INTRODUCTION

Network tracking using Wireshark and Google Earth is a powerful approach to visualize and analyze network activity in a geographical context. Wireshark, a robust network protocol analyzer, allows the capture of packet-level data, revealing the intricacies of data exchanges within a network. This information can then be correlated with geographical locations using Google Earth, providing a comprehensive view of network traffic patterns.

In this methodology, Wireshark serves as the primary tool for capturing, inspecting, and exporting network data. The captured packets include essential details such as source and destination IP addresses, timestamps, and other relevant information. To enhance the analysis, accurate timestamps and, when available, GPS data associated with the captured packets are crucial.

The integration with Google Earth involves mapping the network activity to physical locations. This process may require additional steps, including the conversion of IP addresses to geographical coordinates or utilizing geolocation services to associate IP addresses with specific locations.

By combining Wireshark's packet-level insights with the visual representation provided by Google Earth, network administrators and analysts gain a deeper understanding of network behavior. This approach aids in identifying potential security threats, optimizing network performance, and visualizing the flow of data across different geographical areas.

However, it's important to note that while network tracking can be a valuable tool, it should be performed ethically and in compliance with legal regulations. Understanding the nuances of network traffic and its geographical implications contributes to a more comprehensive and informed network management strategy.

NETWORK TRACKING

Network tracking, often referred to as network tracing or monitoring, involves the systematic observation and analysis of data flow within a computer network. The primary goal is to gain insights into network behavior, identify potential issues, and ensure optimal performance.

Here are key aspects of network tracking

1. Data Collection: Network tracking involves collecting data related to network activities. This can include information about data packets, bandwidth usage, device connections, and communication protocols.

2. Packet Analysis: Analyzing individual data packets is a fundamental aspect of network tracking. Tools like Wireshark are used to capture and inspect the contents of packets, providing details about the communication between devices

3. Performance Monitoring: Network tracking helps in monitoring the performance of a network. This includes assessing factors such as latency, packet loss, and overall network responsiveness.

4. Security Monitoring: Network tracking is essential for detecting and preventing security threats. By monitoring network traffic, anomalies and suspicious activities can be identified, aiding in the early detection of potential security breaches.

5. Traffic Flow Analysis: Understanding how data flows within a network is crucial. Network tracking tools can visualize the paths taken by data packets, helping administrators optimize network architecture for efficiency.

6. Resource Utilization: Tracking network usage helps in assessing resource utilization. This includes monitoring bandwidth consumption, identifying resource-intensive applications, and ensuring a balanced distribution of resources.

7. Policy Enforcement: Network tracking allows organizations to enforce policies related to data usage, access control, and compliance. It helps ensure that network activities align with established rules and regulations.

In summary, network tracking is a comprehensive process that involves observing, capturing, and analyzing data related to network activities. This proactive approach is essential for maintaining a secure, efficient, and well-functioning computer network

WIRESHARK

Wireshark is a widely used open-source network protocol analyzer. It allows users to capture and examine the data traveling back and forth on a computer network in real-time. Originally known as "Ethereal," Wireshark provides a detailed view of the data packets, displaying information about various protocols and their interactions within a network.

Key features of Wireshark include:

1. Packet Capture:Wireshark captures packets from the network and presents them in a human-readable format.

2. Protocol Analysis:It supports a broad range of protocols, allowing users to inspect and understand the details of network communications.

3. Live Capture and Offline Analysis:Wireshark can capture data live from a network or analyze pre-recorded capture files.

4. Filtering and Search: Users can apply filters to focus on specific types of traffic or search for particular packets based on various criteria.

5. Packet Decoding: Wireshark decodes and interprets packet contents, revealing details such as source and destination addresses, packet size, and protocol-specific information.

6. Graphical User Interface (GUI):Wireshark provides a user-friendly interface for both beginners and experienced network analysts.

7. Cross-Platform Compatibility: It is available for various operating systems, including Windows, macOS, and Linux.

Wireshark is commonly used for network troubleshooting, security analysis, protocol development, and educational purposes. It plays a crucial role in understanding and diagnosing network-related issues by providing insights into the traffic patterns and communication protocols within a network.

Advantages of wireshark:

1. Packet-Level Insight: Wireshark provides detailed packet-level information, allowing users to analyze the content of each data packet exchanged within a network. This level of granularity is crucial for troubleshooting and understanding network behavior.

2. Wide Protocol Support: Wireshark supports a broad range of communication protocols, making it versatile for analyzing various types of network traffic, including TCP/IP, UDP, HTTP, DNS, and more.

3. Live Capture and Offline Analysis: Users can capture live network traffic or analyze pre-recorded capture files. This flexibility is valuable for both real-time monitoring and post-incident analysis.

4. Educational Value:Wireshark is widely used for educational purposes in networking and security courses. It provides hands-on experience with real-world network traffic, helping students understand network protocols and behaviors.

Disadvantages of wireshark:

1. Complexity:It can be overwhelming for beginners due to its extensive feature set and complex interface.

2. Resource Intensive: Wireshark can be resource-intensive, especially when capturing and analyzing large amounts of network traffic, potentially affecting system performance.

3. Security Concerns: Misuse of Wireshark for unauthorized network monitoring can pose security risks and privacy concerns.

4. Legal and Ethical Considerations: Using Wireshark on networks without proper authorization may violate privacy laws and ethical standards, making it crucial to use responsibly.

GPS DATA

GPS, or Global Positioning System, data refers to information gathered by a GPS receiver from satellites to determine the device's geographical location on the Earth's surface.GPS data is widely used in navigation, mapping, geotagging photos, fitness tracking, and various other applications where precise location information is essential. It relies on signals from multiple satellites in orbit to triangulate the user's position accurately.

This data typically includes:

1. Latitude and Longitude: These are the geographical coordinates that pinpoint a specific location on the Earth. Latitude specifies the north-south position, while longitude indicates the east-west position.

2. Altitude: The height above sea level is provided as the altitude, giving a three-dimensional representation of the location.

3. Timestamp: The time at which the GPS data was recorded is crucial for tracking movements over time.

4. Speed and Heading: Some GPS devices also provide information about the speed of the device and the direction it is moving (heading).

