WIPRO NGA Program – LSP Batch

Capstone Project Presentation – 04 June 2024

Linux Network Packet Statistics Display Project

Presented by – Anirban Mazumdar

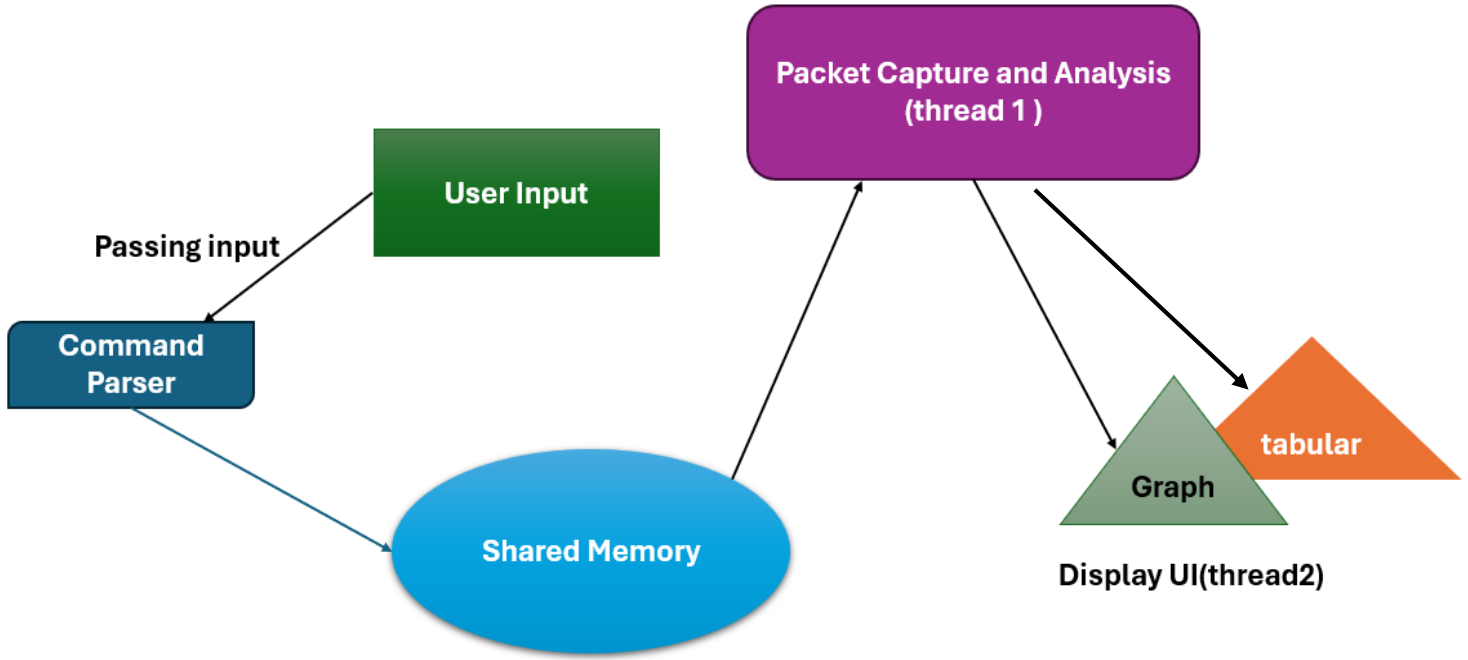
AGENDA

- Introduction
- Design Overview
- Code Functionality
- Test cases
- Conclusion
- Question and Answer

Introduction

- Enhance understanding of packet capture. Gain skills in system programming on Linux.
- Capturing and analyzing network packets on a Linux system.
- Statistics presented in both tabular and graphical formats.
- Updates packet statistics in real-time.
- Command-line parameters allow customization of display options.

Design Overview



Code Functionality

Code Functionality :

Data Structures:

- Packet Statistics: This structure holds various packet statistics such as counts of TCP, UDP, and ICMP packets, as well as arrays to store packet sizes for each type.

Shared Memory:

- The code creates a shared memory segment using shmget to store and share Packet Statistics data between threads.

Packet Capture and Analysis (Thread 1)
(capture_and_analyze_packets):

- This thread simulates packet capture by generating data for TCP, UDP, and ICMP packets.
- It updates the shared memory segment with the generated statistics at regular intervals.

Code Functionality

Code Functionality:

Display (Threads 2) (Display_ui):

- This thread continuously reads the shared memory segment and displays packet statistics either in tabular or graphical format based on the chosen display format (TABULAR or GRAPH).
- The display format and packet types to show (TCP, UDP, ICMP) are determined based on command-line arguments.

Command-line Arguments (parse_arguments):

- Parses command-line arguments to set the display format (tabular or graph) and the packet types to show (tcp, udp, icmp, all).

Test Cases

Basic Functionality Test

Input: Run the program without any command-line arguments.
Expected Output: The program should run with default settings, capturing packets, and displaying statistics in tabular format.

Display Format Test

Input: Run the program with command-line argument graph.
./program graph
Expected Output: The program should display packet statistics in graphical format.

Packet Type Filtering Test

Input: Run the program with command-line argument all
./program all
Expected Output: The program should display statistics for all packet types (TCP, UDP, ICMP).

Test Cases

Specific Packet Type Test
Input: Run the program with command-line argument tcp
./program tcp
Expected Output: The program should display statistics only for TCP packets.

Invalid Command-line Argument Test
Input: Run the program with an invalid command-line argument
./program invalid_argument
Expected Output: The program should print a usage message indicating the correct usage of command-line arguments.

QUESTIONS ?

Thank you