Packet Sniffer with DoS Detection

Final Project Report

ECEN 335 - Computer Networks

Fall 2024

Student's Name :

- Mai Waheed , ID : 202200556
- Zeina Ayman , ID : 202200351
- Lujain Ahmad , ID : 202201738
- Farida Mohamed , ID : 202202579

Submitted to : Dr. Nashwa Abdelbaki

Table of Contents :

1. Introduction

In today's digital world, network security is crucial because online services are seriously threatened by Denial-of-Service (DoS) attacks. Denial-of-Service (DoS) attack, where an attacker overwhelms a target server with an excessive amount of traffic, rendering it unusable to legitimate users. This project develops a Packet Sniffer with DoS Detection tool using Python and the Scapy library to monitor network traffic in real time. By inspecting the packets, the sniffer can identify different protocols (such as ICMP, TCP, and UDP) used in network communications and track the rate at which packets are arriving, and looks for packet flooding that might be a sign of a denial-of-service attack. One important feature is DoS detection, which, when the packet rate exceeds a predetermined threshold, stops sniffing and raises an alert. The Tkinter GUI allows users to view real-time packet statistics, adjust detection thresholds, and monitor network activity. This project integrates network analysis and security, offering a practical solution for detecting DoS attacks while providing valuable learning experience in network security.

2. Project Description

2.1 Key Features and Functionalities

1. **Packet Capturing and Analysis**

   The tool uses Scapy's sniff() function to capture network packets in real time. It analyzes every packet that is captured, extracting important data like the protocol type (e.g., ICMP, TCP, UDP), packet size, and source and destination IP addresses. By analyzing the protocol types and packet details, the tool helps users understand network traffic patterns and detect unusual behaviors.

2. **Denial-of-Service (DoS) Attack Detection**

   Detecting DoS attacks is an essential part of this project. DoS attacks frequently entail sending an excessive number of requests to a target network or server. To identify unusual traffic patterns, the packet sniffer keeps an eye on the rate of incoming packets per second (PPS). The DoS detection threshold is dynamically calculated as 1.5 times the average PPS over a short duration (default 10 seconds). If the packet rate exceeds this threshold, the system triggers an alert and automatically stops sniffing to prevent system overload.

3. **Graphical User Interface (GUI)**

   Tkinter was used in the design of the user interface, providing an interactive platform for users to manage and observe the sniffer. It has attributes like:

   - Threshold Setting: By modifying the detection sensitivity according to network conditions, users can modify the DoS detection threshold using a straightforward input field.
   - Live Log: Real-time details about packet captures, alerts, and status updates are shown in a scrollable log section. This feature guarantees that users can keep a close eye on the process as it unfolds.
   - Start/Stop Sniffing: By pressing a button to initiate or terminate the sniffing process, users have command over the packet capture session.

4. **Packet Flooding (DoS) Alerts**

   When a packet rate exceeding the set threshold is detected, the tool triggers a DoS alert both in the log and through a pop-up message. This alert helps inform users of potential network attacks, allowing them to take necessary actions, such as blocking malicious traffic or investigating the source of the attack. When a DoS attack is identified, the system automatically stops sniffing, avoiding additional strain on system resources and possible network damage.

5. **Average Packets Per Second (PPS) Calculation**

   The tool calculates the average packets per second (PPS) over a short period, which serves as the baseline for determining the detection threshold. This feature ensures the threshold is tailored to the network's normal activity, rather than relying on static values. This dynamic calculation provides a more adaptable and accurate means of identifying DoS attacks, which can vary based on network load and configuration.

2.2 Use Case and Practical Applications

This packet sniffer with DoS detection is valuable in network monitoring environments where it is essential to quickly detect abnormal traffic patterns and protect systems from potential attacks. It can be used in:

- Educational Environments: To share knowledge of traffic analysis, DoS attack detection, and network security concepts.
- SMEs, or small to medium-sized businesses: as a simple network monitoring tool for real-time DoS attack detection and prevention.
- Network administrators: To keep an eye on the overall condition of the network, spot problems with performance, and protect against attacks.

2.3 System Workflow

1. Initialization: To create a baseline packet rate, the program first determines the average PPS over a 10-second period.
2. Packet Capture: After sniffing is started, the tool starts to record packets and shows important information (such as IP addresses and protocols) in the log section.
3. Real-Time Statistics Update: Logs are continuously refreshed, and protocol statistics are updated in the GUI as new packets are captured.
4. DoS Detection: Sniffing is automatically stopped, and an alert is shown if the packet rate exceeds the threshold.
5. Threshold Update: Users can update the DoS detection threshold through the GUI, which will adjust the detection sensitivity accordingly.

2.4 Technologies Used

- ✓ Scapy: A Python library for network packet manipulation and sniffing.
- ✓ Tkinter: A Python library for creating graphical user interfaces.
- ✓ Python: The primary programming language used to develop the tool.

3. Key Components

Variables:

- protocol_count: A Counter that tracks packet counts by protocol type.
- large_packet_threshold: Defines the size above which a packet is considered "large."
- protocol_names: Maps protocol numbers (1, 6, 17) to their string equivalents (ICMP, TCP, UDP).
- packet_times: Stores timestamps of incoming packets for DoS detection.
- dos_threshold: The number of packets per second that triggers DoS detection.
- stop_sniffing: A flag to stop the sniffing process.
- average_pps: Stores the calculated average packets per second.

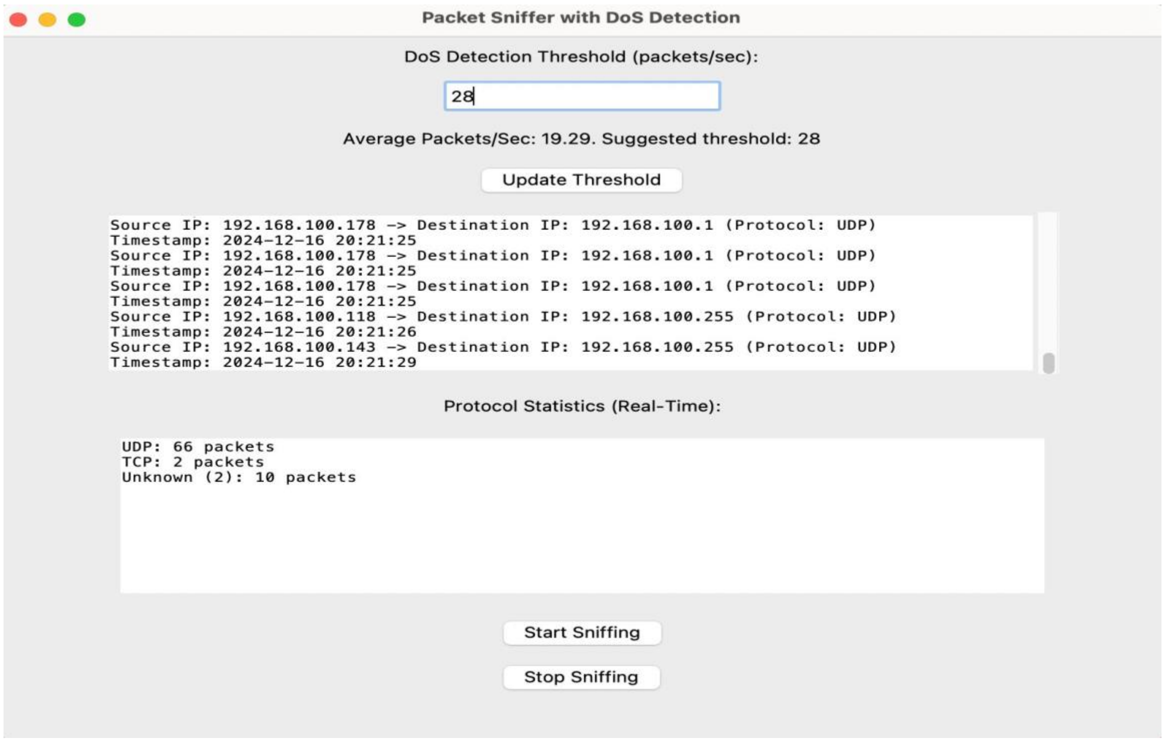
Functions:

- **calculate_average_pps(duration):**
  - Sniffs packets for a set duration and calculates the average packets per second.
  - Action: Calls the sniff function to capture packets and computes PPS.
- **update_threshold():**
  - Allows the user to update the DoS detection threshold through the GUI.
- **display_log(message):**
  - Displays messages in the GUI's scrollable text area.
- **update_protocol_stats():**
  - Refreshes the GUI to display real-time protocol counts.
- **stop_sniffing_handler():**
  - Stop the sniffing process when invoked.
- **detect_flood():**
  - Monitors packet timestamps to detect DoS attacks.