USES OF GPS DATA

1. Navigation: GPS data is fundamental for navigation systems in vehicles, ships, and aircraft. It provides real-time location information, helping users find the most efficient routes to their destinations.

2. Mapping:GPS data is used in mapping applications to create detailed Earth, update geographic information systems (GIS), and improve the accuracy of cartographic representations.

3. Location-based Services (LBS): Many mobile apps use GPS data to offer location-based services, such as finding nearby restaurants, businesses, or services, and providing personalized recommendations.

4. Emergency Services: GPS data assists emergency services in locating people in distress, accidents, or natural disasters, enabling quicker response times.

5. Fitness Tracking: Wearable devices and fitness apps utilize GPS data to track and analyze activities like running, cycling, or hiking, providing users with detailed information about their routes and performance.

6. Geotagging: GPS data is embedded in photos or other media files to geotag them, indicating the exact location where they were captured.

7. Fleet Management: GPS tracking is employed in fleet management systems to monitor the location, speed, and behavior of vehicles, optimizing routes and enhancing overall efficiency.

8. Surveying and Geology: Surveyors and geologists use GPS data for accurate mapping, land surveying, and monitoring changes in the Earth's crust.

9. Agriculture: Precision farming utilizes GPS data to optimize crop management, monitor field conditions, and plan planting and harvesting activities more efficiently.

10. Research and Environmental Monitoring:Scientists use GPS data to study movements of wildlife, track environmental changes, and conduct research in various fields.

CORRELATION

In the context of network tracking using Wireshark and Google Earth, "correlation" refers to the process of aligning or matching the timestamps from captured network packets (analyzed with Wireshark) with the timestamps from GPS data. The goal is to synchronize network events with specific geographical locations on a map.By correlating Wireshark captures with GPS data, you can gain insights into where specific network events occurred in physical space. This correlation is valuable for tasks such as tracking the movement of devices, analyzing network performance in different locations, and understanding the geographic distribution of network activities.

Here's a breakdown of the correlation process:

1. Timestamp Alignment:- Ensure that the timestamps from Wireshark captures and GPS data match or can be aligned. This step is crucial for accurately associating network activities with specific points in time.

2. Matching Devices:- Identify the devices or network nodes in the captured packets using Wireshark. This could involve examining MAC addresses or IP addresses associated with the network traffic.

3. Geographical Coordinates:- Extract the geographical coordinates (latitude and longitude) from the GPS data corresponding to the timestamps when network packets were captured.

4. Plotting on Google Earth:- Use the Google Earth API or a similar mapping solution to plot the identified geographical coordinates on a map. Each point on the map corresponds to a specific location where network activity occurred.

5. Visualization:- Create a visual representation that overlays the network events on the map. This could involve markers, lines, or other annotations to indicate the correlation between network activity and geographical positions.

MAPPING SOFTWARE

In the context of network tracking using Wireshark and Google Earth, "mapping software" refers to applications or tools that facilitate the visualization of network-related data on geographical Earth, often using services like Google Earth.By using mapping software, you can merge the spatial information from GPS data with network packet details from Wireshark, creating a visual representation that helps in understanding how network activities correspond to specific locations. This visualization can be valuable for tasks such as tracking the movement of devices or analyzing network performance across different geographical areas.

Here's how mapping software is utilized in this scenario:

1.Google Earth API: - Incorporate the Google Earth API into your tracking system. The API allows you to embed Google Earth into your application and provides functionality for customizing and interacting with the map.

2. Integration with Wireshark Data: - The mapping software needs to integrate with the data collected using Wireshark. This includes the timestamps of network events, the IP addresses or MAC addresses of devices involved, and any other relevant information.

3. Geographical Coordinates:- Extract geographical coordinates (latitude and longitude) from the GPS data associated with the timestamps of network events. This data serves as the basis for plotting points on the map.

4. Visualization: - Utilize the mapping software to visually represent network activities on the map. This could involve placing markers or annotations at specific geographical locations corresponding to where network events occurred.

5. Real-time Tracking (Optional): - Some mapping software may support real-time tracking, allowing continuous updates on the map as new network events are captured by Wireshark. This feature can be useful for dynamic tracking scenarios.

6. Customization and Interaction: - Depending on the capabilities of the mapping software, you may have options to customize the appearance of the map and enable user interactions, enhancing the overall user experience.

VISUALIZATION

Visualization refers to the creation of visual representations of data or concepts. It can involve charts, graphs, Earth, or other graphical elements to help people understand complex information more easily. In various fields, visualization is used to analyze trends, communicate ideas, and make data more accessible.It refers to the presentation of information or data in a visual format, such as charts, graphs, or other graphical representations. It helps individuals understand complex concepts, patterns, and trends more easily by translating numerical or abstract data into visual forms that can be quickly interpreted.

In network tracking with Wireshark and Google Earth, visualization involves mapping network activity onto geographical locations. Wireshark captures and analyzes network traffic data, while Google Earth provides a platform for displaying this information spatially.

For example, you can use Wireshark to capture network packets and extract relevant data like IP addresses. Then, through a process known as geolocation mapping, you can plot these IP addresses on Google Earth to visualize the network traffic's geographic origin or destination. This can be useful for identifying the source of network issues, monitoring global network activity, or enhancing cybersecurity efforts.

Several tools and techniques exist to integrate Wireshark data with Google Earth, allowing for a more comprehensive and visual understanding of network traffic patterns.

Uses of visualization in "Network tracking using wireshark and google Earth".

1. Geographical Mapping: Overlaying network data on Google Earth can show the geographical locations of network nodes, helping identify potential issues related to specific locations or regions.

2. Traffic Patterns:Visualization allows you to see traffic patterns, spikes, or anomalies over time, helping in the identification of potential security threats or performance issues.

3.Identifying Latency: By mapping network latency to specific locations, you can pinpoint areas with high latency and investigate potential causes, such as network congestion or hardware issues.

4. Network Topology:Visualizing the network topology on a map helps in understanding the structure of the network, including the location of routers, switches, and other devices.

5.Troubleshooting:Visualization can assist in troubleshooting by providing a clear overview of the flow of data between different nodes, making it easier to identify bottlenecks or problematic connections.

