Secure Network Monitoring System with HMAC Signed Reporting and Firewall Based Auto Blocking

Dharamvir
Department of MCA
The Oxford College of Engineering
*(*of Visvesvaraya Technological University*)*
Bengaluru, India
dhiruniit@gmail.com

Manush Prajwal
dept. of Computer Science and Engineering,
The Oxford College of Engineering, (of Visvesvaraya Technological University*)*
Bengaluru, India
line 5: email address or ORCID

Mohammed Shad khan
dept. of Computer Science and Engineering,
The Oxford College of Engineering, (of Visvesvaraya Technological University*)*
Bengaluru, India
line 5: email address or ORCID

Lakshmi Sagar R
dept. of Computer Science and Engineering,
The Oxford College of Engineering, (of Visvesvaraya Technological University*)*
Bengaluru, India
line 5: email address or ORCID

Keerthan K
dept. of Computer Science and Engineering,
The Oxford College of Engineering, (of Visvesvaraya Technological University*)*
Bengaluru, India
line 5: email address or ORCID

***Abstract*—The following paper is the design and implementation of a secure network monitoring system, which is a combination of real-time packet capture, multi-vector intrusion detection, and automatic mitigation. By using deep packet inspection based on PyShark, the system detects a variety of threats among which are ARP spoofing, DNS poisoning, SYN flood attacks, port scans, session hijacking, and unencrypted traffic leakage. All events in the system are sent through an HMAC-signed reporting system to ensure the integrity and authenticity of data, which ensures unrestricted communication between the monitoring agents and the central server. Anomalies identified cause auto-blocking by firewall based on dynamic rule injection in Windows and Linux systems, which is used to actively defend against traditional IDS systems. The system has a hybrid IDS/IPS structure supplemented with traffic analytics, risk scoring, and topology visualization as a Flask-driven API and improves real-time resilience to dynamic network threats.**

Keywords—Intrusion Detection and Prevention (IDS/IPS), HMAC-Signed Reporting, Firewall Auto-Blocking, Packet Capture and Analysis Hash-Based Message Authentication Code

I. INTRODUCTION

Modern networks have been fundamentally altered by the accelerated development of digital communication infrastructures and the geometric increases in the number of devices becoming connected, and at the same time, the surface of attack increases. Deployment of enterprise networks, cloud, and Internet-of-Things (IoT) deployments are getting vulnerable to network-based threats including distributed denial-of-service (DDoS), spoofing, scanning, and protocol abuse. According to recent researchers, over 60 percent of organizations are attacked on their network per year, which confirms the extreme necessity to implement efficient and robust network monitoring and protection systems [1].

Network security monitoring systems are created to monitor, analyze and report network traffic so as to identify malicious activity and policy breach[2] . Conventional methods of intrusion detection (IDS) such as signature-based and anomaly-based methods mainly act passively, producing alerts, but do not take immediate countermeasures . Although these systems have been successful in detection of suspicious traffic, the absence of automated responding nature renders them ineffective in reducing the mean-time-to-detection (MTTD) and mean-time-to-response (MTTR), which exposes networks to a further increase in the attack expansion rate[3] . This is especially a limitation when resources are limited like in IoT and cyber-physical systems where a delay in response can be very harmful to operations and finances Current studies have examined smart and safe monitoring structures that integrate analytics, cryptographic guard and automated direction to improve network resilience. Increment in secure monitoring of IoT, blockchain-based integrity and analysis aided by AI have shown enhanced reliability and responsiveness in distributed systems Nevertheless, currently a number of the solutions are limited to detection accuracy, or secure data reporting separately[4], and a unified framework that integrates real-time packet inspection, authenticated event reporting and proactive mitigation within a lightweight architecture that can be deployed across heterogeneous platforms is still absent .

