Intrusion Detection Dashboard - Comprehensive Explanation

1. Introduction

This document provides an in-depth explanation of the real-time Intrusion Detection Dashboard developed using Python. The dashboard captures live network traffic, analyzes packet details, and visualizes network activity using graphs and tables. It utilizes Dash for UI, Plotly for visualization, and Scapy for packet sniffing.

2. Libraries and Tools Used

- Dash: Web application framework for Python.
- Dash Bootstrap Components: Provides UI components with Bootstrap styling.
- Plotly: Enables interactive graphing and visualization.
- Scapy: Powerful packet manipulation library.
- Pandas: Used for data manipulation and processing.
- Threading: Allows background execution of packet sniffing while keeping the UI responsive.

3. Packet Sniffing Mechanism

Packet sniffing is implemented using Scapy's `sniff()` function, which captures live packets on the network. Each captured packet is processed, extracting relevant details such as:

- Timestamp
- Source IP Address
- Destination IP Address
- Protocol Type
- Packet Length

The data is stored in a list and is continuously updated in real time.

4. Dashboard Layout and Components

The dashboard consists of three main sections:

1. **Live Packet Monitoring Graph**
   - Displays packet trends over time.
   - Shows real-time fluctuations in packet size.

2. **Protocol Breakdown Pie Chart**
   - Analyzes the percentage of different protocols in the captured traffic.

  - Helps identify suspicious protocol activity.

3. **Packet Details Table**
  - Displays all captured packets in a tabular format.
  - Includes source and destination IPs, protocol type, and packet size.

5. Auto-Refresh and Real-Time Updates

The dashboard refreshes every 2 seconds using Dash's `dcc.Interval()` component. This ensures that the latest captured packets are displayed dynamically without requiring manual refreshing.

6. Running the Application

To run the application:

1. Install the required dependencies using `pip install dash dash-bootstrap-components plotly scapy pandas`.
2. Execute the Python script.
3. Open a web browser and go to `http://127.0.0.1:8050`.
4. The dashboard will display live network traffic data.

7. Output Explanation

After running the script, the following outputs are observed:

- **Web-based Dashboard Opens**
  - A browser window opens displaying the dashboard.

- **Live Packet Monitoring Graph**
  - A real-time updating line graph shows packet sizes over time.

- **Protocol Breakdown Pie Chart**
  - Displays a pie chart representing the distribution of different network protocols.

- **Packet Details Table**
  - A dynamically updating table lists all captured packets with timestamps, IP addresses, and protocol details.

These outputs provide real-time insights into network activity, enabling users to detect anomalies.

8. Security and Performance Considerations

1. **Security Concerns**
   - Running this script requires administrator privileges.
   - Unauthorized packet sniffing can violate legal and ethical guidelines.
   - Always ensure compliance with network security policies.

2. **Performance Considerations**
   - Packet storage is limited to avoid excessive memory usage.
   - The table is optimized to display only recent 1000 packets.
   - Multi-threading ensures smooth execution without lagging the UI.

9. Use Cases

This Intrusion Detection Dashboard can be used for:

- **Real-time Network Monitoring:** Identify suspicious activities and track live traffic.
- **Cybersecurity Research:** Analyze network behavior to detect anomalies.
- **Traffic Analysis:** Understand data flow and protocol distribution in a network.
- **Security Awareness Training:** Educate professionals about packet analysis and intrusion detection.

10. Conclusion

This dashboard provides an interactive and real-time solution for monitoring network traffic, making it useful for cybersecurity professionals, network engineers, and ethical hackers. The combination of Scapy, Dash, and Plotly ensures a robust and efficient intrusion detection system.