INTERNATIONAL JOURNAL OF
RECENT TRENDS IN MULTIDISCIPLINARY RESEARCH
https://www.doi.org/10.59256/ijrtmr.20260602032
March-April 2026, Vol 6(02), 231-239.
ISSN No: 2583-0368

MINISIEM: A Log Analysis & Security Monitoring System

Dr. V. Dhanakoti[1], Charan Raj K[2], Akash D[3], Hamsahaasan G[4]
[1] *Professor, Computer Science and Engineering, SRM Valliammai Engineering College, Chennai, Tamilnadu, India.*
[2, 3, 4] *Computer Science and Engineering, SRM Valliammai Engineering College, Chennai, Tamilnadu, India.*

OPEN ACCESS
Article Citation:
Dr. V. Dhanakoti[1], Charan Raj K[2], Akash D[3], Hamsahaasan G[4] "MINISIEM: A Log Analysis & Security Monitoring System", International Journal of Recent Trends in Multidisciplinary Research, March-April 2026, Vol 6(02), 231-239.

Abstract: Cybercriminals now have access to a much larger attack surface due to the ever-increasing reliance on internet-connected devices. This has resulted in an increase in network-based attacks such as port scanning, brute-force logins, distributed denial-of-service attacks, and attempts to gain unauthorized access [1]. Firewalls usually serve as a first line of defence for most networks, however, they generate a lot of log data which is not often properly analysed due to difficulties with manual analysis [6]. The objective of this study is to design and implement a Firewall Security Analytics System that will collect and analyse firewall logs to identify potentially malicious network activity, classify the attacks, and provide real-time monitoring of security events. The general design includes a client-server architecture using a Flask backend and a React-based Security Operations Center (SOC) dashboard. The firewall logs are collected from an Ubuntu server using SSH-based secure log ingestion into a structured database format. The classification of attack patterns is accomplished by an analysis engine that utilizes rules to determine whether or not an event is suspicious (for example: port scanning, brute-force attempts, and abnormal connection activity). If a security threshold is exceeded, an alerts engine will generate an alert to notify the administrator. In addition, the solution includes a dashboard where real-time analytics can be viewed to visualize current attacks on the network. Ultimately, the results indicate that the proposed platform is able to convert raw data from firewalls into action-based intelligence about the state of your network, giving the administrator a better understanding of the current status of their network, allowing for faster detection of potential threats.

Keywords: Cyber security, Firewall Monitoring, Security Analytics, Intrusion Detection, Flask, React, Attack Classification, Network Security.

1. Introduction

As a result of the rapid growth of digital infrastructure, organizations relying primarily on the internet have become increasingly vulnerable to cyber attacks. An increasing number and types of critical services available through these networks provides attackers with significantly more opportunities to exploit system/application weaknesses supporting these services. Organization's use firewalls as their first layer of protection from both authorized/unattributed access to their protected resources by filtering incoming/outgoing network traffic through network traffic filtering techniques. While firewalls effectively block incoming malicious network traffic, firewalls generate large amounts of logs that contain detailed timestamps, connection attempts, blocked/non-blocked packet statistics, source/destination addresses, and port numbers. Due to the high volume of firewall logs generated, manually analyzing them is exceedingly difficult for networking staff.

Firewalls have traditionally provided very little analysis of network activity beyond filtering traffic, so some of the threat patterns could go undetected until they cause extensive damage to the network infrastructure. As a result, there is a very real

and urgent need for intelligent monitoring systems that can continuously monitor firewall log data and identify suspicious activity in real time.

The System firewall Security Analytics solution automates how you collect, analyze and visualize the firewall log as part of a methodology to enhance your ability to manage your organization's network activity and respond to potential threats in a timely manner by collecting, analyzing and visualizing firewall logs from your organization's monitored workstations so that administrators can monitor their organization's network activity and alert them to possible suspicious activity quickly.

2. Related Work

The Firewall Security Analytics System's Architecture is built using many of today's current principles in the development of systems for Network Security Monitoring (NSM), Intrusion Detection Systems (IDS), and Log Analytical Platform [6]. This section will discuss previously published research related to firewall monitoring, attack detection methods, security visualization utilities etc.

