Geolocation-Based Source Tracking for Threat Identification on Telegram

Mukilan R.^{1,3}, Dr. T. V. Ananthan^{2,3}, Dr. S. Geetha^{2,3}, S. Ram Sundar^{4}

[1]Postgraduate Student, [2]Professor,
[3]Department of Computer Science and Engineering, Dr. M.G.R. Educational and Research Institute,Chennai.

[4]Managing Director and Cybersecurity Researcher, HebeSec Technologies, India & Malaysia

Abstract **–**With the increasing popularity of instant messaging platforms like Telegram, the need for efficient and reliable call source tracking has become crucial in various domains, including law enforcement, cybersecurity, and intelligence gathering. This project focuses on utilizing Wireshark, a powerful network packet analysis tool, to capture and analyze network traffic in order to track the source of Telegram calls. The foundational stage involves setting up Wireshark to intercept packets traversing through network interfaces, capturing both incoming and outgoing data exchanges to create a comprehensive record of network activity. Subsequently, specific packets of interest are isolated through meticulous filtering processes within Wireshark. Once these packets are extracted, the project shifts towards location identification by analyzing the content within them, including IP addresses. By leveraging Wireshark's detailed protocol dissections, the project aims to derive location-specific data associated with the captured Telegram call activities, facilitating source tracking and enhancing network security measures.

Keywords: *Geolocation databases, Cybersecurity, Wireshark, Telegram calls tracking, Packet capturing, Folium*

1.Introduction

Packet capture at the network level and source call tracking form the basis of the network analysis and security in this complex field. Such techniques contribute immensely in the overall understanding of dealing and improving networks against probable vulnerabilities and threats. The concept of network packet capture is based on the listening to and thorough recording of the data packets that are being transmitted through the infrastructure of a network. It is not just data acquisition process but a scrutiny of the complex network anomalies and glitches that is also part of it. Packet inspecting in detail, not only revealing clues about network characteristics but also helping in correcting errors and thereby boost network performance.

Alongside, source call tracking is developed as an indispensable proficiency in a network which helps trace the origins and trajectory of individual calls or packets within the system architecture. Through the use of network tracers, network administrators acquire vital information into the sequence of interaction and the complex routing paths the data take. These intelligence findings play a critical role in the detection of possible threat sources, abnormalities, or a suspicious activity within the network infrastructure. The tracking of sources is a powerful instrument that makes it possible to identify malicious activities and unauthorized intrusions very quickly and to respond effectively in order to guard the network integrity against all possible threats.

The incorporation of these approaches constitutes a base for network monitoring and protection. As well as the ability to analyse the packets in real time or retrospectively packets offer an overview of the network that enables a proactive approach to security. With a thorough examination of the exchanged packets, deviations, trends, and patterns are pinpointed and addressed in a proactive manner, thus providing an important improvement to the security position.

The packets, which is small pieces of data, travel via the internet. Meanwhile, with every single instance of the user data sent in these data packets, all of these packets therein are then disassembled into separate ones followed by which each of these packets is converted into a sequence of individual bits. Within this maze, data packets called bytes are the infinte tiny travelers that make their way across the internet thru routers and switches. The next step is seamless as the bits navigate through the socket and reconstruct the original intended meaning that is interpreted by brains; either human or animals. These stages of packaging and passing on brought in proper and efficient transmission systems for high speed and successful transmission of the data down long lines of networks, thereby creating the sharing of information worldwide.

The value of source call tracking detection being a significant part of investigation can hardly be underrated as the incidents of network foresight analysis. With a perfect tracking of packets or calls that are individual, security analysts can recreate the timeline of events that took place before the intrusion and abnormalities started. It is also the primary function of forensic fraternity which helps in determining the technique used in the cyber attack, and leads to the development of fruitful countermeasures.

In fact, the application areas of these concepts go across the whole range from corporate network safety to nationwide infrastructure security and beyond. Companies gain a lot from the deep packet inspection when monitoring their networks because their performance is tested and if any bottlenecks exist they will be identified during the process and the network efficiency is thus kept at the optimum level. Just as in the cybersecurity space, phone tracking has been proven to be effective by assisting establishment of the source and track of any calls. This enables identification and confinement of security breaches that can be minimized before they go out of hand and cause great loss of assets.

Network packet capturing and source call tracking are inseparable means for network analysts and network security specialists. Such skills synergy can allow for proactive monitoring, speedy incident reaction, and network defence from emerging disasters. The discussion of these approaches as well as the practical implementation of them shows the critical importance they have to ensure reliable and secure network resources in the face of the ever growing networked digital world. This article explain the techniques and methods that are used in the network traffic capture and source call tracking, indicating their relevance and providing examples of their applications.

2. Related Works

2.1. NETWORK PACKET CAPTURE

Network packet capture, which is key element used in the fabric of network management and security, works at the center of network understanding, monitoring, and security. Its major function is the intrusion and storage of network packets constituting a plethora of information elements—starting from the origins and destination of data down to the very contents of the pipelines. The underlying principle is a variety of tools and techniques which begins with packet sniffers and network analyzers, and peacefully concludes with the archiving and analysis of these packets.

Network packet capture is more than just data storage; it is the base building block for a whole range of critical network management job functions. First of these is network troubleshooting, where these stored packets can be a resource of valuable information. Administrators dive into these packets to detect and resolve various types of network disorders such as congestion points, transmission errors and connectivity issues. Gleaning insights from captured packets gives the power to do troubleshooting in swiftness and effectiveness, network performance and reliability being optimized.

Secondly, as a security specialist, network packet capture is also essential in the case solving process. The analysis of the network packet contents enables the administrators not only to discover the suspicious network traffic in a proactive way but also to prevent the abnormal network activity within the eco-system. Irregularities in data sets or dispose of unauthorized access surgeries usually use these packets to run preemptive activities against malicious actions that might escalate. On the flipside, that power comes with a ton of responsibility, and privacy is an issue that can arise. Collected information can be sensitive, containing only confidential data, which will need to reconcile without compromising usefulness and privacy.

As with every approach, ethical and legal concerns become a key part of implementing a network data packet capture method. Inculcating the tact of gathering captured data to create a safe and optimized network and ethical guidelines as well as the legal framework requires extra care. Data security also needs to go hand in hand with measures to protect captures data from unauthorized access and misuse, and deploy applications responsibly with transparency to direct the purpose tasks that is; network management and security improvement.

Network packet capture is a term that has relevance in uniting different sectors, servicing a wider industries base. Packet analysis serves as an optimization process for businesses, helping with network maintenance and problem-solving. Critical service sectors make use of that strategy which is designed for the hardening of the networks against the possible threats; therefore, these services are protected as essential ones. Also in the cybersecurity domain, packet save is a forward-thinking solution which contains the power to proactively predict and successfully counteract new threats.

Deep packet inspection remains the key element of modern deployment and security practices. Not only does it help spot network performance issues but it also offers additional security to protect the networks from hackers by serving as a crucial pillar in sustaining well-functioning and resilient networks. While issues like ethics and privacy do require that the data be safely stored, used, and destroyed, responsible helming and respectful management of this data are mandatory. Regulatory frameworks anchored on compliance to laid down laws and transparency in operations would be applied to practice in order to realize the positive attributes of packet capture while still preserving human rights and organizational position.

2.2. METHODS OF NETWORK PACKET CAPTURE

Network packet capture is a crucial element in the analysis, examining, and assuring the safety of the network communication. Different approaches create the basis of this vital method, which is equipped with unique strengths and weaknesses. Such techniques include monitoring applications such as Wireshark and tcpdump, as well as using hardware tools and software-defined networking technologies. The totality of these solutions ensures effective analysis and management of network traffic.

Software-Based Capture: Wireshark and tcpdump: Wireshark / tcpdump - it is a software-based packet capture tools, allowing to intercept and analyze network traffic on the fly. Network administrators can take advantage of the intuitive features to keep watch on, filter, and scan packets using different approaches. Providing the ability to discern control and data-plane protocols as well as source, target IPs, and port numbers, these tools enable user-interactive tracking, assisting in troubleshooting, optimization and security analysis.

Hardware-Based Capture: Network Taps and Port Mirroring: Contrarily, the functioning of the hardware-based packet capture devices such as network taps and port mirroring devices is passive, that is, they just duplicate the whole stream of traffic of the network segment. Unlike the encapsulative mode, their non-intrusive nature improves packet capturing without affecting network performance making them ideal for high speed networks. Nevertheless, the introduction of this equipment may be problematic and sometimes challenging to be adapted in all network setups.

Software-Defined Networking (SDN) Approach: Additionally, software-defined networking (SDN) could be very useful, for the packets are captured from the network switches and routers themselves. Besides the improved efficiency and scalability there is also the possibility of real-time views and sifting at the network edge. This integration in network infrastructure architecture is the dawn of network traffic management that is quick and ready for any contingency, as it allows dynamic and scalable packet capture.

Passive Network Monitoring: Through non-active network monitoring means traffic observation and analysis is performed but without intervening or modifying the flow of the packets are allowed. It enables monitoring to be performed without any mention of it or disturbing the natural course of network communication. Using the port mirroring, network taps, or span ports on switches, chiefly for the purposes of streamlining troubleshooting, security monitoring, and performance tuning of the traffic, administrators can intercept and analyze the traffic. Active Network Probing: Unlike passive monitoring, active probing network fosters the active process where sending probing (packets or signals) into the network in order to check the responsiveness, performance and integrity. It is flawless. Tools such as ping sweeps, traceroute, or ping command contribute to the assessment of the routing paths, delays and reachability by sending controlled network traffic. These techniques help in the determination of network issues, locating possible bottlenecks and confirmation of network arrangements.

Statistical Traffic Analysis: Statistical traffic analysis methods consist of statistical-algorithms and models to detect the patterns, trends, as well as the anomalies of network traffic. By employing many statistical approaches, such as anomaly detection, clustering, or regression analysing, network administrators can differentiate between normal and abnormal traffic patterns and also discover behaviors that differ from the expected ones. Through these techniques, more sensitivity can be achieved in detection of anomalies while malicious actors or scenarios are recognized.

Signature-Based Packet Inspection: Signature-based packet filtering matching packets to existing signatures or patterns spotting known malicious behavior or a particular type of traffic. The IDSes and IPSes are the systems, which utilize the signature-based detection and prevention mechanism to detect and prevent threats, such as malware, viruses or specific network attack patterns. This approach is based on big data repositories and syntactic rules to recognize and block the threats.

Behavioral Packet Analysis: In behavioral packet analysis, behaviour of the network packets is seen and scrutinized in order to detect traffic deviations from their normal sequences. The baseline of expected behavior is created. All abnormal traffic volumes, anomalies in the communication patterns, and packet characteristics are then flagged for in-depth investigation. Behavioral analysis pinpoints zero-day attacks or any other known threats by alerting the security team of any abnormal patterns.

Hybrid Approaches: Hybrid approach is about the joining of the packet capture and analysis methods which use the combined synergies to get a comprehensive perception of the overall network behavior. These multilayer approaches could be combined by deep packet inspection, statistics analysis, and behavioral monitoring, which make threat detection more accurate and reduce false positives.

These methods, when compared, show the range of expertise and virtues. Applications like Wireshark and tcpdump are the best in their field with regard to their interface design and analysis features but will degrade system performance. Hardware-based devices offers non-intrusive capture but it happens with some hardware requirements. SDN-based methodologies are scalable and in real-time, but the existing SDN infrastructure should be in place.

The diverse packet captures of these methods find an application both in different sectors and different cases of usage. From enterprise networks to critical infrastructure, the data taken from packet analysis is used for troubleshooting, performance optimization, and providing protection against security threats. Whether discovery of anomalies, resolving performance bottlenecks, or preventing security threats, such methods play an essential role in ensuring healthy operationality of networks.

Nevertheless, the way in which they are undertaken poses certain problems. The ethical ramifications that arise from data privacy as well as the legal frameworks that have to be taken into account, along with the technical complexities and resource availability should be considered with meticulousness. However port selection also depends on network topology, traffic size and the scope of the packet analysis.

The responsible deployment of packet capture methodologies necessitates adherence to ethical guidelines and legal frameworks, particularly concerning data privacy. Balancing the need for comprehensive analysis with privacy concerns mandates transparent practices and stringent security measures to protect captured data from unauthorized access or misuse.

The array of network packet capture methods underscores the dynamic nature of network analysis and management. Each method offers unique advantages, fostering flexibility in capturing and analyzing network traffic. Whether through software tools, hardware devices, or SDN-driven approaches, these methods empower network administrators to troubleshoot, optimize performance, and fortify network security. Their collective utility unravels the intricate communication patterns, enabling robust and resilient network infrastructures. Yet, the judicious selection and implementation of these methods must align with ethical standards, legal obligations, and technical requirements to harness their full potential while safeguarding individual privacy rights.

2.3. TECHNIQUES FOR NETWORK PACKET CAPTURE

The network packet capture yields a major function of understanding of monitoring and regulating network interaction. A plethora of methods are developed to perform with the packet capture and the analysis, through different approaches and technologies. The measures adopt a software-based snoopers as well as a hardware approach. These measures offer remarkable powers and are essential in network assessment and debugging.

Network Sniffer Tools: Assessing Software-Based Models: Applicable Tools: Network sniffers and packet analyzers software is the primary approach for capturing packets. These provide the operation by passing the NIC into promiscuous mode and making it the capture all the packets passed through the network. Their exemplifiers are tcpdump and Wireshark ready to perform a packet trace to specific hosts or network interfaces. With more granular details on packet data in mind, application-level insights become possible including source and destination addresses, protocols used, and actual data in the packets, crucial for in-depth network analysis.

Hardware-Based Approaches: The more advanced alternatives like the network taps and port mirroring facilitate the passive way of the monitoring of net traffic. The network taps physically intercept and copy network traffic as they go by with no latency or interference. As well, switch port mirroring mirrors the traffic in a port for monitoring and analysis. These techniques guarantee that the process of packet acceptance is not insertive, which is suitable for very fast networks and may, however, necessitate additional hardware infrastructure for implementation.

Host-Based Packet Capture Tools: On the other hand, network-centric methods are accompanied by host-based packet capture tools like tcpdump and Wireshark that enabled the resolution of interesting capture from an individual host or specific network interfaces. Such tools render themselves as potent allies in host diagnosing the network issues, supplying the exact packet information that is vital for determining the problems of specific machines or interfaces.

Comprehensive Packet Insights: Irrespective of the method used, network packet capture tools provide valuable information indispensable for network analysis. They disclose fine details, like source and destination addresses, protocols used, and payload data allowing administrators to spot network abnormalities, detect possible threats and enhance network efficiency.

Deep Packet Inspection (DPI): One of the cutting-edge methods is Deep Packet Inspection (DPI) which gives an ability to fully check packet contents at granular level. DPI looks beyond the traditional packet header examination to the structure of the payload and extracts the application-specific information. Through the process of checking packet payloads, DPI makes it possible to determine particular applications, protocols, or even the extraction of file content. But the DPI's comprehensive analysis needs heavy processing power and might come into privacy consideration because of its intrusive nature.

Flow-Based Packet Capture: Flow-based packet capturing seeks to consolidate and examine network flows rather than individual packets. This is realized in form of a flow, which is a sequence of related packets that makes it possible to have an abstracted view of network traffic. These approaches can use either NetFlow or sFlow, which are methods that provide aggregated data about traffic patterns to facilitate network traffic analysis, anomaly detection, and capacity planning. With the flow-based capture, the storage requirements decreases significantly compared to the full packet capture, thereby providing an overview of traffic flows without deep packet analysis.

Ring Buffer Capture: Ring buffer capture techniques consider the continuous overwriting of captured packet data on a circular buffer. This method makes it possible to continuously capture packets by discarding old data as new captured packets are captured. This cyclic memory encourages ongoing capture without the risk of losing data from storage limitations. On the other hand, this could cause missing the specific packets during high traffic periods.

Packet Sampling: The technique of packet sampling refers to the only the part of the network packet collection rather than the transmitting of the full packet. It does this by allowing the system to sift through the most important data and helps the network analyst to draw important conclusions on the network performance. Packet sampling can be implemented using other techniques like random sampling or time-based sampling that gives network admins great insights into the trends and behaviours of the network without needing to capture every individual packet.

Encapsulated Traffic Analysis: With the encapsulated traffic analysis, we are in the days when networks transfer via other protocols. This strategy implies packet dissection and analysis of the structure which carries all the internet works. The approach can either be the Virtual Private Network (VPN) traffic or even protocols that are contained within tunnels. We so are able to see the packet itself after the decapsulation, thus we have a visibility over the network communication that might have been otherwise not clear due to the opacity of the packets.

Resource Requirements and Considerations: Just like other type of process, network packets capturing needs massive storage and computing power due to the abundance of information generated. These large volumes may overload the storage systems resulting in the process being more time-consuming.

The selection of network packet technologies employed in different industries (from banks to healthcare) reinforces the reality of the business. The knowledge gained can help in identifying network problems and finding optimal performance as well as safeguards the network from possible attacks. Alongside, these methods are really very essential to maintain the good networks' infrastructures without stability and security. Whilst the adoption of packet capture devices has a number of benefits, the implementation process is often encountering a few challenges. On the contrary, ethical issues of privacy and the legal structure are the keys, what should not be left out is a strong enforcement. Also, technological intricacies, size restrictions, and device management play a big role in bringing out these technologies to all users.

These network packet capture techniques showcases the diversity and versatility inherent in network analysis methodologies. Whether through software-based sniffers, hardware-centric approaches, or host-based tools, these techniques unravel the complexities of network communication. Their collective utility empowers administrators to diagnose issues, optimize performance, and bolster security across network infrastructures. Yet, their effective implementation mandates careful consideration of resource requirements, ethical obligations, and technical intricacies to harness their full potential in maintaining robust and resilient networks.

2.4. SOURCE CALL TRACKING

Source call tracking method involves a detailed tracing of an individual call or packet that take place within the network infrastructure. The route trace function reveals to the network administrators the deep network paths used by data packets and helps them in locating traffic disturbances, the origin of disruptions or security threats. Through the explanation of how a call travels, source call tracking gives the administrators the ability to take highly focused and effective measures when trouble shooting network problems.

The original utility of source call tracking is to reveal the complex network of devices and systems that a call traverses during its transmission journey. This high-level view of the network's behavior furnishes administrators with a visual depiction of data flow, enabling them to identify anomalies or deviations from the normal patterns that occur. The visualization of data flow patterns allows anomalies, inconsistencies and unauthorised access attempts to be identified, revealing possible network security threats or operational inefficiencies.

Also, source call tracking does more than observing; it is a proactive tool for maintaining the integrity of the network. The ability to monitor calls or packets flow in this way makes it possible to identify abnormalities early on and take the appropriate action immediately to prevent further disruptions or security problems. This preventive approach enables administrators to intercept problems right in their beginning, thus reducing the damage to operations and improving the network resistance.

Source call tracking is multidimensional in various types of network management and it brings great benefits to different departments operating in divergent domains. Tracing is a method that guides administrators to find the exact point of the network disruption and fix the problem. Such identification becomes more specific enabling targeted interventions and quick resolution. Yet, in the cybersecurity sphere, the question of where a call was made and by whom becomes the most crucial aspect. Source call tracking is a fundamental input to the forensic analysis of cybersecurity breaches and the identification of possible attack vectors and also offers insights into the attack roles and the methods of such malicious activities. Through the section on the transmission path, administrators get a vital piece of information that they use it to implement preventive measures that help keep the network safe from future attacks.

While source call tracking can be effective when supported by advanced and comprehensive network monitoring mechanisms, its effectiveness depends on integration of such tools and methods. Adequate and advanced data capture mechanisms are among the essentials in the transmission path tracing. Further, the scalability as well as the complexity of the modern day networks is among the factors which make the capturing and analysis of huge volumes of data challenging leading to need for sophisticated data handling and processing systems.

Adding source tracking to the core functions of network operations and security is very important. By being able to trace call origins and routes within the network, administrators are provided with unrivalled informational resources in the assessment of the network behavior and possible weaknesses. The methodology that visualizes data flows and anomalies alerts early and facilitates proactive troubleshooting. It improves network security and resilience. However, the successful application of source call tracing needs the integration of powerful monitoring tools, improved data analysis capabilities, and highly nuanced understanding of network intricacies so as to fully realize its potential in providing security and optimal network infrastructural enhancements.

2.5. METHODS & TECHNIQUES OF SOURCE CALL TRACKING

Network Packet capture is a primary technique applied in networks to track the route of the source calls in an infrastructure. This technique is characterised by the precise interception and recording of data packets at different points of the network. This would make it possible to examine not only the content but also the origin, the destiny and the routes followed by all the calls thus captured.

Techniques and Tools for Network Packet Capture: Network packet capture mostly utilizes tools such as packet sniffers and network analyzers that are embedded both in software and hardware platforms. These instruments help catch and store network packets to be further investigated. Software approaches like Wireshark and tcpdump as well as network taps or port mirroring devices allow for packet capture without any impact on network performance, giving a detailed look into the data being transferred through the network.