6.Security Analysis:Detecting unusual patterns or unexpected connections visually can aid in identifying security threats, such as suspicious network activity or unauthorized access.

7. User Behavior Analysis: By mapping network activities to specific geographical regions, you can analyze user behavior and identify any deviations from normal usage patterns.

8. Capacity Planning:Visualization can assist in capacity planning by showing areas with high network usage, helping organizations anticipate and address potential scalability issues.

SECURITY AND PRIVACY

When it comes to representing security and privacy in network tracking using Wireshark and Google Earth, **it's crucial to consider the following aspects:**

1. Encryption and Decryption: Visualizing encrypted traffic can highlight secure communication channels, emphasizing the importance of encryption for maintaining privacy and securing sensitive data during transmission.

2. Anomaly Detection: Visualization can aid in identifying anomalous patterns in network traffic that might indicate security threats or privacy breaches, helping security professionals react promptly to potential issues.

3. Privacy Concerns: Representing areas of the map where sensitive data is transmitted can draw attention to potential privacy risks. It emphasizes the need for proper encryption and secure transmission methods to protect user data.

4. Access Points and Authentication: Mapping network access points and authentication mechanisms can showcase the security layers in place, illustrating how devices connect securely to the network and ensuring authorized access.

5. Firewall Rules: Visualizing firewall rules on the map can help demonstrate how network security policies are implemented, restricting or allowing traffic based on specific criteria to protect against unauthorized access.

6. Incident Response: During a security incident, visualizing real-time data on a map can assist in tracking the source and spread of the threat, facilitating a more effective incident response.

7. Compliance Monitoring: Visualization can be used to demonstrate compliance with privacy regulations by showcasing secure data transmission paths and adherence to privacy standards in different geographical locations.

8. Masking Sensitive Information: While visualizing network traffic, it's essential to ensure that any personally identifiable information (PII) or sensitive data is appropriately masked to uphold privacy standards and comply with data protection regulations.

9. VPN Connections: Illustrate Virtual Private Network (VPN) connections on the map, showcasing secure tunnels for private communication. This helps emphasize the importance of using VPNs to enhance privacy and security, especially when accessing networks remotely.

10. Anonymizing User Data: Highlight measures taken to anonymize or pseudonymize user data in transit. This can involve masking personally identifiable information (PII) to protect user privacy during network tracking.

11. Privacy Policies and Compliance: Integrate visual elements that represent adherence to privacy policies and compliance standards. This can include markers on the map to indicate regions where data protection regulations are particularly stringent.

12. Access Control and Authentication: Illustrate the implementation of robust access control measures and authentication protocols. Visualizing secure access points and user authentication events emphasizes the importance of controlling who can access the network.

By incorporating these elements into the visual representation, you can effectively communicate the commitment to security and privacy in network tracking, reassuring users and stakeholders that measures are in place to safeguard sensitive information throughout the network

SYSTEM REQUIREMENTS

- HARDWARE REQUIREMENTS

i. Laptop or pc
ii. Windows 7 or higher
iii. i3 processor or higher

- SOFTWARE REQUIREMENTS

i. Python 3.11
ii. Wireshark
iii. Google earth
iv. GeoLightCity,Dat

EXISTING SYSTEM VS PROPOSED SYSTEM

The combination of Wireshark and Google Earth for network tracking isn't a standard or built-in system, but people have come up with creative ways to use these tools together. Wireshark is primarily a network protocol analyzer, while Google Earth is a mapping service.

One possible approach is to capture network traffic with Wireshark, analyze it to extract location-related information (like IP addresses or geolocation data), and then map this information onto Google Earth manually or using a custom script or tool. This could help in visualizing the geographical distribution of network traffic.

Keep in mind that such activities need to be done ethically and in compliance with privacy and legal regulations. Tracking network activity without proper authorization is not only unethical but may also be illegal.

PROPOSED SYSTEM

A proposed system for network tracking using Wireshark and Google Earth could involve a combination of packet analysis, data extraction, and mapping.

Here's a generalized outline of how you might approach it:

1. Packet Capture with Wireshark:- Use Wireshark to capture network packets, analyzing the data flowing through the network.

2. Extract IP Addresses:- Extract relevant IP addresses from the captured packets. These could be source or destination IP addresses.

3. IP Geolocation:- Utilize an IP geolocation service or database to map IP addresses to

geographical locations. This helps in determining the physical location associated with each IP.

4. Data Correlation:- Correlate the geographical information with the corresponding network data, linking IP addresses to specific locations.

5. Integration with Google Earth:- Use the Google Earth API or another mapping tool to visualize the correlated data on a map. This step involves integrating the geographical information into a mapping service for a user-friendly display.

6. User Interface:- Develop a user interface that allows users to interact with the mapped data, providing a comprehensive view of network activity along with geographic context.

7. Real-time Tracking (Optional):- Implement real-time tracking capabilities if needed, allowing users to monitor network activities as they occur.

8. Security and Privacy Considerations:- Ensure the system complies with security and privacy regulations, as network tracking involves sensitive information.

