Real-Time Traffic Analysis and Filtering for Dos Attacks Detection and Mitigation in Cloud Environment

Dumnamene JS Sako
Department of Computer Science
Rivers State University, Port Harcourt, Nigeria
dum.sako@ust.edu.ng

James Tonye Amachree
Department of Computer Science
Rivers State University, Port Harcourt, Nigeria

Joy Tochukwu Nnodi
Department of Computer Science & Informatics
Federal University, Otuoke, Bayelsa State, Nigeria
nnodijt@fuotuoke.edu.ng

DOI: 10.56201/ijasmt.vol.11.no3.2025.pg14.24

Abstract
This paper proposed a system that detects and mitigates DoS attacks in the cloud in real time. Known DoS attacks in the Internet generally conquer the target by exhausting its resources, that can be anything related to network computing and service performance. This necessitates robust mitigation strategies capable of defending against a diverse array of attack vectors in real time. This response significantly reduces the impact of DOS attacks and safeguards cloud resources from resource exhaustion attacks. The system uses statistical analysis, payload and signature-based capturing and pattern matching algorithms to recognize malicious activities within the network traffic. Modular system architecture enables dynamic adaptation to new attack methods through Python's filtering capabilities. A user-friendly web interface empowers security personnel with real-time attack monitoring and visualization tools. By integrating real-time analysis, advanced filtering, user-centric design, and scalability, this comprehensive DoS mitigation system offers a robust defense against evolving threats, ensuring the uninterrupted operation of critical cloud services.

Keywords: *DoS mitigation, Network traffic filtering, real-time analysis, cloud computing.*

I. Introduction

In the realm of network security, the proliferation of Internet technology and online services has introduced significant vulnerabilities, notably evidenced by the rise of Denial of Service (DoS) attacks, especially in a cloud computing environment. Cloud computing embodies a formidable set of tools and techniques that provide for the on-demand access of computing resources and services via the internet, harnessing software and hardware systems housed in data centers (Armbrust et al. 2020). With cloud computing, we get the computing power or storage we need, without having to own or manage the physical hardware ourself. Compared to traditional on-premises Information Technology (IT), where a company owns and maintains physical data centers and servers to access computing power, data storage and other resources, cloud computing offers many benefits, including: cost-effectiveness, increased speed and agility, unlimited scalability, enhanced strategic value, etc (Susnjara & Smalley, n.d.).

In spite of these obvious benefits associated with cloud computing, the issue of security remains a significant hurdle to the broad adoption of cloud computing, with Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks posing serious threats to system availability (Ali et al., 2015; Ouhssini, et al. 2024)

DoS attacks, which are a system-on-system attack, flood the application, network, or service with traffic with traffic or by sending the target information that triggers a crash. These malicious activities aim to disrupt targeted services by overwhelming service provider resources with false requests, rendering them inaccessible to legitimate users. Such attacks have become increasingly prevalent and sophisticated, inflicting substantial financial losses on their targets (Sachdeva et al., 2016; Deshmukh and Devadkar, 2015). In these instances, the DoS attack deprives legitimate users of the service or resource they expected (Jacques & Christe, 2020)

Known DoS attacks in the Internet generally conquer the target by exhausting its resources, that can be anything related to network computing and service performance, such as link bandwidth, TCP connection buffers, application/service buffer, CPU cycles, etc. Individual attackers can also exploit vulnerability, break into target servers, and then bring down services (Gu & Liu, n.d.)

DoS attack can happen both at the network and application levels. As oppose to traditional DoS attacks, application layer DoS attacks are perceived as stealthy, sophisticated and practically undetectable at the network layer (Shiravi et al, 2012). With focus on specific characteristics and vulnerabilities of application layer protocols, application layer DoS attacks are capable of inflicting the same level of impact as traditional flooding DoS attacks at a much lower cost (Jazi et al, 2017).

The evolution of DoS attacks necessitates robust mitigation strategies capable of defending against a diverse array of attack vectors. This study proposed a DoS mitigation system that incorporates effective traffic analysis and filtering mechanisms for real-time detection and prevention of DoS attacks, thereby improving the overall availability and reliability of critical infrastructure. The system comprises of key components aimed at detecting and mitigating such attacks. These include traffic capture, traffic analysis, attack detection and filtering, logging and alerting, attack monitoring, and visualization of network traffic patterns, among others.

