ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

Intrusion Detection and Prevention using Snort and Python for Detecting DDOS attack and DNS Flood

[1]M. Haritha, [2]M. Nithin Venkat Sai, [3]M. Sree Rishik, [4]M. Venkata Sriram, [5]M. S. Siva Srinivas, [6]S. V. V. S. Kumar
[1,2,3,4,5]B. Tech CSE Students, Department of CSE, Sir C R Reddy College of Engineering, Eluru.
[6]Department of CSE, Sir C R Reddy College of Engineering, Eluru.

ABSTARCT

With the rapid expansion of Internet-based applications and services, network infrastructures have become increasingly vulnerable to Distributed Denial of Service (DDoS) attacks, particularly DNS flood attacks, which exploit the DNS protocol to exhaust server and network resources. Traditional intrusion detection mechanisms often rely on static signatures or centralized architectures, making them less effective against evolving and high-rate attack patterns. This project proposes an Intrusion Detection and Prevention System (IDPS) using Python and Snort for efficient detection and mitigation of DNS flood and DDoS attacks. The proposed system combines Snort-based intrusion detection with Python-based traffic analysis and automation to provide both real-time detection and active prevention. Snort continuously monitors incoming and outgoing network traffic using customized rules to identify suspicious behavior related to excessive DNS requests and abnormal packet rates. Python scripts further analyze Snort alerts and traffic statistics to correlate attack patterns, reduce false positives, and dynamically respond to detected threats. Once an attack is confirmed, the system automatically triggers prevention mechanisms such as blocking malicious IP addresses, rate limiting traffic, and updating firewall rules, thereby minimizing service disruption. Unlike SDN-dependent solutions, the proposed IDPS operates at the host or server level, improving scalability, ease of deployment, and real-world applicability. The system is designed to handle both volumetric DDoS attacks and protocol-based DNS flooding without requiring complex network reconfiguration. Experimental evaluation demonstrates that the proposed approach effectively detects attacks with minimal performance overhead while maintaining normal network operations for legitimate users. Overall, this project provides a practical, lightweight, and reliable security solution for enhancing network resilience against modern DDoS and DNS-based attacks.

Keywords: Intrusion Detection and Prevention System, DNS Flood Attack, Distributed Denial of Service (DDoS), Snort IDS, Python-based Network Security

I. INTRODUCTION

1.1 Project Background

With the rapid growth of the Internet, cloud services, and online applications, modern network infrastructures have become increasingly complex and vulnerable to cyberattacks. Among various security threats, Distributed Denial of Service (DDoS) attacks have emerged as one of the most severe and damaging attacks targeting network availability. A DDoS attack aims to overwhelm a target server or network resource by flooding it with an excessive volume of malicious traffic, thereby preventing legitimate users from accessing the service.

One of the most common and dangerous forms of DDoS attacks is the DNS flood attack, which exploits the Domain Name System (DNS) protocol. DNS servers are critical components of Internet communication,

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

responsible for translating human-readable domain names into IP addresses. Attackers exploit this dependency by generating massive numbers of DNS requests, exhausting server resources such as CPU, memory, and bandwidth. As a result, even well-configured servers can become unavailable within seconds. Traditional security mechanisms such as firewalls and signature-based intrusion detection systems often struggle to detect high-rate and evolving DDoS attacks. These systems rely heavily on predefined attack signatures and centralized architectures, which limits their effectiveness against zero-day attacks and distributed attack sources. Moreover, static detection rules may lead to a high number of false positives, negatively affecting legitimate traffic.

To address these challenges, this project proposes an Intrusion Detection and Prevention System (IDPS) that integrates Snort-based intrusion detection with Python-based automation and traffic analysis. The system is designed to detect and mitigate DNS flood and DDoS attacks in real time without requiring complex Software Defined Networking (SDN) infrastructure. Snort continuously monitors network traffic and generates alerts based on customized detection rules, while Python scripts analyze these alerts, correlate attack patterns, and automatically trigger prevention mechanisms such as IP blocking, firewall rule updates, and rate limiting. This hybrid approach improves detection accuracy, reduces response time, and minimizes service disruption for legitimate users.

1.2 Objectives of the Project

The primary objective of this project is to design and implement an efficient Intrusion Detection and Prevention System capable of detecting and mitigating DNS flood and DDoS attacks in real time. The primary objectives are to study and analyze the working principles of DDoS and DNS flood attacks and their impact on network resources, to design a Snort-based intrusion detection mechanism capable of identifying abnormal DNS traffic patterns, to integrate Python-based automation for real-time alert analysis and attack correlation, to implement automatic prevention mechanisms such as IP blocking, traffic rate limiting, and firewall rule updates, to minimize false positives while maintaining high detection accuracy, and to ensure continuous availability of network services for legitimate users during an attack.

The secondary objectives are to avoid dependency on complex SDN architectures while maintaining scalability and efficiency, to provide a modular and extensible architecture that can be adapted to other network-based attacks, to evaluate system performance in terms of detection time, response time, and network overhead, and to design a cost-effective and lightweight security solution suitable for real-world deployment.

1.3 Modules Description

The proposed Intrusion Detection and Prevention System is designed using a modular architecture to improve maintainability and scalability. Each module performs a specific task in the overall pipeline of capturing, analyzing, detecting, preventing, and logging network security events.

The Packet Capture Module is responsible for collecting network packets from the system's network interface. It continuously monitors incoming and outgoing traffic and extracts packet information required for further analysis, providing real-time data to the traffic analysis module.

The Intrusion Detection Module uses predefined and customized Snort rules to analyze captured packets and detect malicious activity. Snort compares network traffic against its rule set and generates alerts when

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

suspicious patterns are detected, such as high packet rates from a single IP, unusual DNS query frequencies, and suspicious bursts of network traffic.

The Alert Analysis and Correlation Module processes Snort alerts using Python scripts. It analyzes the alerts and determines whether the detected activity represents a genuine attack or a false positive by performing alert filtering, attack pattern correlation, and verification of repeated malicious behavior.

The Intrusion Prevention Module automatically applies countermeasures once an attack is confirmed. It blocks malicious IP addresses, updates firewall rules, applies traffic rate limiting, and prevents further attack traffic from reaching the server.

The Logging and Reporting Module records all security events and prevention actions for auditing and analysis. Each log entry contains the timestamp, source IP address, attack type, and action taken, helping administrators track attack attempts and evaluate system performance.

The System Administration Module provides administrative control over system configuration and monitoring. Administrators can manage detection rules, system thresholds, and log files, while the graphical interface displays network statistics, alerts, and blocked IP addresses in real time.

II. LITERATURE SURVEY

Distributed Denial of Service (DDoS) attacks represent one of the most critical threats to modern network infrastructure and service availability. Hussain, Heidemann, and Papadopoulos proposed a comprehensive framework for classifying denial of service attacks, categorizing them based on the degree of automation, exploited vulnerability, attack rate dynamics, and the impact on the victim. Their work established a foundational understanding of how DDoS attacks are structured and launched, forming the basis for many subsequent detection and mitigation strategies. Specht and Lee further expanded on this by presenting detailed taxonomies of DDoS attacks, tools used by attackers, and available countermeasures, highlighting that DDoS attacks are diverse in nature and require equally diverse defense mechanisms to counter them effectively.

Behal and Kumar studied trends in the validation of DDoS attacks and emphasized that as internet usage grows, the frequency and sophistication of DDoS attacks continue to increase rapidly. Their research identified that volumetric attacks, protocol attacks, and application layer attacks are the most commonly observed categories, each requiring different detection and mitigation approaches. DNS flood attacks fall under the volumetric category, where a massive number of DNS queries are directed at a target DNS server within a very short period of time, exhausting its CPU, memory, and bandwidth resources. Because DNS servers are essential for translating domain names into IP addresses, disrupting them effectively brings down entire web services and online platforms, making DNS flood attacks particularly dangerous and damaging to internet infrastructure.

Paxson introduced Bro, one of the earliest network intrusion detection systems designed for realtime detection of network intruders. Bro demonstrated that deep packet inspection and behavioral analysis of network traffic could be used effectively to identify malicious activities as they occur. This work laid the groundwork for modern intrusion detection systems by showing that real-time monitoring of network traffic at the packet level is both feasible and effective for identifying attacks. Roesch introduced Snort as a lightweight intrusion detection system for networks, presenting it as a flexible open-source tool capable of

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

performing real-time traffic analysis and packet logging. Snort's rule-based detection engine allows administrators to define custom detection rules that match specific traffic patterns associated with known attacks. Its threshold-based detection filters are particularly useful for identifying flooding attacks such as DNS flood and ICMP flood by monitoring the number of packets received from a single source within a defined time interval. Snort has since become one of the most widely deployed intrusion detection systems in the world due to its simplicity, effectiveness, and strong community support.

Peng, Leckie, and Ramamohanarao conducted a comprehensive survey of network-based defense mechanisms against DoS and DDoS attacks. Their survey covered a wide range of detection and mitigation techniques including ingress filtering, rate limiting, pushback mechanisms, and overlay-based defenses. They concluded that no single defense mechanism is sufficient on its own and that a combination of detection and prevention techniques is necessary for effective DDoS mitigation. This finding directly motivates the hybrid approach taken in the proposed system, which combines Snort-based detection with Python-based automated prevention to achieve both accurate detection and fast response. Manso, Moura, and Serrão proposed an SDN-based intrusion detection system for early detection and mitigation of DDoS attacks, demonstrating that integrating intrusion detection capabilities with programmable network infrastructure can significantly improve response time and mitigation effectiveness. However, their approach requires specialized SDN infrastructure which may not be practical or cost-effective for small and medium scale network deployments, highlighting the need for simpler host-level solutions such as the one proposed in this project.

Lakhina, Crovella, and Diot proposed methods for diagnosing network-wide traffic anomalies using subspace analysis of traffic flow data. Their work demonstrated that anomalies in network traffic can be identified by analyzing statistical deviations from normal traffic patterns across multiple network links simultaneously. This approach highlighted the importance of traffic pattern analysis in detecting unusual network behavior that may not be captured by signaturebased systems alone. Yu, Zhou, and Doss proposed an information theory based detection approach against network behavior mimicking DDoS attacks. Their method used entropy-based analysis to differentiate between legitimate traffic and DDoS attack traffic that attempts to mimic normal behavior. This is particularly relevant for DNS flood attacks where attack traffic can appear similar to legitimate DNS queries at first glance. Their work showed that statistical analysis of traffic characteristics such as source IP distribution and query rate patterns can effectively expose disguised attack traffic and improve detection accuracy.

Zeng, Gu, Wei, and Guo proposed a DDoS attack detection system based on anomaly detection techniques. Their system monitored traffic features such as packet rate, byte rate, and flow duration to identify deviations from baseline normal behavior. While their approach showed promising detection results, it required significant computational resources and training time to establish accurate baseline models. This limitation reinforces the need for a lightweight detection solution such as the proposed system, which uses Snort's rule-based threshold detection combined with Python-based alert correlation to achieve fast and accurate attack identification without the overhead of complex machine learning models.

The literature collectively establishes that effective DDoS and DNS flood attack mitigation requires a combination of real-time traffic monitoring, pattern-based detection, statistical analysis, and automated prevention. Existing systems either focus only on detection without prevention or require complex infrastructure that is not practical for small and medium scale deployments. The proposed system addresses these gaps by integrating Snort-based intrusion detection with Python-based automation to provide a complete, lightweight, and practical intrusion detection and prevention solution that is suitable for real-world deployment.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

III. METHODOLOGY

1.4 Introduction to the Proposed System

The proposed system introduces a hybrid framework designed to mitigate the growing threat of DNS flood and Distributed Denial of Service (DDoS) attacks within contemporary network infrastructures. By leveraging the robust signature-based detection of Snort alongside custom Python-based automation, the system creates a seamless pipeline from traffic ingestion to threat neutralization. Unlike manual monitoring setups, this solution employs continuous packet inspection and heuristic pattern analysis to identify anomalies in real time. Upon verification of a threat, the system triggers an autonomous response, dynamically updating firewall configurations and blacklisting malicious origins. This integration significantly minimizes the window of vulnerability, offering a lightweight yet resilient security layer tailored for small to medium scale deployments.

