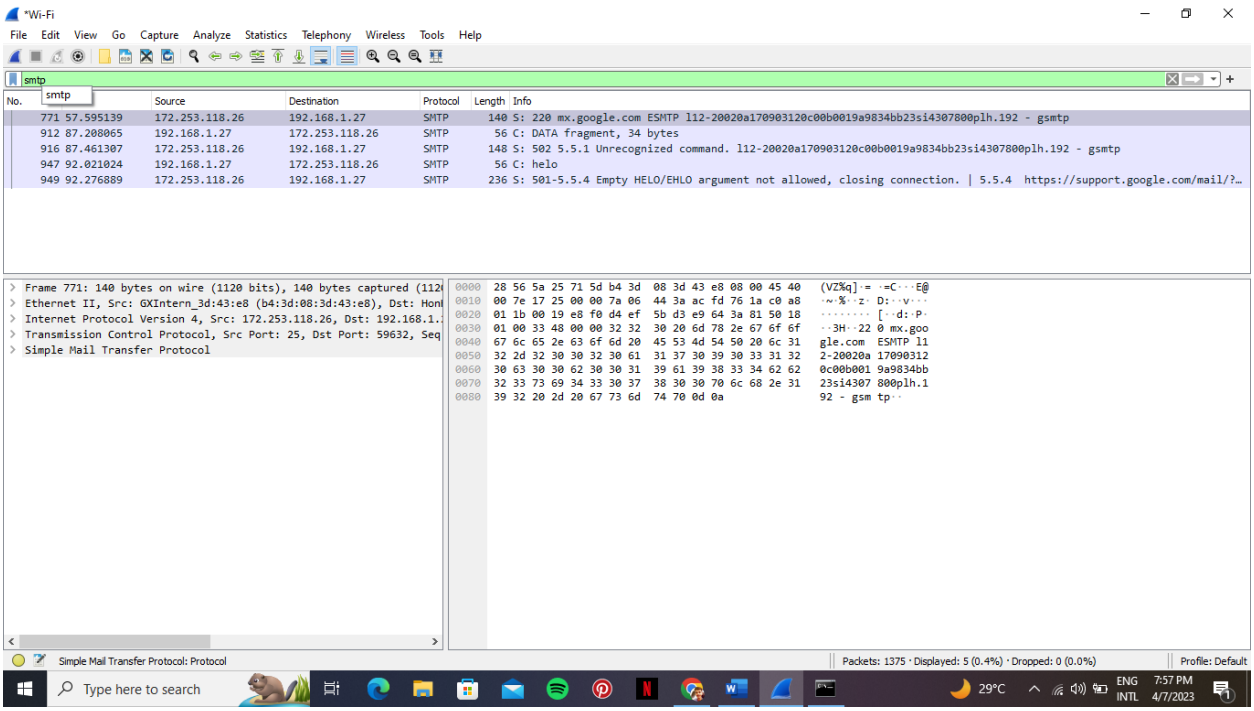
(IP address)


*Wi-Fi
tcp.stream eq 18
No. Time Source
767 57.231983 192.168.1
768 57.333851 172.253.1
769 57.333927 192.168.1
771 57.595139 172.253.1
774 57.644195 192.168.1
802 73.301586 192.168.1
806 73.400326 172.253.1
810 73.522767 192.168.1
814 73.621020 172.253.1
828 76.502834 192.168.1
Frame 771: 140 bytes on wire
Ethernet II, Src: GXIntern_3d
Internet Protocol Version 4,
Transmission Control Protocol
Simple Mail Transfer Protocol
wireshark_Wi-Fi780621.pcapng
Wireshark · Follow TCP Stream (tcp.stream eq 18) · Wi-Fi
220 mx.google.com ESMTP l12-20020a170903120c00b0019a9834bb23si4307800plh.192 - gsmtp
hw .. ell... ..... .helloworld
502 5.5.1 Unrecognized command. l12-20020a170903120c00b0019a9834bb23si4307800plh.192 - gsmtp
helo
501-5.5.4 Empty HELO/EHLO argument not allowed, closing connection.
501 5.5.4 https://support.google.com/mail/?p=helo l12-20020a170903120c00b0019a9834bb23si4307800plh.192 - gsmtp
38 client pkts, 3 server pkts, 4 turns.
Entire conversation (402 bytes)
Show data as ASCII
Stream 18
Find:
Find Next
Filter Out This Stream
Print
Save as...
Back
Close
Help
Profile: Default
Type here to search
29°C
ENG INTL 8:02 PM 4/7/2023