The remainder of the paper is structured as follows: Section 2 provides an overview of related works, while the proposed approach is detailed in section 3. Section 4 presents the implementation, results and discussions. The paper concludes in Section 5.

2. Review of Related Work

Several studies have been done on detection and prevention of DoS and DDoS attacks. Lundqvist, and landsberg, (n.d.), in their work, employed proof of work protocol to mitigate DoS attacks by implementing their protocol on top of UDP in order to make a server more secure against DoS attacks. Sharma et al. (2007) used k-nearest neighbours (k-NN) to classify the DDoS attack.

The work by Cheng et al. (2009) presented a technique that employs Internet Protocol (IP) Flow Interaction (IFI) algorithm to analyse time series with multi-features of normal flow.

Zargar et al. (2013) proposed algorithm that analyses the historical traffic data to predict the expected behaviour. They also used regression models and autoregressive integrated moving average (ARIMA) models for forecasting the time series data.

Jazi et al, 2017 explored several types of application-layer DoS attacks and proposed a generic detection approach based on nonparametric CUSUM algorithm which can precisely

detect flooding attack as well as stealthy attacks. Behal and Kumar (2017) used theory-based entropy and divergence to differentiate between legitimate flow, DDoS attack and flash event.

Shidaganti et al (2020) proposed DDoS detection and prevention in cloud environments using the Selective Cloud Egress Filter (SCEF). The approach has some promising outcomes as the system can take specified corrective measures when an attack is discovered. However, it is difficult to choose an appropriate threshold due to the statistical nature of the processes.

Bamasag et al, (2022) proposed real-time DDoS monitoring and detection using machine learning techniques (Naïve Bayes, K-nearest neighbors, decision tree, random forest).

Recently, Ouhssini et al, (2024) proposed a system called DeepDefend, a framework for real-time detection and prevention of DDoS attacks, to protect cloud computing infrastructure against DDoS threats. This integrated approach combines time series analysis, genetic algorithms, and deep learning techniques, notably CNN-LSTM-Transformer networks, to predict network traffic entropy and detect potential DDoS attacks.

3. The Proposed System

The methodology proposed in this paper presents a detailed strategy for detecting and preventing DoS attacks that consists of several key interconnected components/modules, including traffic capture, traffic analysis, filtering, logging and alerting, and a user-friendly interface as presented in figure 1.

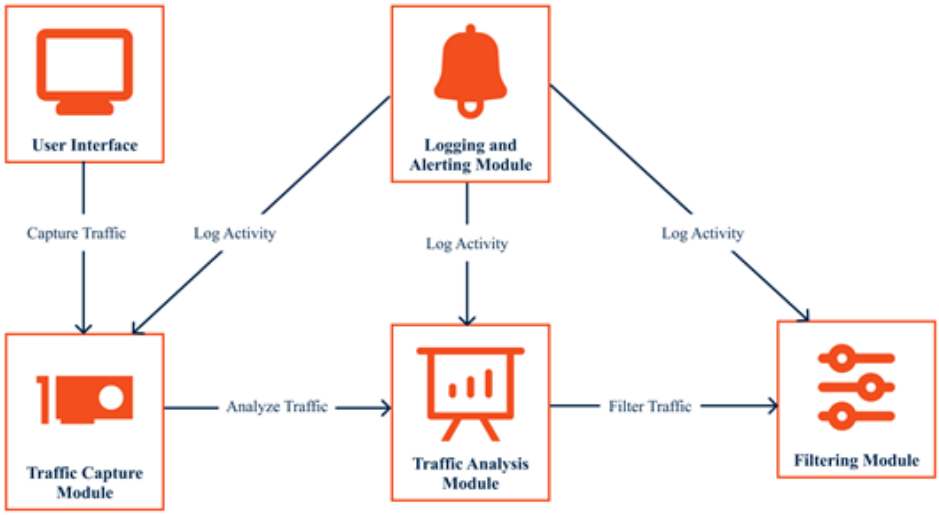

Figure 1: An overview of the functioning of the proposed system

Each module plays a crucial role in detecting, analyzing, and mitigating DoS attacks in real-time. This ensures that the DoS mitigation system operates efficiently, analyzing traffic data in real-time, identifying DoS attacks, and taking decisive actions to protect cloud services.