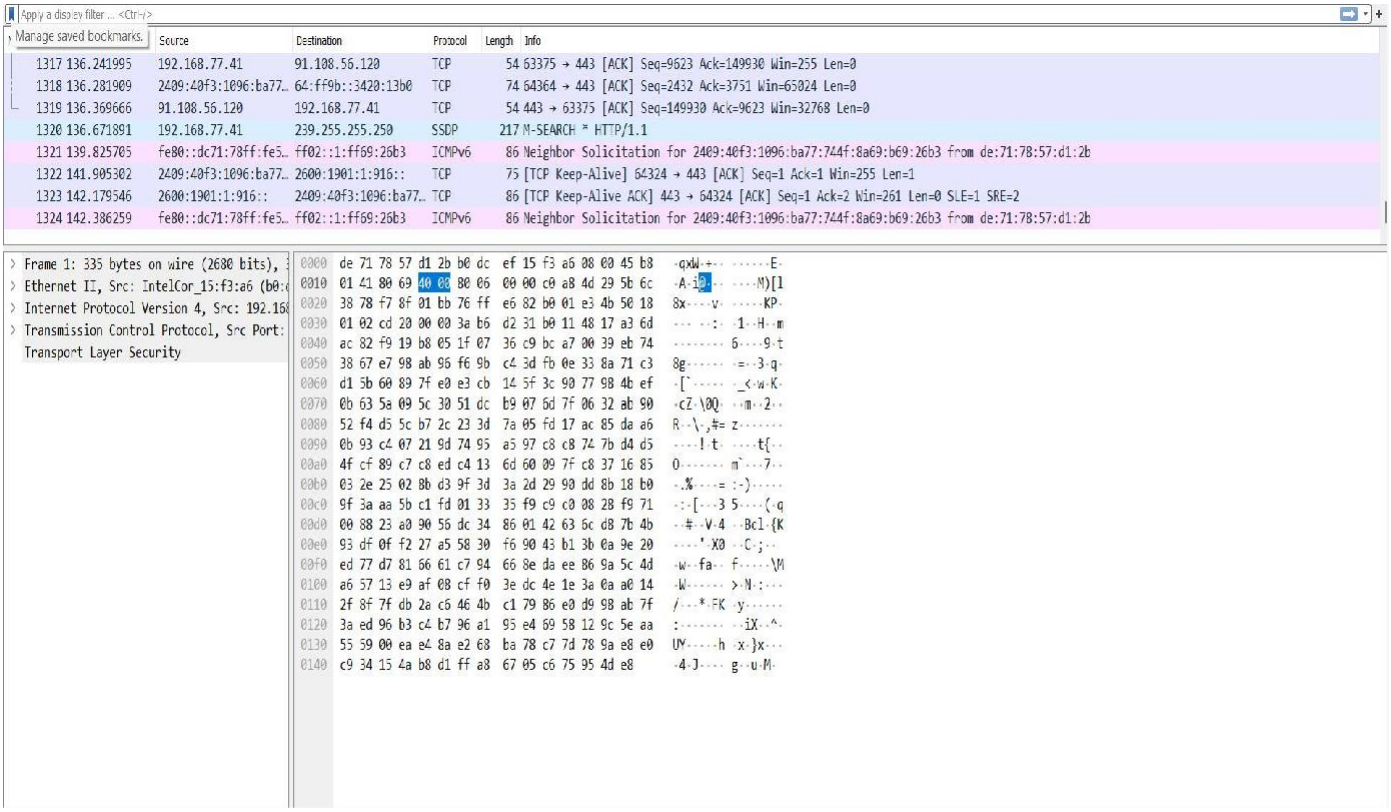
CODE

```
import dpkt              ( A Python library for working with packet data.)
import socket            (Provides low level network communication.)
import pygeoip           (Allows geographic IP lookups using MaxMind  GeoIP
databases.)

gi = pygeoip.GeoIP('GeoLiteCity.dat')

def retKML(dstrip, scrip):        (This function takes two arguments : dstrip
(destination IP) and scrip (source IP).
    dst = gi.record_by_name(dstrip)
    src = gi.record_by_name('x.xxx.xxx.xxx')
    try:
        dstlongitude = dst['longitude']
        dstlatitude = dst['latitude']
        srclongitude = src['longitude']
        srclatitude = src['latitude']
        kml = (
            '<placemark>\n'
            '<name>%s</name>\n'
            '<extrude>1</extrude>\n'
            '<tessellate>1</tessellate>\n'
            '<styleUrl>#transBluePoly</styleUrl>\n'
            '<linestring>\n'
            '<coordinates>%6f,%6f\n%6f,%6f</coordinates>\n'
            '</linestring>\n'
            '</placemark>\n'
        ) % (dstrip, dstlongitude, dstlatitude, srclongitude, srclatitude)
```

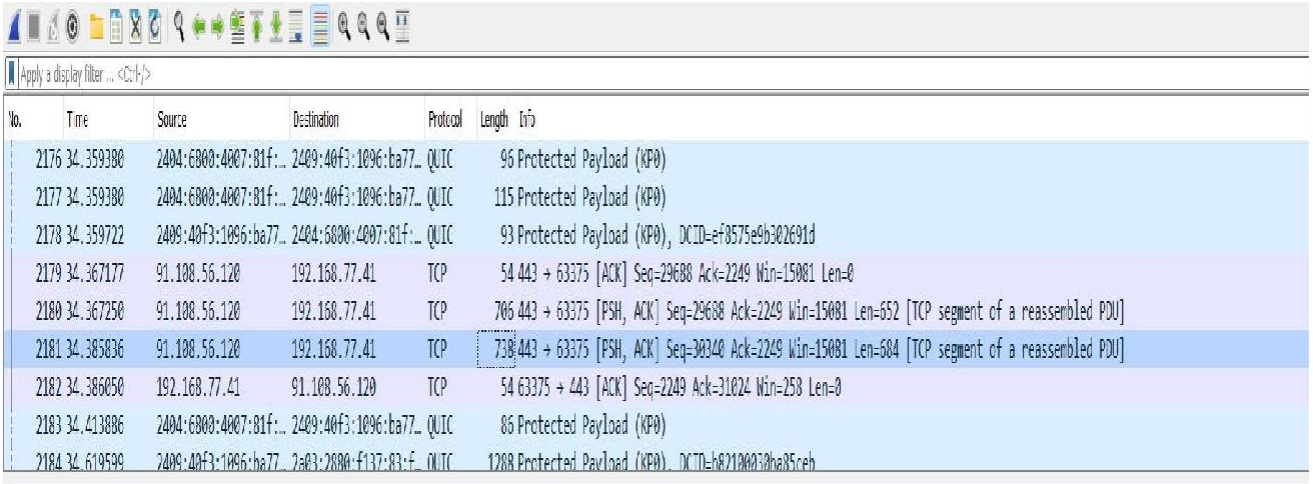
```
        return kml
    except:
        return ''

def plotIPs(pcap):    (This function takes up pcap (packet capture) file as an argument.)
    kmlpts = ''
    for (ts, buf) in pcap:
        try:
            eth = dpkt.ethernet.Ethernet(buf)
            ip = eth.data
            src = socket.inet_ntoa(ip.src)
            dst = socket.inet_ntoa(ip.dst)
            KML = retKML(dst, src)
            kmlpts = kmlpts + KML
        except:
            pass
    return kmlpts

def main():    (This function opens a pcap file('wire.pcap') and reads it content using
dpkt's pcap.Reader.)
    f = open('wire.pcap', 'rb')
    pcap = dpkt.pcap.Reader(f)
    kmlheader = '<?xml version="1.0" encoding="UTF-8"?> \n<kml
xmlns="https://www.google.com/Earth/@10.5442763,76.1383668,7z?entry=ttu/kml/2.
2">\n<Document>\n' \
                '<Style id="transBluePoly">' \
                '<LineStyle>' \
                '<width>1.5</width>' \
                '<color>501400E6</color>' \
                '</LineStyle>' \
                '</Style>'
```

```
    kmlfooter = '</Document>\n</kml>\n'
    kmldoc = kmlheader + plotIPs(pcap) + kmlfooter
    print(kmldoc)

if _name_ == '_main_':    ( This ensures that the main function is executed when the script is run directly.)
    main()
```

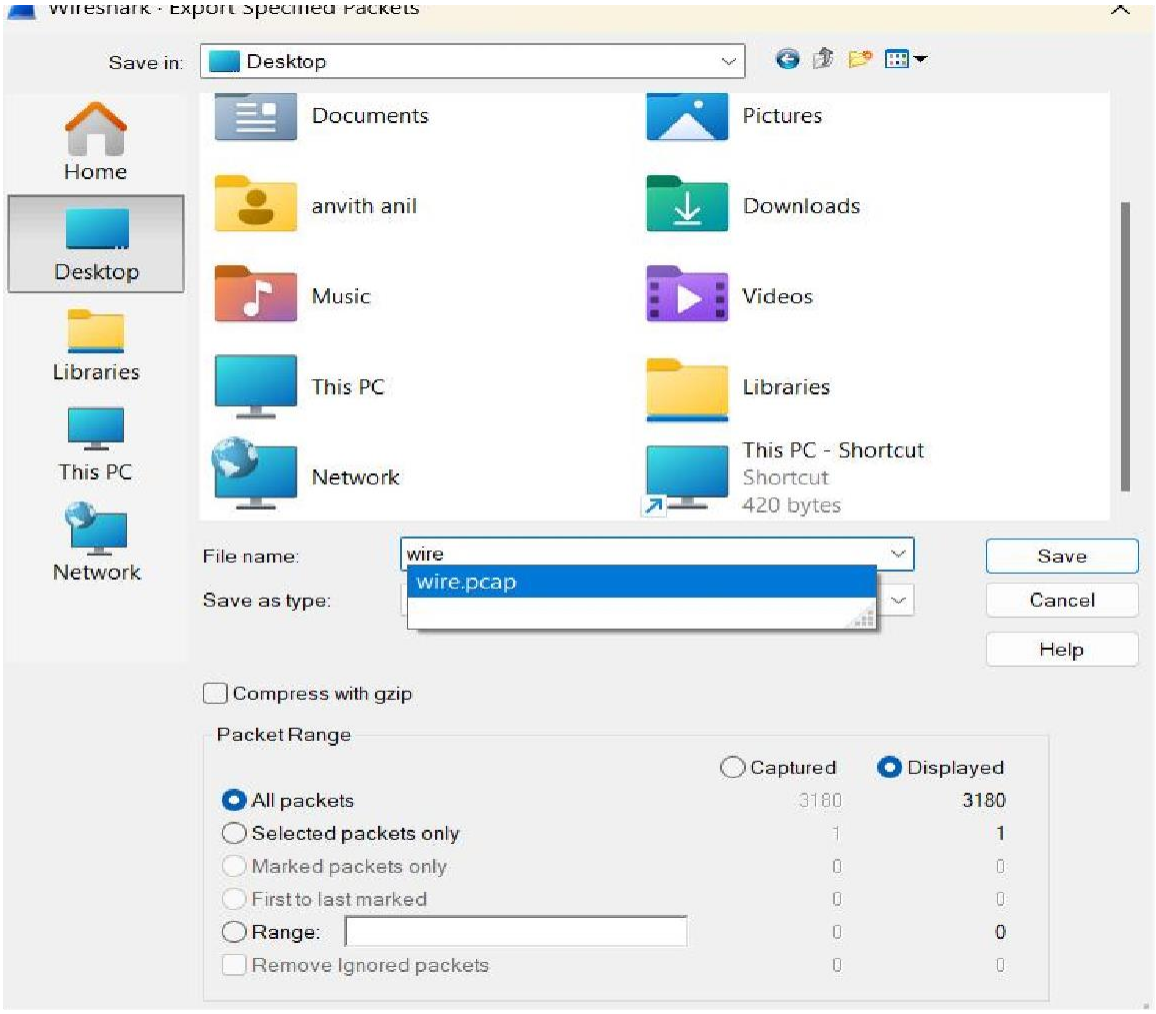
OUTPUT



: search for packets in wireshark



: select and capture the packets from the wireshark



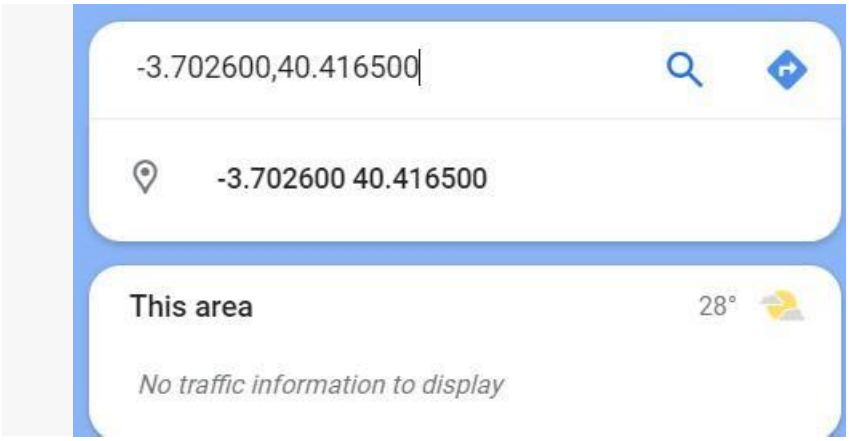
: save the captured packets in desktop after saving run the program

