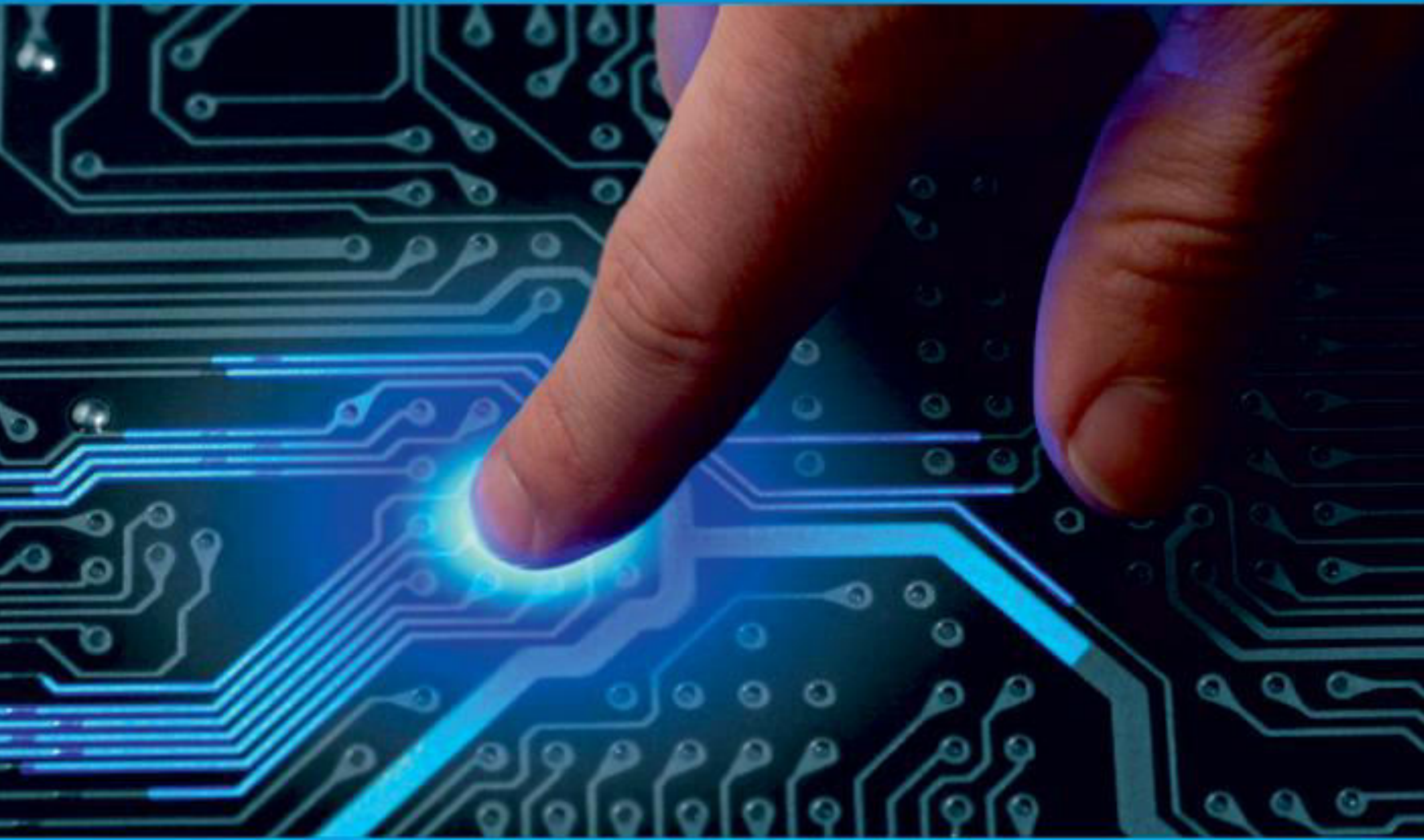
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 3, March 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1203046 |

Network Traffic Virtualization Using Wireshark and Pygeoip

Ansar Dheen N[1] , Haripriya V[2]

PG Student, School of CS & IT, Jain(Deemed-to-be-University), Bengaluru, India[1]

Assistant Professor, Department of School of CS & IT, Jain (Deemed-to-be-University), Bengaluru, India[2]

ABSTRACT: Security analysts possess the requisite expertise to discern various types of network traffic, detect instances of unusually high activity, and pinpoint the specific network nodes responsible. Furthermore, network traffic visualization aids users in pinpointing servers and tracking data flows, thereby enabling researchers to identify potential online threats. Leveraging Wireshark's functionalities as both a packet tracer and capture tool, the process of network analysis hinges on the selection of appropriate packet capture parameters. In this study, Wireshark serves as the cornerstone for creating a network tracking system. By employing Wireshark to capture data packets, analyze them, and subsequently translate the IP addresses into latitude and longitude coordinates, researchers can plot the source and destination IPs on a Google map. This innovative approach allows users to visualize network traffic patterns on a geographical scale and effectively monitor them.

KEYWORDS: Virtualization, Internet Protocol (IP), Source IP, Destination IP, Networks, Wireshark, Google Maps, Packets, Data, Latitude, Longitude, Traffic.