i. **Traffic Capture Module:** This component acts as the system's first line of defense. It continuously captures incoming network traffic directed toward the cloud service, providing a raw data stream for further analysis.
ii. **Traffic Analysis Module:** Captured traffic is then passed to the analysis module. Here the network traffic patterns are dissected, identifying anomalies and suspicious activity that might indicate a DoS attack.
iii. **Attack Detection and Filtering Module:** Based on the analysis results, the filtering module takes decisive action. If the analysis module identifies traffic patterns consistent with DoS attacks (e.g., excessive packet volume, SYN floods), it triggers an attack

detection alert. If malicious traffic is identified, the filtering module selectively blocks suspicious packets. These filters can target specific source IP addresses, ports, or traffic patterns associated with DoS attacks, effectively mitigating the attack and protecting the cloud service from resource overload.

iv. **Logging and Alerting Module:** The system maintains a comprehensive log of network activity, including detected attacks, filtering/mitigation actions taken and system events. This log serves as a valuable audit trail for security personnel and allows for post-mortem analysis of attack attempts. Additionally, the system can be configured to generate real-time alerts when DoS attacks are detected, notifying security personnel of potential threats.

v. **User Interface (UI):** A user-friendly web interface provides a central hub for system configuration, attack monitoring, and visualization of network traffic patterns. Security personnel can utilize the UI to manage the system, view real-time attack statistics, and configure filtering rules to adapt to evolving threats.

The DoS mitigation system operates in a continuous cycle, constantly monitoring and analyzing network traffic for signs of malicious activity. This continuous cycle of traffic capture, analysis, filtering, and system management ensures that the DoS mitigation system remains vigilant and proactive in protecting cloud services from denial-of-service attacks.

The DoS mitigation system primarily relies on network traffic data as its input. The traffic capture module continuously captures incoming network traffic directed towards the cloud service. These captured packets, containing information like source IP address, destination IP address, port numbers, and payload data, serve as the raw material for analysis.

The algorithms for the proposed system are presented in algorithms 1-4. These algorithms represent the core functionalities of the DoS mitigation system in a concise and understandable manner.

Algorithm 1 Traffic Capture and Preprocessing [captureAndPreproces]

```
Input: Network Data
Loop (continuously):
    packet = CaptureNextPacket()
    Extract relevant data from packet (source IP, destination IP, ports, etc.)
    Preprocess data (e.g., convert to standard format)
    Store preprocessed data in Network Traffic table
End Loop
End
```

Algorithm 2 Traffic Analysis [analyzeTraffic]

```
Input: Packet data
Analyze packet data for anomalies (e.g., high packet rate, suspicious payload)
Compare traffic patterns with Attack Signatures table
If match found with a known DoS attack signature:
    Set attack_detected = True
    Identify attack type based on signature
Else
    Set attack_detected = False
End If
Return attack_detected, attack_type (if applicable)
```

Algorithm 3 Attack Detection and Filtering [DetectAndFilterAttack(packet_data]

```
attack_detected, attack_type = AnalyzeTraffic(packet_data)
If attack_detected:
    Check Filtering Rules table for matching criteria (source IP, port, etc.)
   If matching rule found:
     Take action based on rule (Block packet, Log event, Trigger alert)
   Else
     Log potential new attack pattern
End If
End
```

Algorithm 4 Logging and Alerting **[**LogAndAlert(event_type, event_details]

```
Create new entry in System Logs table
 Store event type (e.g., attack detected, packet filtered)
Store event details (e.g., timestamp, packet data, rule triggered)
If event type requires alert (e.g., DoS attack detected):
      Generate real-time alert notification for security personnel
End If
End
```

The class diagram for the proposed system is shown in Figure 2. It displays the classes and their relationships within the system. Classes include those related to traffic capture, analysis, attack signatures, filtering rules, system logs, and user interface.

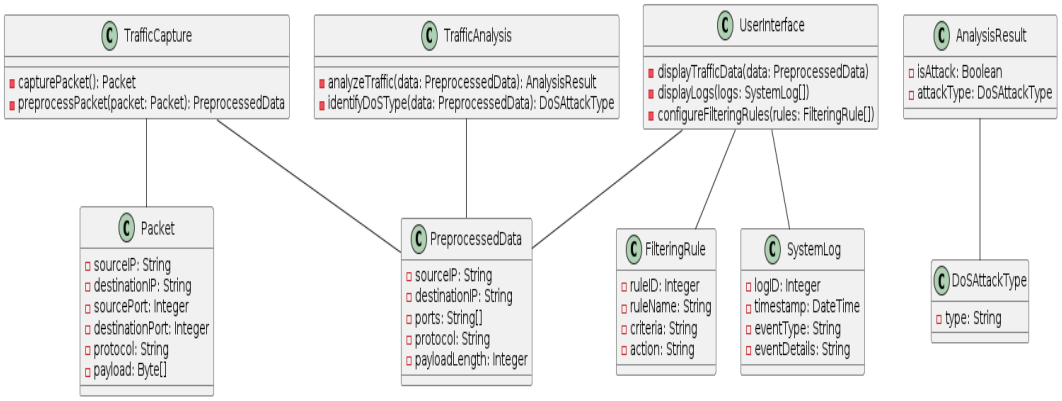

Figure 2: UML class diagram of the proposed system

This sequence diagram, presented in Figure 3, depicts the system's response to detecting and mitigating a DoS attack. It shows how the system captures and preprocesses network traffic, identifies a DoS attack pattern, checks for matching filtering rules, blocks malicious packets, logs the event, and optionally alerts security personnel through the user interface.

The proposed system requires a database to store critical information for analysis, logging, and system management. The proposed system's database schema, presented in Figure 4, consists of four main tables: Network Traffic, Attack Signatures, Filtering Rules, and System Logs. The Network Traffic table stores detailed information about incoming packets, while the Attack Signatures table contains criteria for identifying known DoS attack patterns. The Filtering Rules table includes user-defined criteria for blocking malicious traffic, and the System Logs table records all significant events for auditing and analysis.

