Genesis Grant

CTEC 402

In Class Assignment 7

Part 1:

A packet sniffer is a tool used to capture and analyze data packets as they travel across a network. To complete this task, I chose to implement a Java-based packet sniffer, using resources like CodeWithC for guidance. First, I set up my development environment by installing a Java Development Kit (JDK) and an IDE like IntelliJ IDEA. After that, I downloaded libraries such as JPCAP, which are essential for interacting with network interfaces and capturing packets. I then created a project in my IDE and wrote code to access the network adapter, capture packets, and extract details such as IP addresses, protocols, and data payloads. Once the code was complete, I ran the program to capture and display live network traffic. To validate its functionality, I compared the results with outputs from tools like Wireshark. I also explored Python for packet sniffing, using the Scapy library and video tutorials, which made understanding packet structures more accessible.

Part 2:

Microsoft Message Analyzer is a powerful tool for troubleshooting network issues by capturing and analyzing network traffic and protocol messages. After downloading and installing the software, I started by launching it and creating a new capture session. I selected my network interface as the source for capturing live traffic, but the tool also supports importing pre-captured files in formats like .pcap. Once the session began, the program displayed captured packets in an organized table, showing details like the source, destination, protocol type, and timestamps. I applied filters to focus on specific protocols, which helped make the data easier to analyze. Double-clicking on packets provided a detailed view of their structure and payload. The program also offered visualization tools like charts to understand traffic patterns better. This step-by-step process allowed me to troubleshoot network problems effectively and gain insights into how different devices communicate.