A. Log Analysis and Parsing Techniques

Since distributed computing environments can create a large number of different types of logs, performing log analysis has become an essential part of modern cybersecurity monitoring systems. Logs contain important information about the behaviour of the system, any operational problems that may have occurred, and security events that have occurred. A systematic mapping study by Partovian et al. identifies log analysis methods used for smart troubleshooting within Industry 4.0 environments. They found that automated log analysis methods (especially when using machine learning) are most frequently used to find anomalies and diagnose system failures in very large cyber-physical systems [1].

Log parsing transforms unstructured log data into structured representations, which are necessary to support effective analysis. Automated log parsing frameworks have been developed by Marlaithong et al. specifically for the ALICE O2 computing infrastructure supporting the analysis of large volumes of runtime log data produced by distributed scientific computation environments [2]. The authors propose that using TF–IDF-based feature extraction and genetic programming allows for the automatic generation of log templates and improved accuracy of log parsing. These and other effective methods for log parsing allow unstructured log entries to be converted into structured events that can be used in analysing for anomalous activity or monitoring of systems.

B. Log Anomaly Detection and AI-Based Security Analytics

As distributed systems have become increasingly complex, researchers have begun applying machine learning and deep learning techniques so that they can detect anomalies in their corresponding system logs. Through their research, Raeiszadeh et al. developed ALogSCAN, which incorporates a self-supervised dual-network architecture that can adaptively detect anomalies within cloud computing environments. In doing this, the authors utilize dynamic frequency-based log filtering to be able to filter out infrequent but critical log events while also reducing the dependency on labeled training data [3].

Horváth et al. developed and tested several methods for detecting anomalies in log files by using real-time analyses of log data, including deep models based upon neural networks, clustering algorithms, and standard statistical techniques. Their evaluation study found that the more advanced neural networks were equally effective at detecting anomalies as the simpler statistical methods, depending upon the complexity of the system and the operational constraints under which it was maintained [4].

Hybrid deep learning models have recently been examined to boost the effectiveness of anomaly detection systems. Alzamil et al. introduced DualBERT, a novel hybrid approach for anomaly detection based on using time-based features through the use of a Long Short Term Memory (LSTM) regression model combined with a transformer-based analysis of logs using DualBERT itself to analyse event sequences [5]. The incorporation of symbolic event sequences into the temporal analysis of events leads to a substantial improvement in the accuracy of anomaly detection and the percentage of false positives associated with these systems [5].

Similarly, in relation to the analysis of firewall logs, Aboudrar et al. introduced a new AI-based security information and event management (or SIEM) system that utilize deep-learning models capable of analysing information from firewall logs and detecting malicious activity on the network [6]. The proposed system integrates both event correlation and event classification based on neural networks to provide improved threat detection capabilities and a reduction in the number of false positives experienced with cybersecurity monitoring systems [6].

C. Firewall Monitoring, Visualization and Security Optimization

In addition to using techniques to detect anomalies, many studies have been performed using firewall performance analysis as well as monitoring security events. Schufrin et al. (2010), developed an interactive visual log analysis system for performing firewall log analysis through visual analytics techniques. The system utilizes a number of different visual analytic interfaces to provide security analysts with access to similar functionality, including both an overview of the network security events as well as a detailed analysis of the network security events represented by firewall logs [7].

Lee et al. (2013), analysed firewalls event logs for high-performance computing networks and proposed a filtering mechanism to limit the amount of processing performed by the firewall. The results showed that the identification of firewall traffic patterns can be used to improve the performance of a firewall and provide the same (or better) security guarantee as compared to a similar amount of analysis of the same event logs without any performance consideration [8].

Another significant area that many researchers focus on in the field of cybersecurity is developing new datasets used to evaluate intrusion detection systems. For example, Black et al. created the Firewall Attack Detection and Extraction (FADE) dataset, which contains millions of labeled, benign, and malicious network requests to aid in developing and testing firewall attack detection systems [10].