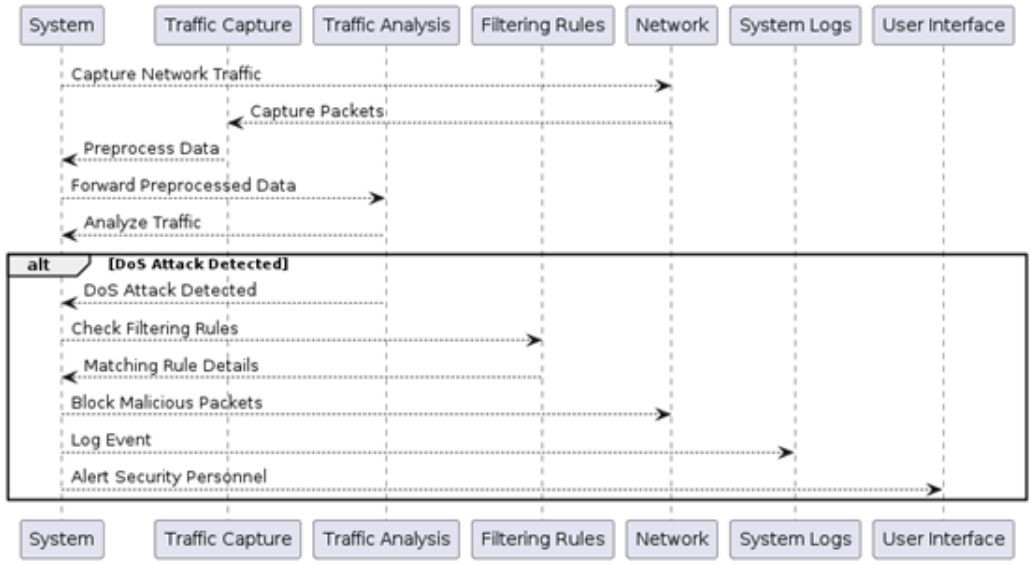

Figure 3: Sequence Diagram for Detecting and Mitigating a DoS Attack

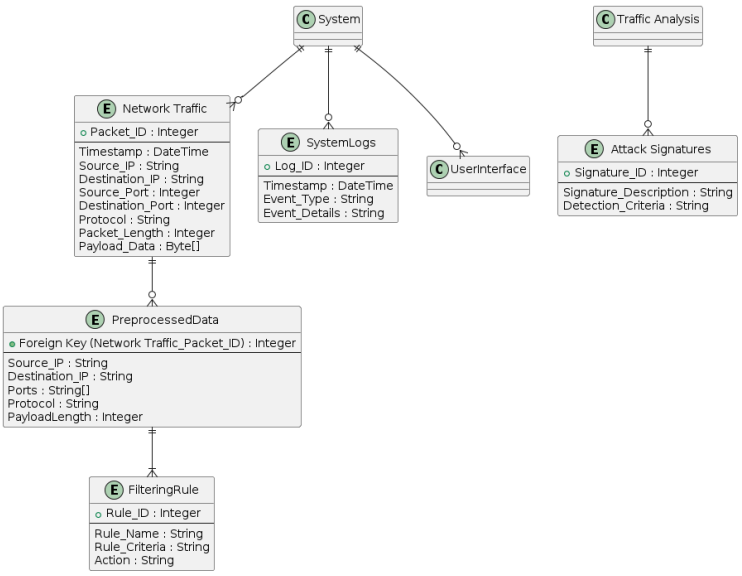

Figure 4: Entity Relationship Diagram (ERD) of the proposed system

4. Implementation and Results

The system was implemented using Python Programming Language. Python was chosen as the primary programming language due to its extensive libraries and frameworks that facilitate rapid development and robust performance. Libraries like Scapy allow for efficient packet manipulation and network traffic analysis, while Flask provides a lightweight yet powerful web framework for developing the user interface. The web-based user interface was developed using HTML, CSS, and JavaScript to ensure accessibility and ease of use.

The system generates various outputs (results) that provide valuable insights into system behavior, attack attempts, and overall network security posture. These include filtered network traffic, attack detection real-time alerts, system logs, real-time visualization of network traffic patterns.

1. Dashboard. The dashboard, presented in Figure 5, offers an overview of the system's status, including real-time traffic statistics, attack alerts, and system health indicators. It provides a quick glance at the current state of network security.

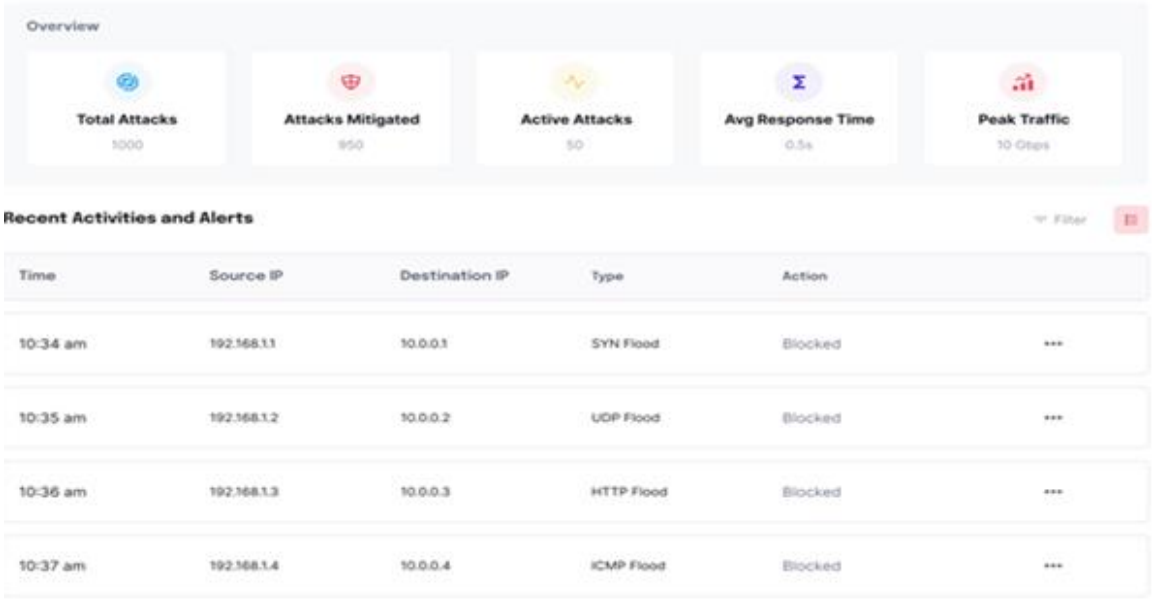

Figure 5: Dashboard Interface

2. Traffic Analysis. This screen displays detailed traffic analysis, including traffic volume over time, traffic composition, and detected anomalies. Graphical representations such as line charts and pie charts provide a clear understanding of network activity. Traffic analysis interface is shown in Figure 6.