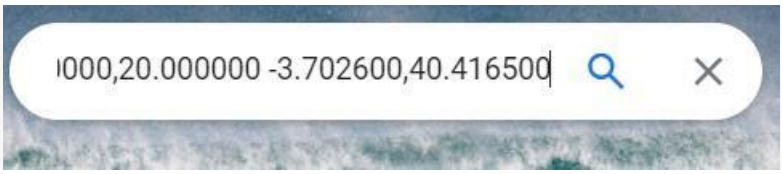
```
<extrude>1</extrude>
<tessellate>1</tessellate>
<styleUrl>#transBluePoly</styleUrl>
<linestring>
<coordinates>77.000000,20.000000
-3.702600,40.416500</coordinates>
</linestring>
</placemark>
<placemark>
<name>49.44.136.88</name>
<extrude>1</extrude>
<tessellate>1</tessellate>
<styleUrl>#transBluePoly</styleUrl>
<linestring>
<coordinates>77.000000,20.000000
-3.702600,40.416500</coordinates>
</linestring>
</placemark>
<placemark>
<name>49.44.136.88</name>
<extrude>1</extrude>
<tessellate>1</tessellate>
<styleUrl>#transBluePoly</styleUrl>
<linestring>
<coordinates>77.000000,20.000000
-3.702600,40.416500</coordinates>
</linestring>
</placemark>
<placemark>
<name>49.44.136.88</name>
<extrude>1</extrude>
<tessellate>1</tessellate>
<styleUrl>#transBluePoly</styleUrl>
<linestring>
<coordinates>77.000000,20.000000
-3.702600,40.416500</coordinates>
</linestring>
</placemark>
<placemark>
<name>49.44.136.88</name>
<extrude>1</extrude>
<tessellate>1</tessellate>
<styleUrl>#transBluePoly</styleUrl>
<linestring>
```


:we will get the coordinates of the packets we captured

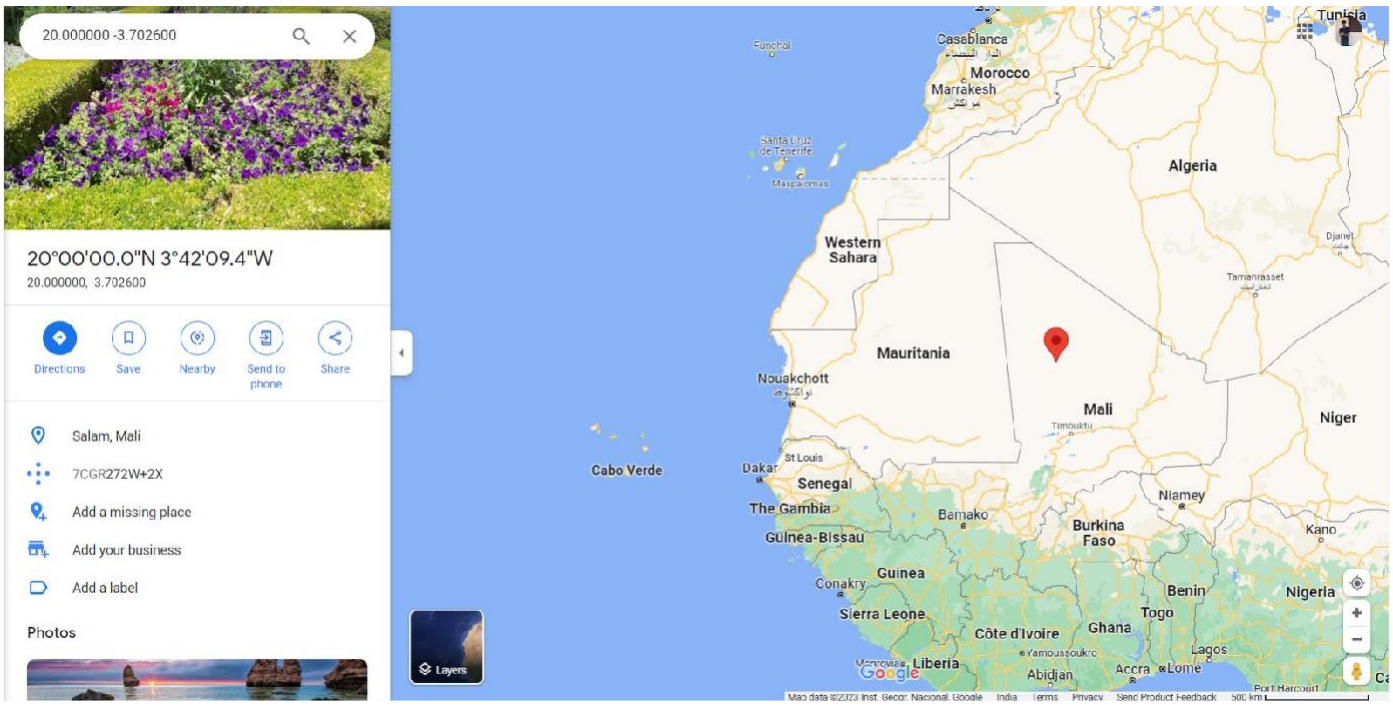
```
<coordinates>77.000000,20.000000
-3.702600,40.416500</coordinates>
```

:select the coordinates and run it in google earth.





: we will get the exact location of the packets we captured


20.000000 -3.702600
20°00'00.0"N 3°42'09.4"W
20.000000, -3.702600
Directions
Save
Nearby
Send to phone
Share
Salam, Mali
7CGR272W+2X
Add a missing place
Add your business
Add a label
Photos
Layers
Tunisia
Casablanca
Morocco
Marrakesh
Algeria
Western Sahara
Mauritania
Nouakchott
Mali
Niger
Cabo Verde
Dakar
St Louis
Senegal
The Gambia
Bamako
Guinea-Bissau
Burkina Faso
Niamey
Kano
Guinea
Conakry
Sierra Leone
Liberia
Côte d'Ivoire
Ghana
Togo
Benin
Nigeria
Lagos
Abidjan
Accra
Lomé
Google

Uses of google earth in “NETWORK TRACKING USING WIRESHARK AND GOOGLE EARTH”

1. Geolocation of IP Addresses:

- Wireshark captures IP addresses involved in network communication.
- You can use tools to map these IP addresses to geographical locations.
- Google Earth or Google Earth can then visualize the geographic distribution of network traffic.

