

**CENTER FOR DEVELOPMENT OF ADVANCED
COMPUTING (C-DAC),
THIRUVANANTHAPURAM, KERALA**

**A PROJECT REPORT ON
“To Visualize captured packet as Network Diagram”
Submitted to**



PG-DCSF SEP 2023

BY

Group no. 8

Anshuman Singh

PRN: 230960940006

Palak Kumari

PRN: 230960940033

Vedanti Sonsurkar

PRN: 230960940054

Vinaya Kshirsagar

PRN: 230960940056

Vinayak Kalshetti

PRN: 230960940057

**Under The Guidance Of
Mr. Jayaram P.
Centre Co- Ordinators and Project Guide**

TABLE OF CONTENTS

Contents	Page No.
Abstract.....	3
Introduction.....	4
Scope and Objectives.....	5
Methodology.....	6
Implementation.....	9
Conclusion.....	14

ABSTRACT

The project aims to develop a Network Forensics Tool with the objective of visualizing Packet Captures offline in the form of a comprehensive Network Diagram. This tool focuses on enhancing the analysis of network data by incorporating features such as device identification, highlighting critical communication pathways, and extracting files from the captured packets. By leveraging network forensics tools, the tool intends to process Pcap Files and generate visual representations of the network structure, showcasing hosts, network traffic patterns, and emphasizing significant communication flows. Additionally, the tool will identify and highlight Tor traffic, as well as potential malicious activities, providing a deeper understanding of the data involved in the network communication. This project addresses the need for efficient offline analysis of packet captures, aiding network security professionals in identifying and responding to potential threats more effectively.

Furthermore, the Network Forensics Tool will offer a user-friendly interface, allowing analysts to interact with the visualized network diagram intuitively. The tool aims to provide an enhanced analytical experience by incorporating features that facilitate the identification of anomalies and potential security breaches. By emphasizing crucial communication paths, the tool assists in quickly pinpointing relevant information within the vast dataset of captured packets. Moreover, the capability to extract and analyze files exchanged during network communication enhances the forensic investigation process, aiding in the detection of malicious activities or data breaches. The comprehensive visualization and analysis capabilities of this tool contribute to the efficiency and effectiveness of network forensic investigations, supporting cybersecurity professionals in making informed decisions to ensure the integrity and security of network environments.

INTRODUCTION

In the contemporary landscape of cybersecurity, the "Network Forensics Tool" project emerges as a crucial initiative aimed at fortifying digital defense mechanisms. As cyber threats grow in complexity, the need for advanced tools that enable comprehensive offline analysis of network activities becomes increasingly evident. This project seeks to address this imperative by developing a sophisticated tool capable of visualizing Packet Captures offline, providing network professionals with an insightful Network Diagram that highlights communication nuances, device interactions, and potential security risks.

With the objective of offering a robust solution to network forensic challenges, the tool will leverage innovative techniques to not only plot hosts and network traffic but also identify devices, emphasize critical communication pathways, and extract files exchanged during network transactions. By combining these features, the Network Forensics Tool aims to empower cybersecurity experts with an efficient means of identifying and responding to potential threats, ultimately contributing to the resilience and security of digital networks.

Moreover, as an integral part of the project's approach, the tool will harness the capabilities of PcapXray, a prominent network forensics tool. By incorporating the strengths of PcapXray, known for its robust packet analysis and visualization, the Network Forensics Tool aims to build upon this foundation to offer an even more comprehensive solution. PcapXray's proficiency in decoding packet data and unveiling intricate network patterns will synergize with the advanced features of the new tool, enhancing the overall efficiency of offline packet capture analysis. This collaborative effort seeks to empower cybersecurity professionals with a potent amalgamation of cutting-edge technologies, ensuring a meticulous examination of network activities and facilitating proactive responses to potential cybersecurity threats.

SCOPE AND OBJECTIVES

The scope of the Network Forensics Tool encompasses the offline visualization and analysis of Packet Captures to enhance network security measures. The tool will leverage the capabilities of PcapXray, a proven packet analysis tool, to provide a comprehensive Network Diagram highlighting the interactions between devices, communication pathways, and network traffic patterns. The scope extends to the identification of potential security threats, including the detection of malicious and Tor traffic, ultimately empowering cybersecurity professionals with a robust solution for in-depth network forensic analysis. The tool's application is intended for a broad range of scenarios, aiding professionals in identifying, analyzing, and responding to security incidents effectively.

The primary objective of the project is to develop a Network Forensics Tool that seamlessly integrates with PcapXray to visualize Packet Captures offline. The tool aims to enhance the understanding of network dynamics by generating detailed Network Diagrams, complete with device identification, traffic analysis, and highlighting crucial communication flows. Additionally, the tool seeks to bolster cybersecurity efforts by incorporating features for the identification of potential malicious activities and Tor traffic, leveraging the sophisticated capabilities of PcapXray in the process. The overarching goal is to provide cybersecurity professionals with a user-friendly interface, coupled with advanced analytical capabilities, to fortify network defenses and proactively address emerging threats.

METHODOLOGY

PcapXray provides insights into network communication, device identification, and potential malicious traffic. Here's an overview of its methodology:

1) Purpose and Goal:

Purpose: PcapXray aims to simplify the investigation of packet capture (PCAP) files.

Goal: Given a PCAP file, it creates a network diagram that displays:

- Hosts in the network.
- Network traffic.
- Highlighted important traffic.
- Tor traffic.
- Potential malicious traffic.
- Data involved in communication.

2) Packet Capture and Visualization:

Packet Capture:

- PcapXray reads PCAP files using the Scapy library.
- It captures data packets from the file, including headers and payloads.

Network Diagram:

PcapXray plots a network diagram with the following components:

- Hosts: Represented nodes.
- Traffic Links: Connections between hosts.
- Highlighted Traffic: Important communication paths.
- Tor Traffic: Identifies Tor network usage.
- Malicious Traffic: Flags potential threats.

The diagram provides an overview of the network structure and communication patterns.

3) Usage and Setup:

Installation:

- Install Python 3 and required libraries (Tkinter, graphviz, etc.).
- Clone the PcapXray repository.
- Run the tool with elevated privileges (sudo).

Traffic Options:

PcapXray supports various traffic types:

- Web (HTTP and HTTPS).
- Tor.
- Malicious.
- ICMP.
- DNS.

4) Challenges and Considerations:

GUI Stability:

- PcapXray uses Tkinter for the GUI, which can be unstable.
- Consider using Django or other alternatives for a more robust interface.

Graph Plotting:

- Creating a readable network graph from captured data requires effort.
- PcapXray uses different libraries (such as pyGraphviz and Networkx) to achieve this.

PcapXray streamlines the investigation process, making it a valuable tool for network forensics and security professionals.

5) Components:

1. Network Diagram Generation:

The creation of a comprehensive Network Diagram constitutes a fundamental aspect of the methodology. This involves the utilization of packet analysis tools, including PcapXray, to dissect and interpret the captured data. The tool will employ algorithms to map out the relationships between various network entities, such as hosts, routers, and switches. The resulting Network Diagram will visually represent the structure of the network, providing insights into the communication dynamics and the flow of data. Special attention will be given to enhancing the clarity and intuitiveness of the diagram for effective analysis.

2. Device/Traffic Details and Analysis:

To enrich the forensic analysis, the methodology will delve into the identification and detailing of devices and network traffic. Device identification involves parsing packet data to distinguish and label each device in the network. Concurrently, a thorough analysis of network traffic will be conducted to discern patterns, anomalies, and communication pathways. By extracting key details related to devices and traffic, the methodology aims to provide a nuanced understanding of the network's operational dynamics, facilitating a more profound analysis of communication flows.

3. Malicious Traffic Identification:

A critical component of the methodology involves the implementation of algorithms and heuristics to identify potential malicious traffic within the captured packets. Leveraging known threat indicators and behavior analysis, the tool will flag and highlight suspicious patterns or activities indicative of security threats. This step is essential for assisting cybersecurity professionals in swiftly detecting and responding to potential security breaches, thereby enhancing the overall resilience of the network.

4. Tor Traffic Analysis:

The methodology will specifically focus on the identification and analysis of Tor (The Onion Router) traffic. Tor traffic often poses challenges due to its anonymizing nature. The tool will

employ techniques to recognize Tor communication within the packet captures, enabling the highlighting of these activities in the Network Diagram. This dedicated analysis aims to enhance the tool's capability to identify potentially covert or anonymized network activities.

5. Graphical User Interface (GUI) Development:

The creation of an intuitive GUI is integral to the user experience and accessibility of the tool. The methodology involves the development of a graphical interface with user-friendly options to upload Pcap files. This interface will provide a seamless experience for cybersecurity professionals, allowing them to initiate the analysis process effortlessly. The GUI will also serve as a platform to display the generated Network Diagram, providing interactive elements for users to navigate and explore the analyzed data efficiently. This user-centric design approach enhances the tool's usability and ensures that it serves as a practical asset in real-world network forensic scenarios.

IMPLEMENTATION

The implementation steps for the setup and installation process for the PcapXray tool. Let's elaborate on each step:

1) Install Python3 and Pip:

The initial step involves installing Python3 and Pip, the package installer for Python. This is achieved through the "apt install" command, which fetches and installs the Python3 interpreter and Pip for managing Python packages. These are fundamental prerequisites for running Python-based applications and ensuring compatibility with the subsequent steps.

Cmd: apt install python3-pip

2) Install Python3-tk:

PcapXray utilizes the Tkinter library for creating graphical user interfaces (GUIs) in Python. The "apt install python3-tk" command installs the Tkinter package, ensuring that the tool can create a user-friendly interface for interacting with the packet capture data.

Cmd: apt install python3-tk

3) Install Graphviz:

Graphviz is an essential tool for creating graph visualizations, and it is used by PcapXray to generate network diagrams. The "apt install graphviz" command installs the Graphviz package, providing the necessary tools to render and visualize the network structure.

Cmd: apt install graphviz

4) Install Python3-PIL and Python3-PIL.imageTk:

The Python Imaging Library (PIL) is required for image processing tasks, and the "apt install python3-pil python3-pil.imageTk" command installs the PIL package along with the PIL.imageTk module. This ensures that PcapXray can handle image-related operations, which is crucial for generating graphical representations of network diagrams.

Cmd: apt install python3-pil python3-pil.imageTk

5) Install Dependencies from requirements.txt:

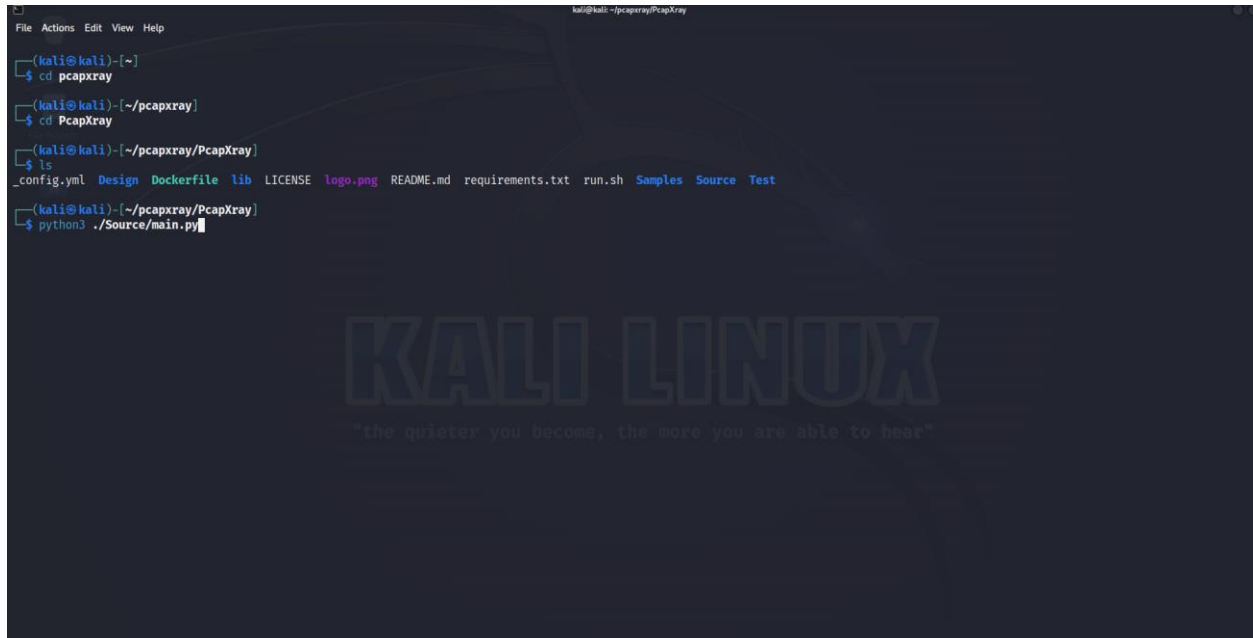
PcapXray relies on specific Python libraries and modules, which are listed in the "requirements.txt" file. The "pip3 install -r requirements.txt" command installs these dependencies, ensuring that the tool has access to the necessary libraries for its proper functioning.

Cmd: pip3 install -r requirements.txt

6) Run PcapXray:

Finally, the "python3 Source/main.py" command executes the main script of PcapXray, launching the tool and making it ready for use. This step initiates the application, enabling users to upload Packet Capture files, visualize network diagrams, and perform in-depth analysis of network traffic.

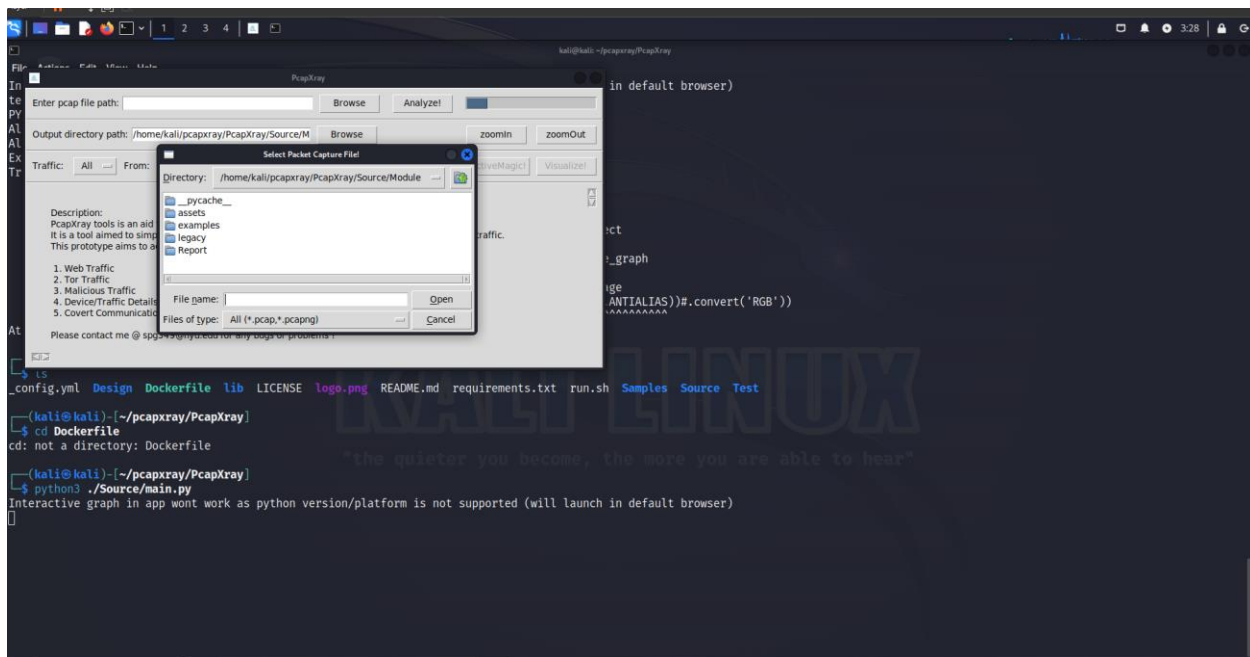
Cmd: python3 Source/main.py



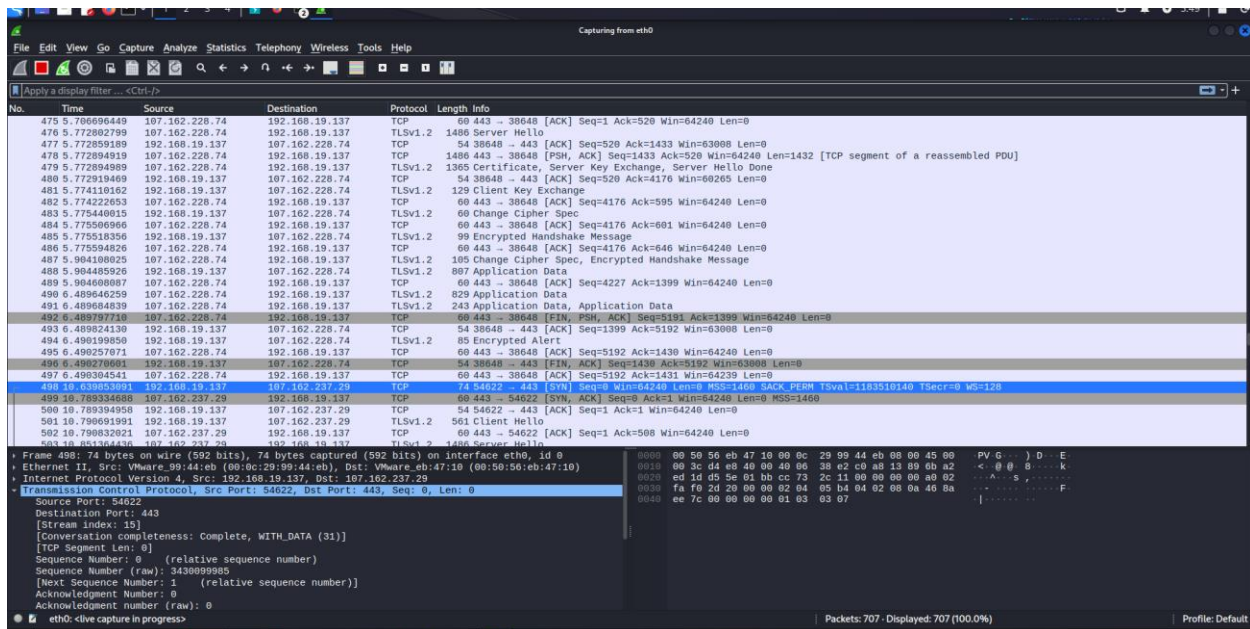
```
File Actions Edit View Help
kali@kali: ~/pcapxray/PcapXray
(kali@kali)-[~]
$ cd pcapxray
(kali@kali)-[~/pcapxray]
$ cd PcapXray
(kali@kali)-[~/pcapxray/PcapXray]
$ ls
_config.yml  Design  Dockerfile  lib  LICENSE  logo.png  README.md  requirements.txt  run.sh  Samples  Source  Test
(kali@kali)-[~/pcapxray/PcapXray]
$ python3 ./Source/main.py
```

The screenshot shows a terminal window with a dark background. At the top, there's a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. Below it, the terminal shows the user navigating through directories: from the home directory to 'pcapxray', then to 'PcapXray'. A 'ls' command lists the contents of the 'PcapXray' directory, including files like '_config.yml', 'Design', 'Dockerfile', 'lib', 'LICENSE', 'logo.png', 'README.md', 'requirements.txt', 'run.sh', 'Samples', 'Source', and 'Test'. Finally, the user runs 'python3 ./Source/main.py'. In the background, a large, faint 'KALI LINUX' logo is visible with the tagline 'the quieter you become, the more you are able to hear'.

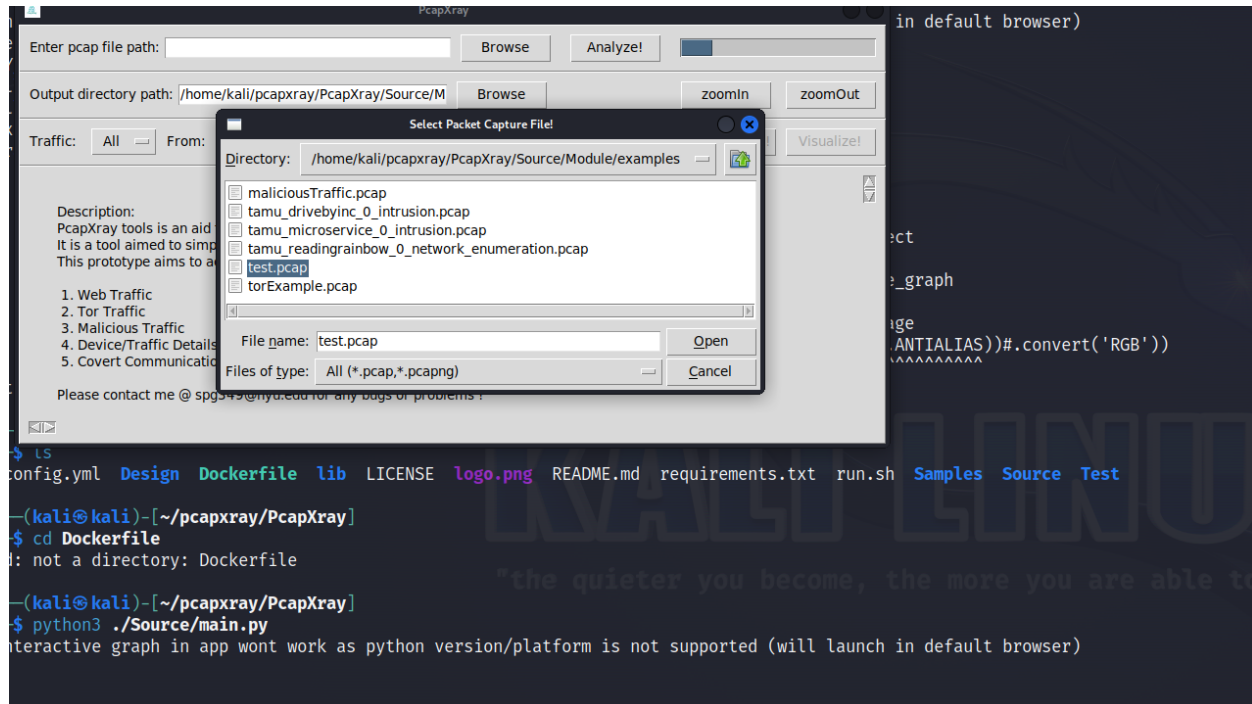
To run the PcapXray we have to use the python cmd and open the main.py file which will result in opening of the GUI of PcapXray tool.