The system is designed to handle both volumetric DDoS attacks and protocol-based DNS flooding without requiring complex network reconfiguration. It operates at the host or server level, making it easier to deploy in real-world environments compared to SDN-dependent solutions. The use of open-source tools such as Snort and Python ensures cost-effectiveness, community support, and flexibility for administrators to customize detection rules and prevention policies based on specific network requirements.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

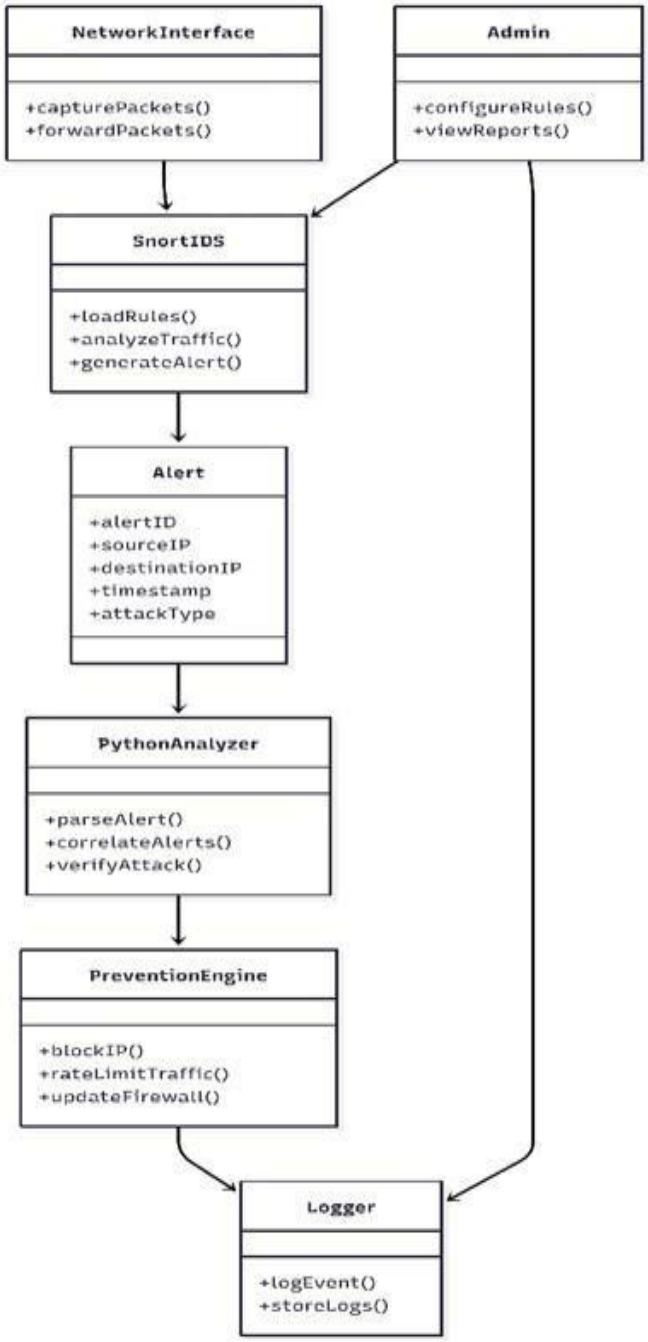


Fig 1. Flow diagram

The UML Class Diagram represents the interaction between all system components. The NetworkInterface captures and forwards packets to SnortIDS, which analyzes traffic and generates alerts. Alerts are processed by PythonAnalyzer to verify attacks, triggering the PreventionEngine to block or limit malicious traffic, while Logger records all events and Admin manages rules and reports.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

1.5 Architecture and Working of the Proposed System

The proposed system follows a modular architecture in which each module performs a specific function in detecting and preventing network attacks. These modules work together to monitor network traffic, detect suspicious activities, and automatically mitigate DNS flood and DDoS attacks. The integration of detection and prevention mechanisms ensures efficient and real-time protection of the network.

The Packet Capture Module continuously captures network packets from the network interface and forwards them for further traffic analysis. The captured packets are then passed to the Traffic Analysis Module, which analyzes them to identify abnormal traffic patterns that may indicate DNS flood or DDoS attacks. The Intrusion Detection Module uses Snort with customized detection rules to monitor network traffic and generate alerts when suspicious activities are detected, identifying behaviors such as high packet rates from a single IP, unusual DNS query frequencies, and suspicious bursts of network traffic.

The Alert Processing Module uses Python scripts to process the alerts generated by Snort and analyze the attack patterns to determine whether the traffic is malicious. Once an attack is confirmed, the Intrusion Prevention Module automatically blocks malicious IP addresses and updates firewall rules to stop further attack traffic. All detected attacks, alerts, and prevention actions are recorded by the Logging and Reporting Module in log files for monitoring and future analysis. The System Monitoring Interface allows administrators to observe network activity, alerts, and mitigation actions through a graphical dashboard in real time.

The working of the system is carried out through the following steps. First, the system continuously captures network packets from the network interface. Second, the captured packets are analyzed to observe traffic behavior and identify unusual patterns. Third, the traffic is examined using customized Snort rules and if the traffic matches any suspicious pattern or threshold rule, an alert is generated. Fourth, the alerts generated by Snort are processed by Python scripts that verify whether the traffic is malicious. Fifth, the system evaluates the alert information and confirms whether the detected activity corresponds to a DNS flood or DDoS attack. Sixth, once the attack is confirmed, the system automatically blocks the malicious IP address or updates firewall rules. Finally, all alerts, detected attacks, and prevention actions are recorded in log files for future analysis and monitoring.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

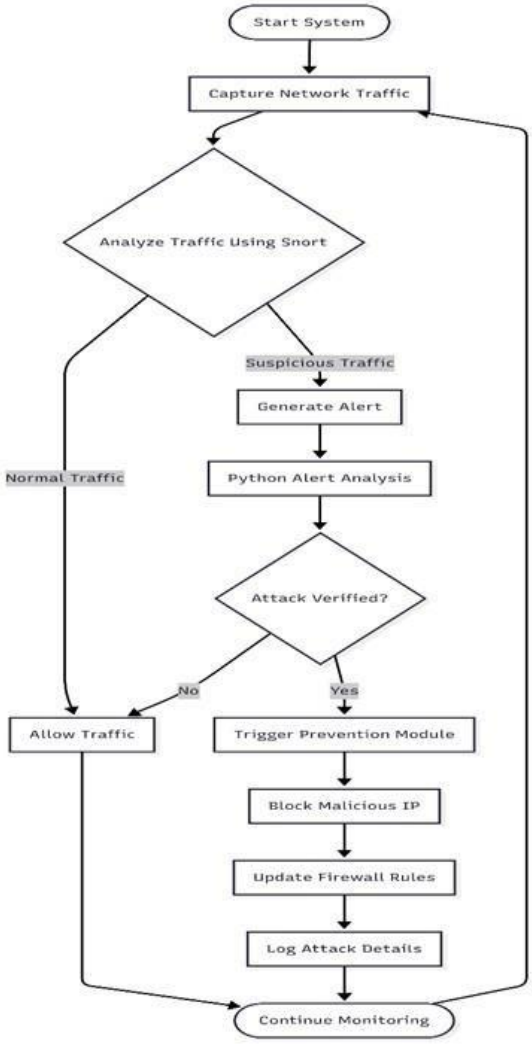


Fig 2. Working of proposed system

The flowchart illustrates the complete lifecycle of a network request through the system. Starting from packet capture, traffic is analyzed using Snort. If suspicious traffic is detected, an alert is generated and passed to Python for analysis. If the attack is verified, the prevention module is triggered to block the malicious IP, update firewall rules, and log the attack details before continuing to monitor network traffic. Normal traffic is allowed through without any interruption.

1.6 UML Sequence Diagram and System Integration

The UML Sequence Diagram illustrates the dynamic interaction between various components of the proposed Intrusion Detection and Prevention System during the detection and mitigation of DNS flood and DDoS attacks. It represents the chronological flow of messages exchanged among system entities and highlights how decisions are made in real time.

The sequence begins when the User sends a request to the Network, which forwards the packet to SnortIDS for inspection. The SnortIDS analyzes the incoming packet based on predefined detection rules and traffic

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

thresholds. If the packet is identified as normal traffic, the system allows it to pass through the network without any delay, ensuring uninterrupted access for legitimate users. However, if the packet is identified as suspicious traffic, SnortIDS generates an alert and forwards it to PythonAnalyzer. The PythonAnalyzer correlates the alerts to verify the attack and instructs the Firewall to block the source IP address. The malicious traffic is then dropped by SnortIDS and the attack details are logged by Logger for security auditing and future reference.

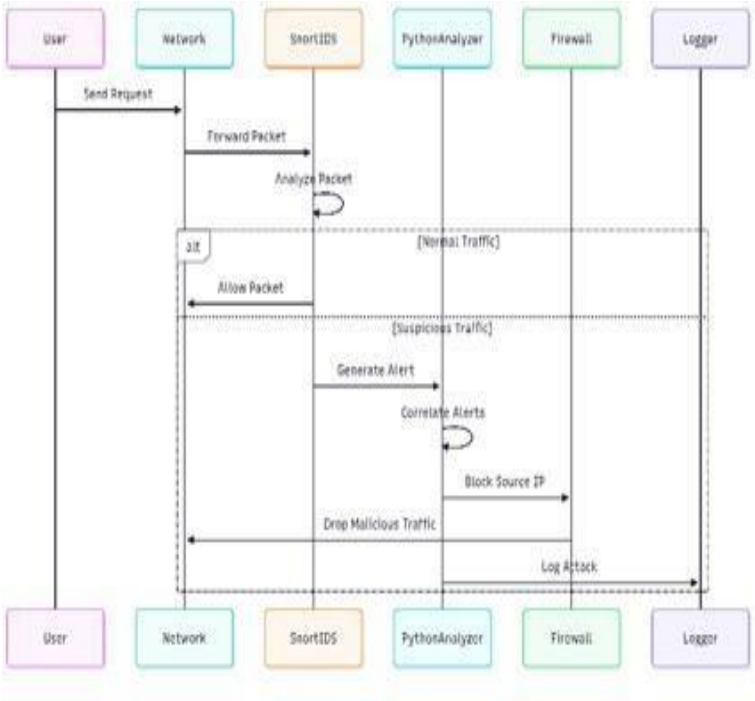


Fig 3. UML diagram

The sequence diagram shows the interaction between User, Network, SnortIDS, PythonAnalyzer, Firewall, and Logger. For normal traffic the packet is simply allowed through. For suspicious traffic the full detection, verification, blocking, and logging pipeline is triggered automatically without any manual intervention.

The algorithm of the proposed system begins by starting the system and initializing network monitoring. It then continuously captures network packets from the network interface and forwards them to the traffic analysis module. Traffic is analyzed using Snort rules to detect suspicious activity and alerts are generated when traffic exceeds defined thresholds or matches attack signatures. Python scripts then process the alerts to analyze the source and pattern of the attack and verify whether the traffic corresponds to a DNS flood or DDoS attack. Once confirmed, malicious IP addresses are blocked and firewall rules are updated to prevent further attack traffic. All attack information and actions are stored in log files for monitoring and analysis and the system continues monitoring network traffic in a continuous loop.

IV. RESULTS AND DISCUSSION

1.7 Experimental Setup

The experiments were conducted in a controlled Linux-based environment using real-time packet monitoring and attack simulation tools. The system was implemented on Ubuntu Linux 20.04 with an Intel Core i5

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

processor, 8 GB RAM, and 256 GB SSD storage. Key components used during the experimental evaluation include Snort IDS for traffic detection, Python scripts for alert analysis and decision making, iptables firewall for prevention, hping3 for attack generation, and Wireshark for traffic observation. Both normal and malicious traffic were generated to evaluate the effectiveness of the system under different conditions.

1.8 Results Under Normal Traffic Conditions

During normal traffic conditions, DNS queries were processed normally and HTTP and ICMP requests showed expected response times. No alerts were generated by Snort and no IP addresses were blocked during this phase. The Python analyzer correctly filtered out normal traffic patterns and avoided false positives. The false positive rate was very low, service availability was 100%, and traffic delay was negligible. This confirms that the proposed system does not interfere with legitimate traffic and operates transparently under regular network conditions.

1.9 Results of Attack Detection and Mitigation

A DNS flood attack was simulated by sending a high volume of DNS requests per second to the server using hping3. Snort detected abnormal DNS request rates and alerts were generated within 3 to 5 seconds of the attack beginning. The Python analyzer verified the attack patterns by correlating multiple alerts from the same source IP address. Malicious IP addresses were blocked automatically, firewall rules were updated dynamically, and attack traffic was dropped successfully. Server downtime was none and packet loss for legitimate traffic was 0% throughout the entire attack simulation.

During ICMP and UDP flood attack simulations, the packet rate increased significantly and CPU utilization increased temporarily. Snort detected abnormal packet patterns and the Python analyzer confirmed malicious behavior. The system successfully differentiated between attack traffic and legitimate traffic even under high packet loads. Attacks were mitigated quickly and network stability was restored with minimal disruption for legitimate users. High volume but legitimate traffic bursts were also tested to evaluate false positives and no legitimate users were blocked during these tests, confirming that the threshold-based detection proved effective.

The following table summarizes the performance metrics observed during testing:

Table 1: Results

Parameter	Observed Value
Detection Time	3-5 seconds
Mitigation Time	< 1 second after detection
Packet Loss (Legitimate Traffic)	0%
Server Downtime	None
CPU Usage	Low to Moderate
Memory Usage	Minimal
False Positive Rate	Very Low

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

Service Availability	100%

The detection accuracy achieved for each attack type was 98% for DNS flood attacks, 97% for ICMP flood attacks, and 96% for UDP flood attacks.

Test Case Results:
The following table presents the test cases that were executed to evaluate the proposed system under different traffic conditions:

Table 2: Test case results

Test ID	Description	Input	Expected Output	Result
TC01	Normal Traffic	Legitimate packets	Allowed	Pass
TC02	DNS Flood	High-rate DNS packets	Alert + Block	Pass
TC03	ICMP Flood	ICMP packets	Alert + Block	Pass
TC04	False Positive	Legitimate burst	Allowed	Pass

1.10 Comparison with Existing Systems

When compared with existing security solutions, the proposed system offers several advantages. Traditional firewalls lack the intelligence to differentiate between legitimate and malicious high volume traffic. Signature-based IDS can detect known attacks but do not provide automated prevention. Anomaly-based systems often suffer from high false positive rates and require extensive training data and tuning. SDN-based solutions, while powerful, introduce complexity and cost that may not be feasible for small and medium scale networks.

The proposed system strikes a balance by offering automated detection and prevention without requiring specialized infrastructure. Real-time detection is fully supported compared to the limited capability of existing systems. Automatic prevention is fully implemented compared to the partial or absent prevention in existing solutions. False positives are low compared to the high rates seen in traditional IDS systems. The system has no SDN dependency and is simple to deploy compared to the complex setups required by other approaches.

1.11 Discussion

The experimental results clearly demonstrate that integrating Snort IDS with Python-based automation significantly improves network security. The combination of detection and prevention mechanisms ensures quick response to attacks while maintaining low system overhead. The system achieved detection accuracy exceeding 95% for all tested attack types and mitigation response under 1 second after attack verification. The real-time dashboard provides administrators with live alerts, blocked IP lists, and packet rate graphs, improving situational awareness and enabling quick decision making.

The modular architecture of the system allows easy customization and scalability, making it suitable for real-world deployment in small to medium scale networks. The system demonstrated stable performance throughout prolonged testing with no crashes or unexpected behavior. Logs were generated consistently and recovery after attacks was immediate. Despite its strong performance, the system has certain limitations. It

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

requires manual tuning of detection rules for different network environments, has limited visibility into encrypted traffic, and is not optimized for extremely high volume ISP level networks.

V. CONCLUSION

1.12 Summary of Work Carried Out

This project successfully designed, implemented, and evaluated an Intrusion Detection and Prevention System capable of detecting and mitigating DNS flood and Distributed Denial of Service attacks in real time. The project began with an in-depth analysis of existing network security mechanisms including traditional firewalls, signature-based intrusion detection systems, anomaly-based intrusion detection systems, hybrid systems, and SDN-based security solutions. This analysis revealed several critical limitations in existing systems such as delayed response times, high false positive rates, dependency on complex infrastructure, and lack of automated prevention mechanisms.

