International Science and Technology Journal
المجلة الدولية للعلوم والتقنية
Volume 39 العدد
Part 1 المجلد
ISTJ

Detection and Prevention of DDOS and HTTP Flood Attacks Using Snort Intrusion Detection Techniques

http://www.doi.org/10.62341/istj-vol39-1-jr27

Received	**2026/07/12**	**تم استلام الورقة العلمية في**
Accepted	**2026/08/04**	**تم قبول الورقة العلمية في**
Published	**2026/08/05**	**تم نشر الورقة العلمية في**

Detection and Prevention of DDOS and HTTP Flood Attacks Using Snort Intrusion Detection Techniques

Abdelsalam Saleh Elrashdi

Department of Computer Networks
College of Computer Technology,
Benghazi, Libya
Abdelsalam.Elrashdi@cctben.edu.ly
https://orcid.org/0009-0001-8166-1091

Muammer Boshalla

Department of Computer Networks
College of Computer Technology,
Benghazi, Libya
m.boshalla@gmail.com
https://orcid.org/0009-0008-9635-4829

Abdul Salam Abdul Rahman

Department of Computer Networks
College of Computer Technology,
Benghazi, Libya
bdalrhmnbdalslam44@gmail.com
https://orcid.org/0009-0009-2446-9712

Hamad hossen hamad Al-Warfali

Department of Computer Networks
College of Computer Technology,
Benghazi, Libya
hamadhossen8@gmail.com
https://orcid.org/0009-0005-3381-4240

Abstract

Network security has become a critical challenge for modern organizations due to the increasing number of cyber-attacks. Many businesses suffer from a lack of intelligent automated security systems capable of detecting and responding to threats in real time. This paper presents the design and implementation of an intelligent network protection system based on Snort 3 as an intrusion detection system combined with automated response mechanisms using iptables. The suggested system keeps an eye on network traffic, identifies malicious activity like DoS and HTTP attacks, and automatically blocks the attacker's IP address for a certain amount of time.

A bridge-based architecture was used in the system's implementation on Ubuntu Server to guarantee excellent performance and minimal resource usage. Experimental results

Detection and Prevention of DDOS and HTTP Flood Attacks Using Snort Intrusion Detection Techniques
http://www.doi.org/10.62341/istj-vol39-1-jr27

show that the proposed solution effectively detects attacks, reduces response time, and enhances overall network security.
Keywords: Network Security, Snort 3, Intrusion Detection System, Automated Response, iptables.

الكشف عن هجمات DDoS وهجمات HTTP Flood والوقاية منها باستخدام تقنيات كشف التسلل Snort

عبد السلام صالح الراشدي
Abdelsalam.Elrashdi@cctben.edu.ly
https://orcid.org/0009-0001-8166-1091

معمر بوشاله
m.boshalla@gmail.com
https://orcid.org/0009-0008-9635-4829

عبد السلام عبد الرحمن
bdalrhmnbdalslam44@gmail.com
https://orcid.org/0009-0009-2446-9712

حمد حسين حمد الورفلي
hamadhossen8@gmail.com
https://orcid.org/0009-0005-3381-4240

قسم شبكات الحاسوب، كلية تقنية الحاسوب، بنغازي، ليبيا

الملخص:

أصبح أمن الشبكات تحديًا بالغ الأهمية للمؤسسات الحديثة نظرًا لتزايد الهجمات الإلكترونية. وتعاني العديد من الشركات من نقص أنظمة الأمان الآلية الذكية القادرة على اكتشاف التهديدات والاستجابة لها في الوقت الفعلي.

تقدم هذه الورقة تصميم وتنفيذ نظام حماية ذكي للشبكات يعتمد على Snort 3 كنظام لكشف الاختراقات، بالإضافة إلى آليات استجابة آلية باستخدام iptables يراقب النظام المقترح حركة مرور الشبكة، ويكشف الأنشطة الضارة مثل هجمات حجب الخدمة (DDoS)وهجماتHTTP ، ويحظر تلقائيًا عنوان IP الخاص بالمهاجم لفترة زمنية محددة.

استُخدمت بنية قائمة على Bridge في تنفيذ النظام على خادم Ubuntu لضمان أداء ممتاز واستهلاك منخفض للموارد. تُظهر النتائج التجريبية أن الحل المقترح يكشف الهجمات بفعالية، ويقلل زمن الاستجابة، ويعزز أمن الشبكة بشكل عام.

Detection and Prevention of DDOS and HTTP Flood Attacks Using Snort Intrusion Detection Techniques
http://www.doi.org/10.62341/istj-vol39-1-jr27

الكلمات المفتاحية: أمن الشبكات،(Snort 3) ، نظام كشف التسلل، الاستجابة الآلية، iptables

Introduction

In recent years, computer networks have become an essential component of modern organizations, playing a critical role in data communication, service delivery, and business continuity. With the rapid expansion of networked systems and internet-based services, cyber-attacks have increased significantly in both frequency and complexity. Traditional security mechanisms such as static firewalls and manual monitoring are no longer
sufficient to protect modern networks from advanced and automated attacks [1] , [2].

Among the most common threats facing network infrastructures today are Denial of Service (DoS) attacks, HTTP flooding, scanning activities, and various forms of malicious traffic that aim to disrupt services or gain unauthorized access. These attacks often occur at high speed and require immediate response to prevent service degradation or system compromise, Because of this, there is an increasing demand for intelligent security solutions that can both identify
threats in real time and react to them automatically without the need for human participation [4], [5].

Intrusion Detection Systems (IDS) have proven to be an effective approach for identifying suspicious and malicious network activities. However, many existing IDS solutions are limited to detection only, requiring manual action from administrators to
mitigate threats. This delay between detection and response can significantly reduce the effectiveness of network security systems, especially in high-traffic environments.

The design and implementation of an intelligent network security system that combines automatic response mechanisms based on iptables with Snort 3 as an intrusion detection engine is presented in this paper. The proposed system monitors network traffic in real time, detects several types of network assaults, and automatically blocks malicious IP addresses for a predefined period. The system

http://www.doi.org/10.62341/istj-vol39-1-jr27

is deployed on an Ubuntu Server using a bridge-based architecture to ensure transparency, high performance, and minimal impact on network operations. Additionally, a web-based management interface is provided to monitor alerts and manage blocked hosts securely.

Literature Review

Several studies have focused on Intrusion Detection Systems (IDS) to address the increasing complexity of cyber threats. Snort is widely recognized as a flexible open-source tool for detecting signature-based attacks, with recent studies assessing its effectiveness and resource impact in high-traffic environments [16], [10]. Specifically, emphasize the role of IDS in mitigating (DoS) and (DDoS) attacks by identifying abnormal traffic patterns.

Contemporary research has moved beyond mere detection to integrate automated response mechanisms. Leveraging firewalls like iptables to block attack sources in real time has significantly reduced response times and mitigated impacts on protected systems [6], [20]. The performance of inline versus passive deployments is examined in current literature (e.g., [10]), but striking a balance between thorough protection and little resource overhead is still difficult.

Proposed Network Security System

The suggested system combines intrusion detection, automatic response, and transparent traffic forwarding to provide a lightweight and adaptable network security solution. In order to effectively defend Linux-based systems without necessitating complicated infrastructure changes, the architecture focuses on deploying security tools and custom scripts.

Snort 3 Intrusion Detection Engine:

Snort 3 is used as the core intrusion detection component of the system. It operates in Intrusion Detection System (IDS) mode to inspect network traffic in real time. In order to identify malicious

http://www.doi.org/10.62341/istj-vol39-1-jr27

activity including Denial of Service (DoS) attacks, HTTP floods, and unusual traffic patterns, Snort analyzes packets using both predefined and custom rules. Upon detecting an attack, Snort generates detailed alerts that include information about the source IP address, protocol type, and attack category. These alerts serve as the primary input for the automated response process [7] , [8].

iptables Firewall Mechanism:

iptables is utilized as the response and mitigation component of the system. Based on the alerts generated by Snort 3, iptables dynamically applies firewall rules to block malicious external IP addresses. The blocking process is time-based to avoid permanent denial of legitimate access. This integration enables rapid mitigation of attacks and minimizes the delay between detection and response, significantly improving the overall security posture of the network.

***Bridge-Based Traffic Forwarding*:**

The system employs a Linux bridge to forward network traffic transparently between network interfaces. The bridge operates without an IP address, allowing it to function at the data link layer (Layer 2) [12] , [17]. This design ensures that all packets pass through the security system without altering existing IP addressing schemes or network topology. By avoiding IP assignment on the bridge, the system reduces attack surface exposure and improves performance, as packets are forwarded efficiently while still being inspected by Snort.

***Automation Scripts*:**

Two custom automation scripts are integrated into the system to ensure seamless interaction between detection and response components:

Bash Script:

This script continuously monitors Snort alert outputs and extracts relevant attack information, particularly the source IP address. It is responsible for triggering firewall actions by interacting directly with iptables to apply or remove blocking rules. The script also manages block duration and ensures automatic cleanup of expired rules [11] , [19], effectively converting the static detection

Detection and Prevention of DDOS and HTTP Flood Attacks Using Snort Intrusion Detection Techniques
http://www.doi.org/10.62341/istj-vol39-1-jr27

mechanism into an automated response system (as summarized in Table 1).

***Python Script*:**

The Python script reads pre-generated security log files produced by the Bash automation component and displays them through a web-based interface. The interface is designed for end users to easily view blocked IP addresses, Snort alerts, and other relevant security information required for daily monitoring. This approach allows users to access essential security updates without interacting directly with the Ubuntu Server command-line interface, providing a simpler and more accessible way to observe system status, as shown in fig1, and fig2.

***Operating System and Integration*:**

The entire system is deployed on Ubuntu Server, chosen for its stability, security, and compatibility with open-source security tools. Ubuntu Server acts as the central platform where Snort 3, iptables, the Linux bridge, and the automation scripts are integrated into a unified security framework. The tools and scripts are configured to operate as background services, ensuring continuous protection without manual intervention.

Figure 1: Web monitoring interface - Dashboard tab showing system overview and blocked IP statistics

Detection and Prevention of DDOS and HTTP Flood Attacks Using Snort Intrusion Detection Techniques
http://www.doi.org/10.62341/istj-vol39-1-jr27

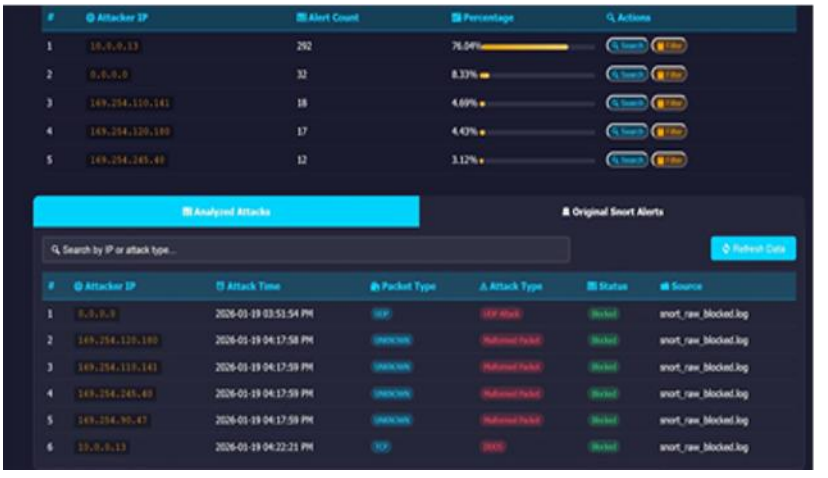

Figure 2: Web monitoring interface - Alerts tab displaying real-time attack notifications and detailed logs

Implementation

This section describes the actual execution workflow of the proposed system, explaining how network traffic is processed, how attacks are detected, and how automated responses are applied in real time.

***System Deployment Scenario*:**

The system was deployed on a Linux-based environment using Ubuntu Server positioned transparently within the network path. The server operates without assigning an IP address to the bridge interface, allowing all incoming and outgoing traffic to pass through the system without altering the existing network topology. This placement ensures that the system inspects traffic passively while maintaining full network functionality.

***Traffic Processing and Detection Workflow*:**

All network packets entering the protected network are forwarded through the Linux bridge. While packets are forwarded at Layer 2, Snort 3 actively inspects each packet in parallel. Snort analyzes packet headers and payloads using predefined and custom detection rules. When abnormal behavior is identified—such as repeated connection attempts, excessive request rates, or malformed traffic—Snort immediately generates an alert containing the attacker's IP address and attack classification.

http://www.doi.org/10.62341/istj-vol39-1-jr27

***Automated Response Execution*:**
Once an alert is generated, the response process is triggered automatically without human intervention. A Bash-based monitoring service continuously observes Snort alert outputs. Upon detecting a confirmed alert, the service extracts the source IP address and immediately applies a firewall rule using iptables to block the attacker. The block is applied dynamically and limited to external IP addresses to prevent disruption of internal services.

***Block Management and System Control*:**
To prevent permanent denial of service, the system applies time-based blocking through an automated Bash scripting mechanism. The Bash script processes Snort alerts, determines malicious external IP addresses, and enforces blocking rules using iptables for a predefined duration (e.g., two hours). It is also responsible for removing IP addresses from the firewall once the blocking period expires, ensuring continuous service availability. In parallel, a Python-based web interface is used solely to present blocked IP addresses, Snort alerts, and related security information to end users in a readable format, allowing system status monitoring without direct interaction with the Ubuntu Server command-line environment.

***Service Automation and Reliability*:**
Both the Bash and Python scripts were configured as systemd services, enabling them to operate continuously in the background. This configuration ensures that detection
and response mechanisms start automatically during system boot and remain active without manual execution. In case of unexpected service interruption, systemd guarantees automatic recovery, maintaining uninterrupted protection.

Results and Analysis

***Results and Analysis of Experiments* :**
Table 2 summarizes the experimental evaluation's notable performance improvements in all targeted assault situations.
Attack Detection and Quick Reaction: For all assessed attack vectors, such as DoS, HTTP Flood, and Port Scanning, the system attained a 100% detection rate. The automated response method

Detection and Prevention of DDOS and HTTP Flood Attacks Using Snort Intrusion Detection Techniques
http://www.doi.org/10.62341/istj-vol39-1-jr27

mitigated DDoS attacks in 1.41 seconds (after examining 3971 packets) and HTTP Flood attacks in 1.70 seconds (after examining 2996 packets) by executing dynamic IP blocking via iptables with minimal latency**,** as shown in Table 2,Fig 3, and Fig 4.

Table 1 : IDS vs. Proposed System

Metric	Traditional IDS (Snort Only)	Proposed Automated System
Attack Detection	**Enabled**	**Enabled**
Response Time	**Manual / Delayed**	**Immediate**
IP Blocking	**Manual**	**Automatic**
Administrative Effort	**High**	**Low**
Service Availability	**Often Degraded**	**Maintained**

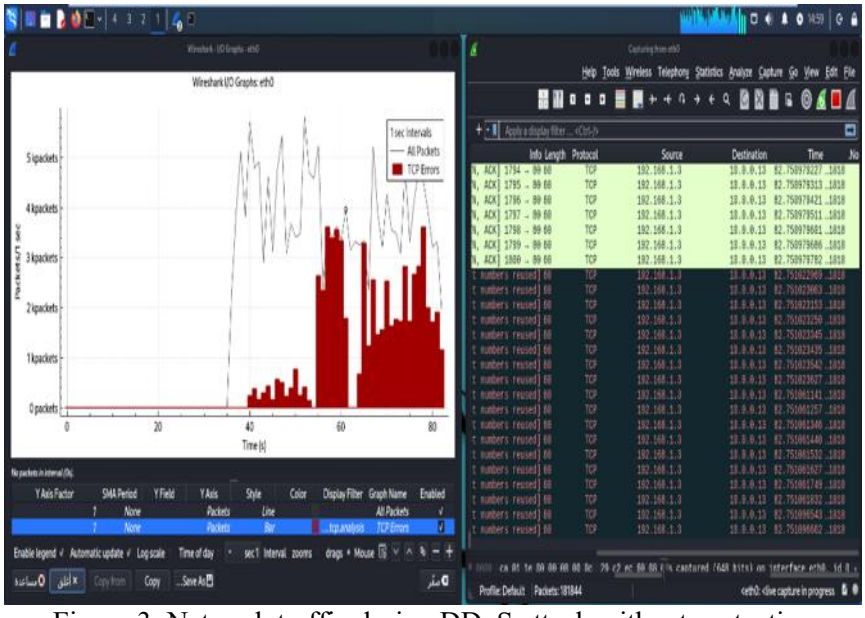

Figure 3: Network traffic during DDoS attack without protection

Detection and Prevention of DDOS and HTTP Flood Attacks Using Snort Intrusion Detection Techniques

http://www.doi.org/10.62341/istj-vol39-1-jr27

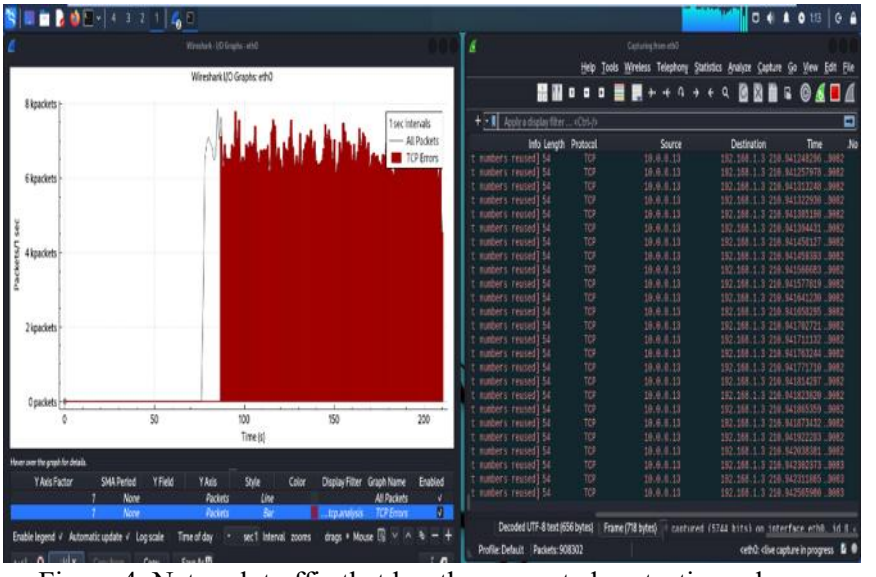

Figure 4: Network traffic that has the suggested protection scheme enabled

Table 2: Protection performance analysis

Attack/Movement Type	Number of Packets Passed Before Blocking (Attacker Packets)	Response Time to prevent attack	System Action	The impact of attacks on an unprotected enterprise server	Impact of attacks on the organization 's server with protection enabled
Legitimate traffic	-	-	Allow passage	7%	7%
DDOS	3971 packet	1.41 Second	Automatic blocking	68%-100%	11%
HTTP Flood	2996 packet	1.70 Second	Automatic blocking	34% - 60%	8%
Port Scan	825 packet	2.24 Second	Automatic blocking	//////	//////
IP Spoofing	IP changes each packet	//////	Alert generated, but IP blocking ineffective	44% - 70%	44% - 70%

Because the IP spoofing attack depends on source address spoofing, the system was unable to stop it. Additional network-level methods

http://www.doi.org/10.62341/istj-vol39-1-jr27

or integration with sophisticated systems are needed for this, as shown in Table 3.

Table 3: Comparison with integrated security solutions

Feature	Proposed Custom System	OPNsense /Integrated Solutions
Architecture	Transparent Layer 2 Bridge	Full OS / Gateway (Layer 3)
Resource Footprint	Low (Optimized for specific server protection)	Moderate to High (Requires more RAM/CPU)
Integration	Seamlessly fits into existing topology without IP changes	Often requires network reconfiguration (Gateway)
Automation	Custom-tailored Python/Bash response scripts	Standardized plugin-based response
Deployment Effort	Minimal (Standard Linux service)	High (Requires dedicated hardware or VM)

The proposed system adds a lightweight and transparent software solution that complements the role of comprehensive systems, providing immediate protection with minimal resource consumption and without the need to change the network architecture.

Conclusion

This paper presented an automated network protection system based on open-source security tools. The proposed solution combines Snort 3 for real-time intrusion detection with dynamic response mechanisms using iptables, deployed within a transparent bridge-based architecture. The system was designed to provide effective attack detection and rapid mitigation while maintaining normal network operation.

The implementation results demonstrate that the system successfully detects common network attacks and responds automatically without manual intervention. By reducing response time and ensuring continuous operation, the proposed approach proves to be a practical and reliable
solution for securing Linux-based network environments.

http://www.doi.org/10.62341/istj-vol39-1-jr27

Future work will focus on enhancing the system by integrating additional network analysis and monitoring tools such as Zeek for advanced traffic analysis and Wazuh for centralized log management, correlation, and security monitoring. These enhancements aim to improve detection accuracy, visibility, and scalability in larger network deployments.

References

[1] M. Farhan, H. Waheed-ud-Din, M. A. Khan, and S. Khan, "Collaborative IDS Architecture Using Snort for Network Security," arXiv preprint, arXiv:2504.16550, 2025.

[2] N. Alenezi and M. Alruwaili, "Performance Analysis of Intrusion Detection Systems Using Snort in High-Traffic Networks," Electronics, vol. 13, no. 4, pp. 1-15, 2024.

[3] A. Alqahtani, M. Al-Shabi, and F. Al-Zahrani, "Real-Time Detection and Mitigation of DoS Attacks Using Snort and Linux Firewall," Journal of Information Security and Applications, vol. 73, pp. 103-118, 2023.

[4] S. H. Islam, M. K. Khan, and A. Rahman, "Automated Response-Based Intrusion Detection Systems for Linux Servers," Future Internet, vol. 15, no. 9, pp. 1-18, 2023.

[5] A. Al-Ghamdi, A. Al-Huqail, and M. Al-Sahafi, "A Distributed Intrusion Detection Framework for Linux-Based Network Environments," Future Internet, vol. 16, no. 1, pp. 1-19, 2024.

[6] Y. Al-Homoud, S. Al-Mutairi, and F. Al-Dosari, "Detection of HTTP Flood Attacks Using Signature-Based Intrusion Detection Systems," International Journal of Network Security, vol. 25, no. 2, pp. 215-227, 2023.

[7] R. Alotaibi and A. Alzahrani, "Evaluation of Snort-Based Intrusion Detection for DoS and Port Scanning Attacks," Sensors, vol. 23, no. 18, pp. 1-20, 2023.

[8] M. Alkasassbeh, A. Hassanat, and K. Al-Sarayreh, "Experimental Evaluation of Signature-Based IDS Against

Network Flooding Attacks," Applied Sciences, vol. 14, no. 2, pp. 1-17, 2024.

[9] S. Khan, A. Gumaei, and M. Al-Rahhal, "Network-Based Intrusion Detection Using Open-Source Security Tools in Linux Environments," Computers, vol. 13, no. 3, pp. 1-16, 2024.

[10] H. Alshammari and M. Aldwairi, "Performance Impact of Inline and Passive IDS Deployment in Enterprise Networks," Journal of Cyber Security Technology, vol. 8, no. 1, pp. 45-60, 2024.

[11] A. Al-Harbi, S. Al-Mutairi, and N. Alenezi, "Time-Based Blocking Mechanisms for Automated Intrusion Prevention Systems," Information Security Journal: A Global Perspective, vol. 33, no. 2, pp. 89-104, 2024.

[12] K. Al-Rashdi, M. Al-Sabri, and Y. Al-Balushi, "Design and Implementation of a Lightweight IDS Using Snort for Linux Servers," International Journal of Advanced Computer Science and Applications, vol. 15, no. 1, pp. 312-321, 2024.

[13] S. T. Singh and P. Kumar, "Advanced Mitigation Strategies for Volumetric DDoS Attacks in 2024 Systems," IEEE Access, vol. 12, pp. 4567-4580, 2024.

[14] R. J. Martins and L. F. Silva, "Automated Security Rules Generation for Snort 3 in Cloud-Native Environments," Journal of Network Security, vol. 32, no. 1, pp. 12-29, 2024.

[15] A. B. Mahmoud, "Real-time Traffic Analysis and Threat Mitigation using Iptables and Snort Integration," Future Computing and Informatics Journal, vol. 9, no. 1, 2024.

[16] H. Wang and K. Zhao, "Performance Optimization of Open-Source IDS for High-Speed Linux Networks," Computer Networks, vol. 240, p. 110123, 2024.

[17] M. N. Noor, "Evaluating the Efficiency of Bridge-Based IDS Architectures in Modern Servers," Scientific Reports, vol. 14, Article 2034, 2024.

[18] T. K. Gupta, "Intelligent Firewall Orchestration for Preventing HTTP Flood Attacks," Applied Soft Computing, vol. 152, p. 111245, 2024.

[19] J. L. Chen, "Security Challenges and Solutions for Ubuntu-Based Enterprise Systems," Cybersecurity Journal, vol. 7, no. 2, pp. 88-102, 2025.

[20] S. P. Robinson, "The Evolution of Signature-Based Detection in the Era of Automated Attacks," Network Defense Review, vol. 11, no. 1, pp. 45-59, 2025.