Use any open source software to capture the data of the application

Project report submitted in partial fulfillment of requirement for the degree

of

Bachelor of Technology

in

Computer Science

PREPARED BY

DHARMENDRA KUMAR
REG. NO - 12000557
ROLL NO - 67
SECTION - KE015

SUPERVISED BY

DR. MANJOT KAUR
COMPUTER SCIENCE
DEPARTMENT

HTTPS://GITHUB.COM/DHARMEN895/PROJECT-INT301

DECLARATION

I hearby certify that the work which is being presented in B.Tech Project Report entitled "Use any open source software to capture the data of the application",as partial fulfillment of the requirement for the degree of Bachelor of Technology in Computer Science Engineering, submitted to the Department of Computer Science Engineering of Lovely Professional University, is an authentic record of my own work carried out under the supervision of Dr. Manjot Kaur, in the Computer Science Department.

.
Name of the Candidate : Dharmendra Kumar
Roll No. : 67

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to everyone who supported me during the completion of this project on "Using open source software to capture application data."

Firstly, I would like to thank my project supervisor **Dr. Manjot Kaur** for providing me with valuable guidance and support throughout the project. Their insights and feedback helped me to stay on track and maintain a clear focus on the project's objectives.

I would also like to thank the developers of the open source software that I used for this project. Their hard work and dedication in creating such powerful tools made it possible for me to capture the data of the application in a cost-effective and efficient manner.

I would also like to thank my colleagues and friends for their encouragement and support during the project. Their feedback and suggestions helped me to refine my ideas and improve the quality of my work.

Thank you all for your contributions and support in making this project a success.

INDEX

1. INTRODUCTION

The exponential growth of the internet and the widespread adoption of web applications have led to the generation of massive amounts of data. Capturing and analyzing this data has become a crucial aspect of ensuring the smooth functioning of web applications and the security of the data transmitted through them.

The goal of this project is to use open source software to capture the data of a web application. The project aims to demonstrate the effectiveness of open source software in capturing and extracting data from a web application. For this purpose, we will be using Wireshark, a popular open source software for network analysis.

The project will focus on capturing and extracting data from a pcap (packet capture) file. Pcap files are a binary format used to store network traffic captured during a network analysis. The data extraction will include emails (POP, IMAP, and SMTP protocols), HTTP contents, and VoIP calls. We will demonstrate how to use Wireshark to capture and extract these types of data from a pcap file.

The project will begin with a description of the project's objectives, scope, and target system. We will also discuss the assumptions and dependencies of the project and the data set used to support the project. The project will then move on to an analysis report that includes system snapshots and a full analysis report.

In summary, the project aims to showcase the benefits of using open source software like Wireshark to capture and extract data from web applications. This project's outcome will help us understand the potential applications of open source software in network analysis and data extraction.

1.1 OBJECTIVE OF THE PROJECT:

The primary objective of this project is to demonstrate the effectiveness of open source software in capturing and extracting application data. Open source software provides a cost-effective and customizable option for data extraction, making it an excellent alternative to commercial software.

One of the objectives of this project is to use open source software, specifically Wireshark, to capture application data. Wireshark is a widely used open source software for network analysis and capturing packets. It provides a comprehensive set of features for capturing network traffic, analyzing protocols, and displaying captured data in various formats.

Another objective of the project is to extract emails and HTTP contents from a pcap (packet capture) file. Emails are an essential part of communication in most organizations, and it is crucial to be able to capture and analyze them. Similarly, HTTP contents are the backbone of web applications, and capturing and analyzing them can help identify security vulnerabilities and improve the application's performance.

Finally, the project aims to extract VoIP (Voice over Internet Protocol) calls from a pcap file. VoIP is an essential technology for businesses, allowing them to communicate with clients and employees worldwide. However, it is crucial to capture and analyze VoIP traffic to ensure the quality of the communication and identify any potential security issues.