I. INTRODUCION

A computer network is formed when two or more computers or devices are interconnected to facilitate data exchange. These connections can be established either through wired cables or wireless Wi-Fi technology. Various hardware components such as routers, switches, ports, along with software elements like operating systems and user applications, are utilized to establish and manage computer networks. Within a computer network, nodes can encompass personal computers, servers, networking hardware, or other specialized hosts. Each of these devices is assigned a hostname and network address for identification purposes. Computer networks can be classified based on different criteria, including the transmission media used, bandwidth, communication protocols, network size, topology, traffic control mechanisms, and organizational objectives. Computer networks serve a multitude of purposes, including accessing the World Wide Web, streaming digital media, sharing application and storage servers, printing documents, sending faxes, and utilizing email and instant messaging software, among other services and applications. Ensuring the security of these networks is crucial, involving procedures and measures aimed at protecting against unauthorized access or misuse. By implementing robust network security measures, potential threats can be mitigated effectively. Thieves from stealing confidential data housed on a network as shown in figure 1

Fig. 1. Network traffic analysis.

Moreover, it ensures prompt and secure accessibility to essential data and information possessed by authorized network users. Various components such as firewalls, Virtual Private Networks (VPNs), antivirus software, among others, are integral parts utilized for network protection. These elements, coupled with effective network security protocols, serve as defenses against malware, spyware, botnets, and other intrusions, whether they are intentional or accidental, thereby fortifying networks against vulnerabilities. Packet capture devices play a pivotal role in monitoring data transmission across both physical and wireless networks by intercepting and recording all data being transmitted. In contrast, network switches or specialized probing appliances are often deployed to gather NetFlow data, which is then forwarded to a centralized NetFlow collection server for analysis as summary information on data flows. Due to its reduced storage and bandwidth requirements during collection, NetFlow summary data can be utilized more extensively across the network. The safeguarding of sensitive data and critical systems from online threats falls under the purview of cybersecurity. Cybersecurity measures, also known as Information Technology (IT) security, are aimed at thwarting attacks on networked systems and applications, whether originating from internal or external sources.

II.RELATED WORKS

Network traffic analysis refers to the process of acquiring, storing, and scrutinizing the data packets constituting network traffic. Its applications span various areas, encompassing the optimization of network performance, capacity planning, traffic management, identification of security issues, and troubleshooting network issues. Additionally, network traffic analysis provides insights into real-time or historical network activity and behavior. The preceding studies on Wireshark's role in network traffic analysis encapsulate the groundwork laid in this field.

A. Malicious Traffic analysis using Wireshark by the collection of Indicators of Compromise

The study conducted by Bindu Dodiya and Umesh Kumar Singh delves into the analysis of malicious traffic using Wireshark through the collection of Indicators of Compromise (IoCs). Packet analysis stands as a fundamental technique in network forensics, also referred to as packet sniffing or protocol analysis. It involves the real-time gathering and examination of data traversing a network to comprehend its operations thoroughly. This method can unveil various insights, including the reconstruction of files, documents, email attachments, and other transmitted data, aiming to uncover signs of malicious activities such as data breaches, unauthorized access to websites, malware infections, and intrusion attempts. Packet sniffers, which capture raw network data, are commonly employed for this purpose. Wireshark, an open-source tool, has proven invaluable in analyzing network packets and their behaviors. It facilitates the recognition and classification of different attack signatures, empowering network administrators with detailed insights into network activities. The objective of this study is to demonstrate how Wireshark aids in network protocol diagnostics and its effectiveness in detecting fundamental indications of malware compromises.

B. A Review Based on Application of SNORT and Wireshark in Network Traffic Analysis.

G. Jain and Anubha provide insights into the implementation of network traffic analysis utilizing Wireshark and SNORT. In today's rapidly expanding network landscape, the internet plays a pivotal role in various aspects of daily life, including finance, education, research, business, and media. However, this reliance on the internet also exposes networks to numerous intrusions. To address this challenge, it's crucial for systems to incorporate a detection engine capable of automatically identifying intrusions without human intervention. While Wireshark serves as an essential tool for network packet analysis, it lacks built-in intrusion detection capabilities. Nevertheless, Wireshark can be effectively utilized as a network analyzer, protocol analyzer, packet sniffer, network troubleshooting tool, and even as a network intrusion detection tool.

C. Review based on Capability of Wireshark as Intrusion Detection System

Sakshi Singh explores the utilization of Wireshark as an intrusion detection system (IDS). Given the indispensable role of the internet in modern lifeIn networking, an intrusion detection system is crucial for ensuring secure data transmission by analyzing potential breaches. Wireshark emerges as a prominent tool for packet analysis, offering the capability to intercept and identify encrypted network communication. Although Wireshark's effectiveness in intrusion detection is somewhat limited, it can still discern various types of attacks, including Denial of Service (DoS) and

Distributed Denial of Service (DDoS) attacks . SNORT, on the other hand, operates by collecting live internet packets. A Review Based on The Analysis and Design for Network Protocol Analysis System Based on Wincap. Lili Jiang, Xiaohui Yang, and Tao Li discuss protocol analysis using WinCap.

III. METHODS

This approach employs location tracking using Wireshark to ascertain the origin and destination of network data, aiding in troubleshooting, security measures, or research endeavors. Nonetheless, it's essential to acknowledge that the reliability of IP geolocation information may be compromised due to the dynamic nature of IP addresses and the varying quality of databases.

