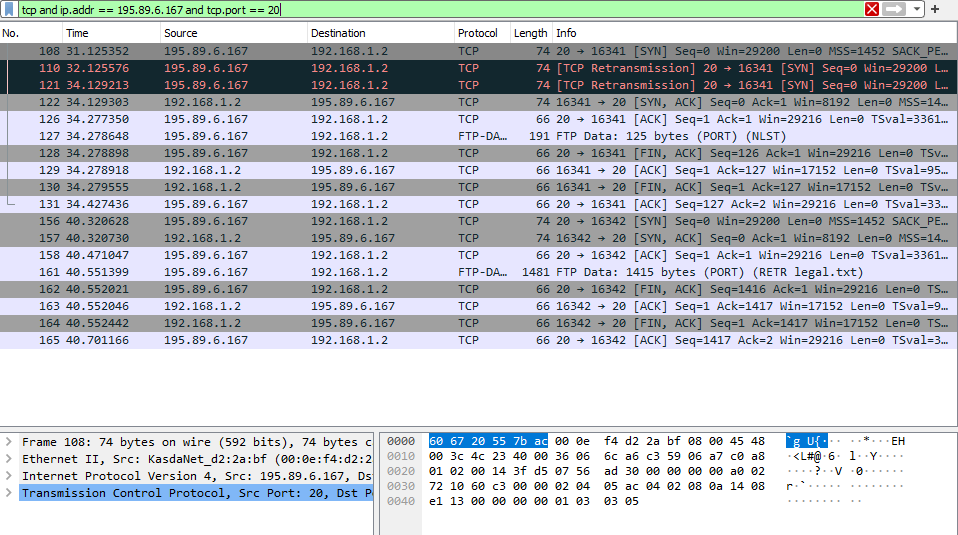
**Lab-4**

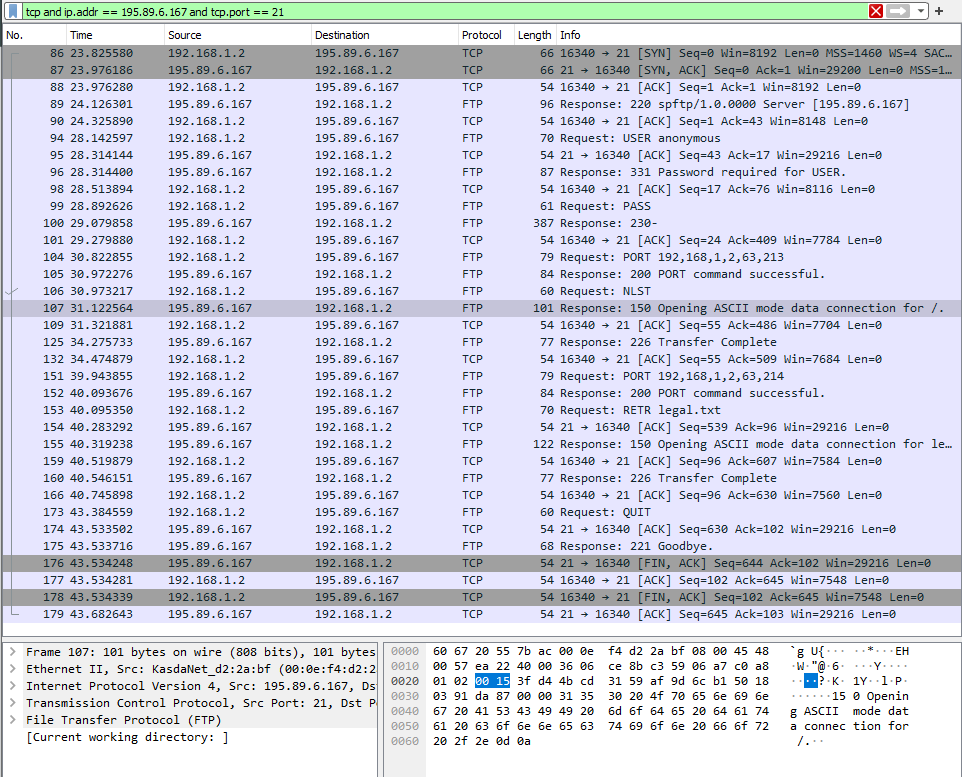
# **Lab statement 1**

## **1)** **FTP uses two port numbers: 20 and 21. Apply tcp.port==20 and tcp.port==21. Analyze the result and write down the purposes of these two ports for FTP.**

* *Port: 20* is used to send data through the network using FTP-DATA protocol.

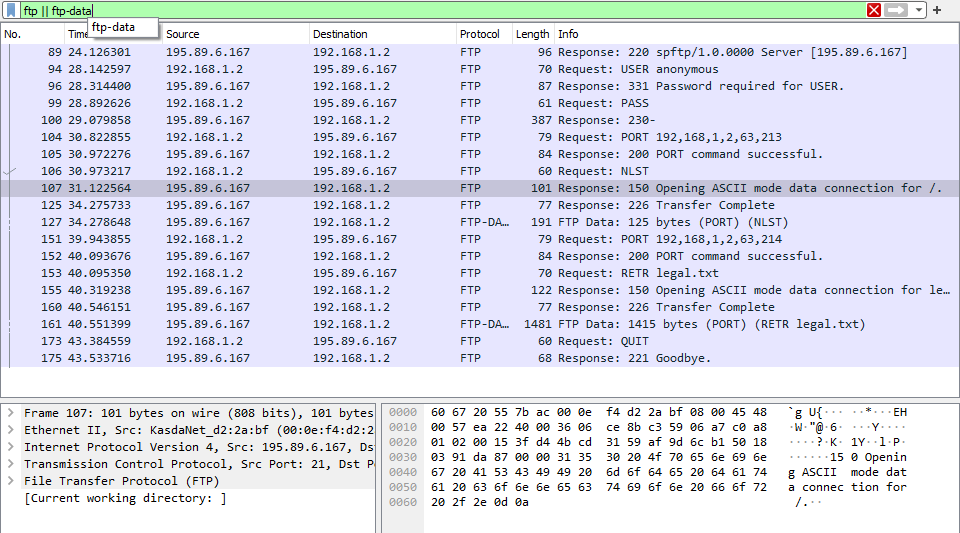


* Port : 21 is used to send control bits using FTP protocol and to establish the connection.



## **2) Filter out each packet using either FTP or FTP-DATA Protocol (using ftp || ftp-data filter). Mention each packet number and its purpose with reference to request made and response received in the above mentioned FTP Session in command line to get file legal.txt (screenshot show above). Also look for Response Code and Response Arg in the FTP Header for each packet**

## 



**Packet 89:** server tells that service is ready for new user

**Packet 94:** client enters anonymous in User

**Packet 96:** server responds with user name okay, need password

**Packet 99:** client enters empty password

**Packet 100:** server accepts the login credentials and process with displaying

230-

230- ---------------------------------------------------------------------------

230- WARNING: This is a restricted access system. If you do not have explicit

230- permission to access this system, please disconnect immediately!

230 ----------------------------------------------------------------------------

**Packet 104:** client asks server to send the data on IP:192.168.1.2 and Port:16341

**Packet 105:** server tells the client that port:16341 is connected successfully.

**Packet 106:** client enters ls command which sends NLST request to the server

**Packet 107:** server tells the client:  
 Opening ASCII mode data connection for /.

**Packet 125:** server tells that transfer was successful and is closing the connection

**Packet 127:** server sends 11 line-based text data to the client which was asked using NLST

**Packet 151:** client asks server to send the data on IP:192.168.1.2 and Port:16342

**Packet 152:** server tells the client that port:16342 is connected successfully.

**Packet 153:** client asks to download legal.txt file by sending request RETR legal.txt

**Packet 155:** server tells the client:  
 Opening ASCII mode data connection for /.

**Packet 160:** server tells that transfer was successful an is closing the connection

**Packet 161:** server sends 1415 bytes long legal.txt file to client

**Packet 173:** client asks to end the communication using QUIT

**Packet 175:** server ends the connection and says Goodbye to the client

**Lab statement 2**

|  |  |  |
| --- | --- | --- |
| 1 | Are ICMP messages sent over UDP or TCP? | ICMP is a connectionless protocol. Messages are not sent over UDP or TCP |
| 2 | What is the link-layer (e.g; Ethernet) address of the host? | Link-layer address of hosts are:  **Host 1:** Tp-LinkT\_87:05:fe (c0:4a:00:87:05:fe)  **Host 2:** IntelCor\_55:7b:ac (60:67:20:55:7b:ac) |
| 3 | Which kind of request is sent through these ICMP packets? | Echo (ping) request are sent through these ICMP packets to test connectivity |
| 4 | How many requests are sent through the host? | Four Echo requests are sent through the host in packet no 48, 77, 101, 135 |
| 5 | What is the IP address of your host? What is the IP address of the destination host? | IP address of host: 192.168.33.110  IP address of dest: 172.217.27.36 |
| 6 | Why is it that an ICMP packet does not have source and destination port numbers? | ICMP packets do not have source and destination port numbers because ICMP operates at the network layer of the OSI model, while source and destination port numbers are associated with transport layer protocols like TCP and UDP. |
| 7 | What values in the ICMP request message differentiate this message from the ICMP reply message? | The “ttl” value differentiates ICMP request message from ICMP reply message  For request: ttl = 128  For reply: ttl = 52 |
| 8 | Examine one of the ping request packets sent by your host. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields? | ICMP type: 8  ICMP code: 0  Moreover it has Checksum, Identifier, Sequence number and Data.  Number of bytes of checksum: 2  Number of bytes of identifier: 2  Number of bytes of sequence number: 2 |
| 9 | Examine the corresponding ping reply packet. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and  identifier fields? | ICMP type: 0  ICMP code: 0  Moreover it has Checksum, Identifier, Sequence number and Data.  Number of bytes of checksum: 2  Number of bytes of identifier: 2  Number of bytes of sequence number: 2 |
| 10 | Examine the packet no 56. What are the ICMP type and code numbers? Why is the IP and TCP  Header included in the ICMP Header? What does these headers depict? | ICMP type: 3  ICMP code: 3  It includes a bit of the original packet that had problems. This is like which packet had issues while traveling. It helps the sender figure out what went wrong. The included TCP header in the ICMP message helps identify which communication protocol and specific port might have had problems. |

# 

# **Lab statement 3**

**a. What is a packet sniffer in the context of computer networks, and how does it work briefly?**

A packet sniffer, in the context of computer networks, is a software or hardware tool designed to capture and analyze network traffic. It allows users to inspect the data packets traveling over a network, which can be useful for various purposes, including network troubleshooting, security monitoring, and performance analysis.

**b. Explain the primary purpose of using packet sniffers like Wireshark in network troubleshooting and monitoring.**

Packet sniffers like Wireshark are primarily used for network troubleshooting and monitoring. They help network managers and technicians diagnose underlying problems in their networks. By capturing and analyzing packets of data that flow through a particular network, sniffers can provide insights into network performance, identify bottlenecks, and detect security vulnerabilities. Sniffers can be configured to capture all possible packets or only those containing specific data components. They are valuable tools for monitoring network traffic, identifying abnormal behavior, and ensuring the smooth operation of computer networks.

**c. In Wireshark, what is the significance of a “capture filter” and when might you use it during packet capture?**

In Wireshark, a capture filter is a feature that allows you to selectively capture network traffic based on specific criteria, such as source or destination IP addresses, port numbers, or protocols. It helps you focus on capturing only the packets that are relevant to your analysis, reducing the amount of captured data and improving performance.

**d. Describe a potential ethical concern associated with the use of packet sniffers in network security and privacy.**

One potential ethical concern associated with the use of packet sniffers in network security and privacy is the invasion of privacy. Packet sniffers can capture and analyze network traffic, including sensitive information such as passwords, personal data, and confidential communications. If used maliciously or without proper authorization, packet sniffers can be used to intercept and exploit this information for nefarious purposes, violating individuals’ privacy rights and compromising their security.

**e. What are some common protocols or technologies that Wireshark can analyze and decode within captured network packets? Provide examples of a few.**

Wireshark is a powerful network protocol analyzer that can analyze and decode various protocols and technologies within captured network packets. Here are a few examples:

* HTTP
* TCP/IP
* DNS.
* FTP
* SMTP

These are just a few examples of the many protocols and technologies that Wireshark can analyze and decode. It supports a wide range of network protocols, making it a valuable tool for network troubleshooting and analysis.