ISSN: 3107-6513

NetSentinal: An Automated DDoS Traffic Network Identifier for Real-Time Cybersecurity Defense

Avinash N, Dr. M. Usha Devi
Department of Computer Science, Rathinam College of Arts and Science (Autonomous), Coimbatore, Tamil Nadu, India.

Abstract – Distributed Denial of Service (DDoS) attacks pose a critical and escalating threat to modern network infrastructures, overwhelming targeted systems with malicious traffic and causing severe service disruptions. Traditional standalone detection tools suffer from fragmented analysis, limited scalability, and high rates of false positives and negatives. This paper presents NetSentinal, a centralized and automated DDoS Traffic Network Identifier designed to detect, classify, and analyze malicious traffic patterns in real-time. The proposed system integrates multiple traffic analysis techniques including packet inspection, flow-based monitoring, and anomaly detection within a unified, modular framework. NetSentinal continuously monitors live network traffic and offline PCAP datasets, identifying abnormal traffic spikes, suspicious request patterns, and protocol misuse indicative of DDoS attacks. The system further categorizes attack types such as volumetric, protocol, and application-layer attacks, and generates structured reports to aid rapid incident response. Implemented using Python on a Linux-based environment, NetSentinal demonstrates high detection accuracy, reduced response time, and improved resilience against evolving DDoS attack strategies.

Keywords – **DDoS Detection, NetSentinal, Network Traffic Analysis, Anomaly Detection, Packet Inspection, Cybersecurity, Flow-Based Monitoring, Real-Time Detection, Intrusion Detection, Network Security.**

1. Introduction

The exponential growth of internet-connected devices and online services has created an increasingly complex cybersecurity landscape. Among the most disruptive threats in this environment are Distributed Denial of Service (DDoS) attacks, which overwhelm targeted network infrastructure or services with abnormally high volumes of traffic, rendering them unavailable to legitimate users. These attacks are highly adaptable, ranging from volumetric floods and protocol exploitation to sophisticated application-layer intrusions, and they continue to evolve in complexity and frequency.

DDoS traffic identification involves the analysis of multiple network parameters including packet rate, traffic volume, protocol behavior, request patterns, and source distribution. The central challenge lies in accurately distinguishing malicious traffic from legitimate high-volume activity in real-time. Signature-based detection methods can identify known attack patterns using predefined rules, while anomaly-based approaches monitor deviations from established baselines to detect emerging or previously unknown threats. A robust detection mechanism requires both methods working in concert within a unified, scalable framework.

Traditional traffic analysis systems rely on separate, independently operated tools such as packet analyzers, intrusion detection systems (IDS), and firewall-based filters. While each tool may function adequately in isolation, their lack of integration leads to fragmented outputs, inconsistent data formats, and increased manual effort for security analysts. These limitations result in delayed detection, higher rates of misclassification, and significant operational overhead, particularly in enterprise-scale or real-time environments.

To overcome these limitations, this paper introduces NetSentinal, a DDoS Traffic Network Identifier that provides a centralized and automated solution for detecting malicious network activity. By integrating traffic capture, preprocessing, feature extraction, anomaly detection, and structured reporting within a single modular framework, NetSentinal eliminates the fragmentation inherent in traditional approaches. The system enables continuous monitoring, real-time anomaly identification, and actionable insights that significantly reduce response time and improve the overall security posture of network systems.

2. Related Works

The detection of Distributed Denial of Service attacks has been the subject of extensive research, with approaches evolving from simple rule-based systems to sophisticated machine learning and hybrid frameworks. Early DDoS mitigation techniques relied on static threshold-based filtering and access control lists, which were effective against basic volumetric attacks but

unable to adapt to changing traffic patterns or more nuanced protocol-level intrusions.

Subsequent research introduced statistical anomaly detection methods that establish traffic baselines and flag deviations as potential attacks. Techniques such as entropy-based detection, Principal Component Analysis (PCA), and autoregressive models provided better sensitivity to unknown attack vectors. However, these approaches often required significant tuning and were prone to false positives in environments with highly variable legitimate traffic.

Machine learning-based approaches have since become prominent in DDoS detection literature. Algorithms including Decision Trees, Random Forest, Support Vector Machines (SVM), and Naive Bayes have been applied to classify traffic flows based on extracted features. These methods demonstrated improved accuracy and adaptability over rule-based systems. Deep learning architectures, including Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN), have further enhanced detection by automatically learning temporal and spatial patterns from raw traffic data.

Network-based detection techniques focus on analyzing DNS records, IP address behavior, traffic flow characteristics, and protocol patterns to identify suspicious activities. Tools such as Wireshark, Tcpdump, and Scapy-based frameworks have been widely used in academic and practical settings for packet-level analysis. Flow-based monitoring solutions leveraging NetFlow and sFlow protocols enable scalable traffic summarization suitable for high-bandwidth environments.