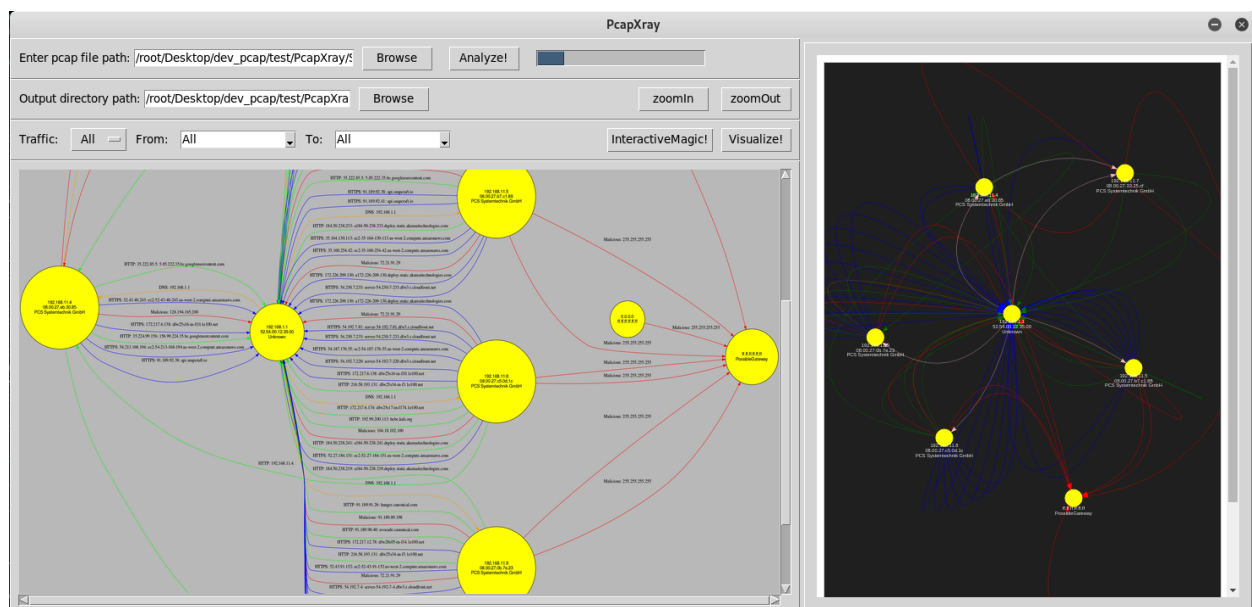
After running the command, we get the above GUI asking for a pcap file to analyze. For that purpose we used Wireshark to capture packets/Traffic of various things such as Tor, Web traffic, HTTP, ICMP, DNS packets.

