PROJECT REPORT ON PACKET ANALYSER

Prepared for

Mr. Shoaib Raza

Lecturer, Computer Networks

National University, FAST NUCES

By

Mr. Osama Irfan

Student, National University, FAST NUCES

and

Mr. Saad Hassan

Student, National University, FAST NUCES

DECEMBER 4, 2023

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Network Traffic Analyzer

OSAMA IRFAN. AUTHOR[1], (21k-4772), SAAD HASSAN. AUTHOR[2], (21K-3605)
[1]Department of Cyber Security, FAST NUCES, Karachi, Pakistan (e-mail: k214772@nu.edu.pk)
[2]Department of Cyber Security, FAST NUCES, Karachi, Pakistan (e-mail: k213605@nu.edu.pk)

ABSTRACT This project presents the development and implementation of a network traffic analyzer utilizing the Python Scapy library. The primary objective is to capture network packets comprehensively, organize the captured data, and store it in a structured log file. Additionally, a web interface is integrated to facilitate a user-friendly display of the captured traffic. The system provides detailed information for each packet, including date-time stamps, source and destination IP addresses, source and destination ports, raw data, and protocol. Furthermore, it incorporates functionality to detect specific credentials such as usernames and passwords within the captured logs.

I. INTRODUCTION

Network traffic analysis is a critical aspect of network security and performance monitoring. The proposed project aims to address this by implementing a robust network traffic analyzer. The tool utilizes the Python Scapy library, a powerful packet manipulation tool, to capture and dissect network packets.

II. METHODOLOGY

A. PACKET CAPTURE

The system employs the Scapy library to capture network packets. It listens on the specified network interfaces (Wi-Fi, Ethernet) and captures packets in real-time. The captured packets are processed and relevant details are extracted.

B. PACKET DETAILS EXTRACTION

Upon capturing packets, relevant information is extracted, including:

- Date and time of capture
- Source and destination IP addresses
- Source and destination ports
- Raw data payload
- Protocol type (TCP, UDP)

C. DATA ORGANIZATION AND STORAGE

Captured packet details are organized systematically and stored in a log file for future reference and analysis. The log file maintains a structured format, enabling ease of interpretation and retrieval of information.

D. WEB INTERFACE INTEGRATION

The system features a user-friendly web interface powered by Flask, a robust Python web framework. Flask enables the creation of an interactive display showcasing detailed packet information. Through Flask's integration with HTML, CSS, and potentially JavaScript, users can easily explore captured data, including date-time stamps, IP addresses, ports, protocol types (TCP, UDP), and detected payloads. This seamless integration ensures real-time updates and smooth communication between the backend capturing system and the frontend interface, facilitating efficient analysis of the captured network traffic data.

FIGURE 1. Packet Analyser Web Interface

III. EXPERIMENTAL SETUP

Following is the experimental setup for this project:

A. COMMAND EXECUTED:

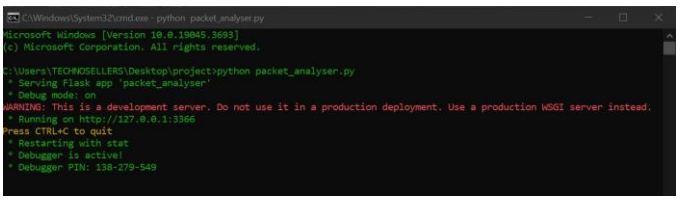

FIGURE 2. Command Executing Script

B. GETTING SERVER'S IP ADDRESS:

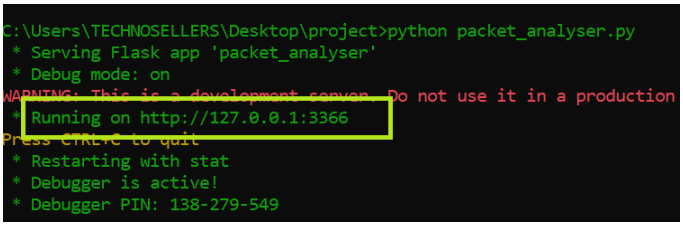

FIGURE 3. Getting Server's IP Address

C. RUNNING SERVER'S IP ON THE WEB BROWSER:

FIGURE 4. Running IP On Web Browser

D. SELECT INTERFACE:

FIGURE 5. Selecting Interface

E. START CAPTURE:

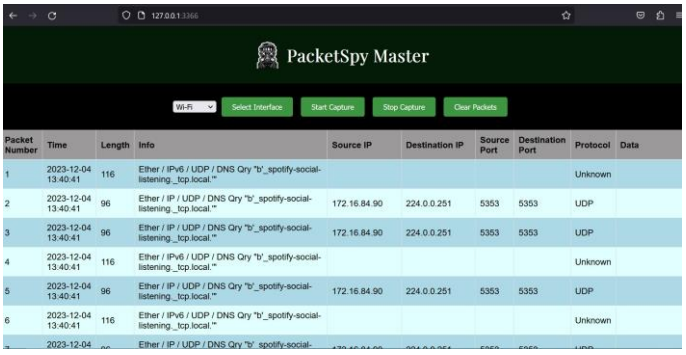

FIGURE 6. Starting Packet Capture

F. STOP CAPTURE:

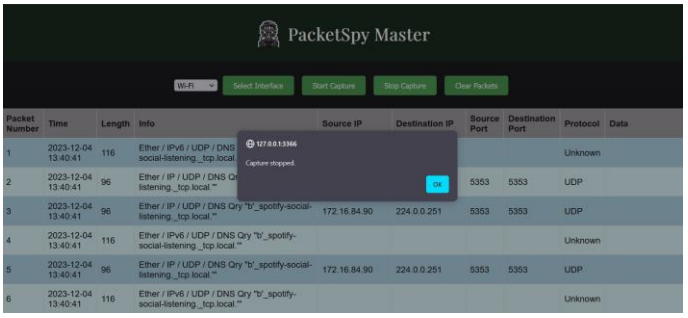

FIGURE 7. Stopping Packet Capture

G. ANALYSING TXT FILE:

Captured packets will be saved automatically once the capture is stopped and will be saved in an automatically generated TXT file.

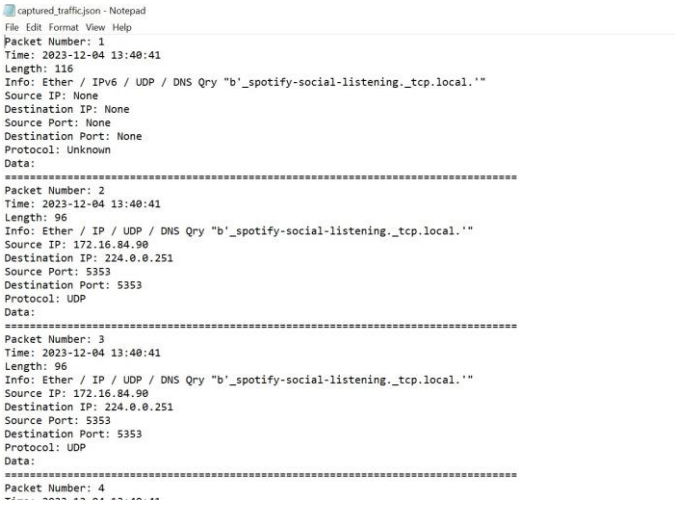

FIGURE 8. TXT file View

H. ANALYSING DETECTED CREDENTIALS:


captured_traffic.json - Notepad

File Edit Format View Help

```
================================================================================
                              Detected Credentials:
================================================================================
Packet Number: 158
Time: 2023-12-04 13:56:39
Length: 594
Info: Ether / IP / TCP 172.16.91.30:52575 > 44.228.249.3:http PA / Raw
Source IP: 172.16.91.30
Destination IP: 44.228.249.3
Source Port: 52575
Destination Port: 80
Protocol: TCP
Raw Data: POST /userinfo.php HTTP/1.1

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Content-Type: application/x-www-form-urlencoded

Content-Length: 22

Origin: http://testphp.vulnweb.com

Connection: keep-alive

Referer: http://testphp.vulnweb.com/login.php

Upgrade-Insecure-Requests: 1


uname=osama&pass=irfan
================================================================================
```

FIGURE 9. Detected Credentials

IV. RESULTS AND DISCUSSION

The developed network traffic analyzer successfully captures and organizes network packets. Each captured packet is detailed with relevant information, facilitating analysis and identification of network activities. The web interface offers a convenient way for users to access and visualize the captured traffic, promoting ease of use and interpretation.

V. CONCLUSION

In conclusion, the implemented network traffic analyzer uti- lizing the Python Scapy library effectively captures, orga- nizes, and presents network packet details. The integration of a web interface enhances user accessibility and provides a seamless platform for analyzing network traffic. The system's ability to detect specific credentials within captured logs contributes to its utility in network security applications.

REFERENCES

[1] Python Scapy Library - https://scapy.net/

[2] Flask Web Framework - https://flask.palletsprojects.com/

[3] Python Official Website - https://www.python.org/

[4] YouTube - https://www.youtube.com/