iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79506

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

pfSENSE: Intrusion Detection System for Advanced Network Security

Sujitha Jammula[1], Gochalam Vinod[2], Rishi Priya Nainamoina[3], Ms. K. Parveena[4]
Computer Science and Engineering (Cyber Security), Institute of Aeronautical Engineering, Hyderabad, India

Abstract: ***Network defense has moved beyond simple perimeter filtering because modern attacks are faster, stealthier, and more adaptive than traditional rule sets can handle. This paper presents a layered security framework built around pfSense firewall, Snort intrusion detection, and the Isolation Forest anomaly detection algorithm. The design combines three complementary capabilities: access control at the network edge, signaturebased detection for known attacks, and machine learning-based anomaly detection for previously unseen behavior. The result is a practical architecture that can identify both documented threats and suspicious traffic patterns that do not match existing signatures. pfSense acts as the gateway and enforces segmentation between trusted and untrusted zones. Snort inspects live packets, compares them against rule sets, and generates alerts for malicious payloads, scans, and suspicious protocol behavior. Isolation Forest processes log-derived features such as packet size, source and destination addresses, connection rate, and timing patterns to detect deviations from normal activity. Because it is unsupervised, the model can be applied even when labeled attack data is limited. The full system is implemented in a virtualized environment so that it remains cost-effective, repeatable, and suitable for academic use.***

The proposed framework improves visibility, reduces response time, and strengthens the ability to defend against unknown threats. It also provides a basis for automated containment, where suspicious hosts can be blocked through firewall rule updates after detection and validation.

Keywords: Network Security, pfSense, Snort, Intrusion Detection System, Isolation Forest, Anomaly Detection, Firewall, Threat Mitigation.

I. INTRODUCTION

The growth of digital infrastructure has increased the dependence of organizations on networked services, remote access, cloud storage, and real-time communication. This dependence has also expanded the attack surface available to adversaries. Malicious actors now exploit weak passwords, misconfigured services, exposed ports, unpatched systems, and user mistakes to gain access or disrupt operations. The scale of these attacks makes network security a continuous operational requirement rather than an occasional administrative task. Traditional perimeter tools remain important, but they are no longer sufficient on their own. Firewalls are effective when the threat can be described clearly in terms of allowed and denied traffic. Signature-based intrusion detection systems are effective when a known attack has a reliable pattern that can be written as a rule. The problem is that modern attacks often do not behave in a fixed or predictable way. Attackers use encrypted channels, fragmented payloads, polymorphic malware, and low-rate reconnaissance to avoid detection. In such environments, a single defensive layer is not enough. This project addresses the gap by integrating pfSense, Snort, and Isolation Forest into a single layered architecture. pfSense provides the control point for traffic filtering and network segmentation. Snort adds real-time inspection and detection of known malicious traffic. Isolation Forest provides behavioral analysis so that traffic can be flagged even when it does not match a predefined signature. Together, these components implement a defense-in-depth strategy that is both practical and extensible. The project is also intended to demonstrate how open-source technologies can be combined into a usable security stack without large licensing costs. This is important in academic environments, small organizations, and laboratories where the goal is to build a functional and understandable system that can be tested under realistic traffic conditions. The implementation therefore emphasizes clarity, modularity, and measurable detection behavior.

II. EXISTING SYSTEM

Current network protection strategies commonly deploy a firewall, an IDS, and manual analysis as separate layers. In many cases, these tools are configured independently and do not share context. A firewall may block obvious unauthorized traffic, but it cannot determine whether allowed traffic is carrying malicious intent. A signature-based IDS may generate an alert, but the alert often requires a human to interpret it and decide on a response. This separation slows down incident handling and can leave an organization exposed during the period between detection and mitigation.

Existing systems also rely heavily on known signatures. Such systems work well when the attack is already documented and the relevant rule has been distributed. However, they are less effective for zero-day attacks, obfuscated payloads, and novel attack sequences that have not been captured in existing databases. In environments where attackers change tactics frequently, signature updates can lag behind the threat landscape.

Another common issue is centralization. Some deployments depend on cloud-based monitoring or a remote analytics platform to process logs and make decisions. While central visibility can be useful, it may introduce latency, dependency on external service availability, and additional complexity in secure configuration. For organizations that need immediate local response, this is a limitation.

A. Drawbacks of Existing Systems

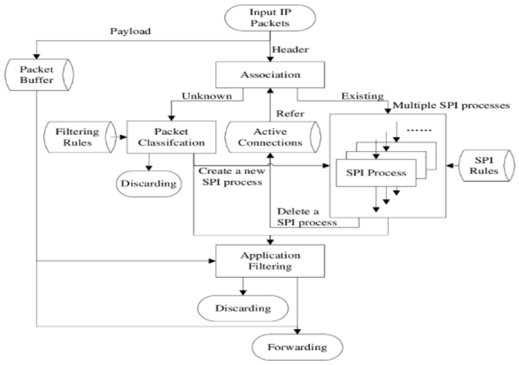
The first drawback is dependence on known attack signatures. If a threat does not resemble a stored rule, the system may not identify it. The second drawback is high false positive rates in some anomaly tools, especially when traffic is dynamic or the detection threshold is poorly tuned. The third drawback is limited automation. Many systems notify administrators but do not take corrective action automatically, increasing dwell time. The fourth drawback is weak correlation between network events and behavioral analysis. As a result, administrators must manually combine evidence from multiple sources.

III. PROPOSED SYSTEM

The proposed system introduces a unified architecture in which pfSense, Snort, and Isolation Forest operate together. pfSense is deployed as the main gateway firewall. It separates internal and external traffic, enforces access policies, and reduces the attack surface by limiting unnecessary communication. Snort runs as the signature-based inspection layer. It checks packets and sessions against a rule set and generates alerts for known threats such as scanning, exploit attempts, and suspicious payloads.

Isolation Forest provides the anomaly layer. It consumes structured logs from the network stack and evaluates whether observed traffic matches normal behavior. Features such as packet length, protocol type, connection count, request timing, and source-destination relationships are used to build the input vector. The model isolates abnormal cases quickly because anomalous patterns tend to differ from the bulk of traffic. This is useful for detecting new attack styles that are not covered by signatures.

The design can support automated response. When a suspicious source is confirmed, a rule can be inserted into pfSense to block the IP address or restrict the connection path. This reduces the delay between detection and containment. The system can also be extended with external threat intelligence sources so that suspicious events can be compared against known malicious reputation feeds before an action is taken.



A. Advantages

The main advantages are broader threat coverage, lower deployment cost, and better visibility. The system identifies both known and unknown threats. It also remains suitable for a virtualized lab because the core components are open source and can run on standard hardware. In addition, the modular structure makes it easier to add dashboards, SIEM integration, or additional machine learning models later.

IV. PROBLEM STATEMENT

The core problem is the lack of a single low-cost security stack that can detect both known threats and unknown anomalous behavior in real time. Firewalls alone focus on policy enforcement, while IDS tools focus on signature matching. Neither approach fully solves the problem of novel attacks, stealthy behavior, and delayed response. A practical solution is needed that can inspect traffic, detect suspicious patterns, and support automated mitigation without requiring expensive proprietary infrastructure.

V. OBJECTIVES

The first objective is to configure pfSense as a secure gateway for network segmentation and traffic control. The second objective is to deploy Snort for live packet inspection and known threat detection. The third objective is to implement Isolation Forest for anomaly detection using traffic logs and behavioral features. The fourth objective is to evaluate the system using simulated attacks and measure detection quality, response time, and false positives. The fifth objective is to demonstrate that open-source tools can be combined into an intelligent and scalable defense mechanism.

VI. LITERATURE SURVEY

Research on network security has increasingly shifted toward hybrid approaches that combine rule-based controls with machine learning. Earlier intrusion detection systems were dominated by signatures because they were simple to deploy and effective against known attacks. Over time, however, researchers observed that these systems fail when attack behavior changes faster than the rule base can be updated. This motivated work on anomaly detection, feature engineering, and hybrid defense models.

Firewall research shows that rule-based traffic control remains foundational. Open-source platforms such as pfSense are attractive because they provide stateful inspection, NAT, VPN support, flexible rules, and segmentation. Their value lies in enforcement and containment rather than attack classification. Studies on firewall design consistently emphasize that segmentation reduces exposure and limits lateral movement if one host becomes compromised.