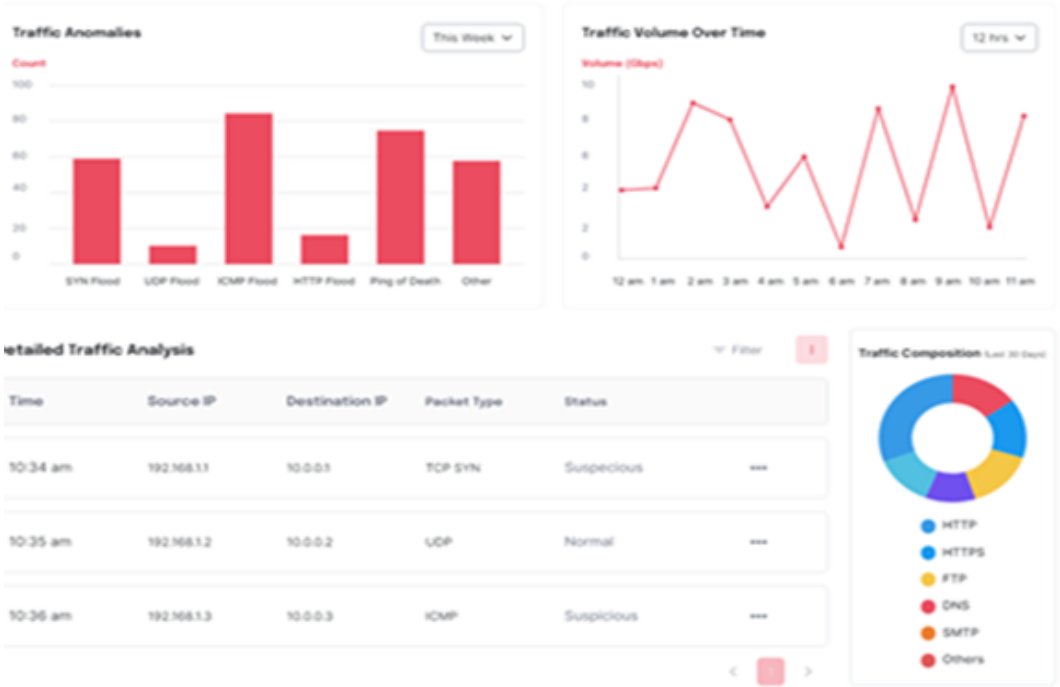

Figure 6: Traffic Analysis

3. Analysis Summary. The analysis summary interface, in Figure 7, consolidates key metrics and trends observed during traffic analysis. It highlights peak traffic times, predominant traffic types, and the frequency of different anomalies, aiding in strategic decision-making.