2. Tracking Network Routes:

- Analyze the packet paths recorded by Wireshark.
- Use Google Earth to map and visualize the routes taken by network packets, helping identify any unusual or unexpected paths.

3. Visualizing Network Latency:

- Analyze timestamps and response times from Wireshark.
- Use Google Earth to create visual representations of network latency, helping identify areas with higher latency or potential issues.

4. Monitoring Network Devices:

- Identify devices communicating on the network through Wireshark.
- Map these devices using Google Earth, providing a comprehensive view of the physical locations of networked devices.

5. Incident Response and Forensics:

- During incident response, use Wireshark to analyze network traffic related to a security incident.
- Use Google Earth to visualize the geographical locations of affected systems, aiding in identifying the scope and potential sources of the incident.

6. Geofencing and Anomaly Detection:

- Establish geographic boundaries using Google Earth.
- Wireshark can be used to detect anomalies in network traffic, and Google Earth can help visualize these anomalies in relation to specific geographic areas.

7. Infrastructure Planning:

- Wireshark data can help in understanding current network usage and identifying areas of improvement.
- Google Earth can aid in planning network infrastructure upgrades based on the geographical distribution of network activity.

8. Collaboration and Reporting:

- Create visual reports using Google Earth that can be shared with stakeholders.
- Use the combination of Wireshark data and geographic visualization to enhance collaboration and communication about network issues.

PROGRAM DESCRIPTION

Python program utilizes the dpkt library to parse network packets from a Wireshark capture file ('wire.pcap'). It extracts source and destination IP addresses, resolves them to geographic coordinates using the GeoLiteCity database through the pygeoip library, and generates a KML file for Google Earth mapping.

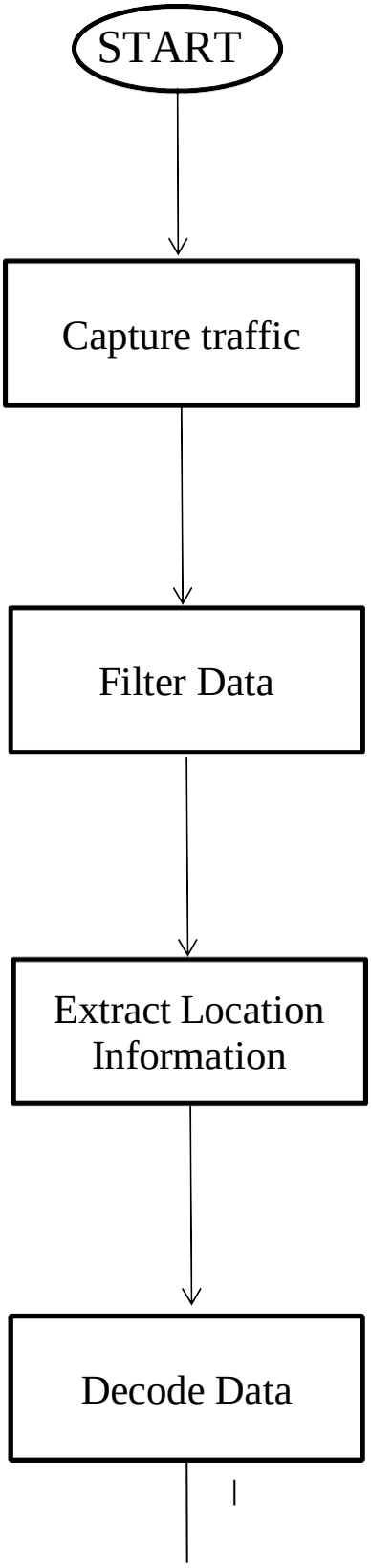
The `retKML` function constructs KML (Keyhole Markup Language) placemark elements with coordinates, creating a line on the map connecting source and destination locations. The `plotIPs` function processes the pcap file, calling `retKML` for each packet, and concatenates the generated KML elements.

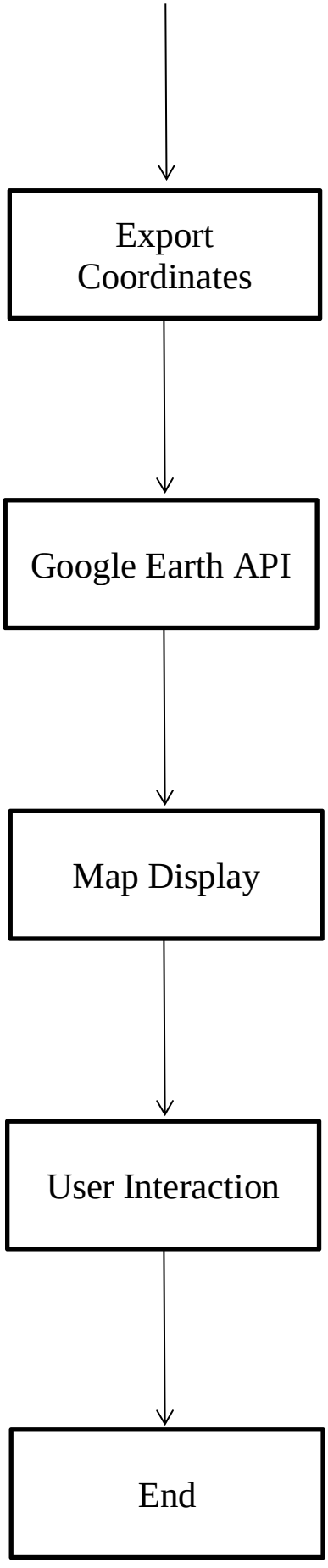
The `main` function initializes the pcap reader, defines KML header and footer, and prints the complete KML document with the plotted IP addresses' connections.

Note: There are a couple of typos in the code. The assignment in the line `kmldoc=kmlheader+plotIPs(pcap)=kmlfooter` seems incorrect, and there's a typo in 'encording' in the kmlheader. Additionally, the `dkpt` module should be corrected to `dpkt`.

Also, there's a typo in `encording="UTF-8"`, it should be `encoding="UTF-8"`. Additionally, the `width` and `color` elements in the `LineStyle` tag within the `transBluePoly` style should be inside double quotes.

WORKING DIAGRAM




Export Coordinates
Google Earth API
Map Display
User Interaction
End

WORKING ENVIRONMENT

The programming language we chose was Python. Python is an extremely useful programming language for cybersecurity professionals because it can perform a multitude of cybersecurity functions, including malware analysis, scanning, and penetration testing tasks. The syllabus demands the project to be done using python. In this project we use python IDE 3.10.5. Coded in 100% pure Python, using the tkinter GUI toolkit. Cross platform: works mostly the same on Windows, Unix, and macOs. Python shell window (interactive interpreter) with colorizing of code input, output, and error messages. Multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features. Search within any window, replace within editor windows, and search through multiple files (grep). Debugger with persistent breakpoints, stepping, and viewing of global and local namespaces. Configuration, browsers, and other dialogs

APPLCAITONS

1. Network Security Analysis:

Packet Inspection: Wireshark allows you to capture and analyze network packets. By examining packet data, you can identify potential security threats, such as suspicious traffic or unauthorized access attempts.

Geographical Mapping: Integrating Google Earth can provide a visual representation of the geographical location of network traffic sources. This can help identify and trace the origin of potential security threats.

2. Troubleshooting Network Issues:

Traffic Analysis: Wireshark helps in diagnosing network issues by capturing and analyzing packet-level details. Integrating Google Earth can assist in identifying the physical location of network components experiencing problems.

Latency Analysis: By examining packet delays and response times, you can pinpoint network latency issues. Mapping these delays onto Google Earth can help identify the geographical location of latency hotspots.

3. IoT Device Tracking:

Device Identification: Wireshark can be used to analyze traffic patterns from IoT devices. Integrating with Google Earth allows you to visualize the locations of these devices, which can be crucial for monitoring and managing IoT networks.

4. Network Optimization:

Traffic Flow Visualization:Analyzing network traffic and mapping it onto Google Earth can aid in optimizing network infrastructure. This includes identifying areas with high traffic concentration and optimizing routing paths.

5. Incident Response:

-Real-time Tracking: Integrating Wireshark with real-time tracking on Google Earth can be beneficial during incident response scenarios. Security teams can quickly identify the source and spread of a network incident.

6. Mobile Network Analysis:

Mobile Traffic Tracking:For mobile networks, Wireshark can capture and analyze data packets. Integrating with Google Earth can provide insights into the locations where mobile devices are actively communicating.

7. Capture Packets with Wireshark:

- Use Wireshark to capture network traffic.
- Analyze packets to identify IP addresses and relevant data.

8. Extract IP Addresses:

- Identify the source and destination IP addresses in the captured packets.

9. Geolocate IP Addresses:

- Use a geolocation service or database to map IP addresses to physical locations.

10. Map IP Addresses on Google Earth:

- Use the obtained geographical coordinates to plot the locations on Google Earth.

This process is commonly employed for:

- Network Security: Identifying and tracking potentially malicious activities.
- Troubleshooting: Locating network issues or unauthorized access

FUTURE ENHANCEMENT

1. Enhanced Protocol Support:

- Wireshark updates may include enhanced support for new and emerging network protocols, ensuring comprehensive packet analysis.

- Improved dissection and decoding of protocols can lead to more accurate and detailed information in the captured packets.

2. Real-time Collaboration:

- Collaborative features might be introduced, allowing multiple users to analyze and track network traffic simultaneously.

- Real-time updates and shared views on Google Earth can enhance collaboration among security teams or network administrators.

3. Integration with Cloud Services:

- Wireshark might integrate with cloud-based services for additional analysis capabilities and storage.

- Google Earth integration could leverage cloud-based APIs for better performance and scalability.

4. Advanced Filtering and Search:

- Enhanced filtering options within Wireshark could make it easier to pinpoint specific types of network traffic.

- Improved search capabilities may allow users to quickly locate relevant packets for analysis.

5. Machine Learning Integration:

- Implementation of machine learning algorithms within Wireshark could automate the identification of abnormal network behavior.

- Google Earth integration might leverage machine learning for predictive mapping based on historical data.

6. Mobile Device Tracking Improvements:

- Wireshark updates may include improved capabilities for tracking and analyzing traffic from mobile devices.

- Google Earth integration might offer more precise location tracking and visualization for mobile devices on the network.

7. Customizable Dashboards:

- Both Wireshark and Google Earth integration might introduce customizable dashboards for better user experience.

- Users may be able to tailor the display to show relevant metrics and information based on their specific needs.

8. Security Enhancements:

- Continuous improvements to ensure the security of the Wireshark application itself.

- Integration of additional security features to protect sensitive information during the network tracking process.

CONCLUDING REMARKS

In conclusion, combining Wireshark for network tracking with Google Earth provides a powerful synergy for visualizing and analyzing network traffic. This integration allows for a comprehensive understanding of geographical patterns in data flow, enhancing network security, troubleshooting, and optimization efforts. By leveraging Wireshark's detailed packet analysis alongside the spatial context provided by Google Earth, one gains a holistic view of network activities, ultimately facilitating informed decision-making and proactive management. Leveraging Wireshark for network tracking, coupled with Google Earth integration, provides a powerful synergy. This combination not only enhances real-time monitoring but also facilitates comprehensive analysis by correlating network activities with geographic locations. The ability to visually map out network traffic adds a valuable layer of understanding, aiding in troubleshooting, security assessments, and optimizing network performance.

The integration of Wireshark for network tracking, coupled with Google Earth, enhances the analysis of network traffic by providing geographical context. This powerful combination not only aids in identifying anomalies and potential security threats but also offers a visual representation of data flows. By leveraging these tools, network administrators can make more informed decisions to optimize performance and strengthen overall network security.

REFERENCE

- https://github.com/mbcc2006/GeoLiteCity-data
- https://www.geeksforgeeks.org/introduction-to-wireshark/
- https://pip.pypa.io/en/stable/installation/
- https://www.youtube.com/results?search_query=how+to+install+pip+in+python+
- https://chat.openai.com/
- https://vinsloev.com/#/