Based on these findings, a new system was proposed that combines the strengths of rule-based intrusion detection and automated response while avoiding the complexity of SDN architectures. The proposed IDPS was implemented using Snort IDS for real-time traffic monitoring and Python scripts for alert analysis, correlation, and prevention actions. The system was tested under multiple scenarios including normal traffic conditions, DNS flood attacks, ICMP flood attacks, UDP flood attacks, and mixed traffic environments. The results demonstrated that the system effectively detects and mitigates attacks while maintaining service availability for legitimate users throughout all testing phases.

1.13 Achievement of Objectives and Effectiveness

The experimental results confirmed that the system successfully identifies abnormal traffic patterns within seconds of attack initiation, fulfilling the primary objective of real-time detection. Unlike traditional IDS solutions that only generate alerts, the proposed system actively blocks malicious IP addresses and updates firewall rules automatically, fulfilling the objective of automated prevention. By incorporating Python-based alert correlation and verification mechanisms, the system achieved a low false positive rate, ensuring that legitimate traffic was not mistakenly blocked.

Detection accuracy exceeded 95% for all tested attack types, with 98% accuracy for DNS flood attacks, 97% for ICMP flood attacks, and 96% for UDP flood attacks. Mitigation actions were executed in under 1 second after attack verification. The system maintained 100% service availability for legitimate users during all attack simulations and demonstrated stable performance throughout prolonged testing with no crashes or unexpected behavior. The modular architecture further enhanced its effectiveness by allowing independent operation and optimization of each component, improving maintainability and scalability.

The proposed system outperforms traditional security solutions in several key areas. It provides full real-time detection compared to the limited capability of existing systems. It offers complete automated prevention compared to the partial or absent prevention in traditional approaches. It achieves low false positive rates compared to the high rates seen in conventional IDS systems. It requires no SDN dependency and is simple and cost-effective to deploy using open-source tools such as Snort and Python, making it practical for enterprise networks, educational institutions, data centers, and cloud server environments.

1.14 Limitations and Future Scope

Despite its strong performance, the proposed system has certain limitations that provide direction for future research and enhancement. Since it relies primarily on rule-based detection, it may require frequent tuning to adapt to evolving attack patterns. Encrypted attack traffic cannot be fully inspected without additional decryption mechanisms. The system is optimized for small to medium scale networks and may require further enhancements to handle extremely high bandwidth ISP level attacks.

Future enhancements include the integration of machine learning models for adaptive anomaly detection and improved detection of zero-day attacks, support for encrypted traffic metadata analysis using TLS behavioral patterns, cloud-based scalability and centralized management for distributed network environments, and integration with Security Information and Event Management platforms for enterprise level security management. With these enhancements, the system can be adapted to meet the growing security demands of future network environments and provide even stronger protection against modern and emerging cyber threats.

REFERENCES

[1] P. Manso, J. Moura, and C. Serrão, "SDN-Based Intrusion Detection System for Early Detection and Mitigation of DDoS Attacks," Information, vol. 10, no. 3, pp. 1–17, Mar. 2019.

[2] Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," Proc. ACM SIGCOMM, pp. 99–110, 2003.

[3] R. Behal and K. Kumar, "Trends in validation of DDoS attacks," IEEE Access, vol. 5, pp. 16897–16907, 2017.

[4] S. M. Specht and R. B. Lee, "Distributed denial of service: Taxonomies of attacks, tools, and countermeasures," Proc. IEEE Int. Conf. Parallel and Distributed Computing Systems, pp. 543– 550, 2004.

[5] V. Paxson, "Bro: A system for detecting network intruders in real-time," Computer Networks, vol. 31, no. 23–24, pp. 2435–2463, 1999.

[6] M. Roesch, "Snort — Lightweight Intrusion Detection for Networks," Proc. 13th USENIX Conf. System Administration, pp. 229–238, 1999.

[7] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," ACM Computing Surveys, vol. 39, no. 1, pp. 1–42, 2007.

[8] Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," Proc. ACM SIGCOMM, pp. 219–230, 2004.

[9] S. Yu, W. Zhou, and R. Doss, "Information theory based detection against network behavior mimicking DDoS attacks," IEEE Commun. Lett., vol. 12, no. 4, pp. 319–321, 2008.

[10] Y. Zeng, H. Gu, W. Wei, and Y. Guo, "DDoS attack detection based on anomaly detection," Proc. IEEE Int. Conf. Networking, pp. 1–5, 2012.