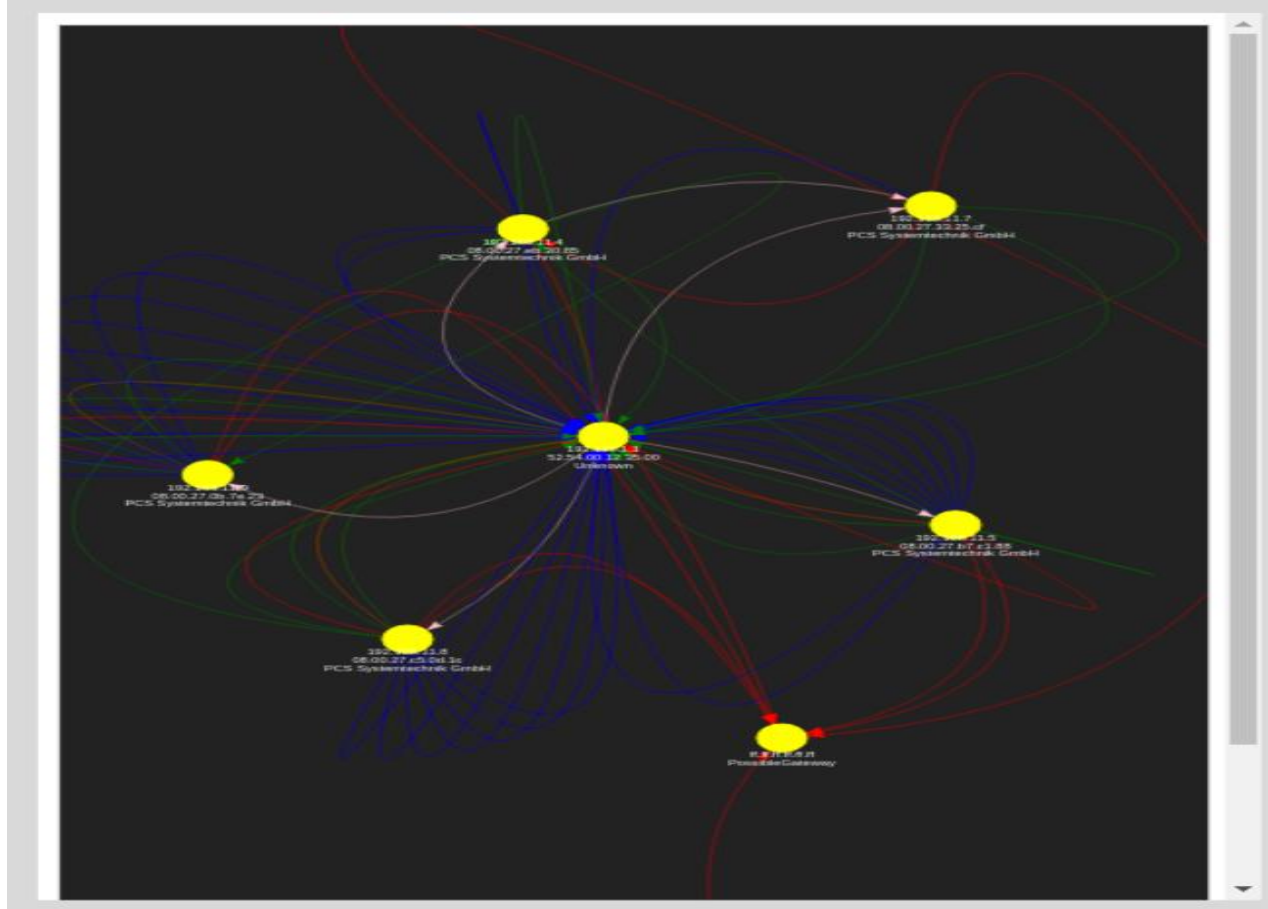
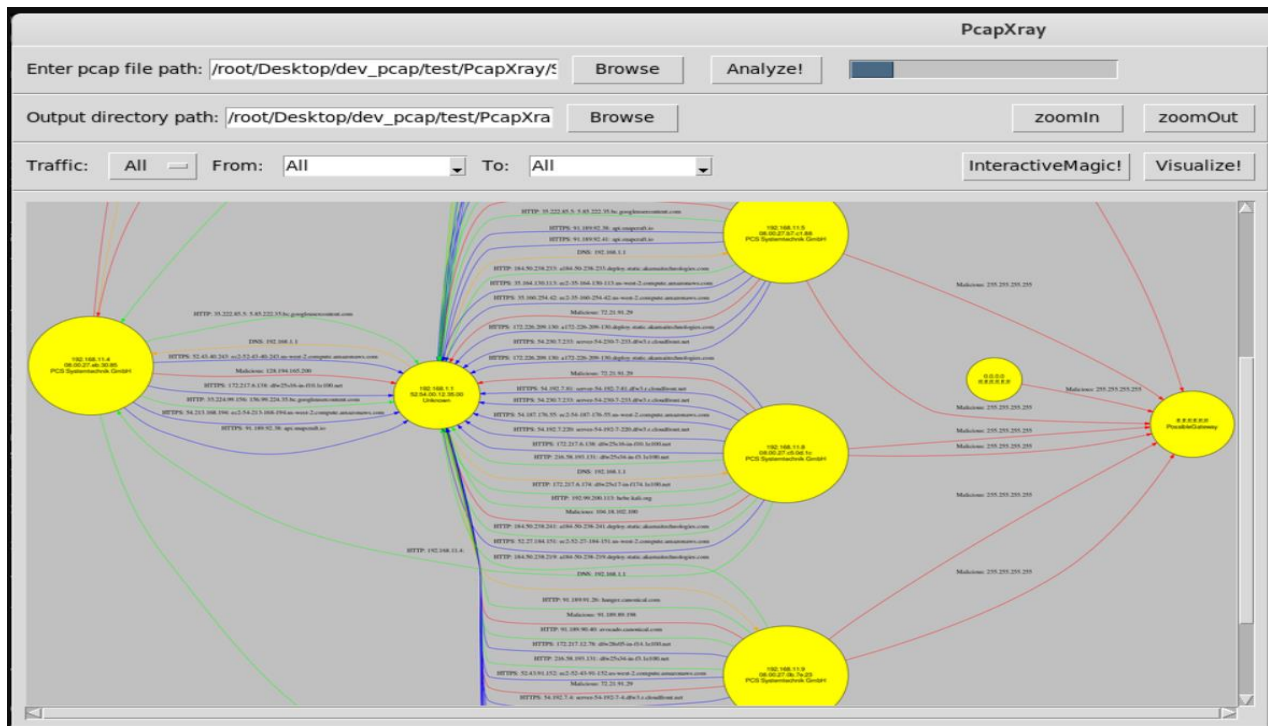


We saved the captured packets as test.pcap file which will be analyzed and used to plot the final network diagram depicting the Traffic for analysis.



In the above diagram we clicked on Browse for Entering the pcap file path, chose the test.pcap file and clicked on Analyze. After clicking on Analyze, the tool takes a bit and comes up with the option to visualize for plotting of the final Network Diagram. After clicking on visualize we get the final result as the below Network diagram. The red lines connecting Malicious packets, the green ones for HTTP and blue ones depicting the HTTPS requests.





CONCLUSION

The culmination of the Network Forensics Tool project marks a significant stride in bolstering the capabilities of cybersecurity professionals for efficient and thorough packet analysis. The seamless integration of Wireshark for packet capture, the subsequent creation of a Pcap file, and the utilization of PcapXray for in-depth analysis and visualization have yielded noteworthy results.

The most striking outcome of this endeavor is the creation of detailed and insightful Network Diagrams that offer a comprehensive visual representation of the network's architecture. The diagrams, generated through the PcapXray tool, serve as a powerful forensic tool, depicting the intricate web of communication pathways between various network entities. From the root node to every node contributing to communication, the tool presents a clear and intuitive depiction of the network's structure.

A notable feature of the visual representation is the incorporation of color-coded lines, which significantly enhance the interpretability of the diagrams. Malicious packets are highlighted with red lines, providing an immediate visual cue to potential security threats. Furthermore, green lines signify HTTP traffic, while blue lines denote HTTPS traffic, aiding analysts in swiftly identifying and differentiating between various types of network activities.

The culmination of these visualizations ultimately streamlines the forensic analysis process, saving valuable time for cybersecurity professionals. The ease with which the tool facilitates the identification of malicious activities, distinguishes between different types of traffic, and presents a holistic view of the network topology contributes to the efficiency of forensic investigations. In essence, the Network Forensics Tool, with its integration of Wireshark for packet capture and PcapXray for analysis and visualization, offers a comprehensive solution for network security professionals. It not only provides a time-saving mechanism but also enhances the ability to focus on critical prospects within the packet captures. As cybersecurity threats continue to evolve, the insights gained from this project contribute significantly to the arsenal of tools available for fortifying network defenses and responding proactively to emerging security challenges.