Research on Snort confirms its strength as a signature-based IDS. It is widely used for packet inspection, alerting, and logging. Its main limitation is that it can only detect what the rule set knows how to identify. When a payload is altered or a technique is new, the IDS may miss it. This limitation has driven many researchers to combine Snort with additional analytics rather than relying on it alone. Machine learning-based anomaly detection has become a major area of study. Isolation Forest is widely used because it is unsupervised, efficient, and effective for outlier detection in high-dimensional data. It does not require every attack type to be labeled beforehand, which makes it suitable for network traffic where attack categories may be incomplete or evolving. Its ability to isolate rare cases quickly makes it appropriate for real-time monitoring.

Recent literature also supports hybrid security systems that correlate firewall, IDS, and analytics outputs. When multiple tools confirm the same event, the confidence of the alert increases. This improves operational usefulness because administrators can focus on higher-value incidents instead of isolated warnings. The proposed system follows this direction by combining packet control, signature matching, and anomaly scoring.

TABLE I: Comparison of IDS Approaches

Approach	Strengths	Limitations
Signature-based	High precision for known threats; easy to deploy	Misses novel attacks; requires frequent updates
Anomaly-based	Detects unknown behavior; adaptable to changing traffic	Can generate false positives; requires tuning
Hybrid	Balances known-threat detection and behav-ioral analysis	More complex to configure and maintain

VII. SYSTEM DESIGN

The architecture consists of two functional nodes. The first node runs pfSense and handles ingress and egress traffic. The second node runs Snort and the anomaly detection module. Traffic is first filtered at the firewall, then mirrored or forwarded for inspection. This arrangement ensures that the firewall can enforce policy while the detection stack can observe events in enough detail to classify them. The data flow begins when a packet reaches the firewall. pfSense decides whether it is allowed to pass, should be blocked, or should be forwarded to the monitoring path. Snort then evaluates the packet stream using its configured rules. If the packet matches a known malicious pattern, an alert is generated and stored with metadata such as time, source, destination, and protocol. These logs are then processed into features for the Isolation Forest model. If the anomaly score crosses the chosen threshold, the event is marked as suspicious even if no signature match was found.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

The use case model is centered on the network administrator. The administrator configures firewall rules, updates Snort rule sets, reviews alerts, and validates the output of the anomaly detector. The administrator also decides whether suspicious hosts should be blocked, isolated, or monitored further. This keeps the system practical because automated analysis is complemented by human oversight.

A. Software Requirements

The software stack includes pfSense, Snort, Python, scikitlearn, pandas, NumPy, and packet analysis tools such as Wireshark. Optional utilities such as Nmap can be used to simulate scanning activity, while log collectors can be used to move Snort output into a structured analysis pipeline. The use of open-source software keeps the design reproducible and affordable.

B. Hardware Requirements

The firewall node should have at least two network interfaces and moderate memory for stable operation. The detection node should have sufficient RAM and CPU to process traffic logs and run the anomaly model. Since Isolation Forest is lightweight compared to deep learning models, GPU acceleration is not required for the current version.

TABLE II: Tools and Technologies Used

Tool	Purpose
pfSense	Firewall, segmentation, traffic control
Snort	Signature-based intrusion detection
Isolation Forest	Anomaly detection on network logs
Python	Feature processing and model execution
Wireshark	Packet tracing and validation

VIII. METHODOLOGY AND IMPLEMENTATION

Implementation begins with a virtualized network setup. One virtual machine is assigned to pfSense, and another is assigned to the detection stack. WAN and LAN interfaces are configured so that traffic passes through the firewall before reaching the rest of the network. Snort is installed on the monitoring host and attached to the traffic stream. Rule sets are enabled so that common attack types can be observed immediately.

The next phase is data collection and preprocessing. Logs from pfSense and Snort are parsed to produce a structured dataset. Features include source and destination IP addresses, port numbers, protocol, packet size, connection duration, packet frequency, and time intervals. Irrelevant or missing fields are removed or standardized. This step is important because anomaly detection depends on the quality of the input features. If the data is inconsistent, the model may produce unstable scores.