Lastly, work is being done by Durante et al. to establish a formal model for distributing packet filtering rules across multiple firewalls within a network. The proposed rule redistribution algorithm will reduce the processing overhead related to firewalls by effectively balancing the filtering workload among all of the cascade firewalls in a given environment; thus, increasing the performance of packet processing within large, network-based infrastructures [9].

3. Proposed System Architecture and Design

The Firewall Security Analytics System uses a Modular Design by integrating multiple components of collecting logs from Firewalls, Processing Events, Detecting Attacks & Visualizing Events. Thus, allowing for capabilities of Scalability, Reliability and Efficiency when analyzing Network Events.

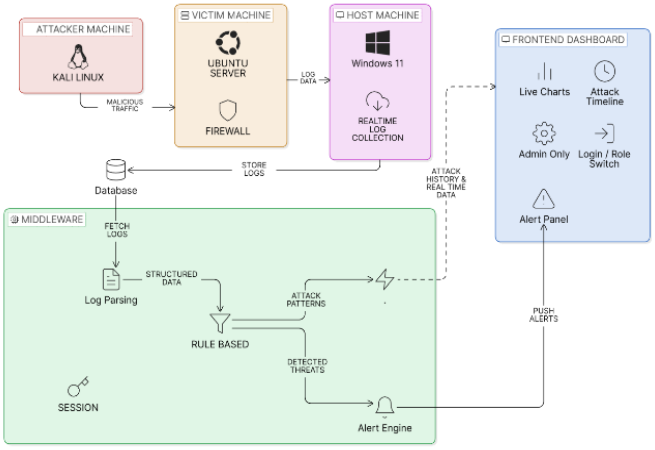

Fig: 3.1 Architecture Diagram

A. Design Principles

The architecture of the system is directed by 3 fundamental design principles.

Modularity and Scalability: The framework consists of 4 distinct modules providing functions for gathering logs, processing data, classifying attacks/attempts, and providing a means to visualize security events. The modular nature of the solution makes it possible to add new security features to the platform without having to modify already-existing modules.

Performance and Efficiency: To support efficient processing of large volumes of log data, network monitoring systems require an appropriate architecture at the backend. This architecture is constructed to allow for continuous log ingestion and classification while maintaining the responsiveness of the dashboard interface.

Security and Reliability: Given that any information transmitted through your systems could potentially include some form of sensitive network traffic, secure communications have been established between the logging service and back-end servers to provide confidentiality. Logging also provides authentication services to ensure that only authorized individuals can view the monitoring dashboard views.

B. System Components

The proposed system consists of three major layers that work together to deliver the platform's functionality.

Table 1: System Component Specifications

Component	Specification
Attacker Machine	Kali Linux
Firewall System	Ubuntu Server with UFW Firewall
Log Collection Host	Windows 11
Backend Framework	Flask (Python)
Database	SQLite
Frontend Framework	React.js
Authentication	JWT-based Authentication
Visualization	Chart.js / Dashboard Analytics
Log Retrieval Method	SSH-based log collection
Dashboard Refresh Interval	5 seconds

Fig: 3.2 System Component Specification

Log Collection Layer: Firewall logs generated by the Ubuntu Uncomplicated Firewall are collected using secure SSH connections. These logs contain information such as source IP addresses, destination ports, protocol types, and timestamps.

Table 2: Firewall Event Log Fields and Descriptions

Field	Description
Timestamp	Time when firewall event occurred
Source IP Address	IP address initiating the connection
Destination IP Address	Target system receiving traffic
Destination Port	Network port targeted by the connection
Protocol	Communication protocol (TCP, UDP, ICMP)
Action	Firewall decision (Allowed or Blocked)
Severity Level	Threat level assigned to event
Attack Type	Classified attack category

Fig: 3.3 Firewall Event Log Fields & Description

Backend Processing Layer: The backend of the application was designed and implemented using the Flask web framework, which parses firewall log files, inserts records of events into an SQLite database, and executes an algorithm that classifies attacks against a network. The backend also exposes a number of REST APIs which provide processed data to the front-end user interface (for example, the dashboard).

Visualization Layer: The front-end dashboard application has been developed using React, and provides visual representations of network activity in nearly real-time. The dashboard displays statistics related to attacks, timelines accompanying logs, and displays geographic maps where possible attacks may occur to assist network/system administrators with monitoring for potential attacks.