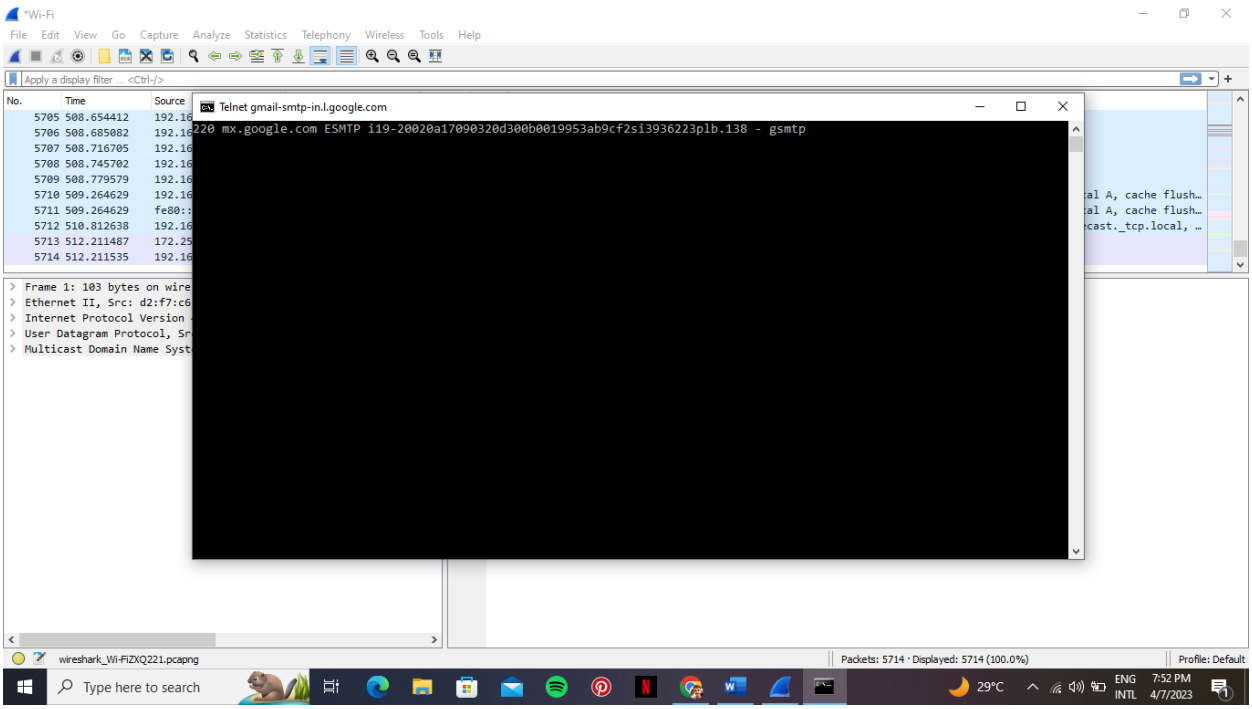
(SMTP)

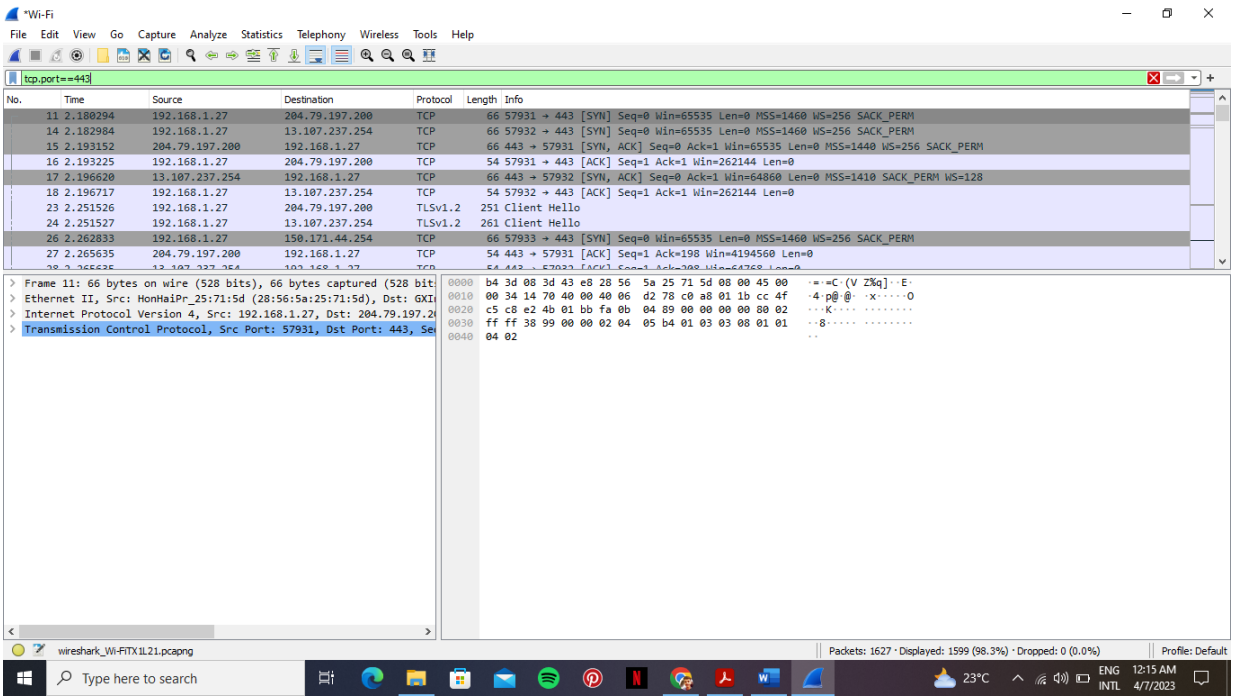

*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
smtp
No. Time Source Destination Protocol Length Info
771 57.595139 172.253.118.26 192.168.1.27 SMTP 140 S: 220 mx.google.com ESMTP l12-20020a170903120c00b0019a9834bb23si4307800plh.192 - gsmtp
912 87.208065 192.16 56 C: DATA fragment, 34 bytes
916 87.461307 172.25 148 S: 502 5.5.1 Unrecognized command. l12-20020a170903120c00b0019a9834bb23si4307800plh.192 - gsmtp
947 92.021024 192.16 56 C: helo
949 92.276889 172.25 236 S: 501-5.5.4 Empty HELO/EHLO argument not allowed, closing connection. | 5.5.4 https://support.google.com/mail/?…
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comments
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window
Frame 771: 140 bytes on wi
Ethernet II, Src: GXIntern
Internet Protocol Version
Transmission Control Proto
Simple Mail Transfer Proto
0000 28 56 5a 25 71 5d b4 3d 08 3d 43 e8 08 00 45 40 (VZ%q]·= ·=C···E@
0010 00 7e 17 25 00 00 7a 06 44 3a ac fd 76 1a c0 a8 ·~·%··z· D:··v···
0020 01 1b 00 19 e8 f0 d4 ef 5b d3 e9 64 3a 81 50 18 ·····[··d:·P·
0030 01 00 33 48 00 00 32 32 30 20 6d 78 2e 67 6f 6f ··3H··22 0 mx.goo
0040 67 6c 65 2e 63 6f 6d 20 45 53 4d 54 50 20 6c 31 gle.com ESMTP l1
0050 32 2d 32 30 30 32 30 61 31 37 30 39 30 33 31 32 2-20020a 17090312
0060 30 63 30 30 62 30 30 31 39 61 39 38 33 34 62 62 0c00b001 9a9834bb
0070 32 33 73 69 34 33 30 37 38 30 30 70 6c 68 2e 31 23si4307 800plh.1
0080 39 32 20 2d 20 67 73 6d 74 70 0d 0a 92 - gsm tp··
Simple Mail Transfer Protocol: Protocol
Packets: 1375 · Displayed: 5 (0.4%) · Dropped: 0 (0.0%)
Profile: Default
Type here to search
29°C
ENG INTL
8:02 PM 4/7/2023


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
smtp
No. Time Source Destination Protocol Length Info
771 57.595139 172.253.118.26 192.168.1.27 SMTP 140 S: 220 mx.google.com ESMTP l12-20020a170903120c00b0019a9834bb23si4307800plh.192 - gsmtp
912 87.208065 192.16 56 C: DATA fragment, 34 bytes
916 87.461307 172.25 148 S: 502 5.5.1 Unrecognized command. l12-20020a170903120c00b0019a9834bb23si4307800plh.192 - gsmtp
947 92.021024 192.16 56 C: helo
949 92.276889 172.25 236 S: 501-5.5.4 Empty HELO/EHLO argument not allowed, closing connection. | 5.5.4 https://support.google.com/mail/?…
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comments
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
DCCP Stream Ctrl+Alt+Shift+E
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream
QUIC Stream
SIP Call
Frame 771: 140 bytes on wi
Ethernet II, Src: GXIntern
Internet Protocol Version
Transmission Control Proto
Simple Mail Transfer Proto
0000 28 56 5a 25 71 5d b4 3d 08 3d 43 e8 08 00 45 40 (VZ%q]·= ·=C···E@
0010 00 7e 17 25 00 00 7a 06 44 3a ac fd 76 1a c0 a8 ·~·%··z· D:··v···
0020 01 1b 00 19 e8 f0 d4 ef 5b d3 e9 64 3a 81 50 18 ·····[··d:·P·
0030 01 00 33 48 00 00 32 32 30 20 6d 78 2e 67 6f 6f ··3H··22 0 mx.goo
0040 67 6c 65 2e 63 6f 6d 20 45 53 4d 54 50 20 6c 31 gle.com ESMTP l1
2-20020a 17090312
0c00b001 9a9834bb
23si4307 800plh.1
92 - gsm tp··
Simple Mail Transfer Protocol: Protocol
Packets: 1375 · Displayed: 5 (0.4%) · Dropped: 0 (0.0%)
Profile: Default
Type here to search
29°C
ENG INTL
8:02 PM 4/7/2023

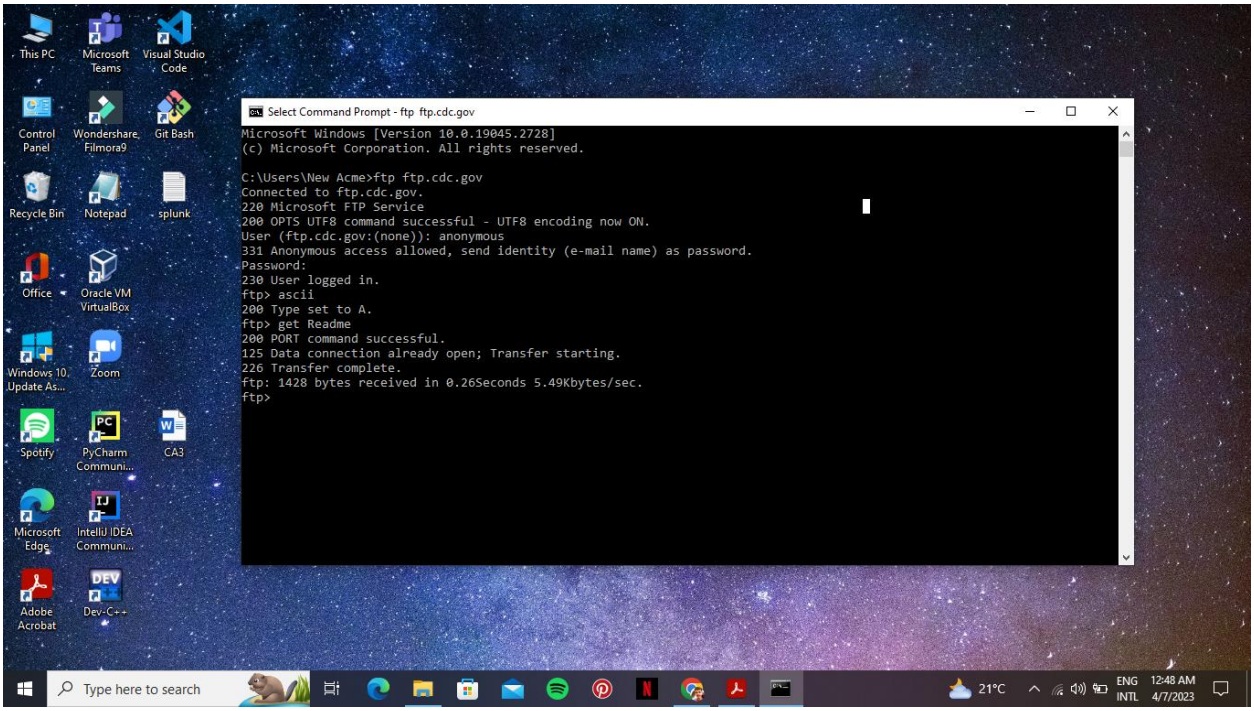

*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
smtp
No. Source Destination Protocol Length Info
771 57.595139 172.253.118.26 192.168.1.27 SMTP 140 S: 220 mx.google.com ESMTP l12-20020a170903120c00b0019a9834bb23si4307800plh.192 - gsmtp
912 87.208065 192.168.1.27 172.253.118.26 SMTP 56 C: DATA fragment, 34 bytes
916 87.461307 172.253.118.26 192.168.1.27 SMTP 148 S: 502 5.5.1 Unrecognized command. l12-20020a170903120c00b0019a9834bb23si4307800plh.192 - gsmtp
947 92.021024 192.168.1.27 172.253.118.26 SMTP 56 C: helo
949 92.276889 172.253.118.26 192.168.1.27 SMTP 236 S: 501-5.5.4 Empty HELO/EHLO argument not allowed, closing connection. | 5.5.4 https://support.google.com/mail/?…
Frame 771: 140 bytes on wire (1120 bits), 140 bytes captured (112
Ethernet II, Src: GXIntern_3d:43:e8 (b4:3d:08:3d:43:e8), Dst: Hon
Internet Protocol Version 4, Src: 172.253.118.26, Dst: 192.168.1.
Transmission Control Protocol, Src Port: 25, Dst Port: 59632, Seq
Simple Mail Transfer Protocol
0000 28 56 5a 25 71 5d b4 3d 08 3d 43 e8 08 00 45 40 (VZ%q].= .=C...E@
0010 00 7e 17 25 00 00 7a 06 44 3a ac fd 76 1a c0 a8 .~.%..z. D:..v...
0020 01 1b 00 19 e8 f0 d4 ef 5b d3 e9 64 3a 81 50 18 .... [..d:.P.
0030 01 00 33 48 00 00 32 32 30 20 6d 78 2e 67 6f 6f ..3H..22 0 mx.goo
0040 67 6c 65 2e 63 6f 6d 20 45 53 4d 54 50 20 6c 31 gle.com ESMTP l1
0050 32 2d 32 30 30 32 30 61 31 37 30 39 30 33 31 32 2-20020a 17090312
0060 30 63 30 30 62 30 30 31 39 61 39 38 33 34 62 62 0c00b001 9a9834bb
0070 32 33 73 69 34 33 30 37 38 30 30 70 6c 68 2e 31 23si4307 800plh.1
0080 39 32 20 2d 20 67 73 6d 74 70 0d 0a 92 - gsm tp..
Simple Mail Transfer Protocol: Protocol
Packets: 1375 · Displayed: 5 (0.4%) · Dropped: 0 (0.0%)
Profile: Default
Type here to search
29°C
ENG INTL
7:57 PM 4/7/2023

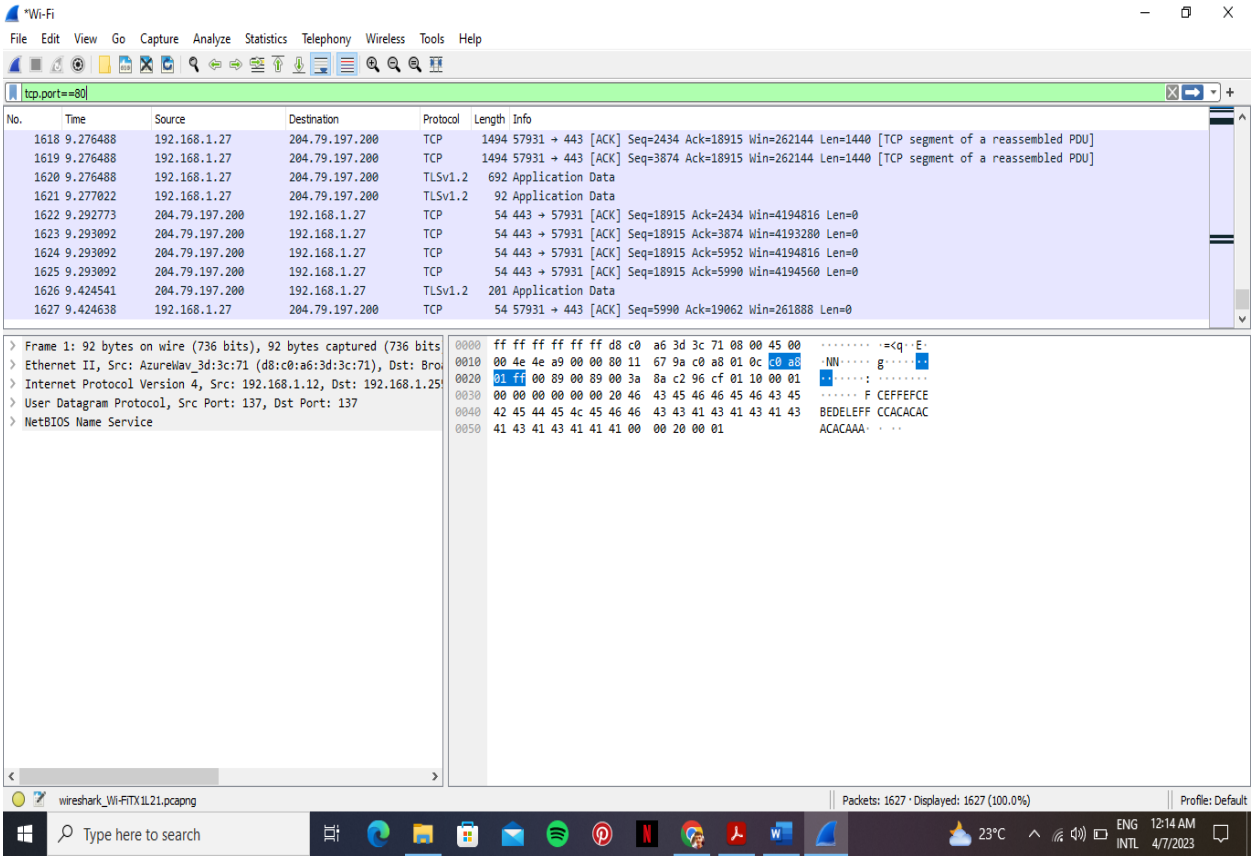
(Mac Address)


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source
5705 508.654412 192.16
5706 508.685082 192.16
5707 508.716705 192.16
5708 508.745702 192.16
5709 508.779579 192.16
5710 509.264629 192.16
5711 509.264629 fe80::
5712 510.812638 192.16
5713 512.211487 172.25
5714 512.211535 192.16
Telnet gmail-smtp-in.l.google.com
220 mx.google.com ESMTP i19-20020a17090320d300b0019953ab9cf2si3936223plb.138 - gsmtp
Frame 1: 103 bytes on wire
Ethernet II, Src: d2:f7:c6
Internet Protocol Version
User Datagram Protocol, Sr
Multicast Domain Name Syst
wireshark_Wi-FiZXQ221.pcapng
Packets: 5714 · Displayed: 5714 (100.0%)
Profile: Default
Type here to search
29°C
ENG INTL
7:52 PM 4/7/2023


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.port==443
No. Time Source Destination Protocol Length Info
11 2.180294 192.168.1.27 204.79.197.200 TCP 66 57931 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
14 2.182984 192.168.1.27 13.107.237.254 TCP 66 57932 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
15 2.193152 204.79.197.200 192.168.1.27 TCP 66 443 → 57931 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM
16 2.193225 192.168.1.27 204.79.197.200 TCP 54 57931 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
17 2.196620 13.107.237.254 192.168.1.27 TCP 66 443 → 57932 [SYN, ACK] Seq=0 Ack=1 Win=64860 Len=0 MSS=1410 SACK_PERM WS=128
18 2.196717 192.168.1.27 13.107.237.254 TCP 54 57932 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
23 2.251526 192.168.1.27 204.79.197.200 TLSv1.2 251 Client Hello
24 2.251527 192.168.1.27 13.107.237.254 TLSv1.2 261 Client Hello
26 2.262833 192.168.1.27 150.171.44.254 TCP 66 57933 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
27 2.265635 204.79.197.200 192.168.1.27 TCP 54 443 → 57931 [ACK] Seq=1 Ack=198 Win=4194560 Len=0
Frame 11: 66 bytes on wire (528 bits), 66 bytes captured (528 bit
Ethernet II, Src: HonHaiPr_25:71:5d (28:56:5a:25:71:5d), Dst: GXI
Internet Protocol Version 4, Src: 192.168.1.27, Dst: 204.79.197.2
Transmission Control Protocol, Src Port: 57931, Dst Port: 443, Se
0000 b4 3d 08 3d 43 e8 28 56 5a 25 71 5d 08 00 45 00
0010 00 34 14 70 40 00 40 06 d2 78 c0 a8 01 1b cc 4f
0020 c5 c8 e2 4b 01 bb fa 0b 04 89 00 00 00 00 80 02
0030 ff ff 38 99 00 00 02 04 05 b4 01 03 03 08 01 01
0040 04 02
wireshark_Wi-FiTX1L21.pcapng
Packets: 1627 · Displayed: 1599 (98.3%) · Dropped: 0 (0.0%)
Profile: Default
Type here to search
23°C
ENG INTL
12:15 AM 4/7/2023

(RDP)


This PC
Microsoft Teams
Visual Studio Code
Control Panel
Wondershare Filmora9
Git Bash
Recycle Bin
Notepad
splunk
Office
Oracle VM VirtualBox
Windows 10 Update As...
Zoom
Spotify
PyCharm Communi...
CA3
Microsoft Edge
IntelliJ IDEA Communi...
Adobe Acrobat
Dev-C++
Select Command Prompt - ftp ftp.cdc.gov
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.
C:\Users\New Acme>ftp ftp.cdc.gov
Connected to ftp.cdc.gov.
220 Microsoft FTP Service
200 OPTS UTF8 command successful - UTF8 encoding now ON.
User (ftp.cdc.gov:(none)): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
ftp> ascii
200 Type set to A.
ftp> get Readme
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
ftp: 1428 bytes received in 0.26Seconds 5.49Kbytes/sec.
ftp>
Type here to search
21°C
ENG INTL
12:48 AM 4/7/2023