Isolation Forest is trained on traffic that represents normal usage. The model learns how benign traffic typically behaves by recursively partitioning the feature space. Instances that are isolated with fewer splits are treated as anomalies. During live analysis, each incoming event is assigned an anomaly score. Events with unusually high scores are flagged for review or response. Since the method is unsupervised, it can operate even when labeled attack data is incomplete.

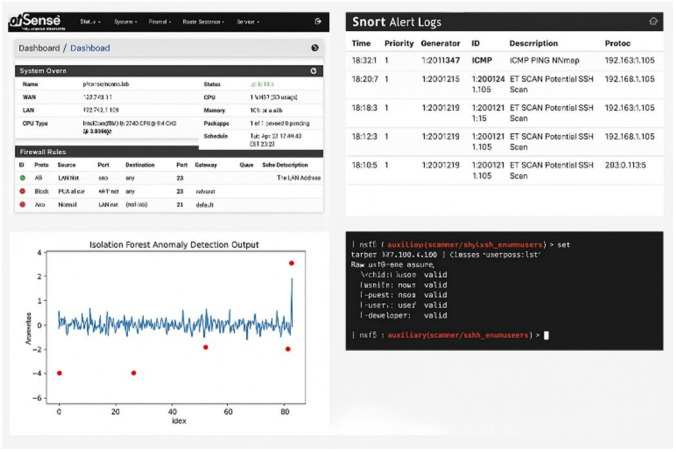
The response stage links detection to action. When Snort confirms a known attack or Isolation Forest flags a suspicious host, the system can log the event, notify the administrator, and optionally update firewall rules to block the source. This gives the architecture a practical mitigation capability rather than a passive alert-only function.

A. Operational Flow

Traffic enters the firewall, is filtered by policy, and is then inspected by Snort. The logs are formatted and passed to the analysis pipeline. The machine learning component calculates anomaly scores and identifies irregular behavior. Alerts are then correlated, and the most serious events are escalated. This sequence supports both preventive and detective security controls.

IX. TESTING AND VALIDATION

Testing is performed by generating controlled traffic that simulates both benign and malicious activity. Common test cases include port scans, malformed packets, suspicious login attempts, and traffic bursts that resemble denial-of-service patterns. The goal is to check whether the firewall blocks unauthorized traffic, Snort raises alerts for known signatures, and Isolation Forest detects traffic that deviates from normal behavior.



Validation focuses on three primary measures. Detection accuracy indicates how often malicious traffic is correctly identified. The false positive rate indicates how often normal traffic is incorrectly marked as suspicious. Response time measures how quickly the system can move from detection to containment. In addition, the integrity of the log pipeline is verified to ensure that no traffic events are lost between components.

The model threshold is adjusted when needed. If the detector is too sensitive, benign traffic may be flagged too often. If it is too conservative, suspicious traffic may be missed. The validation process therefore includes iterative tuning to balance sensitivity and precision. This step is necessary because network traffic patterns vary across environments and usage profiles.

TABLE III: Performance Evaluation Metrics

Metric	Meaning
Detection accuracy	Ratio of correctly classified events
False positive rate	Percentage of benign traffic flagged as malicious
Response time	Delay between alert generation and action
Precision	Reliability of positive detections
Recall	Ability to identify actual attacks

X. RESULTS

The combined system demonstrates that layered security improves both coverage and operational usefulness. pfSense offers immediate traffic control at the perimeter, Snort detects known malicious patterns, and Isolation Forest identifies abnormal behavior that may indicate stealthy or previously unseen threats. This improves visibility compared with a firewallonly deployment and reduces the chance that suspicious traffic will pass without examination. A practical advantage of the system is that it can be deployed in a low-cost virtual environment. This makes it suitable for lab work, small organizations, and prototype testing. The same architecture can also be extended for larger environments by distributing sensors and centralizing logs. The open-source nature of the components makes reproduction straightforward and lowers the barrier to experimentation.