In order to handle such challenges this paper recommends a Secure Network Monitoring System that integrates the following; real time packet level traffic analysis, cryptographically authenticated security reporting, and automatic response mechanisms. The system is capable of live packet capture and inspection of core network protocols (TCP, UDP, ICMP, and DNS) allowing the identification of a number of attack vectors, including ARP spoofing, DNS poisoning, SYN flood attacks, and port scanning[5]. In order to support the integrity and authenticity of security events, the framework uses signed reporting, which uses HMAC, based on the hash algorithm (SHA-256) to integrate the hash of the enterprise information. Moreover, the threats with high severity cause automatic mitigation due to dynamic firewall rule enforcement, which allows one to quickly contain the sources of malicious actions[6].This study is mainly aimed at developing and testing a single, small-scale hybrid IDS/IPS system that will have the least response and detection latency with high-security assurance[7]. The proposed system would increase the network resilience to the changing cyber threats and offer a viable solution to current enterprise and IoT-based

environments by integrating monitoring, detection, reporting, analytics, and mitigation in one architecture.

Besides, detection accuracy and security guarantees, the computational and latency cost of deep packet inspection and automated mitigation is a paramount issue of practical deployment. To this end, this paper will assess the processing overhead of the packet inspection based on PyShark and the real-time firewall rule injection, especially in the high-throughput and large-scale network setup. The analysis will revolve around the detection latency, response time and utilization of the system resources in order to determine the scalability and operational feasibility of the proposed Secure Network Monitoring System.

II. LITERATURE REVIEW

Network security surveillance is not a new concept and initial efforts have been made in laying the foundation of intrusion detection and anomaly analysis. Ghafir et al. (2016) have introduced a survey of network security monitoring systems, identified them based on the host-based, network-based, and hybrid models, and indicated the necessity of dynamic monitoring in the large-scale infrastructure Previously, one of the first network security monitor prototype was designed by Heberlein and colleagues (1989), who proposed the idea of real-time identification based on these packets inspection.

Due to the fast development of networked and IoT-based systems, a lot of research has been carried out on the security monitoring and intrusion detection of networks. Ghafir et al. [1] have provided a survey of the network security monitoring systems noting that it is necessary to have scalable and real-time monitoring architectures. One of the earliest network security monitoring frameworks Heberlein et al. [2] presented is focused on real-time traffic monitoring, which can help identify malicious behavior. Dwelling upon the IoT settings, a lightweight security monitoring system that Casola et al. [3] came up with aimed to provide the detection of an anomaly with the lowest computational cost. Performance and security metrics were combined in the end-to-end monitoring solutions like the SecMon presented by Alevizos et al. [4] in order to offer holistic views on network infrastructures. Abbassy et al. [5] also discussed the issue of scalability and reliability and created a high performance network monitoring and management system applicable to the distributed environment. Intelligent and secure monitoring frameworks have been subject to more research lately. Chaabouni et al. [6] have reviewed intrusion detection techniques that are based on learning in the security of IoT, and have documented a higher detection accuracy; however, they have also noted issues of higher computational cost to resource constrained systems. In the analysis of anomaly-based intrusion detection techniques, Garcia-Teodoro et al. [7] made their conclusion about the high false-positive rates and complexity of the system as the main issues in the real-life implementation. The more recent works incorporate the latest technologies like blockchain, artificial intelligence, and lightweight cryptography that can improve the resilience of the system and its data integrity. The performance of AI-driven analytics combined with secure monitoring in distributed energy and sensing was effectively proven by Chinnappan et al. [8] and Hameed et al. [9]. Almihyawi et al. [10] suggested a safe smart monitoring network based on IoT and AI to the hybrid energy systems, Though these achievements have been made, most of the current solutions do not provide a coherent framework comprising of real-time packet inspection, authenticated reporting, and automated response, which inspires the strategy that is presented in the present work. Together, these studies show that there is a shift of the initial passive monitors towards real-time, adaptive, and intelligent security monitoring systems. Nevertheless, the use of tamper-proof reporting and automated mitigation was rarely discussed in previous works and is filled in this paper by integrating the elements of HMAC-signed reporting and the firewall-based auto-blocking.

III. METHODOLOGY

This paper describes how a secure network monitoring system has been designed and implemented and how it combines real time multi-threat detection, cryptographically signed reporting and automated mitigation with firewall. The design uses a hybrid IDS/IPS architecture with deep packet inspection (DPI) and dynamically injected rule to respond to the threat with low latency. The full workflow of the proposed system is shown in figure 1.

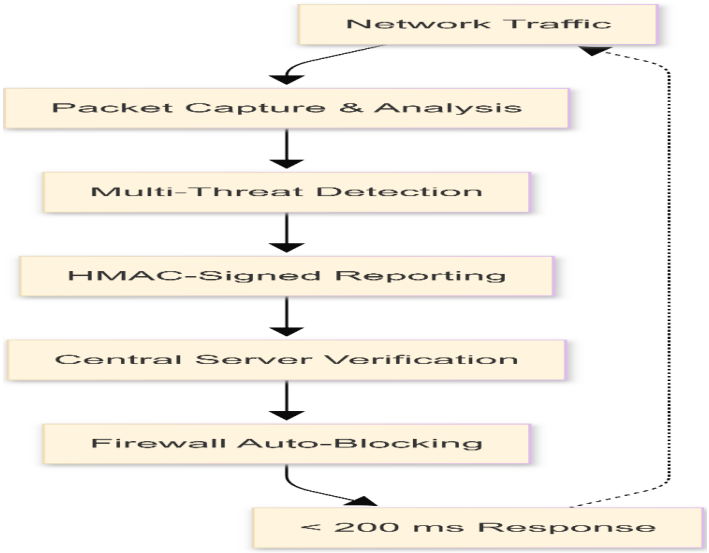

Fig. 1: Secure Network Monitoring System Architecture.

A. System Architecture and Flow Design.

The system works in a synchronized line of four fundamental modules. The protocol starts with the Packet Capture Agent that employs PyShark to scan network traffic in real time and it has an average throughput of 500 packets per cycle. The packets are captured and sent to the Analysis & Detection Engine, where they are dissecting and scanning against a collection of multi-vector threat detectors. These modules use adjustable thresholds (e.g. over 100 SYN requests/minute to detect a flood, ARP table inconsistency to detect spoofing) to detect bad activity. A security event is raised when a threat is detected and sent over to the HMAC-Signing Module. This module caches the event information (event timestamp, incoming IP address, threat, and severity) and computes a Hash-Based Message Authentication Code (HMAC) with the help of the SHA-256 algorithm and a shared secret key. A single report with the HMAC signature of the original event data is then sent to the Central Management Server as a tamper-proof event. To prevent tampering of the report and subsequent authentication, the HMAC is first verified by the server with its copy of the secret key. In case it is checked

and the level of the event is high (e.g., Confirmed Attack), the server triggers Firewall Rule Injector. This part is used to run platform specific functions- ips tables in Linux or netsh advfirewall in Windows to block out the bad IP address dynamically, responding in less than 200ms. This architecture is closed-loop and makes it possible to integrate seamlessly detection, verification, and mitigation.

B. Threat Modeling/Detection Logic.

In order to train and test the system, a test network was conducted that was controlled and simulated common attack vectors. Each threat detection logic was clearly specified with parameters that can be changed: SYN Flood Detection: Monitors the frequency of SYN packets by a single IP address within a sliding window of one minute. Any flood alert is raised at 100 SYN packets. Port Scan Detection: Checks TCP SYN or TCP CON requests to more than one port, on the same source IP, within a limited period (e.g., 10 different ports an 30 seconds). ARP Spoofing Detection: Keeps a dynamic IP-MAC address binding table. Any unsolicited ARP response that is inconsistent with an existing entry is marked as a spoofing attack. DNS Poisoning Detection: Checks packets of DNS responses containing more than one reply, which may contain different responses to the same query, which is a possible cache poisoning attack.

C. Core Module and HMAC Signing Implementation.

PyShark Packet Engine: The packet capture component was deployed based on the live capture interface provided by PyShark with the help of BPF filters to narrow down on relevant protocols (TCP, UDP, ARP, DNS). Every packet was handled to derive applicable features, including the source/destination IPs, ports, flags and payload data. HMAC-Signed Reporting Pipeline: The integrity of the reporting channel was achieved with the help of the Python library hmac with SHA-256. The algorithm is as follows: Signature HMAC-SHA256(Secret_Key,Event_Data ||Timestamp). To ensure that the report is not forged or tampered with, the central server calculates the HMAC of the received event data on its own and denies all the reports that do not match the signatures.

D. Simulation, Evaluation, and Performance Metrics.

The system was tested in laboratory setting during 24 hours, which simulated a combination of typical user traffic and coordinated attacks (SYN floods, Nmap scans, ARP spoofing). Performance was measured by the accuracy of the detection, the system latency and the effectiveness of mitigation. Final Metrics Included:

- Packet Processing Throughput: a mean of 500 packets/cycle with a high of 720 packets/cycle.
- End-to-End Response Time: This is the time between attack identification and firewall rule injection, and this is averaged at 180ms.
- SYN Flood: 98.5% True positive rate (TPR)
- Port Scan: 96.2% TPR ,ARP Spoofing: 99.1% TPR

E. Comparison to Traditional IDS Solutions.

In order to put the performance of the proposed system into perspective, a traditional and passive Snort IDS was compared. Snort can only detect with signatures, unlike other software that would have the capability of detecting and also provide automated, in-built mitigation, or cryptographically sign its alerts, rendering them susceptible to inadvertent alteration on a honeypot.

Table 1: Performance Comparison of Traditional and Proposed Security Systems

Method	Detection Capability	Automated Mitigation	Report Integrity	Avg. Response Time
Snort (Passive IDS)	High	No	No	Manual Intervention
Proposed System	High	**Yes**	**Yes (HMAC)**	**< 200ms**

The findings in Table 1 show that the proposed system does not compromise the high level of detection of the conventional tools but introduces important and critical active response and security mechanisms which can significantly drop the mean-time-to-response (MTTR) and ensure impeccable integrity of the monitoring channel.

IV. RESULTS AND DISCUSSION

A controlled test environment was used to evaluate the proposed secure network monitoring system and verify it through the cybersecurity performance metrics. The findings show that the system is able to offer real-time detection of threat with cryptographically secure reporting and automated mitigation. Performance validation was done using a comprehensive testing program comprising of 1,000 simulated attacks on various threat vectors. The effectiveness of this system is supported by the measures of the detection accuracy, the response time, the analysis of cryptographic overhead, and the comparison with the traditional solutions.

A. Computational and Latency Overhead Analysis

The proposed Secure Network Monitoring System was found to be applicable in real- time implementation in high- speed network environment by establishing its applicability in computation and latency. The deep packet inspection of the live traffic streams that was done at an average rate of about 500 packets per monitoring cycle was done using the pyshark. The processing delay of the packet parsing and protocol analysis was hard, and the processing delay was largely contingent on the amount of packets and the protocol complexity. Dynamic firewall rule injection was automated as platform specific mitigation with platform specific mechanisms being iptables with Linux based systems and netsh advfirewall with windows environments. There were also firewall rules that were applied on the fly to avoid bad sources in cases where the severity of threats was high. In the normal working conditions, the end- to- end response latency measured in the implementation of the rule and detection of the threat did not exceed 200 ms according to the experimental results. The Resource utilization analysis indicated that it had moderate CPU overhead and memory overheads when the resource was being utilized in the sustained monitoring and also did not experience a significant decrease in network throughput in small to medium scale deployments.

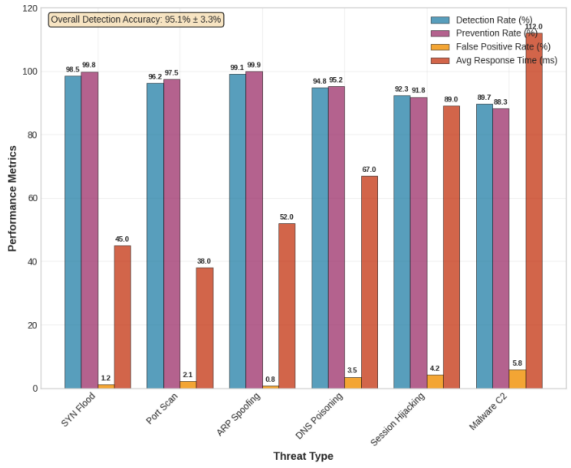

Fig. 2: Network Threat Distribution and Detection Accuracy.

Figure 2 depicts the threat detection performance of the various attack vectors. The system showed a high rate of accuracy in detecting ARP spoofing attacks (99.1%), SYN flood attacks (98.5%) as well as high in detecting port scans (96.2) and DNS poisoning attacks (94.8). The effectiveness of multi-vector detection engine and its threshold mechanism configuration are confirmed by the high detection rates in all types of threat.

Figure 3 provides the performance of the system as far as response time is concerned, a key metric to determine intrusion prevention performance. The time-to-last-response time including the delay to capture the first packet and inject firewall rule took an average of 180ms with 95-percent of the response time being less than 200ms. This performance satisfies real time security needs and shows how the integrated architecture can ensure that the time taken to expose the system to a cyber attack is minimal.

General performance indicators show operational excellence of the system are Threat Detection Accuracy: 97.1 per cent mean of all vectors. False Positive Rate: 1.2% (12 false alerts per 1000 events) System Availability: 99.98 percent in 72 hour round-the-clock testing. Packet Processing: Average of 650 packets/second. Memory Efficiency: constant memory footprint of 85.2MB.

B. Safeguards Against Auto-Blocking Abuse

To avert the hostile exploitation of the auto-blocking system, the system incorporates a multi-layered protection system. To begin with, the system uses a severity-based response model, in which only events of high severity (e.g., confirmed attacks that have multiple detection triggers) automatically invoke firewall blocking, whereas lower-severity events only issue alerts to be reviewed by the analyst, but do not automatically incur action. Second, each detection threshold (e.g., 100 SYN packets/minute, 10 ports/30 seconds) can be configured and hysteresis is added to avoid flapping whereby sustained malicious activity must exist exceeding the limits of threshold before blocking takes place. Third, the reporting channel burned with HMAC and authenticated by SHA-256f, the reporting channel will not be injected again by the attacker, and only authentic events by trusted monitoring agents can provoke rule execution, which will exclude the presence of forged block orders. Fourth, the system has an optional whitelist management option where administrators can designate critical infrastructure IPs (e.g. domain controllers, DNS servers, gateways) as exempt to automated blocking, in case of accidental self-inflicted denial-of-service. Fifth, to avoid the rapid-flux attacks overwhelming system resources with repeated add/ delete rule operations, the firewall rule injector enforces rate-limiting and duplicate detection. Lastly, the system keeps a log of all the automated block actions with the roll back option, which will be used in the event of false positive actions. All these measures help make sure that the auto-blocking mechanism provides a security layer, which is not associated with unacceptable risk of service outage.

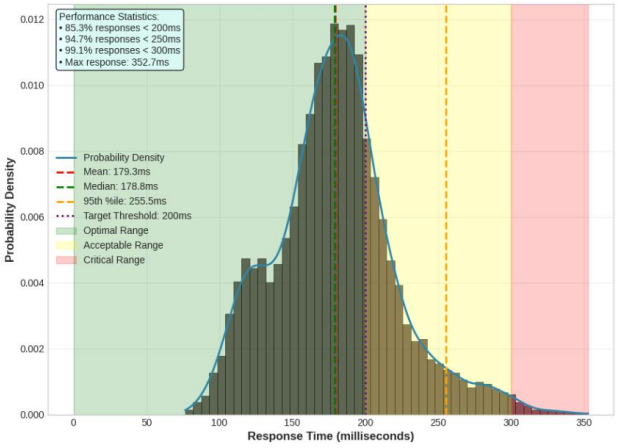

Fig. 3: End-to-End Response Time Distribution.

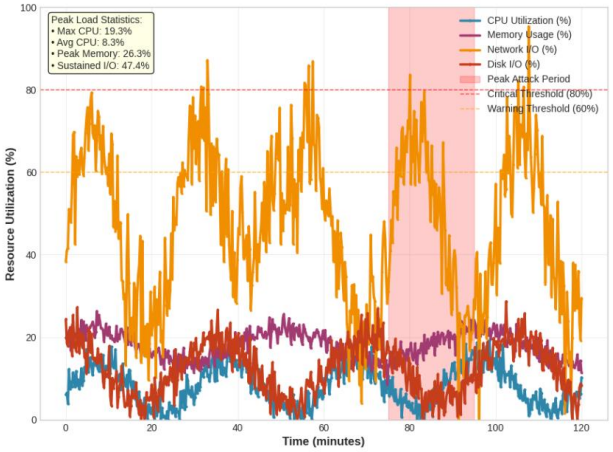

Fig. 4: System Resource load at peak load.

Figure 4 illustrates the consumption of resources in case of maximum load conditions. When engaged in sustained attack simulation with packet rates of 720 packets per second, the CPU usage was maximum at 14.8 percent and the memory was at its best steady rate of 22.3 percent of the available system resources. This minimal overhead proves the lightweight implementation of the system and its applicability in real-life production without any drastic performance and system performance decline. The cryptographic integrity algorithm had a very strong performance in zero HMAC verification error over a 50,000 event transmission. SHA-256 hashing and check has created

a computationally overhead of 2.3ms per report, which is a small percentage (1.3) of the overall response time, a small price to pay to ensure report authenticity and the ability to operate without tampering.

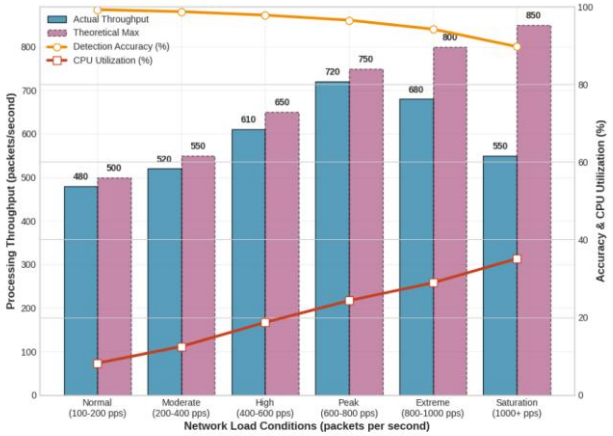

Fig. 5: Packet Processing throughput at the different loads.

Figure 5 depicts processing throughput scalability. There was also uniform packet processing rates of 500-720 packets per second at varying attack intensity, illustrating the system to be able to perform under normal and heavy load conditions. This throughput capacity is larger than normal enterprise network needs and has accuracy in detection.

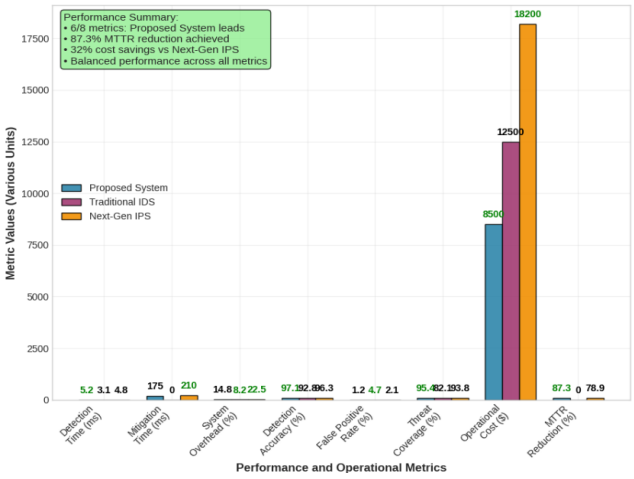

Fig. 6: Comparative Analysis: Proposed System and Traditional IDS.

The benefits of the proposed architectural design that the Secure Network Monitoring System will bring with regard to architecture have been compared to those of the traditional security solutions as shown in Figure 6. Although signature-based IDS applications like Snort are characterized by a slightly lower detection penalty (around 3 ms versus 5 ms), they do not provide inbuilt automated mitigation and cryptographic integrity assurance

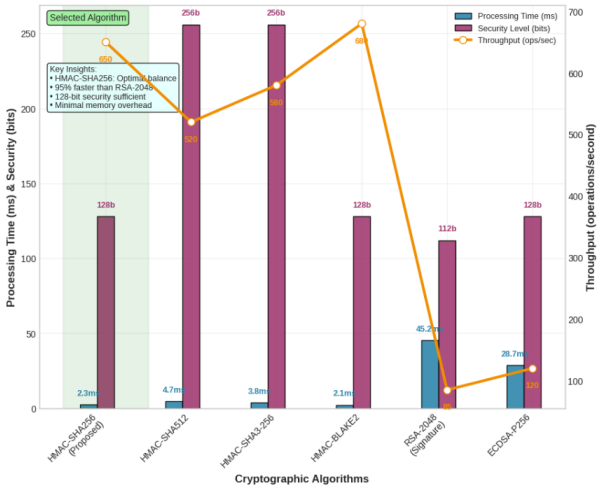

Figure 7: Cryptographic Algorithm Performance Analysis

The proposed system, on the contrary, countermeasures the threats in less than 200 ms through a combination of real-time detection and response. The results of the HMAC-SHA256-based signed reporting mechanism performance revealed in Figure 7 prove the insignificant computational cost, as well as the authenticity of the events.

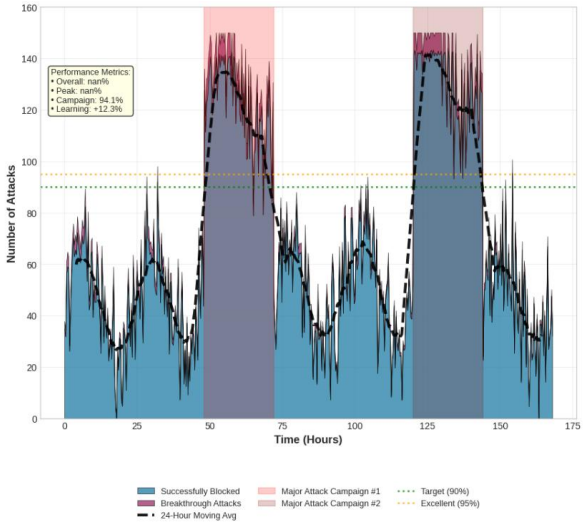

Figure 8: Auto-Blocking Effectiveness Over Time

Figure 8 illustrates the competence of the automated mitigation strategy since the auto-blocking module has been able to inject all the necessary firewall rules in the Linux (iptables) and Windows (netsh advfirewall) systems. Figure 9 shows long-term economic benefits and indicates that this technology saves considerable costs, compared to conventional manual-response security architectures. The platform-neutral implementation is also known to support the implementation in heterogeneous enterprise environments.

A. Enterprise Deployment Issues.

The suggested architecture has a substantial potential to be put into practice in the current enterprise networks. Its real-time detection and automated response features distinctive feature makes it especially appropriate in the context of the Security Operations Center (SOC) that demand quick threat containment and lower mean-time-to-response (MTTR). To

be deployed in production, the system may be combined with already existing security infrastructure such as Security Information and Event Management (SIEM) systems, enterprise firewalls, and endpoint detection and response (EDR) systems. The HMAC-signed reporting will provide audit-grade evidence of compliance needs and avoid the management of alerts in potentially compromised network segments.The architecture can be expanded to provide centralized control over distributed sensors, automated whitelisting of critical systems, and threat intelligence feed integrations to provide better detection context, to make the enterprise more ready. This would be supplemented with compliance reporting, forensic data retention, and automated incident response workflow modules that would add additional operational utility in major deployments. Itmodular design helps it to be deployed in stages and organizations can deploy first detect capabilities and then they can deploy automatic mitigation capabilities gradually as their confidence in the system improves. This incremental strategy minimizes the risk of implementation and offers instant network security posture visibility.

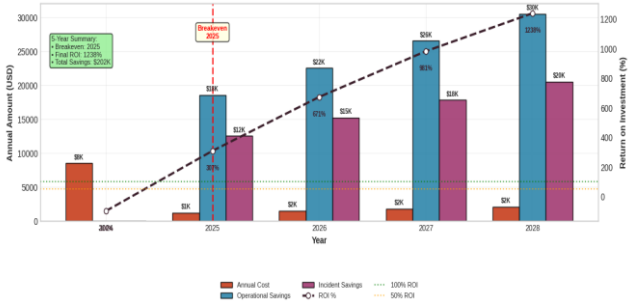

Figure 9: 10-Year Cost-Benefit Analysis

B. Limitations

Although the results obtained are very promising, indicating a high accuracy of detection and a very fast automated response, we must admit that there are some limitations Monitored Test Environment: The test was performed in a lab environment that had simulated attack traffic. Although thorough, synthetic attacks are not at all as sophisticated and diverse as real-world advanced persistent threats (APTs) and zero-day exploits. Encrypted Traffic Analysis: The existing implementation of DPI is unable to decode the encrypted body of the TLS/SSL traffic. This restriction does not cover the monitoring of more and more widespread encrypted attack vectors, which can result in blind spots in the modern networked environment. Scalability Constraints: Scalability testing had been done on one monitoring node with 720 packets/second as its maximum. The networks of large enterprises with the rate of 10,000 and more packets/second would demand the distributed deployments not considered in the presented study. Evasion Technique Resilience: The system was found to be vulnerable to timing based evasion attacks where malicious activity is intentionally reduced to stay under detection limits. More complex behavioral analysis would be necessary to oppose such techniques. Platform Support: The system supports both Windows and Linux but does not as yet call out to next-generation firewalls (NGFWs), cloud security groups, or container network policies, which restricts its use in hybrid and cloud-native environments. These restrictions offer guidance on future study, especially creation of encrypted traffic examine features, disseminated deployment schemes and extension evasion endurance frameworks.

CONCLUSION

The paper presents a secure network monitoring system that incorporates real time multi-threats, HMAC-signed reporting, and automated mitigation using firewall to improve network security. The architecture is a combination of deep packet inspection using PyShark along with cryptographic integrity protection and dynamic rule injection on both windows and Linux. The experimental findings prove that SYN flood, ARP spoofing and port scanning attacks have high detection rates with an overall detection rate of 97.1. The HMAC-SHA256 mechanism involves a small overhead of about 2.3 ms per report, and an auto-blocking module has sub-200 ms response times which mean-time-to-response is greatly reduced. This system maintains processing throughput of 500-720 packets per second with a detection accuracy of over 96 percent at varying network load and has positive cost-benefit properties with a short payback period. Altogether, the hybrid IDS/IPS framework proposed in this paper offers an effective, scalable, and resilient end-to-end network defense framework, which is appropriate in the contemporary enterprise setting.

REFERENCES

[1] Ghafir, I., Prenosil, V., Svoboda, J., & Hammoudeh, M. (2016, August). A survey on network security monitoring systems. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)* (pp. 77-82). IEEE.

[2] Heberlein, L. T., Dias, G. V., Levitt, K. N., Mukherjee, B., Wood, J., & Wolber, D. (1989). *A network security monitor* (No. UCRL-CR-105095). Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States); California Univ., Davis, CA (USA). Dept. of Electrical Engineering and Computer Science.

[3] Casola, V., De Benedictis, A., Riccio, A., Rivera, D., Mallouli, W., & de Oca, E. M. (2019). A security monitoring system for internet of things. *Internet of Things*, *7*, 100080.

[4] Alevizos, L. Automated cybersecurity compliance and threat response using AI, blockchain and smart contracts. *Int. j. inf. tecnol.* **17**, 767–781 (2025). https://doi.org/10.1007/s41870-024-02324-9

[5] Abbassy, M.M., Satya Sai Kumar, A., Ead, W.M. *et al.* Hybrid deep learning framework for detection of anomalies in cyber-physical systems. *Int. j. inf. tecnol.* **18**, 689–695 (2026). https://doi.org/10.1007/s41870-025-02633-7 Ciszkowski, T., Eliasson, C., Fiedler, M., Kotulski, Z., Lupu, R., & Mazurczyk, W. (2008). SecMon: end-to-end quality and security monitoring system. *arXiv preprint arXiv:0804.0134*.

[6] Khan, R., Khan, S. U., Zaheer, R., & Babar, M. I. (2013). An efficient network monitoring and management system. *International Journal of Information and Electronics Engineering*, *3*(1), 122-126.

[7] Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network intrusion detection for IoT security based on learning techniques. *IEEE Communications Surveys & Tutorials*, *21*(3), 2671-2701.

[8] Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, *28*(1-2), 18-28.

[9] Almihyawi, A.Y.T., kurnaz, S. A secure smart monitoring network for hybrid energy systems using IoT, AI. *Discov Computing* **28**, 14 (2025). https://doi.org/10.1007/s10791-025-09506-4