A. Identifying class of IP address.

An Internet Protocol (IP) address serves as a unique identifier for a device within a network. Comprising four integers separated by dots, each ranging from 0 to 255, an IP address is structured. For instance, 192.168.1.1 is a typical example of an IP address. IP addresses are categorized into five classes: A, B, C, D, and E. These classes utilize different methodologies and value ranges for the first number to divide the IP address into network and host components. The network portion specifies the network to which the device is affiliated, while the host portion pinpoints the specific device within that network. By examining an IP address, we can determine its respective class.

172.16.55.11

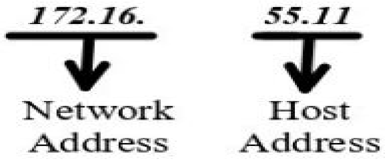


Fig. 2. Parts of IP Address.

Figure 2 describes the parts of the IP address, Prefix and suffix are the two components that make up an IP address. Prefix, also referred to as a network address, is the portion of an IP address that indicates the physical network to which a computer is connected. The host address is another name for the suffix.

TABLE I. CLASSES OF IP ADDRESS

Class First	Octet Value	Subnet Mask	Application
A	0 to 127	8	Large number of hosts
B	128 to 191	16	Medium size networks
C	192 to 223	24	Local Area Network
D	224 to 239	-	Multi-tasking
E	250 to 254	-	R&D purpose

Equation 1 is an IP address, and as it can be seen from the first octet, it falls inside the range of 128 to 192, indicating that it is a Class B IP address. This is used for a medium-sized network and contains 16 subnet masks. The initial octet range, subnet mask, and application of the IP address class are all listed in Table 1 along with information on the IP address class.

B. Tracing Geo Location Using Wireshark and GeoIP Database.

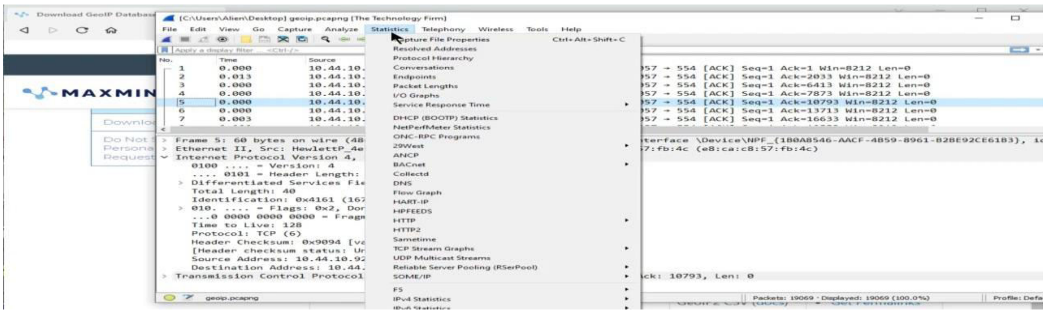
A PyGeoIP database functions as a repository of IP addresses and their associated geographical locations. IP addresses, unique numerical identifiers assigned to internet-connected devices, serve as the basis for this database. Geolocation, the process of pinpointing the precise location of a device or user based on their IP address, relies on PyGeoIP