The results also show the value of correlating rule-based and behavioral evidence. A signature alert by itself may be useful, but when it is combined with a behavioral anomaly, the confidence of the event is higher. This is especially important for incident response, where unnecessary escalation should be avoided while real threats must be contained quickly.

XI. CONCLUSION AND FUTURE SCOPE

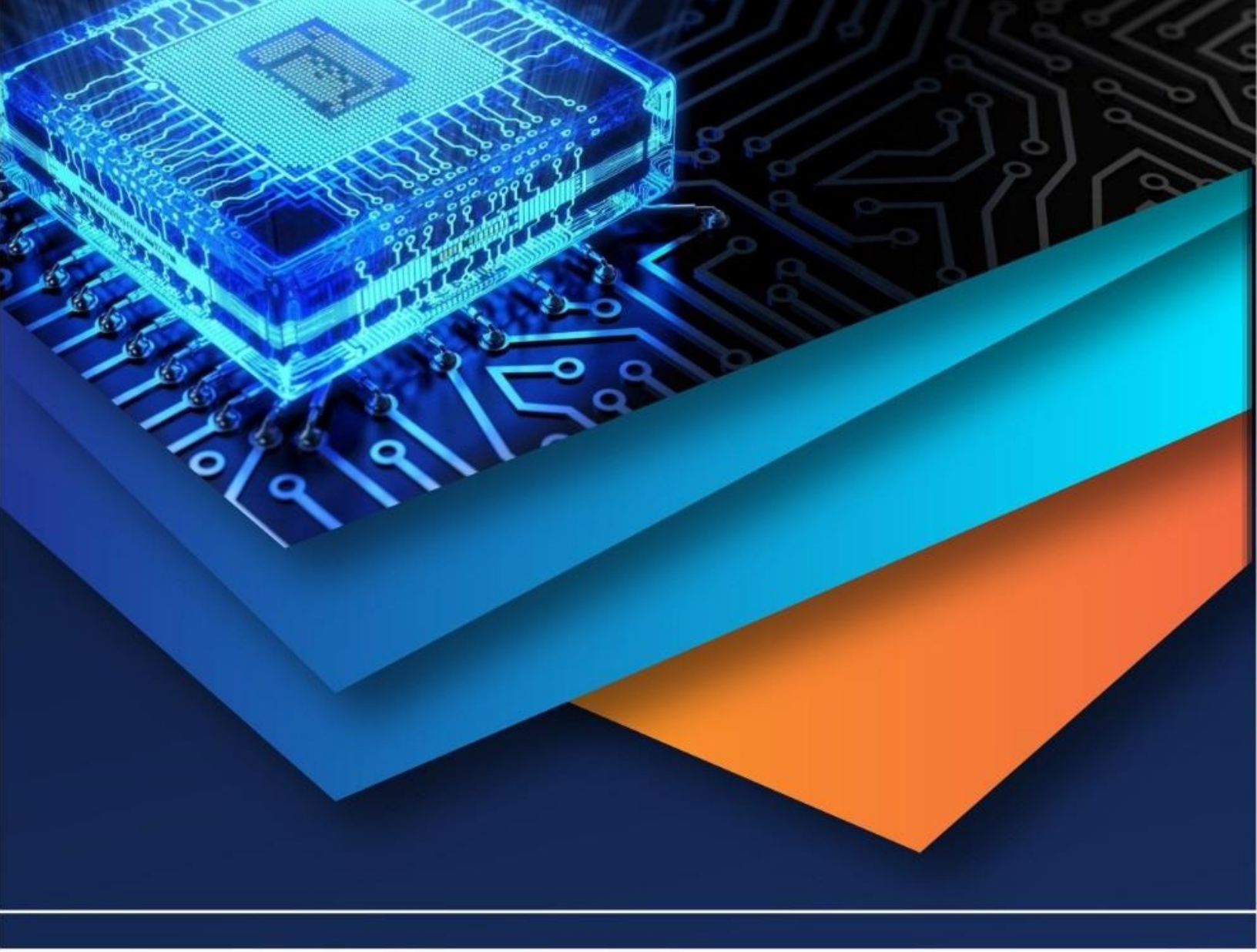
This project shows that an intelligent and practical defense architecture can be created by combining pfSense, Snort, and Isolation Forest. pfSense provides policy enforcement, Snort identifies known attacks, and Isolation Forest adds anomaly detection for unknown or unusual behavior. The resulting framework is more capable than any single component operating alone because it combines prevention, detection, and response in one workflow.

The design is also academically and operationally useful. It demonstrates how open-source tools can be integrated into a functional security stack and evaluated using realistic test traffic. The virtualized deployment keeps the system accessible and makes it easier to repeat experiments or extend the model. The architecture therefore has value both as a teaching model and as a base for further research.

Future work can include more advanced machine learning models, richer feature extraction, and integration with SIEM platforms. Endpoint correlation and threat intelligence enrichment can also be added so that the system can compare local events against broader reputation data. Another useful direction is automated orchestration, where the firewall, IDS, and analytics engine react together with minimal delay. With these extensions, the framework can evolve into a more complete adaptive security platform.

REFERENCES

[1] M. Roesch, “Snort: Lightweight Intrusion Detection for Networks,” project documentation and system overview.
[2] Open Source Community, “pfSense Documentation,” firewall configuration, routing, and segmentation reference.
[3] L. Liu et al., “Isolation Forest,” an unsupervised anomaly detection method used for outlier identification.
[4] Selected studies on hybrid IDS architectures, firewall segmentation, and machine learning-based security analytics.
[5] AlienVault OTX and related reputation services, used for optional threat validation in security workflows.
[6] S. Dai, “Network Intrusion Detection and Protection System Based on pfSense and Snort,” *Network Security and Informatization*, vol. 9, pp. 123–126, 2022.
[7] D. Wang, J. Zhang, and J. Yu, “Research on Intelligent Firewalls for Network Security,” in *Proc. 2nd Int. Conf. Robot., Intell. Control, Artif. Intell. (RICAI’20)*, pp. 255–258, 2020.
[8] W. Stallings, Network Security Essentials: Applications and Standards, 6th ed. Pearson Education, 2018.
[9] M. Roesch, “Snort - Lightweight Intrusion Detection for Networks,” in Proc. 13th USENIX Conf. Syst. Admin., pp. 229–238, 1999.
[10] H. N. Huang, “Implementation and Detection of Denial of Service Attacks Against Snort,” Master’s Thesis, Jilin Univ., 2020.
[11] Z. Zhang, “Design and Implementation of a Snort-Based Intrusion Detection System,” China Univ. Weights Meas., 2020.
[12] T. Liu, K. M. Ting, and Z. H. Zhou, “Isolation Forest: A Fast Anomaly Detection Algorithm,” in Proc. 8th IEEE Int. Conf. Data Min. (ICDM’08), pp. 413–422, 2008.
[13] M. Ahmed, A. N. Mahmood, and J. Hu, “A Survey of Network Anomaly Detection Techniques,” J. Netw. Comput. Appl., vol. 60, pp. 19–31, 2016.
[14] Y. Tang and Y. Hsieh, “Using Machine Learning for Network Anomaly Detection: A Comparative Study,” IEEE Trans. Inf. Forensics Secur., vol. 13, no. 8, pp. 2003–2016, 2018.
[15] P. Ye and Z. Zhang, “Anomaly Detection Method, Apparatus, and Electronic Device Based on Behavioral Whitelist,” China Patent 2018111809412, 2018.
[16] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly Detection: A Survey,” ACM Comput. Surv. (CSUR), vol. 41, no. 3, pp. 1–58, 2009.
[17] [12] J. Li, “Optimization and Implementation of Snort Intrusion Detection Method,” Master’s Thesis, Northeast Normal Univ., 2021.
[18] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A Deep Learning Approach to Network Intrusion Detection,” IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, no. 1, pp. 41–50, 2018.
[19] L. Wang and P. Roger, “A Security Detection Method for Internet Port Scanning Attacks,” Inf. Secur. Technol., vol. 2, pp. 44–45, 2016.
[20] K. Dinakaran, D. Rajalakshmi, and P. Valarmathie, “Efficient Pattern Matching for Uncertain Time Series Data with Optimal Sampling and Dimensionality Reduction,” Microprocess. Microsyst., 2020.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY