# Comprehensive User Guide for Network Traffic Analysis Tool

## Introduction

Welcome to the Network Traffic Analysis Tool. This Python-based application is designed to analyze network traffic data from a raw text file, parse the relevant information, visualize key metrics, and generate detailed reports. This guide provides an in-depth explanation of the tool's features, the underlying logic of its code, and how to use it effectively.

## Features Overview

This tool is equipped with the following features:  
1. \*\*Parsing and Structuring Data\*\*: Reads raw text data using regular expressions and converts it into a structured DataFrame.  
2. \*\*Data Visualization\*\*: Creates various visualizations including packet frequency charts, IP traffic distribution, and TCP flag summaries.  
3. \*\*Report Generation\*\*: Generates detailed reports in PDF and HTML formats for offline analysis.  
4. \*\*Interactive Graphical User Interface (GUI)\*\*: Allows users to explore the data interactively.  
5. \*\*Export Functionality\*\*: Supports exporting the parsed data to CSV format.

## Detailed Explanation of Code

### 1. Parsing Text File

The function `parse\_text\_file\_to\_dataframe(file\_path)` is responsible for reading and extracting structured information from a raw text file using a regular expression. The parsed data includes columns such as timestamps, source and destination IPs, ports, TCP flags, and packet details.

Code Breakdown:

* - \*\*`re.compile`\*\*: Defines a regular expression pattern to match lines of network traffic data.
* - \*\*`file.read()`\*\*: Reads the entire content of the input file.
* - \*\*`finditer()`\*\*: Iterates through the file content to find all matches of the defined pattern.
* - \*\*`groupdict()`\*\*: Extracts matched groups into a dictionary.
* - \*\*`pd.DataFrame`\*\*: Converts the list of dictionaries into a DataFrame for easier manipulation.

### 2. Data Visualization

The tool offers three primary visualizations to help analyze network traffic data:  
a. \*\*Packet Frequency Over Time\*\*: Uses line charts to show how the number of packets changes over time.  
b. \*\*IP Traffic Distribution\*\*: Displays a pie chart to visualize the proportion of traffic handled by various IPs.  
c. \*\*TCP Flags Distribution\*\*: Generates a bar chart showing the frequency of each TCP flag.

Technical Details:

* - \*\*Matplotlib\*\* is used for plotting charts.
* - \*\*`resample('S')`\*\*: Aggregates data by seconds for time-based analysis.
* - \*\*Pie and bar charts\*\* include labeled segments for better interpretability.

### 3. Report Generation

Two types of reports can be generated for detailed analysis:  
a. \*\*PDF Report\*\*: Includes a summary of the data, such as total packets, unique IPs, and the most active IPs.  
b. \*\*HTML Report\*\*: A comprehensive report with tables, charts, and formatted content generated from Markdown.

Key Functions:

* - \*\*`FPDF`\*\*: A library for creating PDF files with text and charts.
* - \*\*`markdown`\*\*: Converts structured Markdown content into HTML.

### 4. Interactive GUI

The GUI is built using `Tkinter`, providing an intuitive interface for users to visualize data, export results, and generate reports.

Interface Details:

* - \*\*Treeview\*\*: Displays the parsed data in a tabular format.
* - \*\*Buttons\*\*: Facilitate exporting data and generating reports.
* - \*\*Dynamic Plotting\*\*: Users can switch between different visualizations interactively.

## Usage Instructions

Follow these steps to use the application:

1. 1. \*\*Launch the Application\*\*: Execute the script in a Python environment.
2. 2. \*\*Select Input File\*\*: Use the file dialog to select a text file containing network traffic data.
3. 3. \*\*View Data\*\*: The parsed data will be displayed in the GUI, along with visualization options.
4. 4. \*\*Generate Reports\*\*: Click the respective buttons to generate and open PDF or HTML reports.
5. 5. \*\*Export Data\*\*: Save the parsed data to a CSV file for further analysis.

## Understanding TCP Flags

The application interprets TCP flags using the following definitions:  
- \*\*SYN (S)\*\*: Synchronize.  
- \*\*ACK (A)\*\*: Acknowledgment.  
- \*\*FIN (F)\*\*: Finish.  
- \*\*RST (R)\*\*: Reset.  
- \*\*PSH (P)\*\*: Push.  
- \*\*URG (U)\*\*: Urgent.  
- \*\*ECE (E)\*\*: Explicit Congestion Notification.  
- \*\*CWR (W)\*\*: Congestion Window Reduced.  
These flags are analyzed to understand packet-level behaviors.

## Advanced Usage

For advanced users, the code can be modified to accommodate custom requirements, such as:  
- Parsing additional fields.  
- Customizing visualizations.  
- Enhancing report formats.

Feel free to extend the code as needed.

## Conclusion

This guide provided a detailed overview of the Network Traffic Analysis Tool, including its features, code structure, and usage. We hope this tool helps you effectively analyze and interpret network traffic data.