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.port==80
No. Time Source Destination Protocol Length Info
1618 9.276488 192.168.1.27 204.79.197.200 TCP 1494 57931 → 443 [ACK] Seq=2434 Ack=18915 Win=262144 Len=1440 [TCP segment of a reassembled PDU]
1619 9.276488 192.168.1.27 204.79.197.200 TCP 1494 57931 → 443 [ACK] Seq=3874 Ack=18915 Win=262144 Len=1440 [TCP segment of a reassembled PDU]
1620 9.276488 192.168.1.27 204.79.197.200 TLSv1.2 692 Application Data
1621 9.277022 192.168.1.27 204.79.197.200 TLSv1.2 92 Application Data
1622 9.292773 204.79.197.200 192.168.1.27 TCP 54 443 → 57931 [ACK] Seq=18915 Ack=2434 Win=4194816 Len=0
1623 9.293092 204.79.197.200 192.168.1.27 TCP 54 443 → 57931 [ACK] Seq=18915 Ack=3874 Win=4193280 Len=0
1624 9.293092 204.79.197.200 192.168.1.27 TCP 54 443 → 57931 [ACK] Seq=18915 Ack=5952 Win=4194816 Len=0
1625 9.293092 204.79.197.200 192.168.1.27 TCP 54 443 → 57931 [ACK] Seq=18915 Ack=5990 Win=4194560 Len=0
1626 9.424541 204.79.197.200 192.168.1.27 TLSv1.2 201 Application Data
1627 9.424638 192.168.1.27 204.79.197.200 TCP 54 57931 → 443 [ACK] Seq=5990 Ack=19062 Win=261888 Len=0
Frame 1: 92 bytes on wire (736 bits), 92 bytes captured (736 bits
Ethernet II, Src: AzureWav_3d:3c:71 (d8:c0:a6:3d:3c:71), Dst: Bro
Internet Protocol Version 4, Src: 192.168.1.12, Dst: 192.168.1.25
User Datagram Protocol, Src Port: 137, Dst Port: 137
NetBIOS Name Service
0000 ff ff ff ff ff ff d8 c0 a6 3d 3c 71 08 00 45 00
0010 00 4e 4e a9 00 00 80 11 67 9a c0 a8 01 0c c0 a8
0020 01 ff 00 89 00 89 00 3a 8a c2 96 cf 01 10 00 01
0030 00 00 00 00 00 00 20 46 43 45 46 46 45 46 43 45
0040 42 45 44 45 4c 45 46 46 43 43 41 43 41 43 41 43
0050 41 43 41 43 41 41 41 00 00 20 00 01
wireshark_Wi-FiTX1L21.pcapng
Packets: 1627 · Displayed: 1627 (100.0%)
Profile: Default
Type here to search
23°C
ENG INTL
12:14 AM 4/7/2023

Reference/Bibliography

1. Baryamureeba V, Tushabe, F (2004) The enhanced digital investigation process model. in Digital Forensic Research Workshop, Utica, New York, USA.
2. Yasinsac A, Manzano Y(2002)Honeytraps, a network forensic tool, in Sixth Multi-Conference on Systemics, Cybernetics and Informatics.
3. https://www.wireshark.org/download.html accessed on 5/04/2023
4. Digital Forensics Research Workshop. "A road map for digital forensics research, technical report", DFRWS, November 2001, pp. 15-20.
5. Institute for Security Technology Studies. "Law enforcement tools and technologies for investigating cyber attacks: Gap analysis report." http://www.ists.dartmouth.edu, February 2004.
6. Sommer P. "Intrusion detection systems as evidence". http://www.raid-symposium.org/raid98/Prog-RAID98/Full- Papers/Sommer-text.pdf, 2002-04-05.