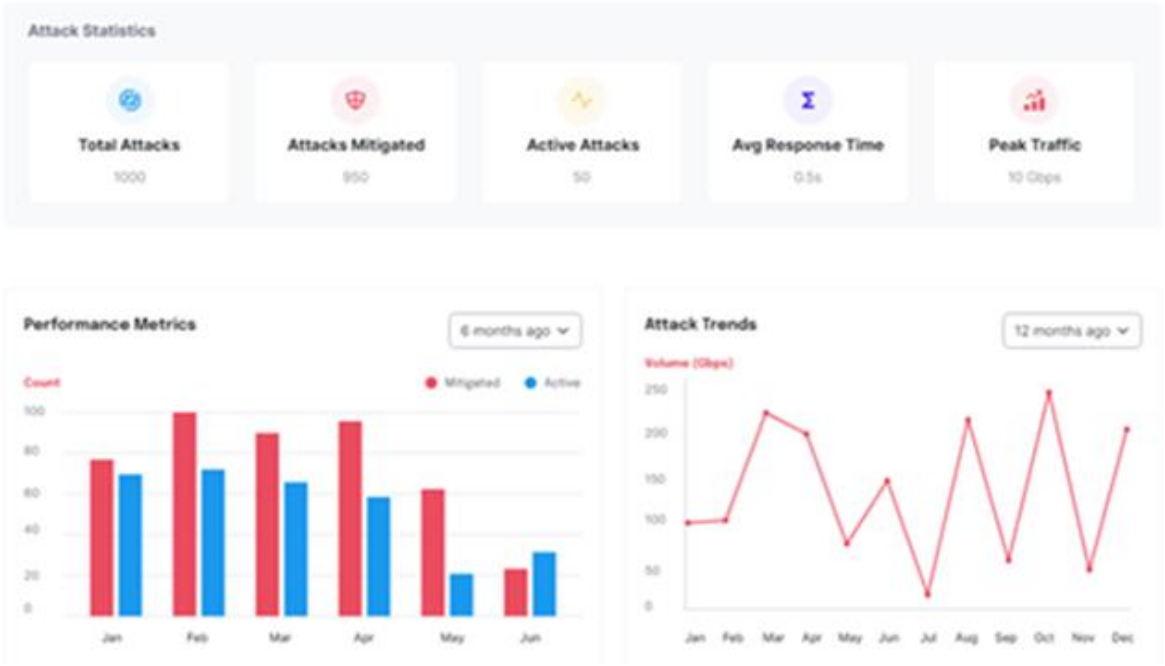

Figure 7: Analysis Summary

4. Logs and Alerts. This screen provides access to detailed logs of network activity, detected attacks, and mitigation actions. It also features real-time alerts to notify security personnel of ongoing threats, ensuring timely responses. This is shown in Figure 8.

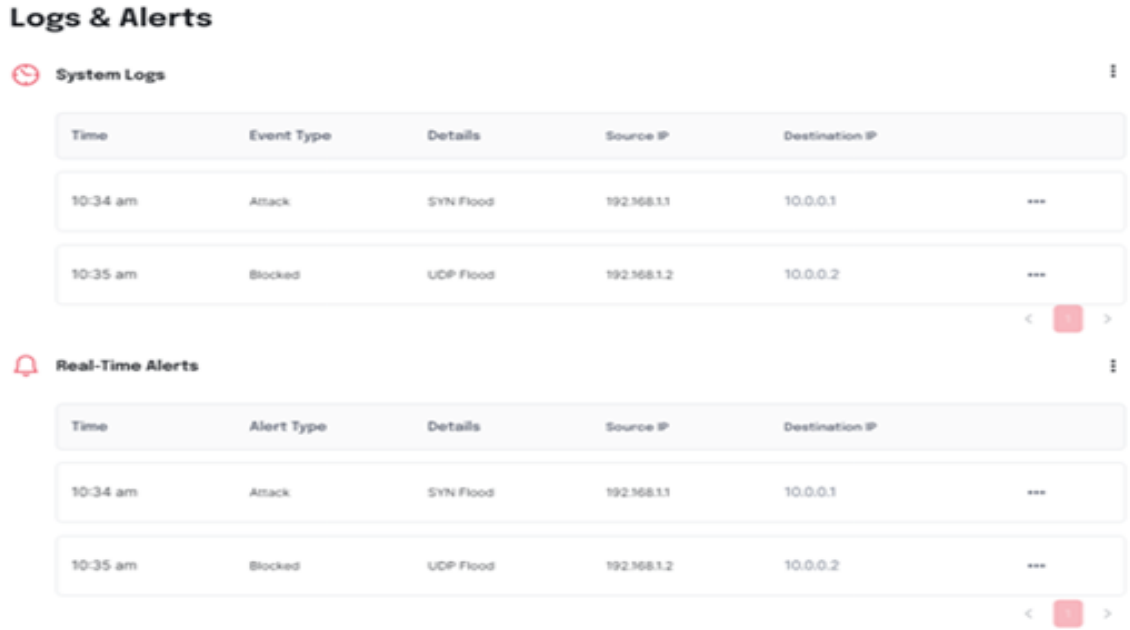

Figure 8: Logs and Alerts

5. Conclusion

The proposed DoS mitigation system offers a comprehensive and adaptable solution for protecting cloud services from denial-of-service attacks. By combining real-time analysis,

efficient filtering, and a user-centric design, the system addresses the limitations of traditional approaches and provides a more robust defense against evolving threats. The system's scalability reduced false positives, improved threat intelligence, and potential for integration further enhanced its value proposition for securing cloud environments.

The system's modular architecture ensures comprehensive protection through its traffic capture, analysis, filtering, logging, and user interface modules. Each component was carefully developed and integrated to provide seamless operation and user-friendly management.

The results of this study demonstrated the system's effectiveness in mitigating various DoS attack vectors, including volumetric, protocol, and application-layer attacks, ensuring the uninterrupted operation of cloud services. The user interface further enhances the system's usability, allowing security personnel to monitor and respond to threats efficiently. The achievements of this study highlight the importance of specialized DoS mitigation systems in maintaining the integrity and availability of cloud services.