Table 3: Comparison of Traditional Firewall Monitoring and Proposed System

Feature	Traditional Firewall Monitoring	Proposed System
Real-Time Analytics	Limited	Yes
Attack Visualization	Minimal	Advanced Dashboard
Event Correlation	Manual	Automated
Geolocation Attack Map	Not Available	Available
Attack Pattern Detection	Limited	Rule-based Classification
Threat Intelligence Support	Limited	Integrated

Fig: 3.4 Comparison of Traditional & Proposed System

4. Implementation & Module Functionality

The MiniSIEM: A Log Analysis & Security Monitoring System was implemented as a modular cybersecurity analytics platform with firewall logs provides real-time threat intelligence by processing the firewall logs into various analytical modules. These analytical modules process the firewall logs collected from any Ubuntu-based firewall, providing insight into threats through multiple analytical modules. Each analytical module contributes to identifying malicious activity and to identifying attack patterns that are helpful in assisting security analysts in identifying the threats on a network.

Table 4: Cybersecurity Analytics Features and Descriptions

Feature	Description
IP Intelligence	Provides contextual information about attacker IP
Targeted Port Analysis	Identifies frequently targeted network ports
Protocol Distribution	Analyzes protocol usage across events
Service Exposure Analysis	Determines exposed network services
Attack Trend Analysis	Observes long-term attack patterns
Attack Growth	Measures increase in attack frequency
Attack Heatmap	Visualizes attack intensity over time
Attack Velocity	Measures rate of attacks within time window
Attack Intensity	Evaluates density of attacks in a given period
Attack Correlation	Links related attack events
Anomaly Detection	Identifies abnormal network behavior

Fig: 4.1 Analytics Features & Description

Fig: 4.2 Real-Time Overview

Attack Timeline: The Attack Timeline Module displays a chronological distribution of the time when security events were detected through the use of timestamps from firewalls whereby security analysts can see the frequency of when attacks occur and can see any abnormal spike in malicious activity at a point in time in order to identify peak attack times.

Severity Distribution: The Severity Distribution Module breaks down the detected events based upon their threat level. Detects will fall under a certain severity based upon the attack method used and the potential impact it may have on a network (low, medium or high). This will produce graphical representations that will aid security analysts in assessing the security health of their network.

Attack Type Distribution: The Attack Type Distribution Module provides insight to the types of attacks that were detected within a monitored environment. By utilizing a rule-based classification engine, the module is able to categorize various types of attacks (port scanning, brute-force attempts and connection bursts) and produce a graphical representation of the frequency of each attack. This will allow analysts to identify their highest, most prevalent threat types.

Top Attacker Ips: Through usage of the Top Attacker IPs module, malicious activity from external IP Addresses can be identified and ranked according to the amount of detected events found within the firewall logs. By using this information, analyst can prioritize their efforts in investigating and mitigating attacks from the most active external IPs.

Fig: 4.3 Dashboard Charts

Attack Map: The Attack Map module shows you visually where attacks are originating from geographically; meaning you can see where cyber attacks originated by looking at the sources IP Address mapped to its approximate geographical location (using available IP geolocation services). This information allows you to see where attacks originated, as well as display how malicious activity is globally distributed within the network that you are monitoring.

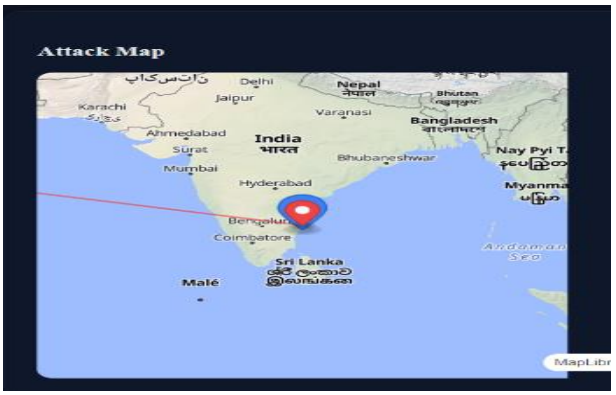

Fig: 4.4 Attack Map

Live Event Feed: The Live Event Feed module provides live feeds of detected security events (threats), with events being displayed on the dashboard immediately after they have been detected (from the backend processing of the firewall logs). This module provides a near-real time view of threats as they happen, similar to a SOC (Security Operations Center) monitoring view.

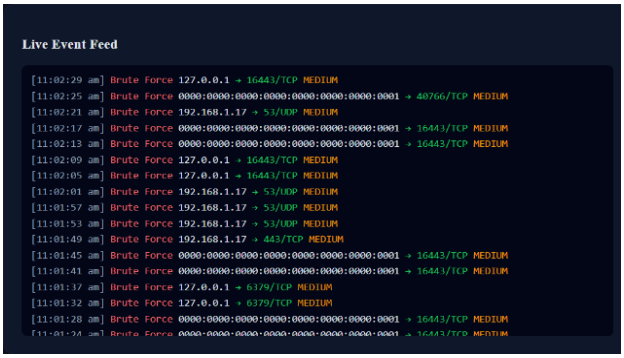

Fig: 4.5 Live Event Feed

Attack Replay: The Attack Replay module allows analyst to look back, at attack sequences, historically; replaying attacks based on one or more attackers OR based on a specific time window. This feature allows for re-creation of an attack timeline, enabling security teams to gain insight into attacker behaviour and to then use this information to improve their security postures against such attacks.

Event Table: The Event Table module offers a well-defined structure to present a list of all events relating to firewalls stored in the system's DB (database). Each entry contains attributes like the time/date stamp, source IP address, destination Port, protocol used, severity level and the type of detected attack. The table allows for easy inspection and filtering of security events.

Suspicious IP Reputation: The Suspicious IP Reputation module measures and rates the level of trust that an IP address has on the network. At the same time, the system tracks IPs that have been consistently causing high numbers of malicious events and then assigns these IPs with a flag marking them as suspicious. This information helps analysts to identify any potential repeating threat actors.

Attacker Persistence: In the proposed system, the Attacker Persistence module tracks the number of repeated connection attempt from a single IP source. Through measuring how often and how long an attack is made, the system can determine whether that source IP is a persistent attacker against the network.

IP Intelligence: The IP Intelligence module collects contextually relevant data on the attacking IP address. This can include information about where the attacking IP is physically located, which network provider is supplying the service to the attacker's IP, and the perceived threat level associated with that IP address. This contextual data improves situational awareness for the incident response team and helps them make more informed decisions during their response to an incident.

Fig: 4.6 Threat Intelligence Card

Targeted Port Analysis: The Targeted Port Analysis module finds the most attacked network ports. By analysing firewall logs' destination port field, the module can determine which services attackers may be trying to test or attack.

Protocol Distribution: The Protocol Distribution module looks at the communication protocols used during events on the network. By reviewing the firewall logs, the module can verify whether or not the events were TCP, UD, or ICMP. Understanding protocol distribution can help identify abnormal traffic patterns.

Service Exposure Analysis: The Service Exposure Analysis module determines which services are most exposed to attempts at external access. By connecting targeted ports to service types, the module can identify those services that may require additional security hardening.

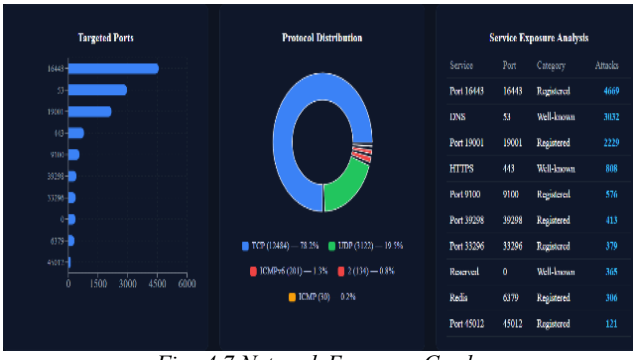

Fig: 4.7 Network Exposure Cards

Attack Trend: The Attack Trend module tracks long-term changes in attack activity. It will analyse historical event data to determine whether the number of attacks over a defined period of time has increased, decreased or remained constant.

Attack Growth: The Attack Growth module calculates the rate of increase in the level of attack activity over a defined period of time. This provides security analysts with a metric for detecting the sudden acceleration of attack campaigns.

Attack Heatmap: The Attack Heatmap Module records times and places (over time) where high volumes of attacks are occurring using a color-coded display.

Attack Velocity: The Attack Velocity Module is used to rate the number of attacks that occur in very short periods of time and can be used to identify an attack that is happening due to an automated attack campaign or a bot.

Fig: 4.8 Temporal Analysis

Attack Intensity: The Attack Intensity Module is used to determine how many attacks are concentrated on a particular system during a defined period of time, with very high attack intensity values indicating either large-scale or coordinated attempts to attack.

Attack Correlation: The Attack Correlation Module identifies how different security events relate to one another. Through the analysis of event attributes such as their source (IP), port (target), and the time (timestamp), this correlations module can correlate and identify events that come from the same attack campaign or individual attack.

Fig: 4.9 Attack Behaviour

Anomaly Detection: The Anomaly Detection Module identifies unusual patterns of traffic at the firewall. The anomaly detection module takes the current activity and compares it to historical activity, using the historical activity to detect any differences that could indicate an undetected or new form of attack.

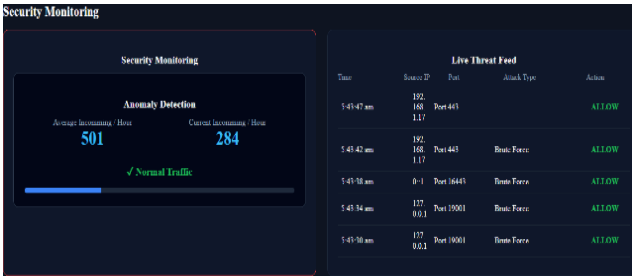

Fig: 4.10 Security Monitoring

5. Experimental Results and Discussion

A controlled laboratory environment was used to evaluate the MiniSIEM: A Log Analysis and Security monitoring System for firewall log monitoring in real-time, attack detection, and security analyticsvisualisation. The experimental setup included the following: An attacker machine generating simulated malicious traffic, a firewall machine running Ubuntu OS configured with Uncomplicated Firewall (UFW) and a host machine managing both the collecting of logs and processing of analytics. Several attack scenarios were created to assess the system's functionality, performance and reliability.

A. Functionality and Correctness

Functional testing was performed in order to establish if the solution would accurately retrieve, store and analyse firewall logs in real-time. The log collector retrieved firewalls logs from the Ubuntu Server using secure SSH communications by forwarding them to the backend processing engine. The log parser created structured security events from the firewall logs by extracting key attributes including source IP address, destination port, protocol type, timestamp and action performed by the firewall.

Multiple attack scenarios (i.e., port scans, repeated connection attempts and abnormal bursts of requests) were simulated as part of the evaluation of the detection capability. The rule classification engine identified each of these attacks correctly based upon the defined detection rules. The classification engine in the proposed system identifies sequential connection attempts from a single IP source across multiple ports as port scanning, and brute-force where repeated attempts to access the same service as potential behaviour of it.

The system generated alerts based upon the detected suspicious activity and displayed the alerts via a real-time dashboard. The support of the visualization modules (e.g., attack timeline, severity distribution and attack types) displayed the processed events with high accuracy. It is clear from these results that the system can interpret firewall logs with accuracy and provide actionable cybersecurity insight based upon them.

B. Performance and Scalability

The evaluation of how well the system processes immense amounts of firewall logs placed priority upon whether or not it does so while responding in real-time. The backend processing engine was that it accomplishes this task by being subjected to continuous logging conditions, along with simulated attack traffic, as opposed to logs simply being processed at certain discrete intervals. The evaluation indicated all log ingestion and processing could occur with almost no delay.

The log parsing function and the event classification function performed well when processing multiple logs/events at the same time. The backend of this application is developed with a lightweight architecture utilizing Flask and SQLite allowing for high performance and low system resource utilization. The dashboard user interface is also created with React, and displays security events (visually) at "near real-time" updates that refresh every five seconds.