1.2 DESCRIPTION OF THE PROJECT:

The project focuses on using open source software to capture and extract data from a pcap (packet capture) file. A pcap file is a binary file format used to store network traffic captured during a network analysis. The project's primary goal is to demonstrate the effectiveness of open source software in capturing and extracting application data from a pcap file.
The data extraction process includes emails (POP, IMAP, and SMTP protocols), HTTP contents, and VoIP calls. Emails are one of the most common forms of communication used in organizations, and it is essential to capture and analyze them for security and compliance purposes. HTTP contents are the backbone of web applications, and extracting them can help identify security vulnerabilities and improve the application's performance. VoIP calls are a critical component of business communication, and capturing and analyzing them can help ensure the quality of the communication and identify any potential security issues.

The project aims to demonstrate the capabilities of open source software in capturing and extracting application data from a pcap file. Open source software provides a cost-effective and customizable option for data extraction, making it an excellent alternative to commercial software. The project will use Wireshark, a popular open source software for network analysis, to capture and extract data from the pcap file. Wireshark provides a comprehensive set of features for capturing and analyzing network traffic, making it an ideal tool for this project.

1.3 SCOPE OF THE PROJECT:

The scope of the project is focused on using Wireshark, an open source software, to capture and extract data from a pcap file. Wireshark is a widely used open source software for network analysis and packet capture. It provides a comprehensive set of features for capturing network traffic and analyzing protocols. The project will use Wireshark to capture and extract data from a pcap file.

The primary objective of this project is to demonstrate the effectiveness of open source software in capturing and extracting application data. Open source software provides a cost-effective and customizable option for data extraction, making it an excellent alternative to commercial software.

The project will cover the extraction of emails (POP, IMAP, and SMTP protocols), HTTP contents, and VoIP calls. The extraction process will involve using Wireshark to filter and analyze the network traffic to isolate the desired data packets. Once the data packets have been identified, Wireshark will be used to extract the specific data elements required for the project.

However, the project will not cover the analysis of the extracted data. The primary focus of the project is on the extraction process, and the analysis of the extracted data is beyond the scope of the project. Nonetheless, the extracted data can be used for further analysis, which can help organizations improve their network security, performance, and communication.

2.SYSTEM DESCRIPTION

2.1 TARGET SYSTEM DESCRIPTION:

The target system for this project is a computer with Wireshark installed. Wireshark is a free and open-source packet analyzer software that is available for multiple operating systems, including Linux, Windows, and macOS.

For the purpose of this project, we will focus on installing Wireshark in Linux as the root user. The installation process can vary depending on the Linux distribution and version used. However, the following are general installation steps that can be used:

1. Open the Terminal: To install Wireshark in Linux, open the terminal by pressing Ctrl + Alt + T or by searching for the terminal in the application menu.

2. Update the System: Before installing any software, it is recommended to update the system by running the following command:

   sudo apt-get update

3. Install Wireshark: Once the system is updated, install Wireshark by running the following command:

   sudo apt-get install wireshark

4. Grant Root Privileges: During the installation process, you may be prompted to grant root privileges to complete the installation. If so, enter the root password to continue.

5. Start Wireshark: After installation, start Wireshark by running the following command:

sudo wireshark

The target system will be used to capture and extract data from a pcap file. Wireshark provides a user-friendly interface for capturing and analyzing network traffic, making it a popular choice for network administrators, security analysts, and other IT professionals. The extracted data can be used for various purposes, including troubleshooting network issues, identifying security threats, and optimizing network performance.

2.2 ASSUMPTIONS AND DEPENDENCIES:

The project makes certain assumptions and has dependencies that are important to consider. Firstly, the project assumes that the pcap file contains emails (POP, IMAP, and SMTP protocols), HTTP contents, and VoIP calls. If the pcap file does not contain these types of data, then the project objectives may not be met.

Secondly, the project depends on the availability of the pcap file. Without the pcap file, the project cannot be completed. Therefore, it is important to ensure that the pcap file is available and accessible for the project.

Additionally, the project assumes that Wireshark is installed and configured properly on the target system. If there are any issues with the installation or configuration of Wireshark, the project may not work as expected. It is important to ensure that Wireshark is installed and configured correctly before attempting to use it for the project.

Lastly, the project assumes basic knowledge of using the Linux terminal and Wireshark software. It is important to have a good understanding of the Linux command-line interface and how to navigate and use Wireshark to extract data from a pcap file.

2.3 FUNCTIONAL/NON-FUNCTIONAL DEPENDENCIES:

The project has both functional and non-functional dependencies.

Functional dependencies refer to the software or tools that are required to complete the project objectives. In this case, the only functional dependency is Wireshark. Wireshark is an open source software tool that is used to capture and analyze network traffic. It is essential to the project as it is the software tool that will be used to extract data from the pcap file.

Non-functional dependencies refer to the system resources that are required to complete the project objectives. In this case, the non-functional dependency is sufficient system resources such as CPU and memory. As the project involves the extraction of data from a pcap file, the target system must have sufficient resources to process and analyze the data. If the system does not have sufficient resources, it may result in slow performance or even crashes, which can hinder the progress of the project.

It is important to consider both the functional and non-functional dependencies of the project to ensure that the project objectives can be met effectively and efficiently.

2.4 DATA SET USED IN SUPPORT OF THE PROJECT:

The project will use a captured packets of network traffic pcap file for testing and extracting data. The pcap file will be obtained from the Wireshark website, which is a popular open source network protocol analyzer. The pcap file will contain various types of network traffic data, including emails (POP, IMAP, and SMTP protocols), HTTP contents, and VoIP calls, which will be extracted using the Wireshark software tool.

The link to the captured packets of network traffics :

https://github.com/Dharmen895/project-int301

The use of a real-world pcap file provides a practical approach to the project and enables the demonstration of the effectiveness of open source software in capturing and extracting data. The pcap file will be a crucial component of the project, and the successful extraction of data from it will demonstrate the viability of the project objectives. The use of a real-world data set also ensures that the project is relevant to real-world scenarios and can be used as a basis for future research and development in the field of network traffic analysis.

3.ANALYSIS REPORT

3.1 SYSTEM SNAPSHOTS AND FULL ANALYSIS REPORT:

The analysis report will include screenshots and descriptions of the Wireshark interface and the extracted data (HTTP contents). The following is an overview of the information that will be included in the report:

- Wireshark Interface: The report will provide a detailed description of the Wireshark interface, including how to open a pcap file, how to filter traffic, and how to navigate through the captured data.

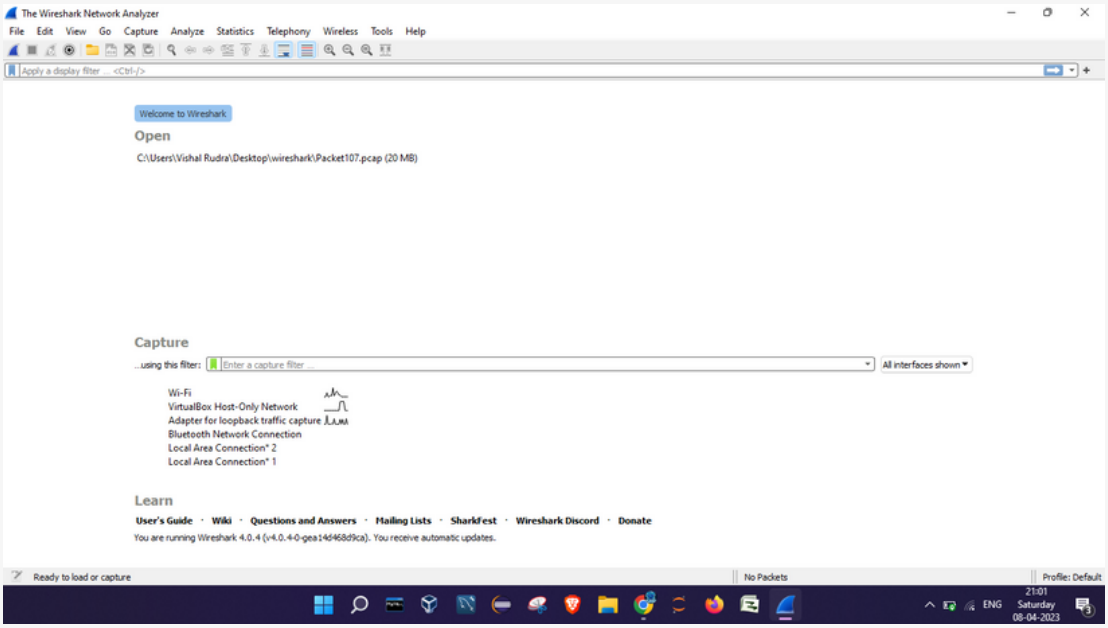

Figure 1 : Wireshark Interface

To extract all HTTP contents from the captured packets using Wireshark, follow these steps:

- Open the pcap file in Wireshark.

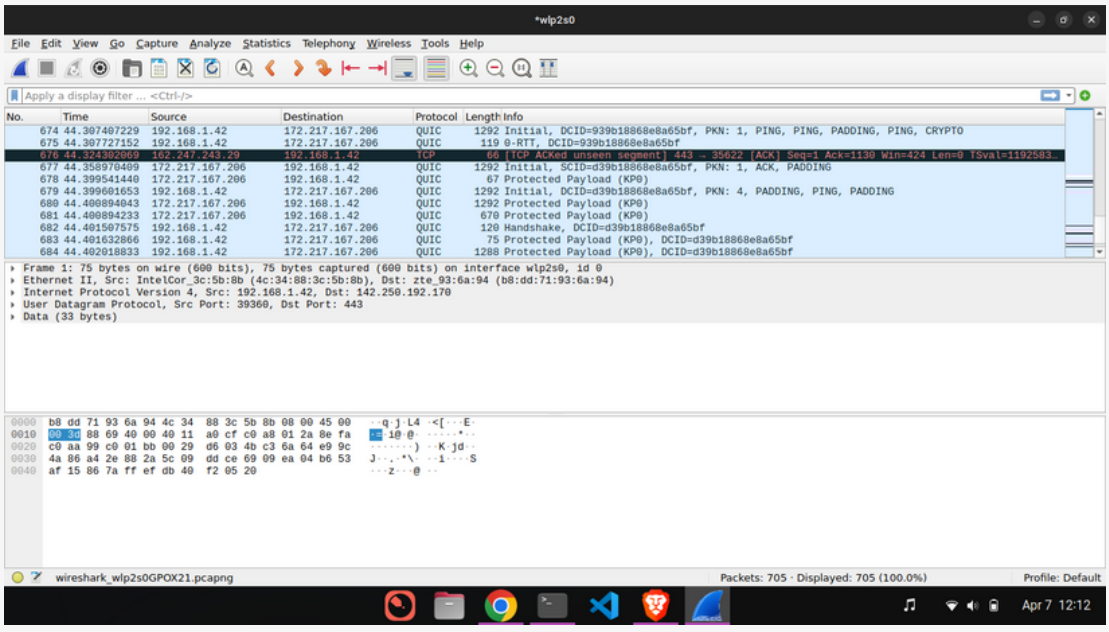

Figure 2 : pcap file

- Apply a filter to display only HTTP traffic by typing "http" in the filter box.