- **packet_handler(packet):**
    - Processes each captured packet:
        - Tracks protocols.
        - Detects large packets.
        - Checks for potential DoS attacks.
- **sniff_stop_filter(packet):**
    - A filter function that stops sniffing once the stop_sniffing flag is True.
- **start_sniffing():**
    - Begins the sniffing process and applies the sniff_stop_filter.
- **start_sniffing_thread():**
    - Runs the packet sniffing in a background thread to avoid blocking the GUI.
- **setup_gui():**
    - Initializes the GUI components, including buttons, text areas, and labels.
- **Main Execution Block:**
    - Calculates the average PPS before setting up the GUI and starting the Tkinter main loop.

4. Results

The tool has been tested in various network environments. Below are the results from the packet sniffing session and DoS attack detection:


Packet Sniffer with DoS Detection
DoS Detection Threshold (packets/sec):
28
Average Packets/Sec: 19.29. Suggested threshold: 28
Update Threshold
Protocol Statistics (Real-Time):
Start Sniffing
Stop Sniffing

```
Source IP: 192.168.100.178 -> Destination IP: 192.168.100.1 (Protocol: UDP)
Timestamp: 2024-12-16 20:21:25
Source IP: 192.168.100.178 -> Destination IP: 192.168.100.1 (Protocol: UDP)
Timestamp: 2024-12-16 20:21:25
Source IP: 192.168.100.178 -> Destination IP: 192.168.100.1 (Protocol: UDP)
Timestamp: 2024-12-16 20:21:25
Source IP: 192.168.100.118 -> Destination IP: 192.168.100.255 (Protocol: UDP)
Timestamp: 2024-12-16 20:21:26
Source IP: 192.168.100.143 -> Destination IP: 192.168.100.255 (Protocol: UDP)
Timestamp: 2024-12-16 20:21:29
```

```
UDP: 66 packets
TCP: 2 packets
Unknown (2): 10 packets
```

Real-Time Traffic Monitoring:

- Displays the count of packets per protocol (TCP, UDP…) and updates dynamically as new packets are captured.

DoS Detection:

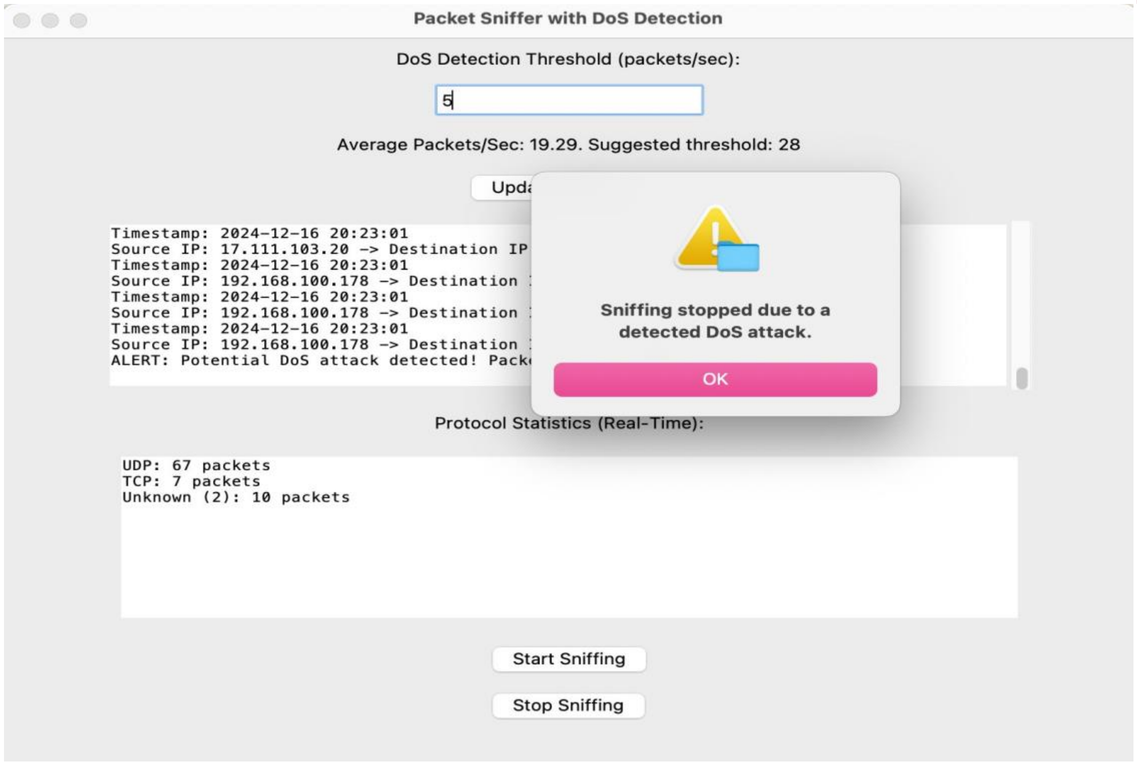
- Detects flooding behavior based on packet rate and stop sniffing when a potential DoS attack is detected.

Large Packet Detection:

- Identifies packets that exceed the 1500-byte size and generates an alert message.

GUI Interactions:

- Provides an intuitive interface for the user to interact with, monitor network traffic, update thresholds, and view logs.


Packet Sniffer with DoS Detection
DoS Detection Threshold (packets/sec):
5
Upd
OK
Protocol Statistics (Real-Time):
Start Sniffing
Stop Sniffing

Average Packets/Sec: 19.29. Suggested threshold: 28

```
Timestamp: 2024-12-16 20:23:01
Source IP: 17.111.103.20 -> Destination IP
Timestamp: 2024-12-16 20:23:01
Source IP: 192.168.100.178 -> Destination
Timestamp: 2024-12-16 20:23:01
Source IP: 192.168.100.178 -> Destination
Timestamp: 2024-12-16 20:23:01
Source IP: 192.168.100.178 -> Destination
ALERT: Potential DoS attack detected! Pack
```

Sniffing stopped due to a detected DoS attack.

```
UDP: 67 packets
TCP: 7 packets
Unknown (2): 10 packets
```

When a potential DoS attack is detected

5. List of Tables

Table 1: Mapping of Protocol Numbers to Protocol Names

Protocol Number	Protocol Name
1	ICMP
6	TCP
17	UDP

Table 2: DoS Attack Detection Thresholds

Detection Method	Threshold Value
Default Threshold	100 packets/sec
Dynamic Threshold (PPS * 1.5)	30 packets/sec

Table 3: Example Packet Log Data (Captured during sniffing)

Timestamp	Source IP	Destination IP	Protocol	Packet Size (bytes)
2024-12-16 14:32:21	192.168.1.5	192.168.1.10	TCP	1500
2024-12-16 14:32:22	192.168.1.8	192.168.1.12	UDP	1350
2024-12-16 14:32:23	192.168.1.5	192.168.1.10	ICMP	64

6. Conclusion

The project successfully implemented a network packet sniffer with basic DoS attack detection. Effective monitoring is made possible by the customizable DoS detection threshold and real-time statistics. The use of Scapy for packet sniffing combined with Tkinter for a user interface makes it accessible and functional for basic network security analysis. Future enhancements could include more sophisticated analysis, such as detecting different types of attacks, and improving the GUI with additional features like packet analysis and filtering.

7. Most Significant References

SCAPY- A powerful interactive packet manipulation program. (2018, December 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/10747338.

Intrusion Analysis System of DDOS Attack Using Python. (2024, September 6). IEEE Conference Publication | IEEE Xplore.https://ieeexplore.ieee.org/abstract/document/10774786?.com

Suroor. (n.d.). GitHub - Suroor101/Packet-Sniffer-tool: A Python-based tool with a GUI for capturing and analyzing network packets. It supports filtering by protocol (TCP, UDP, ICMP) and displays packet details like IP addresses, protocol, length, and payload. Includes packet logging and sorting features. GitHub. https://github.com/Suroor101/Packet-Sniffer-tool

Srii-Exe. (n.d.). GitHub - srii-exe/DOS_Attack_Detector: A simple python code with using of scapy libraries for analyzing network traffic and detect and prevent from the dos attack. GitHub. https://github.com/srii-exe/DOS_Attack_Detector

Author. (2024, September 17). Building A Real-time Packet Sniffer with Python And Scapy. peerdh.com. https://peerdh.com/blogs/programming-insights/building-a-real-time-packet-sniffer-with-python-and-scapy-2?.com

tkinter — Python interface to Tcl/Tk. (n.d.). Python Documentation. https://docs.python.org/3/library/tkinter.html