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2024 ||

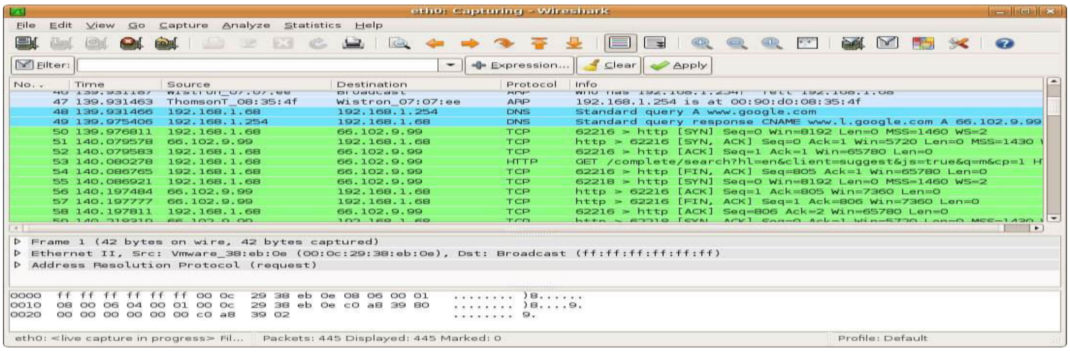
| DOI: 10.15680/IJIRCCE.2024.1203046 |

databases. These databases aggregate data from various sources including internet service providers (ISPs), users, domain name registries, and external providers to compile location information. The accuracy, coverage, and update frequency of PyGeoIP databases vary depending on the data sources and methodologies utilized. While some PyGeoIP databases are proprietary and commercially available, others are open-source and freely accessible. Utilizing Wireshark to locate IP addresses necessitates access to a PyGeoIP database. One approach involves leveraging Maxmind's PyGeoIP tools in conjunction with Wireshark. This database provides information such as city, country, and Autonomous System Number (ASN). It's important to note that the integration process may differ based on the platform used.



```
No.    Time     Source
 1     0.000    10.44.10.        57 → 554 [ACK] Seq=1 Ack=1 Win=8212 Len=0
 2     0.013    10.44.10.        57 → 554 [ACK] Seq=1 Ack=2033 Win=8212 Len=0
 3     0.000    10.44.10.        57 → 554 [ACK] Seq=1 Ack=6413 Win=8212 Len=0
 4     0.000    10.44.10.        57 → 554 [ACK] Seq=1 Ack=7873 Win=8212 Len=0
 5     0.000    10.44.10.        57 → 554 [ACK] Seq=1 Ack=10793 Win=8212 Len=0
 6     0.000    10.44.10.        57 → 554 [ACK] Seq=1 Ack=13713 Win=8212 Len=0
 7     0.003    10.44.10.        57 → 554 [ACK] Seq=1 Ack=16633 Win=8212 Len=0

> Frame 5: 60 bytes on wire (48        terface \Device\NPF_{180A8546-AACF-4859-8961-828E92CE6183}, i
> Ethernet II, Src: HewlettP_4e        7:fb:4c (e8:ca:c8:57:fb:4c)
v Internet Protocol Version 4,
    0100 .... = Version: 4
    .... 0101 = Header Length:
  > Differentiated Services Fie
    Total Length: 40
    Identification: 0x4161 (167
  > 010. .... = Flags: 0x2, Dor
    ...0 0000 0000 0000 = Fragm
    Time to Live: 128
    Protocol: TCP (6)
    Header Checksum: 0x9094 [va
    [Header checksum status: Ur
    Source Address: 10.44.10.92
    Destination Address: 10.44.
> Transmission Control Protocol        ck: 10793, Len: 0
```

Fig. 3. Connecting GeoIP database to Wireshark.



```
No. .  Time         Source              Destination         Protocol  Info
    47 139.931463   ThomsonT_08:35:4f   Wistron_07:07:ee    ARP       192.168.1.254 is at 00:90:d0:08:35:4f
    48 139.931466   192.168.1.68        192.168.1.254       DNS       Standard query A www.google.com
    49 139.975406   192.168.1.254       192.168.1.68        DNS       Standard query response CNAME www.l.google.com A 66.102.9.99
    50 139.976811   192.168.1.68        66.102.9.99         TCP       62216 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
    51 140.079578   66.102.9.99         192.168.1.68        TCP       http > 62216 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 \
    52 140.079583   192.168.1.68        66.102.9.99         TCP       62216 > http [ACK] Seq=1 Ack=1 Win=65780 Len=0
    53 140.080278   192.168.1.68        66.102.9.99         HTTP      GET /complete/search?hl=en&client=suggest&js=true&q=m&cp=1 H
    54 140.086765   192.168.1.68        66.102.9.99         TCP       62216 > http [FIN, ACK] Seq=805 Ack=1 Win=65780 Len=0
    55 140.086921   192.168.1.68        66.102.9.99         TCP       62218 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
    56 140.197484   66.102.9.99         192.168.1.68        TCP       http > 62216 [ACK] Seq=1 Ack=805 Win=7360 Len=0
    57 140.197777   66.102.9.99         192.168.1.68        TCP       http > 62216 [FIN, ACK] Seq=1 Ack=806 Win=7360 Len=0
    58 140.197811   192.168.1.68        66.102.9.99         TCP       62216 > http [ACK] Seq=806 Ack=2 Win=65780 Len=0

▷ Frame 1 (42 bytes on wire, 42 bytes captured)
▷ Ethernet II, Src: Vmware_38:eb:0e (00:0c:29:38:eb:0e), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▷ Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 00 0c  29 38 eb 0e 08 06 00 01   ........ )8......
0010  08 00 06 04 00 01 00 0c  29 38 eb 0e c0 a8 39 80   ........ )8....9.
0020  00 00 00 00 00 00 c0 a8  39 02                     ........ 9.

eth0: <live capture in progress> Fil...   Packets: 445 Displayed: 445 Marked: 0   Profile: Default
```

Fig. 4. Analyzing end point statistics in Wireshark.

IV. RESULTS AND DISCUSSION

The project aims to employ Wireshark for gathering network traffic data from the user's computer, converting it into actionable insights, and utilizing Keyhole Markup Language (KML) and Google Maps to visualize the network traffic in a graphical format. The development and implementation of network traffic visualization rely on tools such as Wireshark, Python, KML, and Google Maps. Specifically, a simulation program script written in Python will serve as the implementation method for executing the project.

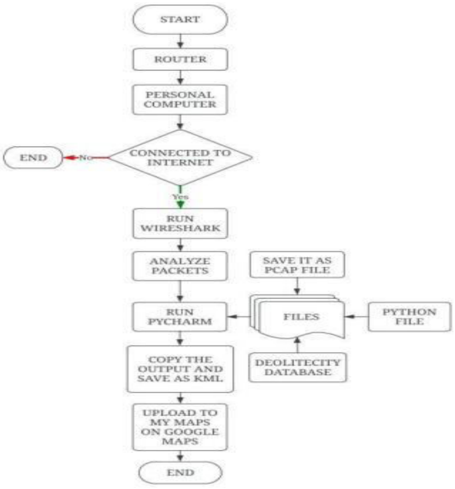


Fig. 5. Implementation flow diagram.