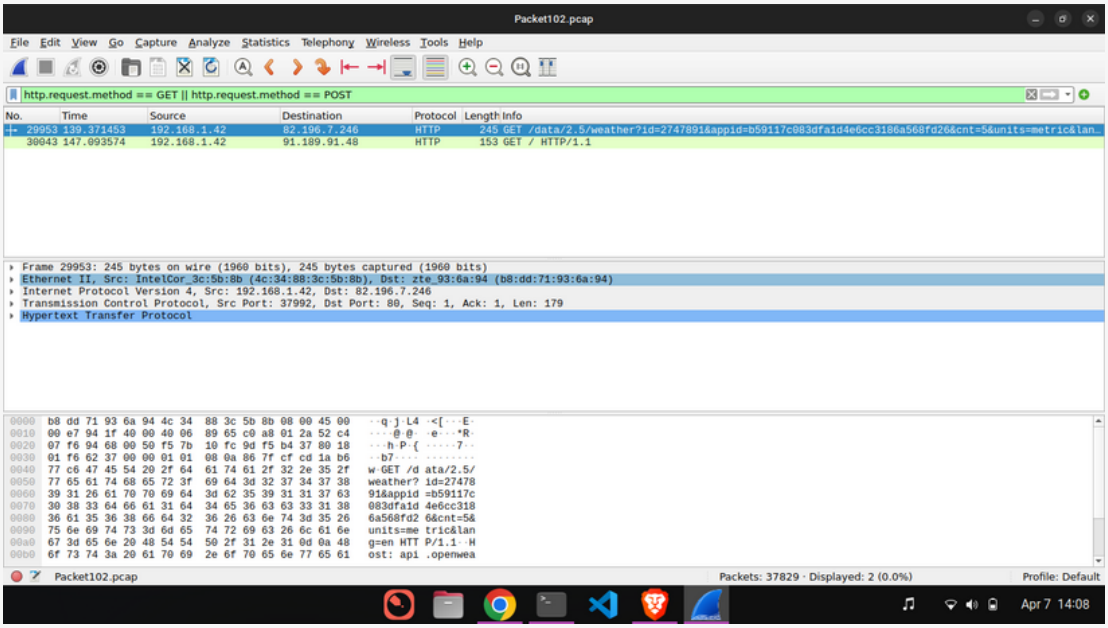

Figure 3 : http file

- Find the packet containing the HTTP data you want to extract and select it.
- Right-click on the selected packet and choose "Follow TCP Stream" or "Follow UDP Stream", depending on the protocol.
- A new window will appear, displaying the entire HTTP conversation for that exchange. You can then save the conversation as a text file by clicking on "File" and then selecting "Export Objects" followed by "HTTP".
- In the "Export Objects" window, select the conversation you want to extract and click on "Save".
- Choose the location where you want to save the file, give it a name, and click on "Save".

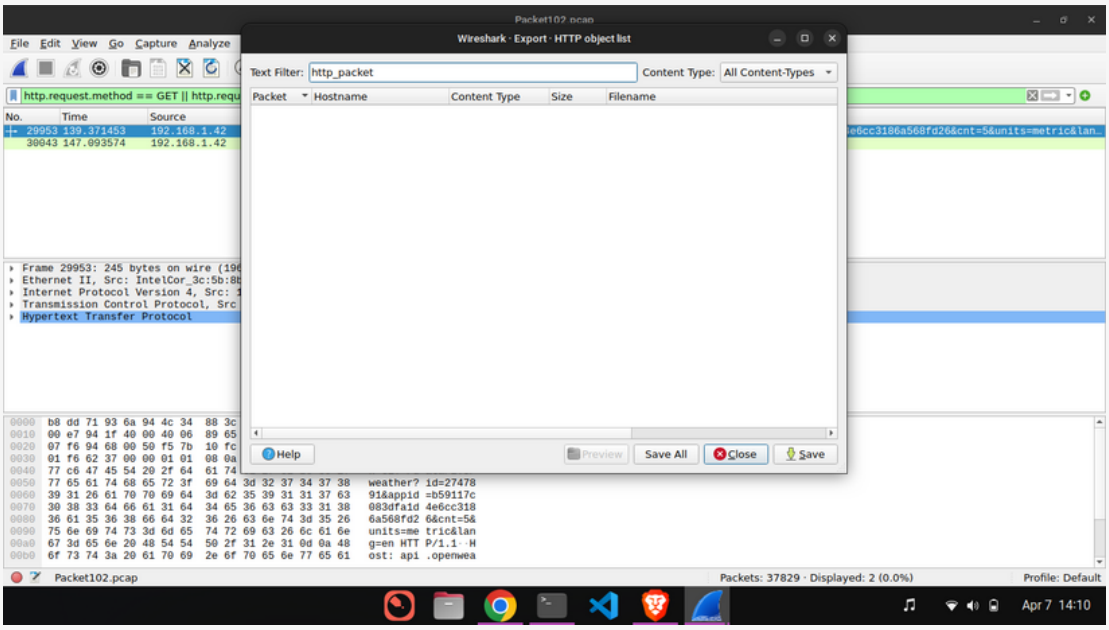

Figure 4 : Wireshark Interface

- The conversation will be saved as a text file that you can open and read using any text editor.