Despite these advances, significant gaps remain. Many existing systems depend on multiple standalone tools that require separate configuration and execution, resulting in fragmented detection pipelines. Others lack real-time responsiveness or scalable architectures suitable for deployment in diverse network environments. The proposed NetSentinal system addresses these limitations by consolidating multiple detection techniques into an automated, centralized, and modular framework designed for practical cybersecurity deployment.

3. System Design

The design of NetSentinal is motivated by the need for an integrated, scalable, and automated DDoS detection solution that overcomes the operational deficiencies of traditional standalone tools. The system architecture follows a modular design philosophy where each component performs a well-defined function, and all components operate within a coordinated workflow. The primary contributions of this system are as follows:

- Automated continuous monitoring of live network interfaces and offline PCAP files for DDoS traffic identification.
- Multi-dimensional feature extraction encompassing packet rate, protocol distribution, source IP frequency, and request patterns.
- Threshold-based and rate-analysis anomaly detection to classify traffic as normal or indicative of a DDoS attack.
- Categorization of attack types including volumetric, protocol-based, and application-layer attacks.
- Centralized, structured report generation to support efficient incident response by security analysts.

3.1 Input Module

The input module serves as the entry point of the NetSentinal framework. It accepts user-specified parameters through a command-line interface, including the detection mode (live traffic or PCAP file analysis), the target network interface or file path, and the anomaly detection threshold. Input validation is performed to ensure the specified interface exists or the PCAP file is correctly formatted, preventing execution errors and ensuring stable operation. This module also supports configurable parameters such as protocol filters and detection sensitivity levels, allowing adaptation to diverse network environments.

3.2 Traffic Capture Module

The traffic capture module is responsible for acquiring raw network data from the specified source. For live traffic analysis, the module employs packet sniffing techniques using the Scapy library to capture packets from the target network interface. For offline analysis, it reads pre-recorded PCAP files and processes them sequentially. This module operates continuously in live mode, ensuring that all incoming traffic is captured without gaps. The collected packets form the foundational dataset for subsequent preprocessing and analysis stages.

3.3 Preprocessing and Feature Extraction Module

The preprocessing module processes raw captured traffic to remove noise and extract meaningful attributes. It standardizes packet data into a consistent format and identifies key traffic features essential for anomaly detection. Extracted features include total packet count, packets per second, unique source IP addresses, protocol distribution (TCP, UDP, ICMP), SYN request frequency, and request rate per IP. These features are computed over defined time windows and passed to the

detection module for analysis. Libraries including NumPy and Pandas are used for efficient data manipulation and organization.

3.4 Detection and Analysis Module

The detection module constitutes the core analytical engine of NetSentinal. It evaluates extracted traffic features against predefined thresholds and detection logic to identify anomalies indicative of DDoS attacks. The module implements rate-analysis techniques to detect sudden traffic spikes and threshold-based methods to flag abnormal request frequencies and suspicious IP concentrations. Detected anomalies are further classified by attack type, including volumetric attacks characterized by high packet rates, protocol attacks identified through SYN flood patterns, and application-layer attacks recognized by repeated HTTP requests. This classification provides deeper insight into the nature of the threat and informs appropriate mitigation responses.

3.5 Automation and Report Generation Module

The automation module coordinates the sequential execution of all system components, managing data flow between modules and ensuring consistent operation without manual intervention. Upon completion of analysis, the report generation module organizes detected anomalies, suspicious IP addresses, traffic statistics, and attack classifications into structured output files stored in a designated directory. Reports are formatted for clarity and readability, supporting documentation requirements for academic submissions, security audits, and incident response workflows.

4. System Architecture

The NetSentinal system follows a layered, modular architecture that ensures clean separation of concerns between data acquisition, processing, detection, and output. As depicted in the system architecture diagram, the user interacts with the framework through a Command-Line Interface (CLI), which validates inputs and routes them to the appropriate processing pipeline.

The Traffic Monitoring Module serves as the central processing unit, incorporating the packet sniffer, feature extractor, and anomaly detector as sub-components. External data sources, including live network traffic and packet capture files, feed directly into this module. Detection techniques employed include threshold-based filtering and rate analysis, enabling the system to identify both known and emerging DDoS signatures. Processed results are persisted in a storage module comprising a structured results database, from which the Report Generation Module derives the final output. The output consists of a detection summary and a set of attack indicators presented in both terminal and file-based formats.

This architecture ensures that each functional layer can be independently tested, maintained, and extended. New detection algorithms or machine learning models can be integrated at the detection layer without modifying upstream or downstream components, preserving the system's overall integrity and scalability.

5. Implementation

NetSentinal is implemented using Python 3.x on a Linux-based operating system, with Kali Linux as the preferred deployment platform due to its comprehensive support for network security tools and libraries. Python was selected for its simplicity, powerful library ecosystem, and strong support for network programming and automation, which are essential requirements for a real-time DDoS detection system.