Gaining Visibility into Network Traffic: The network traffic that passes through the network via packet capture administrators gain an unparalleled insight into. These packets provide a comprehensive understanding of the calls being exchanged, which in turn helps the administrators to scrutinize the packets' details such as content, source and destination addresses. Visibility is the key factor that contributes to the identification of the networks possible sources of issues, anomalies and security threats.

Analyzing Captured Packets for Insights: Decoding those captured packets is the key that will lead to a discovery of a wealth of information about the patterns and trends within the network traffic. Through source call inspection, one can reveal patterns, irregularities or deviations from the expected norm by looking at the content and attributes of communication. These facts help understanding the essence of source calls which forms basis of troubleshooting and proactive security measures.

Detecting Issues and Security Threats: This immediate detection of deviation from standard and possible security concerns among the captured packet contents is made possible by a detailed analysis of captured packets. Anomalies in call content that seem unusual, traffic irregularities, or access attempts that are unknown can be detected through packet analysis and administrators have the ability to handle such kind of problems with speed and then provide security for the network from interruption.

Understanding Source Call Behavior: While packet capture provides an unprecedented vantage point for the contextualization of source calls in terms of the characteristics and behavior, it also allows one to comprehend their flow, context, and timeline better. By exploring the telecommunication channels, administrators receive valuable information about the dynamics of calls and can recognize common paths used, the protocols of the network, and possible inefficiencies and bottlenecks in the transfer of calls.

Behavioral Analysis and Machine Learning: Conducting behavioral analysis involving machine learning algorithms helps indicates irregular or deviated call behavior from the average pattern. Through this method, administrators have the ability to mark behavioral baselines and machine learning models in order to detect unusual behaviors or patterns that are taking place right in the show time. The utilization of such methods develops an ability to determine the deviations or distortions initiated within source calls.

Combining Multiple Analysis Techniques: Hybrid sets of techniques, which all combined, may draw on the strengths of each one to provide a complex picture of source calls. blending stream flow analysis, DPI, metadata, and behavioral analysis are in a way synergic tools that take the precision and depth of sourcing anonymity to an entirely new level.

Enhancing Network Intelligence and Optimization: Packet data from the capture, besides prompting timely problem fixes is also instrumental towards network performance optimization. Through analyzing the behavior and the traits of these calls, an administrator can implement optimal network configurations that space the routing paths, and prepare provisionally in advance to the occurrence of congestion or performance problems.

In spite of the fact that effective packet capture and analysis impose various types of problems on the scale of data volume, storage requirements, and processing power, these challenges can be handled with modern technology. The massive number of packets may flood limited capacity of slots and cause bottlenecks to transmission. This implies the need for efficient storage, powerful processing power and big data analytics to make sense of data.

In essence, network packet capture serves as an indispensable technique in tracking source calls within a network. Its ability to intercept, record, and analyze packets offers unparalleled insights into call behavior, aiding in issue resolution, security enhancement, and network optimization. However, its successful deployment necessitates robust tools, efficient resource management, and adept analysis capabilities to harness its full potential in tracking and understanding source calls within complex network environments.

3. Methods And Materials

3.1. PROPOSED METHODOLOGY

In the proposed system, we aim to enhance security measures within the Telegram messaging platform by implementing call tracking capabilities using Wireshark packet capturing and geolocation databases. The primary objective is to identify the origin of threatening calls received through Telegram by analyzing the IP address associated with the incoming call.

Upon receiving a threatening call on Telegram, the system initiates the process by activating Wireshark, a network protocol analyzer widely used for network troubleshooting, analysis, and communications protocol development. Wireshark is configured to capture packets transmitted over the network, allowing us to intercept and examine the data packets associated with the incoming call.

As the call progresses, Wireshark captures various packets exchanged between the sender and receiver. Of particular interest is the "binding request user" packet, which contains crucial information about the source of the call, including the IP address. This packet serves as a key indicator for tracing the origin of the call and is extracted from the captured packet data.

The next step involves leveraging a geolocation database to determine the physical location associated with the identified IP address. Geolocation databases contain comprehensive information about IP addresses and their corresponding geographic locations.

Using the IP address extracted from the "binding request user" packet, the system queries the geolocation database to retrieve information regarding the location of the caller. By cross-referencing the IP address with the database records, the system can pinpoint the geographical origin of the threatening call.

Once the location data is obtained, it can be presented to the appropriate authorities or security personnel for further action. This information empowers law enforcement agencies or security teams to take appropriate measures to address the threat effectively, such as initiating investigations, enforcing legal actions, or implementing additional security measures to prevent future incidents.

3.2 PROPOSED ARCHITECTURE

The foundational stage of the project centers on leveraging Wireshark, a potent tool for capturing and analyzing network packets. Wireshark serves as the gateway to understanding the intricate communication dynamics within a network. By capturing both incoming and outgoing packets, it provides a comprehensive view of data exchanges, protocols utilized, and the flow of information across the network infrastructure. This phase involves setting up Wireshark to intercept and capture packets traversing through the network interfaces. As data streams through these interfaces, Wireshark captures this information in real-time, creating a detailed record of network activity, laying the groundwork for subsequent analysis.

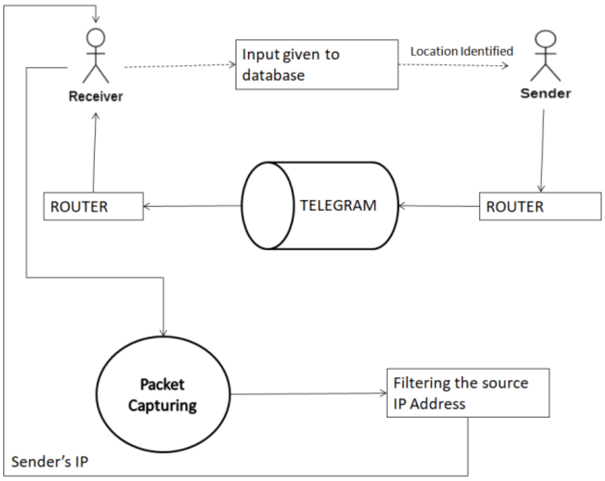


Fig 1. Architecture Diagram

With a trove of captured packets at hand, the focus shifts to isolating specific packets of interest specifically targeting the "binding request user" packet.

With the list of packets captured, the specific packet will be searched manually and identified. Once the packet is identified and IP address is obtained, then the obtained ip address will be given to the geolocation database for location identification. The geolocation database will pinpoint the latitude and longitude of the particular ip address from the information it have about that particular ip address.

4. Implementation

As the GUI initializes using the Tkinter package in Python, it sets the stage for a multifaceted network packet tracing and analysis system. The aim is to capture packets flowing through the network, especially those associated with Telegram calls, and analyze them using Wireshark. Let's dive into the intricacies of how this system operates and the various components involved.

The first step involves the initialization of the graphical user interface (GUI) using Tkinter. Tkinter provides a convenient way to create GUI applications in Python. Through this interface, users can interact with the system and trigger different functionalities seamlessly.
Once the GUI is up and running, the system initiates both the Telegram and Wireshark applications using the os package in Python. This is achieved by executing the respective application binaries or commands within the Python environment.

With Wireshark launched, the system starts capturing network packets in real-time. Wireshark is a powerful network protocol analyzer that allows for detailed inspection of network traffic. By capturing packets at the network interface level, it provides valuable insights into the data being transmitted across the network.

The packet capturing process continues until a predetermined threshold is reached, typically set to a minimum of 1000 packets. This ensures that a sufficient amount of data is collected for analysis, enabling meaningful insights to be drawn from the captured traffic.

As the system monitors the network traffic, it specifically watches for incoming calls within the Telegram application. When a call is detected, the associated packets are captured and displayed alongside the ongoing packet capture in Wireshark.

Amidst the captured packets, the system's next task is to manually identify a particular type of packet known as "binding request user." This packet type is significant in the context of network communication protocols and may contain valuable information about the communication session.

Once the "binding request user" packet is identified, the system extracts the IP address associated with it. This IP address serves as crucial input for the subsequent phase of the analysis.

Returning to the GUI, the extracted IP address is entered by the user. Upon clicking the "trace IP" button, the system initiates a search for the geographical location of the specified IP address.

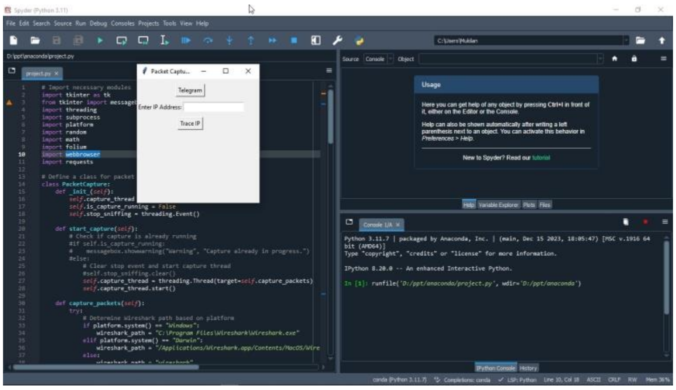
To accomplish this, the system leverages various Python packages, including requests, subprocess, and webbrowser. These packages enable the system to interact with online geolocation databases and retrieve relevant information about the IP address.

Upon retrieving the geographical coordinates associated with the IP address, the system utilizes the folium package to pinpoint the location on a map. Folium is a Python library that allows for the creation of interactive maps using Leaflet.js, providing a visually appealing representation of geographic data.

As the location of the IP address is pinpointed on the map, users can visually inspect the geographic context of the network traffic. This enhances the understanding of the network topology and facilitates further analysis of the communication patterns.

5. Outputs & Discussion

The aim of this study was to develop a method for tracking the source of Telegram calls using a geo-locator. The approach involved executing a specific code, which triggered a sequence of actions leading to the identification of the source IP address of incoming calls on the Telegram application. Subsequently, this IP address was used to trace the geographical location of the caller.


Spyder (Python 3.11)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\ppt\anaconda\project.py
project.py
Packet Captu...
Telegram
Enter IP Address:
Trace IP
C:\Users\Muklan
Source Console Object
Usage
Help Variable Explorer Plots Files
Console 1/A
IPython Console History
conda (Python 3.11.7) Completions: conda LSP: Python Line 10, Col 18 ASCII CRLF RW Mem 36%

```
# Import necessary modules
import tkinter as tk
from tkinter import messageb
import threading
import subprocess
import platform
import random
import math
import folium
import webbrowser
import requests

# Define a class for packet
class PacketCapture:
    def _init_(self):
        self.capture_thread
        self.is_capture_running = False
        self.stop_sniffing = threading.Event()

    def start_capture(self):
        # Check if capture is already running
        #if self.is_capture_running:
        #    messagebox.showwarning("Warning", "Capture already in progress.")
        #else:
            # Clear stop event and start capture thread
            #self.stop_sniffing.clear()
            self.capture_thread = threading.Thread(target=self.capture_packets)
            self.capture_thread.start()

    def capture_packets(self):
        try:
            # Determine Wireshark path based on platform
            if platform.system() == "Windows":
                wireshark_path = "C:\Program Files\Wireshark\Wireshark.exe"
            elif platform.system() == "Darwin":
                wireshark_path = "/Applications/Wireshark.app/Contents/MacOS/Wire
            else:
```

Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
Python 3.11.7 | packaged by Anaconda, Inc. | (main, Dec 15 2023, 18:05:47) [MSC v.1916 64
bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.20.0 -- An enhanced Interactive Python.

In [1]: runfile('D:/ppt/anaconda/project.py', wdir='D:/ppt/anaconda')
```

Fig 2. GUI when the code is executed

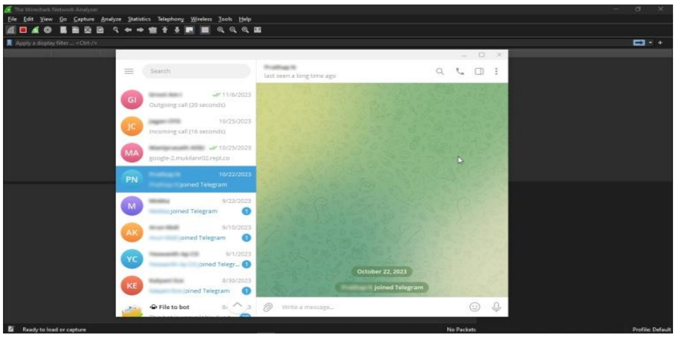
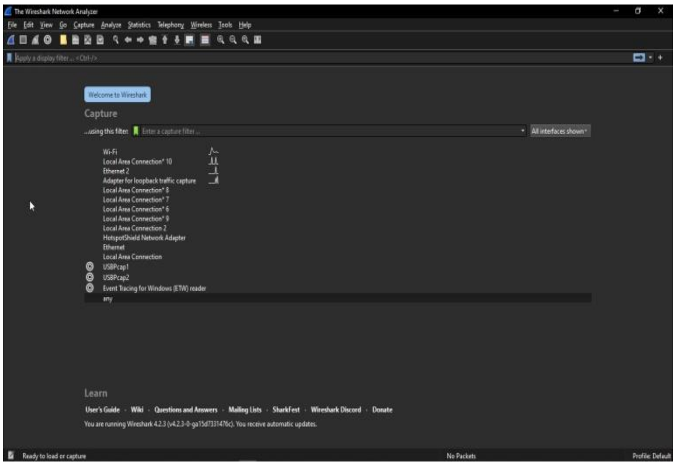
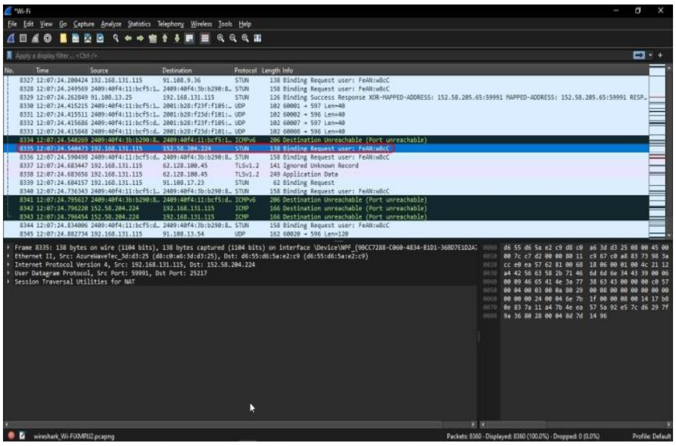

The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Search
GI
Outgoing call (20 seconds)
JC
Incoming call (16 seconds)
MA
PN
M
AK
YC
KE
File to bot
October 22, 2023
Write a message...
Ready to load or capture
No Packets
Profile: Default

Fig 3.1 When the telegram button is clicked telegram application opens.


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
Wi-Fi
Local Area Connection* 10
Ethernet 2
Adapter for loopback traffic capture
Local Area Connection* 8
Local Area Connection* 7
Local Area Connection* 6
Local Area Connection* 9
Local Area Connection 2
HotspotShield Network Adapter
Ethernet
Local Area Connection
USBPcap1
USBPcap2
Event Tracing for Windows (ETW) reader
any
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate
Ready to load or capture
No Packets
Profile: Default

You are running Wireshark 4.2.3 (v4.2.3-0-ga15d7331476c). You receive automatic updates.

Fig 3.2 wireshark application opens at the same time as the telegram opens.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
wireshark_Wi-FiXMPIJ2.pcapng
Profile: Default

```
No.  Time            Source               Destination          Protocol Length Info
8327 12:07:24.200424 192.168.131.115      91.108.9.36          STUN     138 Binding Request user: FeAN:w8cC
8328 12:07:24.249569 2409:40f4:11:bcf5:1… 2409:40f4:3b:b290:8… STUN     158 Binding Request user: FeAN:w8cC
8329 12:07:24.262849 91.108.13.25         192.168.131.115      STUN     126 Binding Success Response XOR-MAPPED-ADDRESS: 152.58.205.65:59991 MAPPED-ADDRESS: 152.58.205.65:59991 RESP…
8330 12:07:24.415215 2409:40f4:11:bcf5:1… 2001:b28:f23f:f105:… UDP      102 60001 → 597 Len=40
8331 12:07:24.415511 2409:40f4:11:bcf5:1… 2001:b28:f23d:f101:… UDP      102 60002 → 596 Len=40
8332 12:07:24.415686 2409:40f4:11:bcf5:d… 2001:b28:f23f:f105:… UDP      102 60007 → 597 Len=40
8333 12:07:24.415848 2409:40f4:11:bcf5:d… 2001:b28:f23d:f101:… UDP      102 60008 → 596 Len=40
8334 12:07:24.540269 2409:40f4:3b:b290:8… 2409:40f4:11:bcf5:1… ICMPv6   206 Destination Unreachable (Port unreachable)
8335 12:07:24.540475 192.168.131.115      152.58.204.224       STUN     138 Binding Request user: FeAN:w8cC
8336 12:07:24.590498 2409:40f4:11:bcf5:d… 2409:40f4:3b:b290:8… STUN     158 Binding Request user: FeAN:w8cC
8337 12:07:24.683447 192.168.131.115      62.128.100.45        TLSv1.2  141 Ignored Unknown Record
8338 12:07:24.683656 192.168.131.115      62.128.100.45        TLSv1.2  249 Application Data
8339 12:07:24.684157 192.168.131.115      91.108.17.23         STUN      62 Binding Request
8340 12:07:24.736343 2409:40f4:11:bcf5:1… 2409:40f4:3b:b290:8… STUN     158 Binding Request user: FeAN:w8cC
8341 12:07:24.795617 2409:40f4:3b:b290:8… 2409:40f4:11:bcf5:d… ICMPv6   206 Destination Unreachable (Port unreachable)
8342 12:07:24.796220 152.58.204.224       192.168.131.115      ICMP     166 Destination unreachable (Port unreachable)
8343 12:07:24.796454 152.58.204.224       192.168.131.115      ICMP     166 Destination unreachable (Port unreachable)
8344 12:07:24.834006 2409:40f4:11:bcf5:d… 2409:40f4:3b:b290:8… STUN     158 Binding Request user: FeAN:w8cC
8345 12:07:24.882734 192.168.131.115      91.108.13.54         UDP      162 60020 → 596 Len=120

› Frame 8335: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface \Device\NPF_{98CC7288-C060-4834-B1D1-368D7E1D2A…
› Ethernet II, Src: AzureWaveTec_3d:d3:25 (d8:c0:a6:3d:d3:25), Dst: d6:55:d6:5a:e2:c9 (d6:55:d6:5a:e2:c9)
› Internet Protocol Version 4, Src: 192.168.131.115, Dst: 152.58.204.224
› User Datagram Protocol, Src Port: 59991, Dst Port: 25217
› Session Traversal Utilities for NAT

0000  d6 55 d6 5a e2 c9 d8 c0  a6 3d d3 25 08 00 45 00
0010  00 7c c7 d2 00 00 80 11  c9 67 c0 a8 83 73 98 3a
0020  cc e0 ea 57 62 81 00 68  18 06 00 01 00 4c 21 12
0030  a4 42 56 63 58 2b 71 46  6d 6d 6e 34 43 39 00 06
0040  00 09 46 65 41 4e 3a 77  38 63 43 00 00 00 c0 57
0050  00 04 00 03 00 0a 80 29  00 08 00 00 00 00 00 00
0060  00 00 00 24 00 04 6e 7b  1f 00 00 08 00 14 17 b8
0070  0e 83 7a 11 a4 7b 4e ea  57 5a 92 e5 7c d6 29 7f
0080  9a 36 80 28 00 04 8d 7d  14 96

Packets: 8360 · Displayed: 8360 (100.0%) · Dropped: 0 (0.0%)
```

Fig 3.3 In the wireshark application give the appropriate network interface to start the capturing of packets.

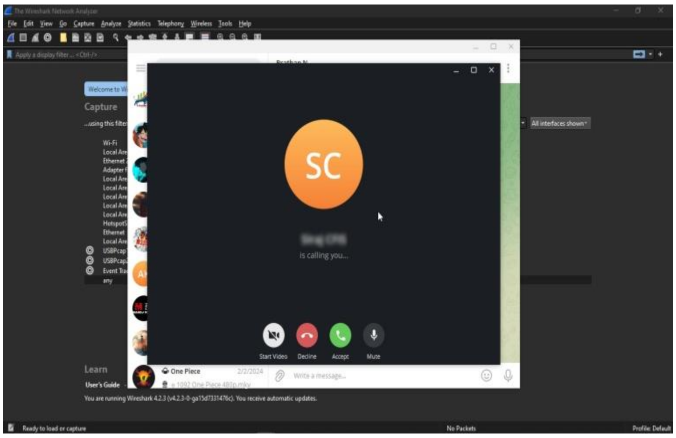
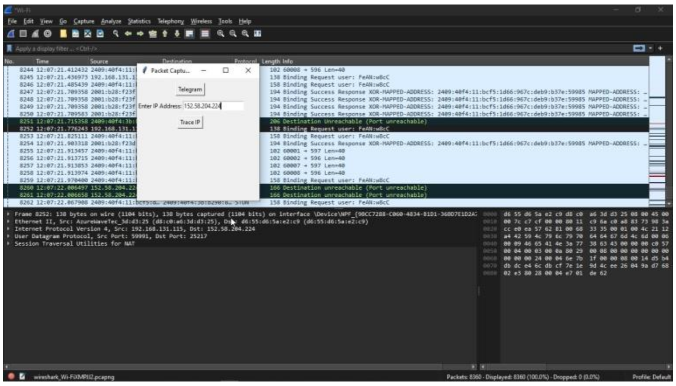


Fig 3.4 Incoming call in telegram application

Upon execution of the code, a graphical user interface (GUI) was presented, as illustrated in Fig 2. The GUI displayed several components, including buttons and input fields. Fig 3 outlines the steps involved in the execution process: (Fig 3.1) Upon clicking the Telegram button on the GUI, the Telegram application was launched. (Fig 3.2) Simultaneously, the Wireshark application also opened. (Fig 3.3) Within the Wireshark application, the user was prompted to select the appropriate network interface to initiate packet capturing. (Fig 3.4)Incoming calls received in the Telegram application.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Packet Captu...
Telegram
Trace IP

Enter IP Address: 152.58.204.224

```
No.   Time               Source                                   Length Info
 8244 12:07:21.412432 2409:40f4:11:       102 60008 → 596 Len=40
 8245 12:07:21.436973 192.168.131.1       138 Binding Request user: FeAN:w8cC
 8246 12:07:21.485439 2409:40f4:11:       158 Binding Request user: FeAN:w8cC
 8247 12:07:21.709358 2001:b28:f23f       194 Binding Success Response XOR-MAPPED-ADDRESS: 2409:40f4:11:bcf5:1d66:967c:deb9:b37e:59985 MAPPED-ADDRESS: …
 8248 12:07:21.709358 2001:b28:f23f       194 Binding Success Response XOR-MAPPED-ADDRESS: 2409:40f4:11:bcf5:1d66:967c:deb9:b37e:59985 MAPPED-ADDRESS: …
 8249 12:07:21.709358 2001:b28:f23f       194 Binding Success Response XOR-MAPPED-ADDRESS: 2409:40f4:11:bcf5:1d66:967c:deb9:b37e:59985 MAPPED-ADDRESS: …
 8250 12:07:21.709583 2001:b28:f23f       194 Binding Success Response XOR-MAPPED-ADDRESS: 2409:40f4:11:bcf5:1d66:967c:deb9:b37e:59985 MAPPED-ADDRESS: …
 8251 12:07:21.715358 2409:40f4:3b:       206 Destination Unreachable (Port unreachable)
 8252 12:07:21.776243 192.168.131.1       138 Binding Request user: FeAN:w8cC
 8253 12:07:21.825111 2409:40f4:11:       158 Binding Request user: FeAN:w8cC
 8254 12:07:21.903318 2001:b28:f23d       194 Binding Success Response XOR-MAPPED-ADDRESS: 2409:40f4:11:bcf5:1d66:967c:deb9:b37e:59985 MAPPED-ADDRESS: …
 8255 12:07:21.913457 2409:40f4:11:       102 60001 → 597 Len=40
 8256 12:07:21.913715 2409:40f4:11:       102 60002 → 596 Len=40
 8257 12:07:21.913853 2409:40f4:11:       102 60007 → 597 Len=40
 8258 12:07:21.913974 2409:40f4:11:       102 60008 → 596 Len=40
 8259 12:07:21.970400 2409:40f4:11:       158 Binding Request user: FeAN:w8cC
 8260 12:07:22.006497 152.58.204.22       166 Destination unreachable (Port unreachable)
 8261 12:07:22.006658 152.58.204.22       166 Destination unreachable (Port unreachable)
 8262 12:07:22.067908 2409:40f4:11:bcf5:d… 2409:40f4:3b:b290:8… STUN 158 Binding Request user: FeAN:w8cC

▸ Frame 8252: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface \Device\NPF_{98CC7288-C060-4834-B1D1-368D7E1D2A7
▸ Ethernet II, Src: AzureWaveTec_3d:d3:25 (d8:c0:a6:3d:d3:25), Dst: d6:55:d6:5a:e2:c9 (d6:55:d6:5a:e2:c9)
▸ Internet Protocol Version 4, Src: 192.168.131.115, Dst: 152.58.204.224
▸ User Datagram Protocol, Src Port: 59991, Dst Port: 25217
▸ Session Traversal Utilities for NAT

0000  d6 55 d6 5a e2 c9 d8 c0  a6 3d d3 25 08 00 45 00
0010  00 7c c7 cf 00 00 80 11  c9 6a c0 a8 83 73 98 3a
0020  cc e0 ea 57 62 81 00 68  33 35 00 01 00 4c 21 12
0030  a4 42 59 4c 79 6c 79 70  64 64 67 6d 4c 6d 00 06
0040  00 09 46 65 41 4e 3a 77  38 63 43 00 00 00 c0 57
0050  00 04 00 03 00 0a 80 29  00 08 00 00 00 00 00 00
0060  00 00 00 24 00 04 6e 7b  1f 00 00 08 00 14 d5 b4
0070  db dc e4 6c db cf 7e 1e  9d 4c ee 26 04 9a d7 68
0080  02 e3 80 28 00 04 e7 01  de 62

wireshark_Wi-FiXMPIJ2.pcapng      Packets: 8360 · Displayed: 8360 (100.0%) · Dropped: 0 (0.0%)      Profile: Default
```

Fig 4. Enter the IP address of the packet captured which has info "Binding Request User"

Fig 4 demonstrates the interface where the user was prompted to enter the IP address of the captured packet called "Binding Request User." This step was crucial in pinpointing the source of the call. After providing the IP address and clicking the "Trace IP" button, as depicted in Fig 5, the location associated with the IP address was mapped. This mapping allowed for the geographical tracking of the caller.

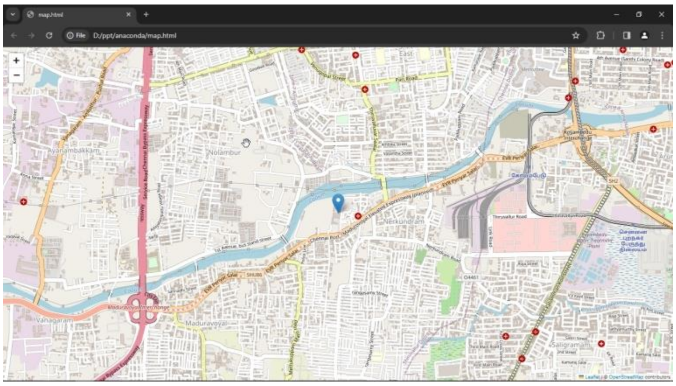

map.html
File D:/ppt/anaconda/map.html
Ayanambakkam
Nolambur
Leaflet | © OpenStreetMap contributors

Fig 5. Once the IP address is provided and "Trace IP" button is clicked, the location is mapped

The successful implementation of the proposed method offers significant implications for tracking the source of Telegram calls. By leveraging network packet analysis and geo-location techniques, the system effectively identifies the origin of incoming calls, thereby enhancing user security and privacy. However, it is essential to acknowledge potential limitations, such as accuracy issues in geo-location mapping due to factors like VPN usage or dynamic IP assignments.

6. Conclusion

In conclusion, this project has demonstrated a method to identify the location of threatening calls made through the Telegram platform by analyzing captured packets using Wireshark and correlating them with geolocation data. While the approach shows promise in identifying the origin of such calls, it comes with limitations that need to be addressed. Firstly, the manual search process for captured packets is time-consuming and requires improvements in automation and efficiency. Additionally, the accuracy of location identification using geolocation databases is not always precise, introducing potential inaccuracies in pinpointing the caller's location.

In summary, future enhancements to this project should focus on automating packet analysis, improving geolocation accuracy, implementing real-time monitoring capabilities, and integrating user feedback mechanisms. By addressing these areas, the system can evolve into a more robust and reliable tool for locating threatening calls in Telegram, ultimately enhancing user safety and security.

7. Future Enhancements

In the future, the Telegram call tracking system will evolve to automate the identification of "binding request user" packets, leveraging sophisticated algorithms for efficient threat detection. This automation will streamline the process, reducing manual effort and enhancing the system's responsiveness to potential security threats. Additionally, improvements in geolocation accuracy will be achieved by integrating with multiple databases and implementing real-time updates, ensuring precise location data retrieval. Integration with other security platforms will be enhanced to strengthen threat detection capabilities, enabling seamless collaboration and information sharing across various security tools and systems. Furthermore, usability enhancements will be prioritized to provide security personnel with intuitive interfaces and streamlined workflows, facilitating more efficient threat mitigation and response actions. Ongoing research into emerging technologies will remain a cornerstone of the system's development, ensuring its readiness to address evolving cyber threats and maintain robust cybersecurity measures within the Telegram messaging platform.

Availability Of Data And Materials

The corresponding author can provide all pertinent data and materials upon a reasonable request related to this study.

Conflict Of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Consent For Publication

The authors provide the consent to publish this research paper in this journal

Ethical Statement

This study did not involve human or animal subjects. Therefore, ethical approval was not required.

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or nonprofit sectors.

List Of Abbreviation

DPI – Deep Packet Inspection

GUI – Graphic User Interface
IDS – Intrusion Detection System
IP – Internet Protocol
NIC – Network Interface Card
IPS – Intrusion Prevention System
SDN – Software – Defined Network
VPN – Virtual Private Network

References

1. S. Arvind, V. K. Silveri, G. Poety, P. Nunavanth, and R. Podishetty, "Network Traffic Virtualization Using Wireshark and Google Maps," in *Proceedings of the International Conference on Distributed Computing and Electrical Circuits and Electronics*, 2023.

2. C.-E. Bogos, R. Mocanu, and E. Simion, "A Security Analysis Comparison Between Signal, WhatsApp, and Telegram," in *Cryptology ePrint Archive*, 2023.

3. Syed Hussain, Dr.PakkirMohideen S, "Advanced Machine Learning Approach for Detection of Multilinguistic Terror Message to save human Lives", Journal of Pharmaceutical Negative Results, Volume 14, Issue 2 , 2023.

4. Julia M. Janssen, Alana McGrath, Rochelle Ereman, Patrick K. Moonan, John E. Oeltmann, Matthew Willis, Stephen A. McCurdy "Use of SMS-linked electronic surveys for COVID-19 case investigation and contact tracing", Elsevier Ltd on behalf of The Royal Society for Public Health, 2021.
5. Mateusz Fedoryszak, Vijay Rajaram, Brent Frederick, ChangtaoZhong "Real-time Event Detection on Social Data Streams", ACM, New York, NY, USA, 9 pages, 2019.
6. Threema, paulroesler, Christian mainka, joergschwenk, "More is Less: On the End-to-End Security of Group Chats in Signal, WhatsApp", 3rd IEEE European Symposium on Security and Privacy (EuroS&P 2018).
7. Nicollas R. de Oliveira, Pedro S. Pisa, Martin Andreoni Lopez, Dianne scherly V. de Medeiros, Diogo M. F. Mattos "Identifying Fake News on Social Networks Based on Natural Language Processing: Trends and Challenges", MDPI Information 2021.
8. VivekTalkhe, ShradheshPandit, Prof.SwapnilSonawane, "Mobile Tracking System Using Short Messaging Service ", IRJET, Volume: 03 Issue: 03 Mar-2016.
9. Oktaf B Kharisma , Mustakim, RianVebrianto, Rice Novita, Hasbullah, Irawati, Yulia Novita, Zaitun, AlwisNazir, Iwan Iskandar, YelfiVitriani, Rina Rehayati and TutiAndriani"Development of location tracking system via short message service (SMS) based on GPS unblox neo-6m and sim 800l module", Journal of Physics: Conference Series, 2019.
10. Sérgio Barbosa and Stefania Milan, "Do Not Harm in Private Chat Apps: Ethical Issues for Research on and with WhatsApp", Westminster Papers in Communication and Culture,2019.
11. Benjamin Fabian, Benedict Bender, and Lars Weimann, "E-Mail Tracking in Online Marketing: Methods, Detection, and Usage", 12th International Conference on Wirtschaftsinformatik,2015.
12. Yu-Min Jeon, Won-Mu Heo, Jong-Min Kim, Kyounggon Kim, "Multimedia Distribution Process Tracking for Android and iOS", arXiv,2023.
13. UpendraDadi, 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.
14. Dong Fang, Cheng Chengqi, GuoShide, “Design and research on GeoIP”, IEEE Xplore, DOI 10.1109/CSCWD.2010.5472009, 24 May, 2010.
15. 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.
16. 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.
17. 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.
18. Ahmad Musa, AliyuAbubakar, 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.
19. 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.
20. G. Sasi, P. Thanapal, V.S. Balaji, G. VenkatBabu, V. Elamaran, “A Handy Approach for Teaching and Learning Computer Networks using Wireshark”, IEEE Xplore, DOI 10.1109/ICISC47916.2020.9171197, 19 August, 2020.
21. 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.
22. George Koutitas, Shashwat Vyas, Chaitanya Vyas, Shivesh Singh Jadon and IordanisKoutsopoulos, “Practical Methods for Efficient Resource Utilization in Augmented Reality Services”, IEEE Access, DOI 10.1109/ACCESS.2020.3042616, December 18, 2020.

23. Waqas Ahmed, Faisal Shahzad, Abdul RehmanJaved, Farkhund Iqbal, Liaqat Ali, "WhatsApp Network Forensics: Discovering the IP Addresses of Suspects", IEEE Xplore, DOI 10.1109/NTMS49979.2021.9432677, 18 May, 2021.

24. ApriSiswanto, 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.

25. 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.

26. MerveOzkan-Okay, Ömer Aslan, RecepEryigit, and RefikSamet, "SABADT: Hybrid Intrusion Detection Approach for Cyber Attacks Identification in WLAN", IEEE Access, DOI 10.1109/ACCESS.2021.3129600, December 3, 2021.

27. BinduDodiya, 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.