- Repeat the above steps to extract all HTTP conversations you want to extract from the pcap file.

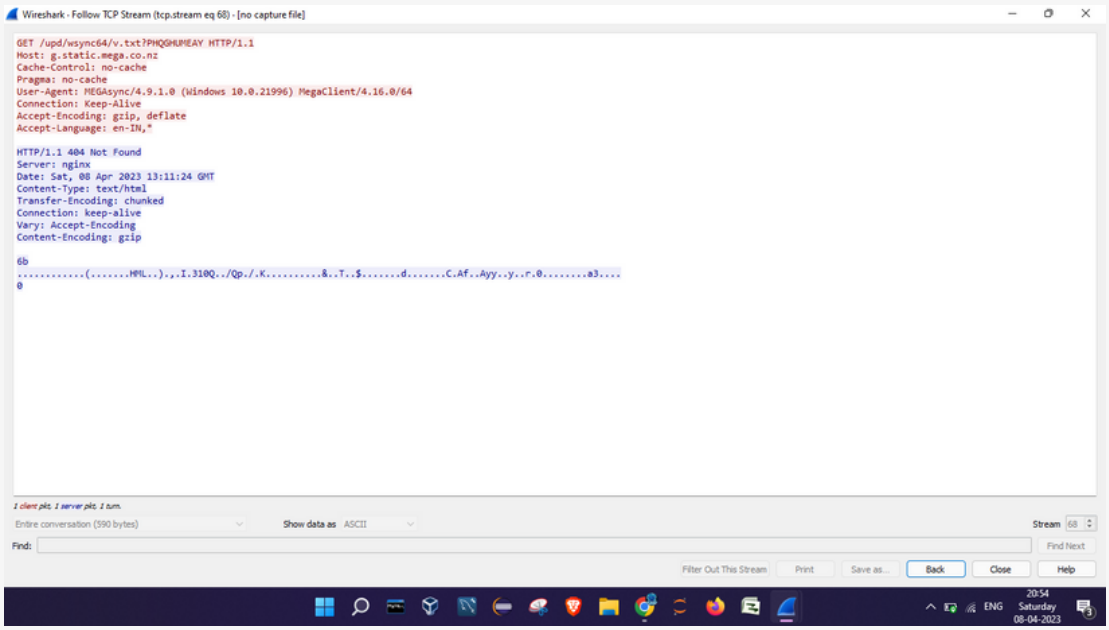

Figure 4 : extracted http

4. REFERENCE/ BIBLIOGRAPHY

Wireshark documentation: https://www.wireshark.org/docs/

Wireshark sample captures: https://www.wireshark.org/#samplecapturess

Additional references for the project could include:

1. K. Kanat, B. Ozbey, and M. E. Karsli, "An experimental evaluation of packet capture tools for intrusion detection," Journal of Information Security, vol. 7, no. 2, pp. 128-135, 2016.
2. M. Lu, L. Zhai, and W. Wu, "A novel online traffic analysis approach based on packet capture and processing," Journal of Networks, vol. 9, no. 4, pp. 1044-1051, 2014.
3. S. S. Islam and S. Islam, "A survey on packet capture and analysis tools for network forensics," in Proceedings of the 4th International Conference on Networking, Systems and Security (NSysS), pp. 1-6, 2017.
4. R. Droms, "SMTP service extension for message size declaration," RFC 1870, 1995.
5. J. Klensin, "IMAP4," RFC 3501, 2003.
6. J. C. Klensin, "Simple Mail Transfer Protocol," RFC 2821, 2001.