References

Ali, M., Khan, S.U., & Vasilakos, A.V. (2015). Security in cloud computing: Opportunities and challenges. Inform. Sci. 305, 357–383. http://dx.doi.org/10.1016/j.ins.2015.01.025.

Bamasag, O., Alsaeedi, A., Munshi, A., Alghazzawi, D., Alshehri, S., and Jamjoom, A. (2022). Real-time ddos flood attack monitoring and detection (RT-AMD) model for cloud computing. PeerJ Comput. Sci. 7, e814. http://dx.doi.org/10.7717/peerj-cs.814.

Behal, S. and Kumar, K. (2017). Detection of DDoS attacks and flash events using novel information theory metrics, *Computer Networks*, 116(4), 96–110.

Cheng, J., Yin, J., Wu, C., Zhang, B. and Liu, Y. (2009). DDoS attack detection method based on linear prediction model, *Emerging Intelligent Computing Technology and Applications*,1004–1013, Springer, Berlin Heidelberg.

Deshmukh RV, Devadkar K. Understanding DDoS attack & its effect in cloud environment. *Procedia Comput Sci* 2015;49:202–10. doi: 10.1016/j.procs. 2015.04.245 .

Gu, Q. and Liu, P. (n.d.), Denial of Service Attacks, Accessed: 12-Jan-2025 https://s2.ist.psu.edu/paper/ddos-chap-gu-june-07.pdf

Jacques, S. and Christe, B. (2020), Information technology, In Introduction to Clinical Engineering, Academic Press, 109-126, https://doi.org/10.1016/B978-0-12-818103-4.00005-3.

Lundqvist, A. and landsberg, J. (n.d.). DoS Mitigation using Proof of Work. Accessed: 12-Jan-2025 https://www.csc.kth.se/utbildning/kth/kurser/DD143X/dkand12/Group5Mikael/final/Jonatan_Landsberg_and_Anton_Lundqvist.pdf

Ouhssini, M., Afdel, K., Agherrabi, E., Akouhar, M., and Abarda, A. (2024). DeepDefend: A comprehensive framework for DDoS attack detection and prevention in cloud computing, Journal of King Saud University - Computer and Information Sciences, 36.

Poonam Gupta, P. & Om Prabha, M.I. (2022). IoT in healthcare ecosystem, In Advances in Biomedical Information, Applications of Computational Intelligence in Multi-Disciplinary Research, Academic Press, 187-204, https://doi.org/10.1016/B978-0-12-823978-0.00003-4.

Jazi, H.H., Gonzalez, H., Stakhanova, N. and Ghorbani, A.A. (2017) Detecting http-based application layer DoS attacks on web servers in the presence of sampling, *Computer Networks,* 121, (7) 25–36.

Sachdeva, M., Kumar, K. and Singh, G. (2016) A comprehensive approach to discriminate DDoS attacks from flash events', *Journal of Information Security and Applications*, 26 (2), 8–22.

Sharma, A., Pujari, A.K. and Paliwal, K.K. (2007). Intrusion detection using text processing techniques with a kernel based similarity measure, Computers & Security, 26(7–8),488–495.

Shidaganti, G.I., Inamdar, A. S., Rai, S.V., & Rajeev, A.M. (2020). SCEF: A model for prevention of ddos attacks from the cloud. Int. J. Cloud Appl. Comput. 10 (3), 67–81. http://dx.doi.org/10.4018/IJCAC.2020070104.

Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A.A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection, Computers & Security, 31(3), 357-374, https://doi.org/10.1016/j.cose.2011.12.012.

Susnjara, S. and Smalley, I. (n.d.). What is cloud computing? Accessed: 10-Jan-2025 https://www.ibm.com/think/topics/cloud-computing#:~:text=Cloud%20computing%20is%20the%20on,pay%2Dper%2Duse%20pricing.

Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2015). Software-defined networking (SDN) and distributed denial of service (ddos) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE Comm. Surv. Tutor. http://dx.doi.org/10.1109/COMST.2015.2487361.

Zargar, S.T., Joshi, J. and Tipper, D. (2013).A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks, *IEEE Communications Surveys & Tutorials*,15(4), 2046–2069.