Figure 5 shows the implementation block diagram of this project. The block diagram's explanation outlines the numerous steps that must be taken in order to obtain the desired result. The most significant tool and well-known resource that practically everyone in the world uses is the Internet. It develops a communication channel for online information sharing and retrieval. Routers are networking devices that operate at the OSI model's layer 3 or network layer.. In a typical Ethernet network, a discrete unit of data is referred to as a packet. The packets will be examined, then saved as a pcap file for additional examination .

The following tools and software are used to develop this project.

Wireshark., PyCharm., Geolitecity database, Notepad++, Python Libraries: Dpkt, Socket and Pygeoip.

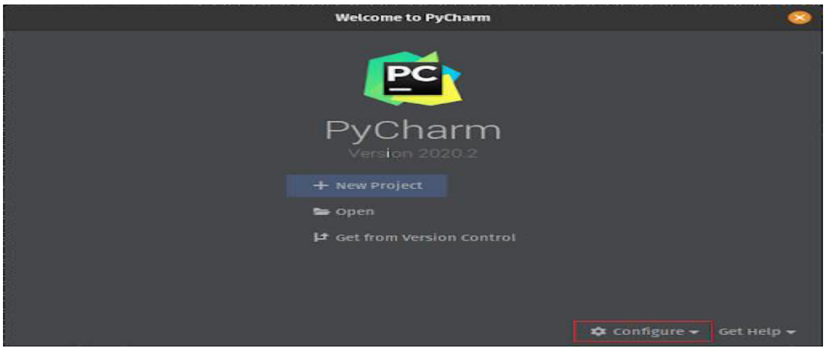
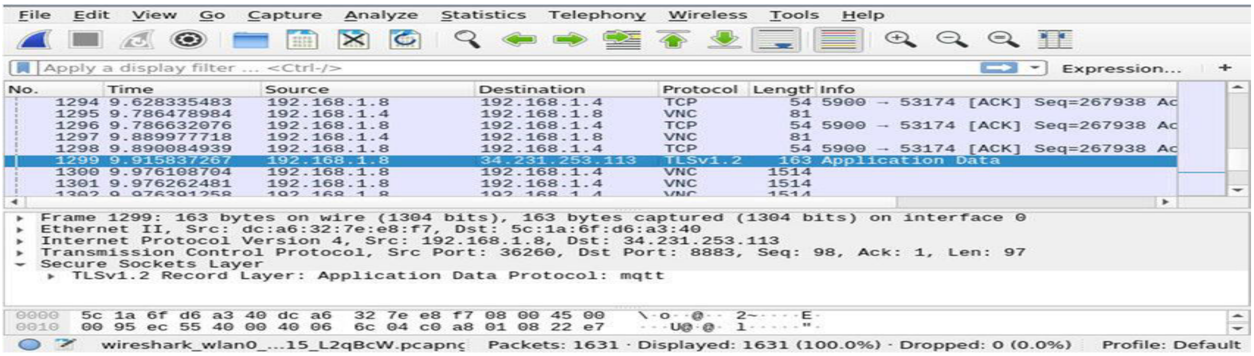


Fig. 6. PyCharm IDE.

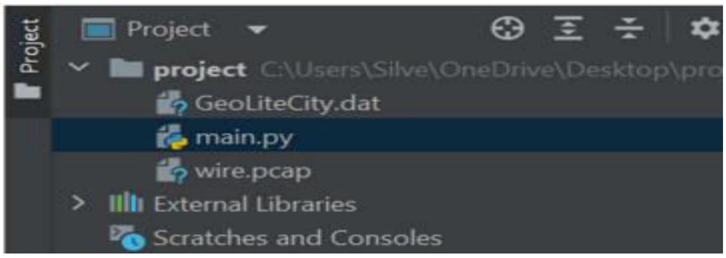
Figure 6 shows the PyCharm IDE Python programming and compatibility with different operating systems, including Windows, Linux, and macOS, were the primary driving forces for PyCharm's creation of this IDE. The IDE includes tools for testing, a debugger, code analysis, and version management.



```
No.     Time         Source          Destination      Protocol  Length Info
   1294 9.628335483  192.168.1.8     192.168.1.4      TCP         54 5900 → 53174 [ACK] Seq=267938 Ac
   1295 9.786478984  192.168.1.4     192.168.1.8      VNC         81
   1296 9.786632076  192.168.1.8     192.168.1.4      TCP         54 5900 → 53174 [ACK] Seq=267938 Ac
   1297 9.889977718  192.168.1.4     192.168.1.8      VNC         81
   1298 9.890084939  192.168.1.8     192.168.1.4      TCP         54 5900 → 53174 [ACK] Seq=267938 Ac
   1299 9.915837267  192.168.1.8     34.231.253.113   TLSv1.2    163 Application Data
   1300 9.976108704  192.168.1.8     192.168.1.4      VNC       1514
   1301 9.976262481  192.168.1.8     192.168.1.4      VNC       1514
   1302 9.976391258  192.168.1.8     192.168.1.4      VNC       1514

▸ Frame 1299: 163 bytes on wire (1304 bits), 163 bytes captured (1304 bits) on interface 0
▸ Ethernet II, Src: dc:a6:32:7e:e8:f7, Dst: 5c:1a:6f:d6:a3:40
▸ Internet Protocol Version 4, Src: 192.168.1.8, Dst: 34.231.253.113
▸ Transmission Control Protocol, Src Port: 36260, Dst Port: 8883, Seq: 98, Ack: 1, Len: 97
▾ Secure Sockets Layer
  ▸ TLSv1.2 Record Layer: Application Data Protocol: mqtt

0000  5c 1a 6f d6 a3 40 dc a6  32 7e e8 f7 08 00 45 00   \·o··@·· 2~····E·
0010  00 95 ec 55 40 00 40 06  6c 04 c0 a8 01 08 22 e7   ···U@·@· l·····"·

wireshark_wlan0_...15_L2qBcW.pcapng   Packets: 1631 · Displayed: 1631 (100.0%) · Dropped: 0 (0.0%)   Profile: Default
```

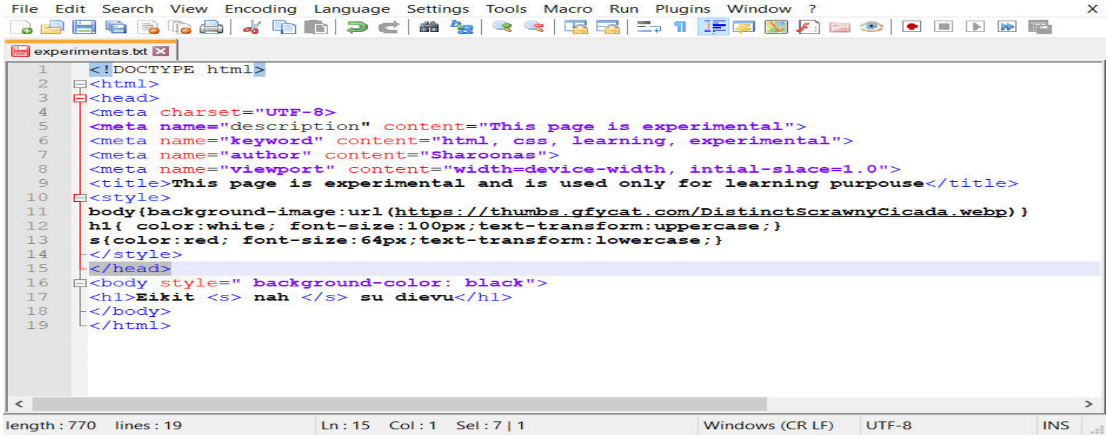
Fig. 7. Wireshark packet analyzer.

Figure 7 shows the traffic analyzer page, which is obtained after selecting the appropriate interface, this screen provides all the information about the traffic, including time, the source and destination IP addresses, the protocol being used, the length of the packet, and other associated details. As a result, Figure 8 displays the files required to run the Python script, make it executable, and prevent run-time errors. The files required for the project are geolitecity database, python script, and pcap file saved from Wireshark.



```
project  C:\Users\Silve\OneDrive\Desktop\pro
    GeoLiteCity.dat
    main.py
    wire.pcap
> External Libraries
  Scratches and Consoles
```

Fig. 8. Files required for this project in PyCharm.



```html
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8>
<meta name="description" content="This page is experimental">
<meta name="keyword" content="html, css, learning, experimental">
<meta name="author" content="Sharoonas">
<meta name="viewport" content="width=device-width, intial-slace=1.0">
<title>This page is experimental and is used only for learning purpouse</title>
<style>
body{background-image:url(https://thumbs.gfycat.com/DistinctScrawnyCicada.webp)}
h1{ color:white; font-size:100px;text-transform:uppercase;}
s{color:red; font-size:64px;text-transform:lowercase;}
</style>
</head>
<body style=" background-color: black">
<h1>Eikit <s> nah </s> su dievu</h1>
</body>
</html>
```

```
length : 770    lines : 19    Ln : 15    Col : 1    Sel : 7 | 1    Windows (CR LF)    UTF-8    INS
```

Fig. 9. Notepad++.

The main screen of Notepad++ is shown in Figure 10. The XML notation known as Keyhole Markup Language (KML) is used to represent geographic maps and three-dimensional Earth browsers. KML was created specifically for Google Earth. A KML file is a type of file that may contain and display several types of information on a map. Keyhole Markup Language, or KML for short, is a method of encoding data using tags and symbols. Many features, including locales, pictures, shapes, models, and links, can be included in a KML file. In addition to names, positions, colors, sizes, and angles, these elements may also have other qualities. Geospatial datasupporting applications like Google

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1203046 |

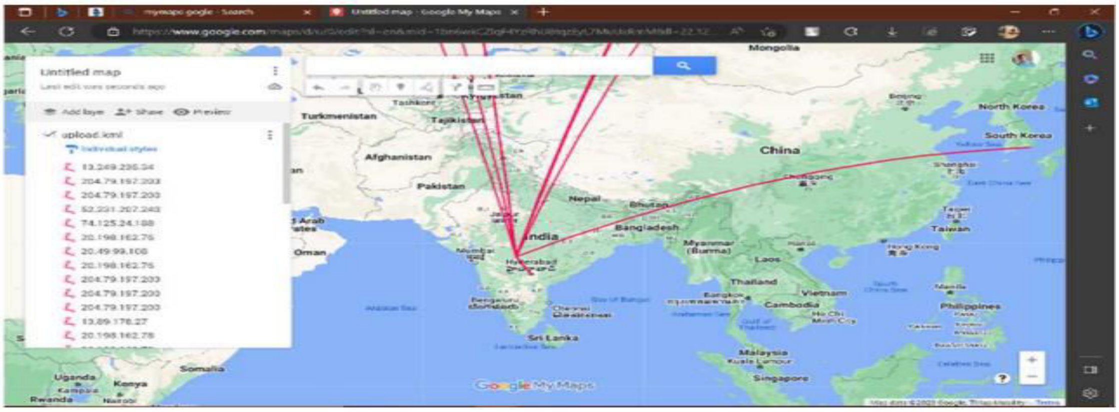
Earth, Google Maps, and ArcGIS can open KML files. For exchanging and visualizing geographic data, a KML file can be helpful.

B. Results.

The innovative method outlined in this invention entails pinpointing the source and destination IP addresses on a Google map, along with establishing a graphical path connecting them. The process unfolds through the following steps:

- Network packets are captured using the Wireshark tool.
- The source and destination IP addresses are extracted from the network packets utilizing a Python script and relevant libraries.
- The Maxmind GeoIP database is queried to acquire latitude and longitude coordinates corresponding to the source and destination IP addresses.
- The obtained coordinates are plotted on Google Maps via "My Maps" in Google Maps, followed by drawing a graphical path between the coordinates using a KML file.

The culmination of the project is depicted in Figure 10, showcasing the fruition of network traffic visualization employing Wireshark and Google Maps.



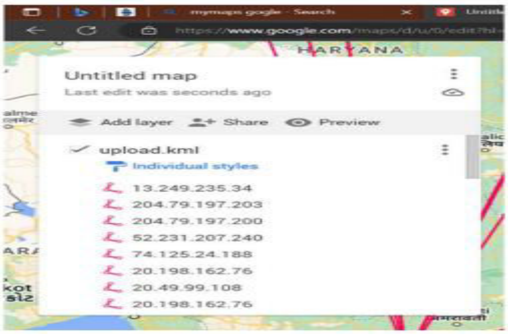
Untitled map

upload.kml

13.249.235.34
204.79.197.203
204.79.197.203
52.231.207.243
74.125.24.188
20.198.162.76
20.49.99.106
20.198.162.76
204.79.197.203
204.79.197.203
204.79.197.203
13.89.178.27
20.198.162.78

Fig. 10. Final output.

The resulting output is a virtual representation of traffic, illustrating a graphical path connecting the source and destination IP addresses alongside their respective locations on Google Maps, akin to the process of pinpointing endpoints using Wireshark within a browser environment. To generate the output on Google Maps, the initial step, as depicted in Figure 9, involves copying the output produced in PyCharm into a text editor like Notepad++ and saving it with a KML extension. These public IP addresses are readily visible on the map, showcasing the graphical path between the source and destination IP addresses. For customized maps, Google Maps supports various file types to convey graphical information, including comma-separated values (CSV), Microsoft XML spreadsheet (XLSX), KML, and GPS exchange format (GPX).



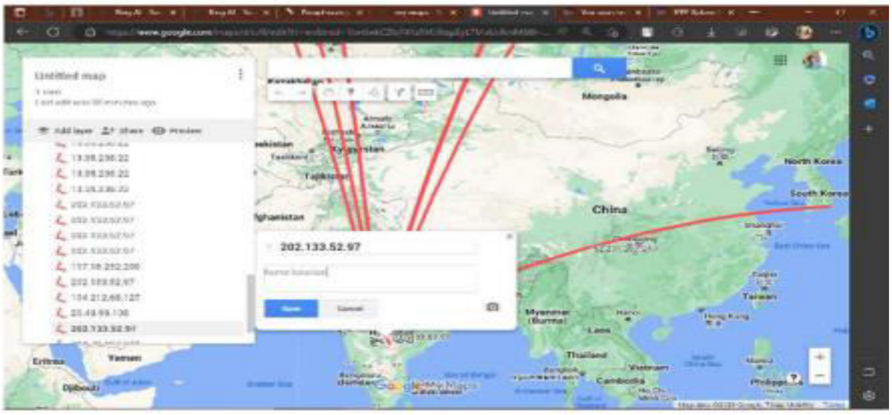
Untitled map
Last edit was seconds ago

upload.kml
Individual styles

13.249.235.34
204.79.197.203
204.79.197.200
52.231.207.240
74.125.24.188
20.198.162.76
20.49.99.108
20.198.162.76

Fig. 11. Map parameters displaying IP addresses.

Figure 11 shows the parameters in the maps, it contains a list of IP Addresses that are extracted previously from Wireshark. All the IP addresses shown in the list are public IP addresses and can be easily accessible on the map. This list highlights the graphical path between the source IP and destination IP and can be modified according to analyst or user requirements by changing color of the track, displaying the IP address for each track, increasing track width, adding a small text note and even images to it as shown in figure 12.



Untitled map

202.133.52.97

Fig. 12. Customization of the map

V. CONCLUSION

Security analysts possess the capability to identify the types of traffic present in a network, detect instances of abnormally high traffic, and pinpoint the network nodes responsible for such activity. This capability is crucial for protecting data and analyzing network traffic to identify cyberattacks like ransomware, fraud, and data breaches. The project aims to achieve this goal through network traffic virtualization using Wireshark and Google Maps, allowing the visualization of data in graphical form to locate the Source IP and Destination IP locations accurately. In essence, traffic virtualization with Wireshark serves as a valuable method for diagnosing and improving network performance in host-only networks. It not only helps in identifying faults or anomalies in network traffic but also provides insights into communication protocols and patterns of virtual machines. However, it does have limitations, including packet loss, storage constraints, and limited coverage of local traffic. As a result, it cannot entirely replace dedicated network appliances or cloud-based monitoring services, which offer more comprehensive and precise analysis of network traffic, especially in cloud computing environments.

