Wireshark Network Tracking with Google Maps

Name: Abhik Das
Registration Number: 22BCE8616
College Email: abhik.22bce8616@vitapstudent.ac.in
Personal Email: abhikdas0811@gmail.com

Documentation

---

1. INTRODUCTION

The growth of global internet infrastructure has led to massive data exchange between devices, applications, and services. This rapid expansion brings not only performance and connectivity benefits, but also increased vulnerability to cyber threats. Network administrators and cybersecurity professionals are constantly faced with the challenge of identifying and mitigating unauthorized or malicious network traffic. Traditional tools like Wireshark are powerful for packet-level inspection but require significant expertise and often lack visual summaries of geographical trends or threat patterns.

The project titled 'Wireshark Network Tracking with Google Maps' addresses this challenge by bridging the gap between raw network data and human-readable, geographically-visualized intelligence. The tool offers a GUI-based solution that simplifies the process of analyzing packet captures, extracting IP addresses, correlating them with threat intelligence sources, and plotting them on an interactive world map. This provides a clearer understanding of external connections, potential attacks, and their geographic origins.

2. EXISTING SOLUTIONS

Existing network monitoring tools like Wireshark and Tshark offer in-depth capabilities for packet capture and protocol analysis. These tools are widely used by professionals for detailed investigations but require users to manually interpret packet data. Visual tools like Splunk or Kibana, when integrated with network logs, offer dashboards and alerts but often require complex setup and enterprise-level infrastructure.

Threat intelligence platforms like AbuseIPDB and VirusTotal are valuable for determining the reputation of IP addresses. However, they are typically standalone services and not integrated with visual maps. Some open-source tools allow geolocation of IPs, but they lack integration with CSV data exported from Wireshark or require programming knowledge.

Our solution builds on these by combining Wireshark output, threat intelligence APIs, and geolocation mapping into a single, easy-to-use application.

3. PROPOSED SOLUTIONS

This project introduces a Python-based GUI that parses CSV data from Wireshark, queries IP reputation from AbuseIPDB and VirusTotal, fetches geolocation using public APIs, and plots the results on an interactive Folium map. The user can filter traffic by country and analyze protocol distribution using pie and bar charts. The system supports real-time threat scoring and provides an intuitive, accessible visualization layer over traditional packet data.

The tool also provides analytics, such as protocol usage distribution and country-wise IP origin charts, through intuitive visualizations like bar and pie charts.
Key features of the proposed system include:

- A Tkinter-based user interface for easy CSV import and interaction.
- Integration with Folium for generating interactive world maps with IP markers.
- Real-time threat lookup using AbuseIPDB and VirusTotal APIs.
- Analytics section to display protocol distribution and malicious IP statistics.
- Country-based filtering to focus on specific geographic areas.

4. SYSTEM REQUIREMENTS AND TOOLS

To run this application, the following software and hardware components are required:

Hardware:

- Dual-core CPU or better (Intel i3/i5/i7 or AMD equivalent)
- Minimum 4 GB RAM (8 GB recommended)
- 200 MB free disk space
- Internet access for API calls

Software:

- Operating System: Windows 10/11, macOS 10.15+, or Linux (Ubuntu 18.04+)
- Python 3.7 or higher
- Required Python libraries: pandas, matplotlib, folium, requests, geopy
- GUI: Tkinter
- Data Source: Wireshark (exported in CSV format)

APIs and External Services:

- AbuseIPDB and VirusTotal (API keys required for usage)

Other:

- CSV files exported from Wireshark should include Source and Destination IP columns for analysis.

5. EXECUTION AND RESULTS

The system starts by importing CSV files generated from Wireshark packet captures. The graphical interface allows users to browse for files, which are then parsed using the pandas library to extract relevant IP addresses.

These IPs are sent to VirusTotal and AbuseIPDB for threat evaluation. Simultaneously, geolocation APIs fetch the country and city details for each IP. The results are mapped using Folium, with different marker colors indicating IP status (malicious, unknown, clean).

The user can also visualize additional charts:

- A pie chart shows the proportion of malicious vs. non-malicious IPs.
- A bar chart displays the number of packets per protocol (TCP, UDP, ICMP).
- Tables and summaries show top offending countries and most frequent IPs.

These outputs make it easy for users to understand traffic trends and identify potential attack patterns quickly.

6. CONCLUSION AND REFERENCES

The Wireshark Network Tracking with Google Maps project successfully integrates network packet analysis with threat intelligence and geolocation services. It simplifies the process of identifying, analyzing, and interpreting suspicious IP activity by representing it on an interactive world map and offering insightful analytics.

- Users can visualize IP sources and threats in real-time.
- The system provides clarity and accessibility to network forensics.
- It is ideal for students, cybersecurity enthusiasts, and early-stage analysts.

This project successfully demonstrates how combining packet capture data with threat intelligence and geolocation tools enhances cybersecurity analysis. The tool is effective in educational contexts and can help new analysts understand network patterns and threats visually.

Future enhancements can include real-time monitoring with auto-refresh, alert-based thresholds, exporting reports in multiple formats, and machine learning integration to predict threat levels.

References:

- https://www.wireshark.org/
- https://www.abuseipdb.com/
- https://www.virustotal.com/
- https://python.org/
- https://python-visualization.github.io/folium/