The core packet capture and manipulation functionality is provided by the Scapy library, which enables granular inspection of network packets at the protocol layer. Traffic data processing and feature computation are performed using NumPy and Pandas, which offer efficient data structures and numerical operations suitable for high-volume traffic datasets. Standard Python modules including os, sys, and subprocess facilitate system-level operations and process management.

The system exposes a command-line interface built using Python's argparse module, supporting two primary operational modes. In live mode, NetSentinal captures packets directly from a specified network interface and performs real-time analysis. In PCAP mode, the system reads a stored packet capture file and performs offline analysis, making it suitable for post-incident forensic examination. The following code segment illustrates the command-line entry point of the framework:

```
import argparse
from core.engine import NetSentinalEngine
from core.logger import log_success

def main():
    parser = argparse.ArgumentParser(
```

 ISSN: 3107-6513

```
        description="NetSentinal - DDoS Traffic Identifier System"
    )
    detect.add_argument("--mode", choices=["live", "pcap"], required=True)
    detect.add_argument("--target", required=True)
    detect.add_argument("--threshold", type=int, default=1000)
    engine = NetSentinalEngine()
    if args.command == "detect":
        engine.run_detection(args.mode, args.target, args.threshold)
        log_success("DDoS Detection executed successfully")
```

Fig 1. NetSentinal Command-Line Entry Point

6.

Results and Output Analysis

The performance of NetSentinal was evaluated across two operational modes: live traffic monitoring on a test network interface and offline PCAP file analysis using pre-recorded attack datasets. The system successfully identified DDoS attack patterns in both modes, demonstrating its versatility and reliability.

6.1 Live Traffic Detection

In live traffic mode, NetSentinal was deployed on a test network interface subjected to simulated DDoS traffic. The system captured 15,320 packets within a 15-second monitoring window, computing a packet rate of 1,250 packets per second against a configured threshold of 1,000 packets per second. The anomaly detection module correctly identified the traffic spike and flagged multiple source IPs exhibiting abnormal request frequencies. Key detection indicators included high packet rate exceeding the configured threshold and multiple concurrent requests originating from the same IP addresses, consistent with a coordinated DDoS attack pattern.

6.2 PCAP File Analysis

In PCAP analysis mode, the system processed a traffic capture file containing 32,540 packets sourced from 120 unique IP addresses. Protocol distribution analysis revealed 70% TCP traffic and 30% UDP traffic. The detection module identified a sudden traffic spike, high request frequency, and repeated SYN requests characteristic of a SYN flood attack. The system generated a structured detection report identifying the primary attack source IPs and summarizing the attack characteristics, providing actionable intelligence for incident response.

Table 1. Detection Output Summary

Parameter	Live Mode	PCAP Mode
Total Packets Captured	15,320	32,540
Packet Rate (pps)	1,250	N/A
Unique IPs Detected	Not specified	120
Protocol Distribution	TCP/UDP Mix	TCP 70%, UDP 30%
DDoS Indicators Found	Yes	Yes
Attack Type Identified	Volumetric / Rate-based	SYN Flood
Detection Result	Suspicious Traffic Detected	DDoS Attack Identified

7.

Objective and Scope

The primary objective of this research is to develop a unified, automated, and efficient system for identifying and analyzing DDoS traffic in network environments. NetSentinal aims to accurately detect malicious traffic in real-time, minimize the manual effort required for network monitoring, and provide actionable intelligence to support rapid incident response. The system is designed to overcome the fragmentation and inefficiency inherent in traditional multi-tool detection approaches by consolidating all analysis stages within a single, cohesive framework.

The scope of this work encompasses the design, implementation, and evaluation of the NetSentinal framework in both live and offline traffic analysis scenarios. The study focuses on threshold-based and rate-analysis detection techniques applicable to

volumetric, protocol, and application-layer DDoS attacks. The system is intended for deployment in academic research environments, small to medium-scale organizational networks, and as a foundational platform for future enhancements incorporating machine learning-based detection.

8. Security and Ethical Considerations

The NetSentinal framework is developed strictly for legitimate cybersecurity research, authorized network testing, and educational purposes. All traffic analysis activities conducted using this system must be performed only on networks for which the user has explicit authorization. Unauthorized network monitoring may violate applicable data privacy regulations and result in legal consequences.

The system is designed to collect only network-level metadata necessary for DDoS detection and does not store or transmit payload data or personally identifiable information. Collected traffic data is handled securely within a controlled local environment. Users of the framework are expected to adhere to ethical hacking guidelines and applicable cybersecurity laws. Additionally, NetSentinal incorporates input validation and access control mechanisms to prevent misuse of the framework itself, ensuring that the system remains a responsible and reliable security tool.

9. Conclusion

This paper presented NetSentinal, an automated and centralized DDoS Traffic Network Identifier designed to address the critical limitations of traditional standalone detection approaches. By integrating packet inspection, flow-based monitoring, and anomaly detection within a unified modular framework, NetSentinal provides real-time detection, structured output, and actionable reporting capabilities that significantly enhance network security operations.

The system demonstrated effective detection of both volumetric and protocol-level DDoS attacks across live and PCAP analysis modes. Its modular architecture ensures adaptability to evolving attack strategies and scalability across different network environments. The use of open-source technologies and minimal hardware requirements makes NetSentinal a cost-effective solution accessible to academic researchers, security analysts, and organizations seeking reliable DDoS defense mechanisms.

Future enhancements to the NetSentinal framework include the integration of machine learning classifiers for improved detection accuracy against sophisticated and zero-day DDoS attacks, the development of a graphical user interface for broader accessibility, and the incorporation of automated mitigation responses such as dynamic firewall rule generation. These improvements will further strengthen the system's capability as a comprehensive, production-ready DDoS defense platform.

10. References

[1] Mirkovic, J. and Reiher, P., "A taxonomy of DDoS attack and DDoS defense mechanisms," ACM SIGCOMM Computer Communication Review, vol. 34, no. 2, 2004, pp. 39–53.

[2] Zargar, S.T., Joshi, J., and Tipper, D., "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, 2013, pp. 2046–2069.

[3] Kumar, S., "Smurf-based distributed denial of service (DDoS) attack amplification in internet," Proceedings of the 2nd International Conference on Internet Monitoring and Protection, 2007.

[4] Bhushan, K. and Gupta, B.B., "Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment," Journal of Ambient Intelligence and Humanized Computing, vol. 10, 2019, pp. 1985–1997.

[5] Liao, Q., Li, Z., and Striegel, A., "Could per-flow queuing be the cause rather than the cure for burstiness?" Proceedings of the IEEE International Conference on Communications (ICC), 2011.

[6] Mahjabin, T., Xiao, Y., Sun, G., and Jiang, W., "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," International Journal of Distributed Sensor Networks, vol. 13, no. 12, 2017.

[7] Ahmed, M.E., Kim, H., and Park, M., "Mitigating DNS query-based DDoS attacks with machine learning on software-defined networking," Proceedings of the IEEE Military Communications Conference, 2017.

[8] Behal, S. and Kumar, K., "Detection of DDoS attacks and flash events using novel information theory metrics," Computer Networks, vol. 116, 2017, pp. 96–110.

[9] Gavrilis, D. and Dermatas, E., "Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features," Computer Networks, vol. 48, no. 2, 2005, pp. 235–245.

[10] Peng, T., Leckie, C., and Ramamohanarao, K., "Survey of network-based defense mechanisms countering the DoS and DDoS problems," ACM Computing Surveys, vol. 39, no. 1, 2007.

[11] Yan, Q. and Yu, F.R., "Distributed denial of service attacks in software-defined networking with cloud computing," IEEE Communications Magazine, vol. 53, no. 4, 2015, pp. 52–59.

[12] Wang, H., Zhang, D., and Shin, K.G., "Detecting SYN flooding attacks," Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2002.

[13] Kumari, P., Jain, A.K., and Gupta, B.B., "A survey on detection and defense against DDoS attacks," International Journal of Security and Its Applications, vol. 9, no. 7, 2015, pp. 201–216.

[14] Agrawal, N. and Tapaswi, S., "Defense schemes for variants of distributed-denial-of-service (DDoS) attacks in cloud computing: A survey," Information Security Journal: A Global Perspective, vol. 26, no. 2, 2017, pp. 61–75.

[15] Cambiaso, E., Papaleo, G., Chiola, G., and Aiello, M., "Slow DoS attacks: A survey of network-level and application-level defense mechanisms," Transactions on Computational Collective Intelligence and Intelligent Systems, Springer, 2015.

11. Acknowledgment

This article is the outcome of the research work carried out in the Department of Computer Science, Rathinam College of Arts and Science (Autonomous), Coimbatore. The authors would like to express their sincere gratitude to the Department for providing the necessary support, resources, and infrastructure to successfully complete this work. Special thanks are extended to Dr. M. Usha Devi, Assistant Professor and project guide, for her invaluable guidance, constructive suggestions, and continuous encouragement throughout the research. The authors also acknowledge the contributions of all faculty members of the department and the management of Rathinam College of Arts and Science for their unwavering support. Gratitude is further extended to Sprout Knowledge Solutions Pvt. Ltd., Coimbatore, for providing the opportunity to implement and validate the proposed system in a practical environment. This research was carried out as part of the B.Sc. (Digital and Cyber Forensic Science) program, and the authors dedicate this work to their parents and colleagues for their constant motivation.