VI. FUTURE SCOPE

In the forthcoming years, Wireshark's traffic virtualization holds promise for advancing network research and optimization in cloud environments. Potential enhancements and innovations could target several areas, including refining and expanding capture and display filters to accommodate diverse network protocols and use cases. Integration of Wireshark with additional network monitoring and security technologies could provide a comprehensive understanding of network traffic in cloud environments, thereby enhancing Wireshark's efficiency and scalability to handle heavy loads of network traffic while maintaining precision and reliability. Moreover, on an advanced research front, there is an opportunity to extend this concept by incorporating features such as network speed monitoring, analysis of incoming and outgoing data, and the development of a comprehensive software suite akin to Cisco's Thousand Eyes and Down Detector, along with the establishment of a personalized tracking system.

REFRENCES

[1] Upendra Dadi, Cheng Liu, Ranga Raju Vatsavai, “Query and Visualization of extremely large network datasets over the web using Quadtree based KML Regional Network Links”, IEEE Xplore, DOI 10.1109/GEOINFORMATICS.2009.5293465, 12 August, 2009.
[2] Dong Fang, Cheng Chengqi, Guo Shide, “Design and research on GeoIP”, IEEE Xplore, DOI 10.1109/CSCWD.2010.5472009, 24 May, 2010.
[3] Lili Jiang, Xiaohui Yang, Tao Li, “The Analysis and Design for a Network Protocol Analysis System Based on Wincap”, IEEE 2014 Communications Security Conference (CSC 2014), INSPEC Accession Number: 14611657, 24, May 2014.
[4] G. Bagyalakshmi, G. Rajkumar, N. Arunkumar, M. Easwaran, K. Narasimhan, V. Elamaran, Mario Solarte, Iván Hernández, and Gustavo Ramirez-Gonzalez, “Network Vulnerability Analysis on Brain Signal/Image Databases Using Nmap and Wireshark Tools”, DOI 10.1109/ACCESS.2018.2872775, September 12, 2018.
[5] Sakshi Singh, Suresh Kumar, “Capability of Wireshark as Intrusion Detection System”, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-5, January 2020.
[6] Ahmad Musa, Aliyu Abubakar, Usman Abdul Gimba, Rasheed Abubakar Rasheed, “An Investigation into Peer-to-Peer Network Security Using Wireshark”, IEEE Xplore, DOI 10.1109/ICECCO48375.2019.9043236, 23 March, 2020.
[7] HyunHo Kim, HoonJae Lee, HyoTaek Lim, “Performance of Packet Analysis between Observer and WireShark”, IEEE Xplore, DOI 10.23919/ICACT48636.2020.9061452, 09 April, 2020.
[8] G. Sasi, P. Thanapal, V.S. Balaji, G. Venkat Babu, V. Elamaran, “A Handy Approach for Teaching and Learning Computer Networks using Wireshark”, IEEE Xplore, DOI 10.1109/ICISC47916.2020.9171197, 19 August, 2020.
[9] Sharath Kumar, S. Pallavi, Ramyashree, “An Effective Network Monitoring Tool for Distributed Networks”, IEEE Xplore, DOI 10.1109/I-SMAC49090.2020.9243344, 10 November, 2020.
[10] George Koutitas, Shashwat Vyas, Chaitanya Vyas, Shivesh Singh Jadon and Iordanis Koutsopoulos, “Practical Methods for Efficient Resource Utilization in Augmented Reality Services”, IEEE Access, DOI 10.1109/ACCESS.2020.3042616, December 18, 2020.
[11] Waqas Ahmed, Faisal Shahzad, Abdul Rehman Javed, Farkhund Iqbal, Liaqat Ali, “WhatsApp Network Forensics: Discovering the IP Addresses of Suspects”, IEEE Xplore, DOI 10.1109/NTMS49979.2021.9432677, 18 May, 2021.
[12] Apri Siswanto, Abdul Syukur, Evizal Abdul Kadir, Suratin, “Network Traffic Monitoring and Analysis Using Packet Sniffer”, IEEE Xplore, DOI 10.1109/COMMNET.2019.8742369, 21 June, 2021.
[13] G Jain and Anubha, “Application of SNORT and Wireshark in Network Traffic Analysis”, IOP Conference Series: Materials Science and Engineering, ISSN: 1119 (2021) 012007, doi:10.1088/1757-899X/1119/1/012007, November 2021.
[14] Merve Ozkan-Okay, Ömer Aslan, Recep Eryigit, and Refik Samet, “SABADT: Hybrid Intrusion Detection Approach for Cyber Attacks Identification in WLAN”, IEEE Access, DOI 10.1109/ACCESS.2021.3129600, December 3, 2021.
[15] Bindu Dodiya, Umesh Kumar Singh, “Malicious Traffic analysis using Wireshark by the collection of Indicators of Compromise”, International Journal of Computer Applications (0975 – 8887) Volume 183 – No. 53, February 2022.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING