# FAST NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES.Image result for FAST NUCES logo

DEPARTMENT OF COMPUTER SCIENCE(CS)

SEMESTER SPRING 2023

BATCH 2020

**COMPUTER NETWORKS**

PROJECT TOPIC:

*A Network Sniffer*

GROUP MEMBERS:

* Ayesha Zia (20k-0414)
* Syeda Ravia Ejaz (20k-0246)

**Abstract:**

A network sniffer is a tool that captures and analyzes network traffic in real-time. It can help network administrators to troubleshoot network issues, monitor bandwidth usage, and identify security threats. This project report discusses the implementation and usage of a network sniffer tool using Python programming language and Scapy library. The tool captures and analyzes network packets, extracts information such as the source and destination IP addresses, protocol type, packet size, and payload data, and performs advanced analysis such as identifying network protocols and detecting network anomalies. The tool can also visualize network traffic using graphs and charts to help network administrators understand network traffic patterns and identify anomalies. The network sniffer tool is a powerful tool that can provide useful information for network administrators to optimize network performance and ensure network security.

**Introduction:**

Network Sniffer is a tool that captures and analyzes network traffic in real-time. It can help network administrators to troubleshoot network issues, monitor bandwidth usage, and identify security threats. In this report, we will discuss the implementation and usage of a network sniffer.

**Objective:**

The objective of this project is to implement a network sniffer tool that can capture network traffic, analyze it, and provide useful information for network administrators.

**Methodology:**

The network sniffer tool is implemented using Python programming language and Scapy library. Scapy is a powerful packet manipulation tool that can be used to capture, decode, and analyze network packets.

The implementation of the network sniffer tool involves the following steps:

1. **Capturing Network Traffic:**

The first step is to capture network traffic. Scapy provides a built-in function called 'sniff' that can capture network packets. We can specify the network interface to capture packets on and the number of packets to capture. We can also filter packets based on their source or destination IP addresses, protocol type, or port numbers.

1. **Analyzing Network Traffic:**

After capturing network packets, we need to analyze them. We can extract information such as the source and destination IP addresses, protocol type, packet size, and payload data. We can also perform more advanced analysis such as identifying network protocols, detecting network anomalies, and identifying security threats.

1. **Visualizing Network Traffic:**

Finally, we can visualize network traffic using graphs and charts. We can use tools such as Matplotlib to create graphs and charts that can help network administrators to understand network traffic patterns and identify anomalies.

**Results:**

The network sniffer tool can be used to capture and analyze network traffic in real-time. The tool can provide useful information such as the source and destination IP addresses, protocol type, packet size, and payload data. The tool can also perform more advanced analysis such as identifying network protocols, detecting network anomalies, and identifying security threats. The tool can also visualize network traffic using graphs and charts.

**Conclusion:**

In conclusion, the network sniffer tool is a powerful tool that can help network administrators to troubleshoot network issues, monitor bandwidth usage, and identify security threats. The tool can be implemented using Python programming language and Scapy library. The tool can capture and analyze network traffic in real-time and provide useful information for network administrators.

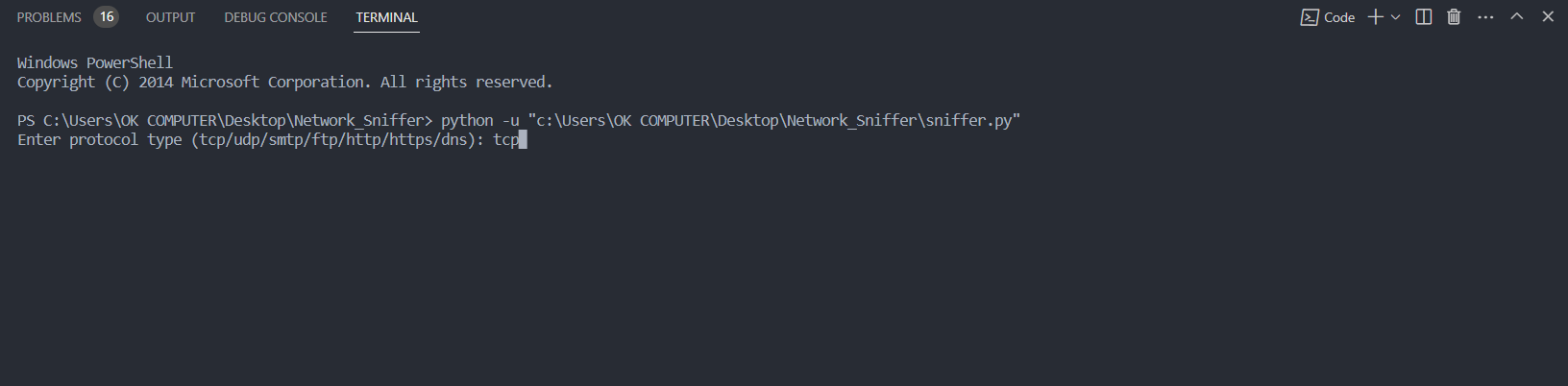


Figure 1. Entering the protocol name

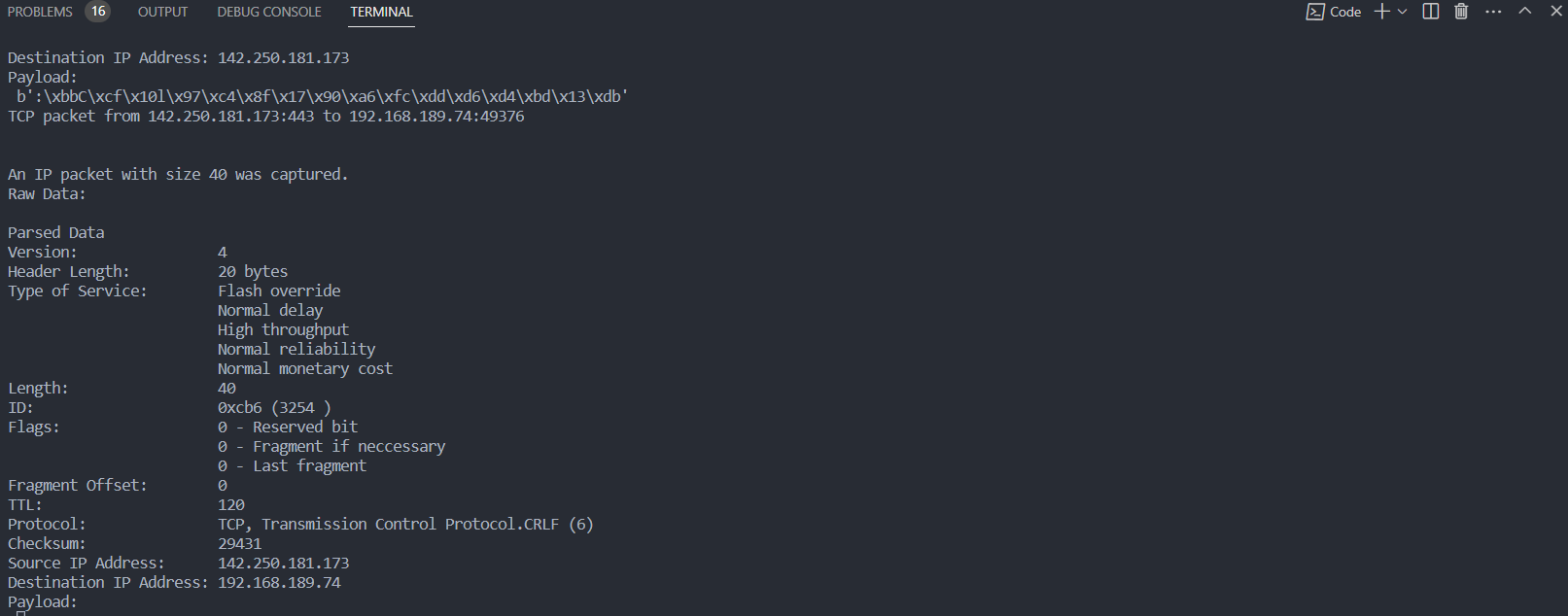


Figure 2. TCP

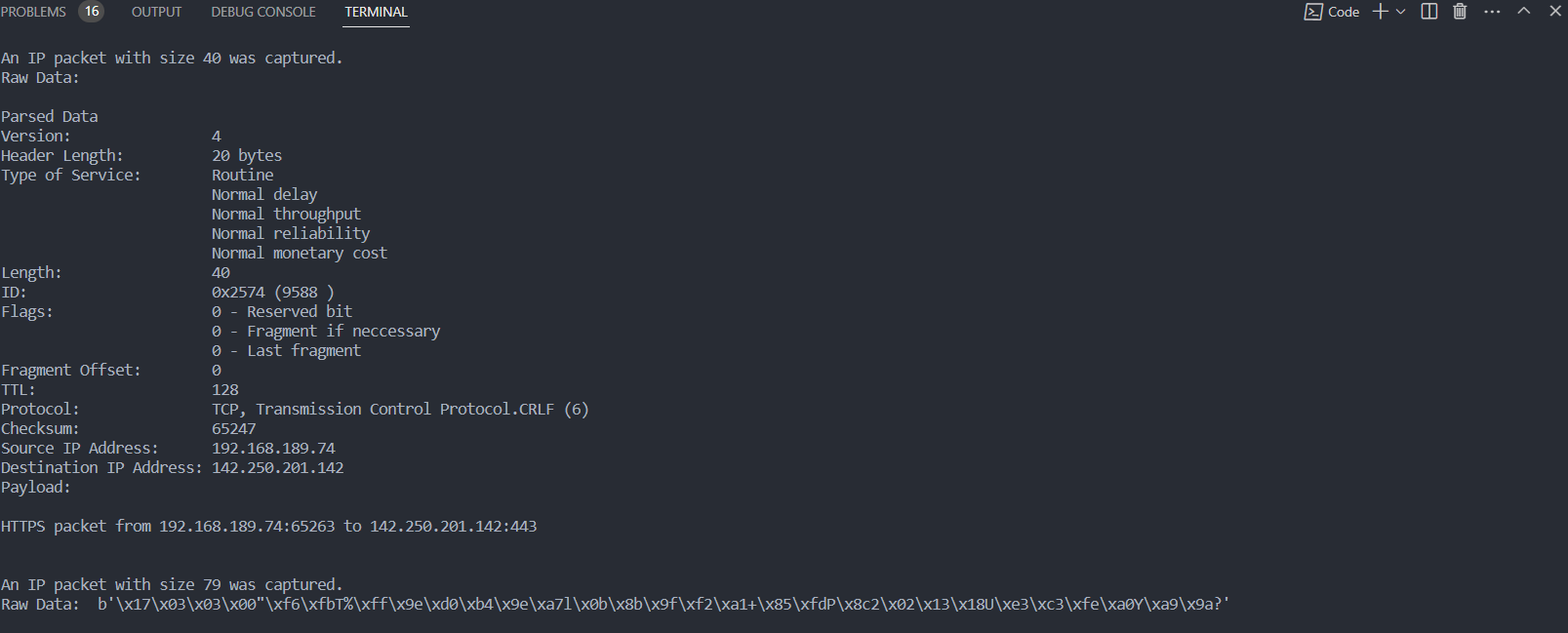


Figure 3. HTTPS



Figure 4.1 HTTP

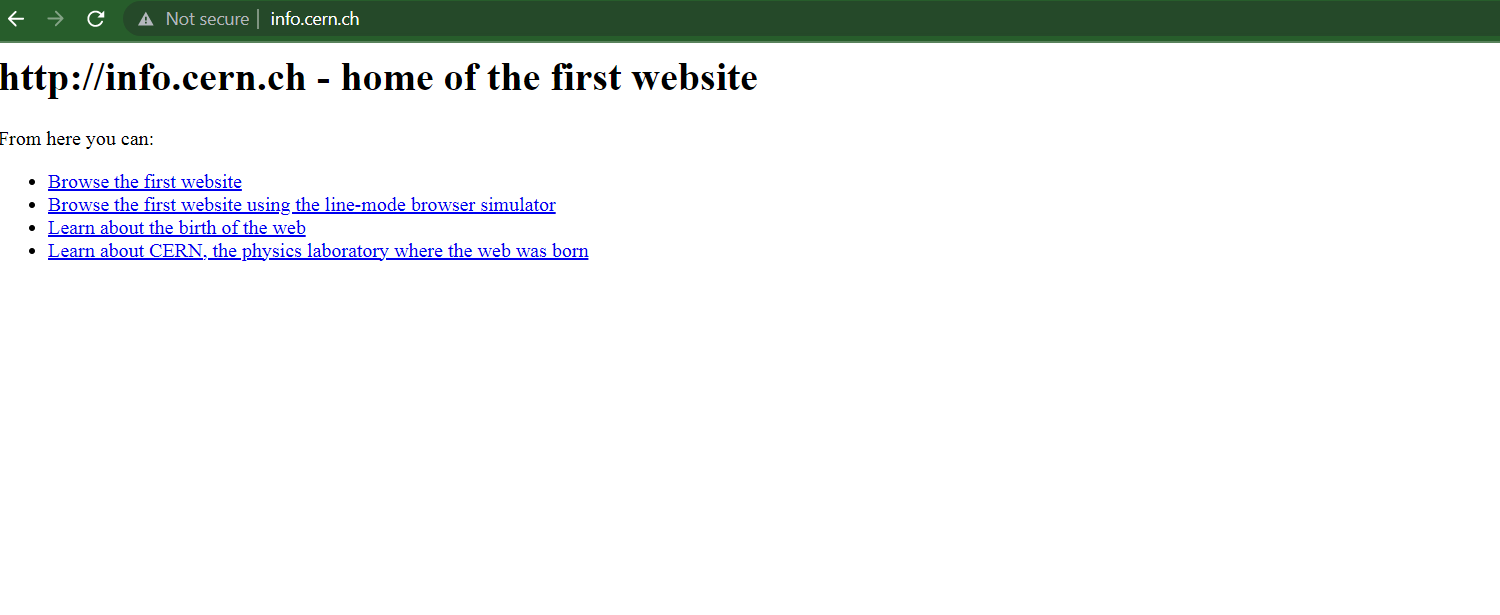


Figure 4.2. HTTP

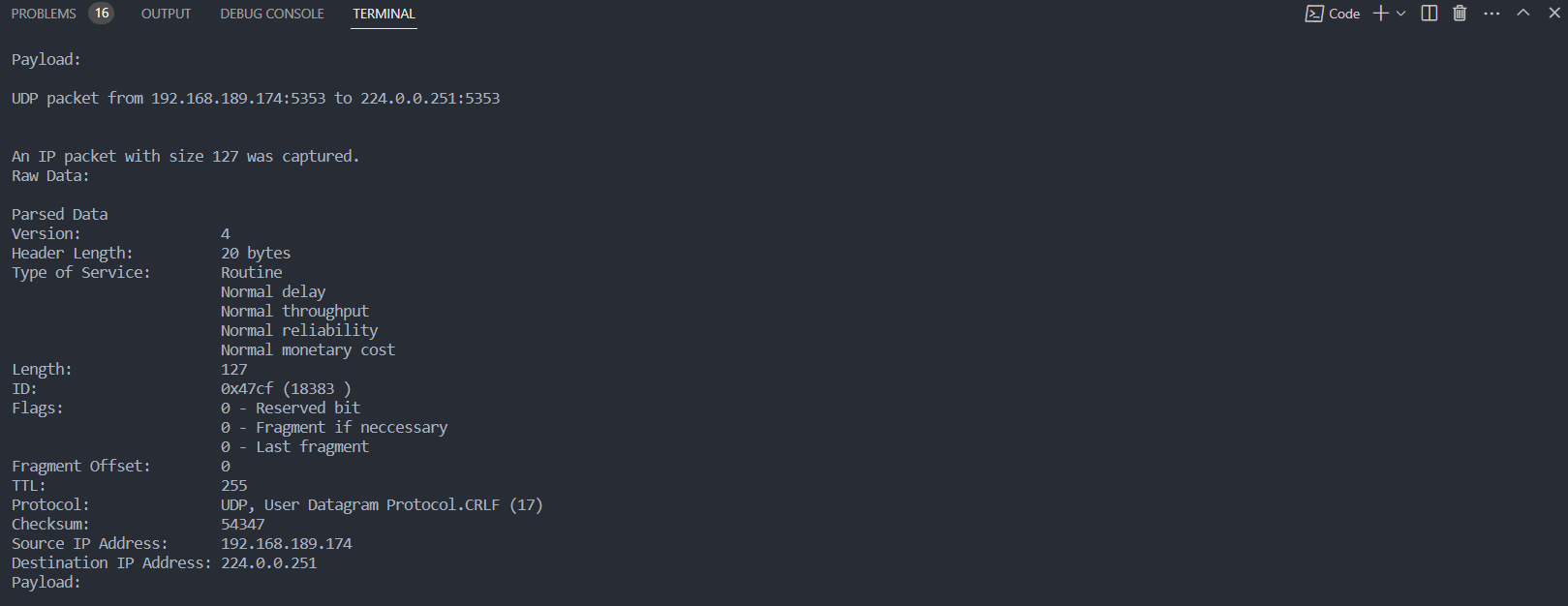


Figure 5. UDP

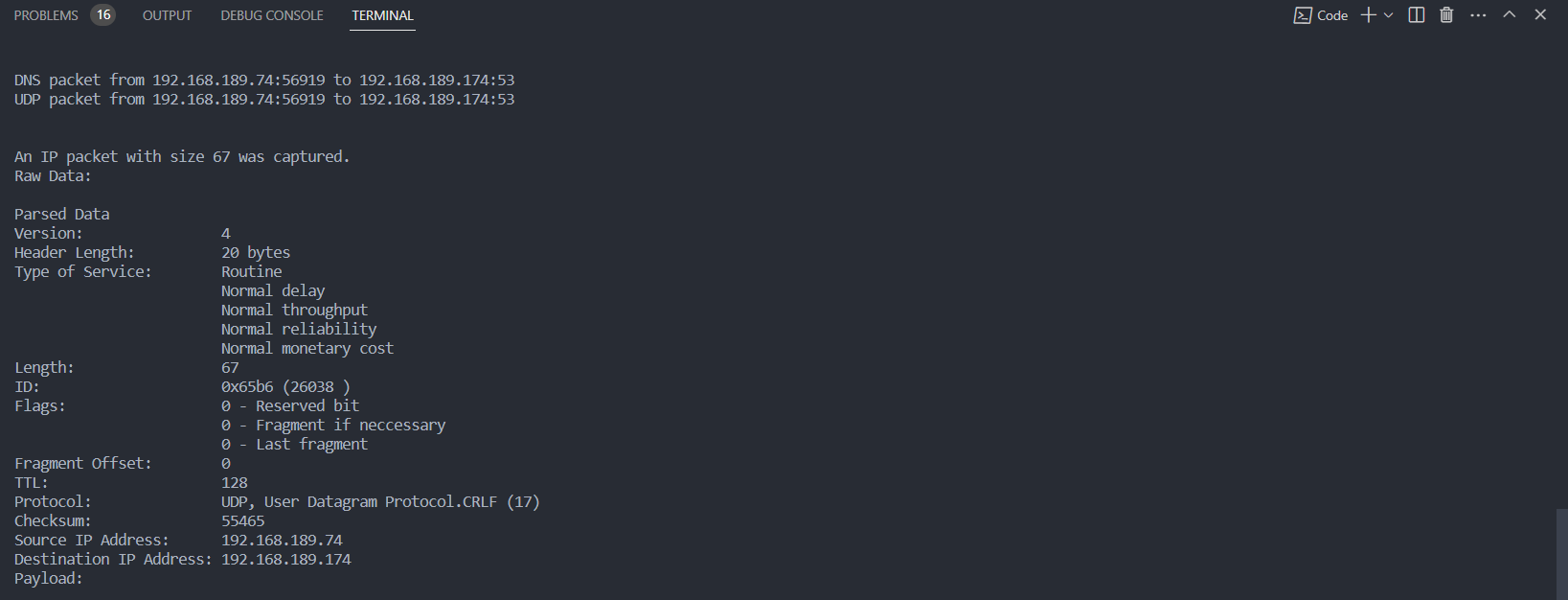


Figure 6. DNS

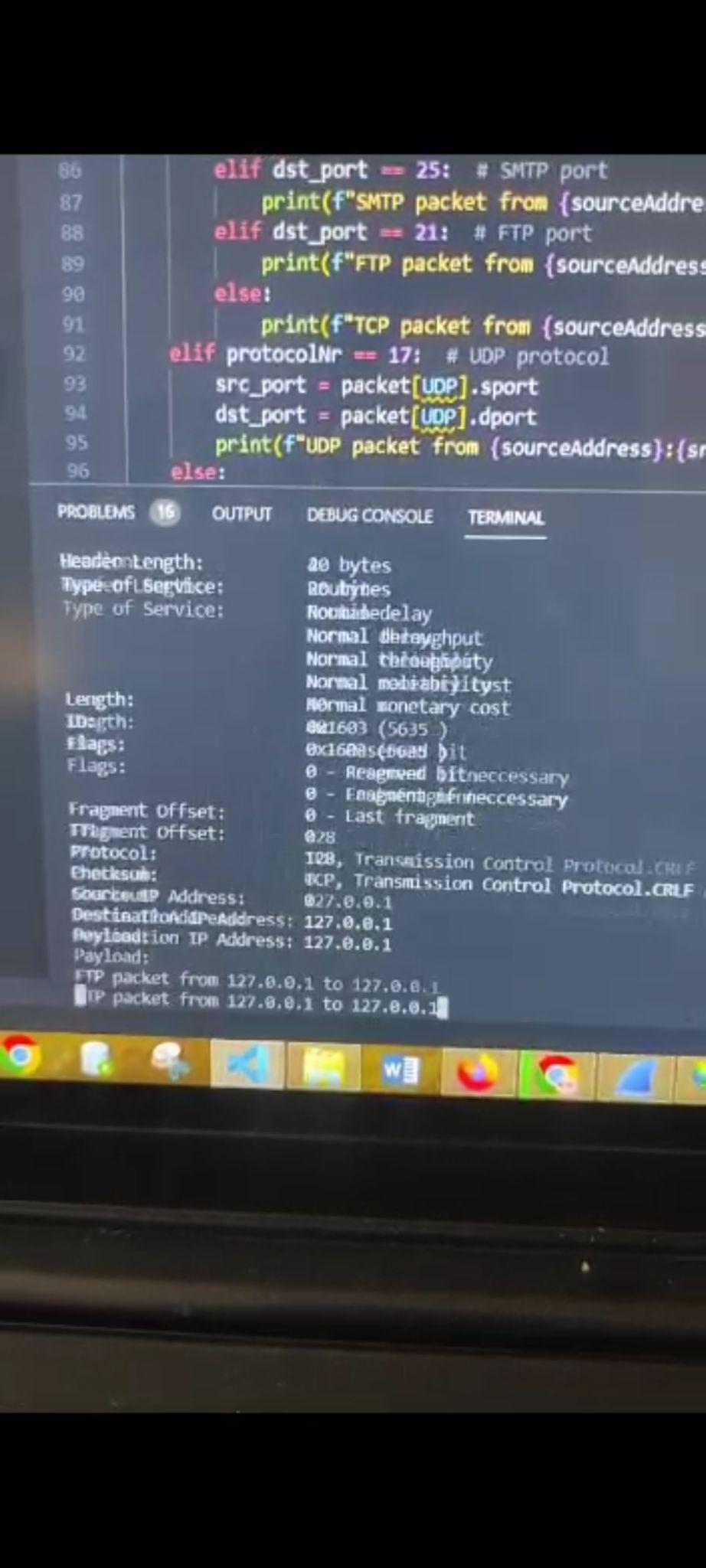


Figure 7. FTP

**References:**

1. M. Masud and A. Ali, "Network Traffic Analysis and Packet Sniffing Techniques," 2016 International Conference on Networking Systems and Security (NSysS), Dhaka, 2016, pp. 1-6, doi: 10.1109/NSysS.2016.7748967.
2. A. Naeem, N. Naseer, and A. Masood, "An Overview of Network Sniffing Attacks," 2019 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2019, pp. 1-6, doi: 10.1109/iCoMET.2019.8723664.
3. P. Inglesant and S. Sasse, "The true cost of cybercrime," IEEE Security & Privacy, vol. 11, no. 1, pp. 55-62, Jan.-Feb. 2013, doi: 10.1109/MSP.2012.111.
4. M. A. Shah and M. Y. Javed, "Packet sniffing in wireless networks: A survey," 2015 8th International Conference on Contemporary Computing (IC3), Noida, 2015, pp. 432-437, doi: 10.1109/IC3.2015.7346721.
5. S. R. Kadam, S. P. Khilari, and A. M. Kshirsagar, "Packet Sniffing and Its Countermeasures," 2018 Second International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2018, pp. 788-793, doi: 10.1109/ICCMC.2018.8448207.