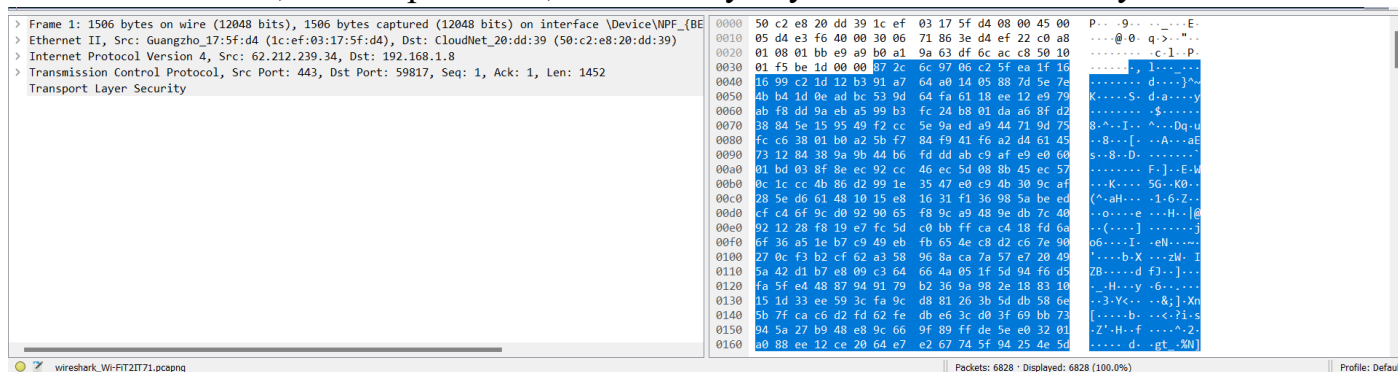
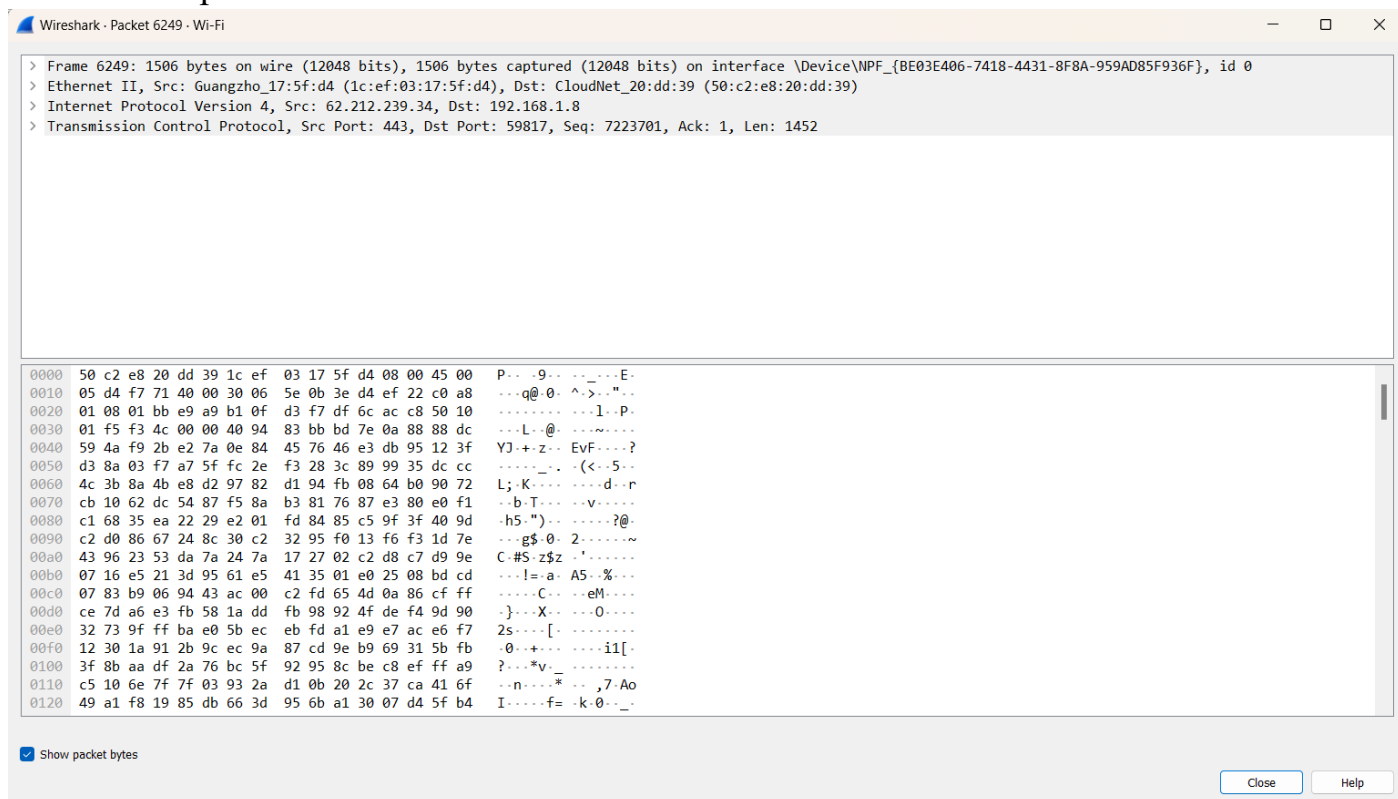


5. **Monitor Packet Capture:** Let Wireshark capture packets for a specific duration or until you have gathered sufficient data for analysis.
6. **Stop Packet Capture:** Stop the packet capture process in Wireshark once you have captured enough packets.

7. Analyze Captured Packets: Explore the captured packets within Wireshark to analyze network traffic, dissect protocols, and identify any anomalies or security issues.



8. Interpret Results: Interpret the results obtained from Wireshark analysis, paying attention to packet details, protocols used, source and destination addresses, and any observed patterns or abnormalities.



9. Document Findings: Record the details of the analysis, including notable findings, suspicious activities, or any other relevant observations.

Result:

By installing and utilizing Wireshark, we successfully captured and analyzed packets from the network traffic. After launching Wireshark, we captured packets using the specified capture filters. Once the capture was complete, we analyzed the captured packets within Wireshark, dissecting protocols, and examining packet details. The analysis yielded notable findings, including unusual network behaviors, suspicious traffic patterns, or security issues, which were documented for further investigation and action.

Conclusion:

Wireshark is a powerful network protocol analyzer that allows for the detailed analysis of captured packets. Through its intuitive interface and robust features, we were able to capture and examine network traffic, dissect protocols, and identify potential security issues. Wireshark proves to be a valuable tool for network troubleshooting, performance analysis, and security monitoring.

Future Scope:

1. Deep packet inspection: Utilize Wireshark's advanced features to perform deep packet inspection, including extracting and analyzing application-layer data, payloads, and specific protocol behaviors.
2. Network forensics analysis: Apply Wireshark's capabilities for network forensics investigations, including identifying attack patterns, malicious activities, or evidence of data breaches.
3. Statistical analysis: Leverage Wireshark's statistical tools and plugins to perform statistical analysis on captured packets, such as traffic patterns, packet sizes, or network latency.
4. Integration with other tools: Explore the integration of Wireshark with complementary tools, such as intrusion detection systems (IDS), security information and event management (SIEM) platforms, or traffic visualization tools, for enhanced network analysis and correlation of events.
5. Stay updated with Wireshark: Regularly update Wireshark to benefit from the latest features, protocol dissectors, and security enhancements, ensuring efficient and accurate packet analysis.