Because of its modular architecture, this solution can be expanded and adapted to support new types of detection modules and/or analytic modules for use with firewalls and firewall logs (shown below). The design of the system allows for these new modules to be added without affecting the core functionality of the system as a whole, therefore, making it easy to use in high volume, continuously generated, real-time firewall log monitoring environments.

C. Security and Reliability

The objective was to conduct a security evaluation to assess both the soundness of the authentication mechanisms utilized by the system and the reliability of the alert generation process. There is a JSON web token (JWT)-based authentication system in place that prevents unauthorized users from accessing the analytics dashboard.

Based on test results, unauthorized access attempts were blocked successfully. Communication channels between the application and server were appropriately secured so that data could be considered safe.

An alert generation system was designed to be capable of accurately detecting and reporting suspicious activity. For example, anytime the classification engine detected a possible attack pattern, the alert generation system produced an alert and created an entry in the event record. All alerts were shown immediately on both the live event feed and in alert panels within the analytics dashboard.

In addition to the real-time alert generation capabilities, the event record provides a historical record of security-related events that are available for use by security analysts as reference material when conducting investigations into past incidents or for evaluating patterns associated with cyber-attacks over an extended period of time. The reliable storage and retrieval of event records ensure that security-related data can be used as evidence in forensic investigations and incident response efforts.

6. Future Work

The proposed MiniSIEM: A Log Analysis & Security Monitoring System provides a solid foundation for real-time firewall monitoring, attack detection, and security analytics; nonetheless, there are many possible improvements that could make the system work better and allow it to do more. For example, one area of potential future development would be to integrate machine learning (ML) and artificial intelligence (AI) techniques into the attack classification engine, which would allow for intelligent detection of complex and unknown patterns of attacks. Through the use of historical log data collected from firewalls, ML models could discover abnormal behaviour and patterns of behaviour that traditional, rule-based detection systems would not have found. This would improve accuracy of detections and decrease false positive detections. In addition, the system could potentially support third-party threat intelligence sources (feeds) and third-party (IP) reputation services to provide additional contextual information about potentially malicious IP addresses, resulting in more accurate threat assessments. Future versions of the platform could also support automated incident response processes (e.g., updating firewall rules to block malicious IP addresses in real-time or temporarily restricting (black-listing) traffic from an IP address if that IP address has shown abnormal behaviour) and allowing automated incident response actions. Lastly, additional functionality could be added to the system to support multiple firewall vendors, distributed log collection from large enterprise networks, and providing additional forms of advanced analytics (e.g., predictive modelling of threats and long-term trends of attacks). The enhancements proposed for establishing an extensive, intelligent cybersecurity monitoring framework would enable the implementation of a much larger and more dynamic monitoring platform that would adequately fulfill today's security operations centers growing requirements for a robust technology base to support their operational needs.

7. Conclusion

The proposed Firewall Security Analytics System is aimed at providing improved methods of enhancing the monitoring of network security through real-time analysis of firewall logs along with smart rendering. Automated log collection; structured event processing; attack classification by rules; event-trigger alerts; and support for interactive analytics dashboards are all features that provide a means to convert general firewall log data into high-quality cybersecurity intelligence. The established architecture provides for continuous monitoring of network activities, while also providing the security administrator with an overview of attack patterns, threat levels, and potential unsafe behaviours. Through the use of the analytical modules of attack timeline, severity distribution, attacker IP analysis, and geographic location based risk visualization of attacks, the administrator is able to quickly identify anomalies and respond to potential cybersecurity incidents. Both the performance based evaluation and test results show that the system is effective at processing firewall logs to provide high degrees of accuracy for identifying suspicious activities, ultimately resulting in actionable information being provided through real-time dashboards. The proposed modular/scalable architecture will allow it to be adaptable as the requirements associated with cybersecurity continue to evolve and to support additional analytical functionality over a more extended period of time. This proposed solution offers an inexpensive, practical framework for enhancing the effectiveness of firewall-based threat detection, thereby enhancing the effectiveness of proactive cyber security defense mechanisms for today’s networks.

Funding Declaration

The authors received no financial support for the research, authorship, and/or publication of this article.

REFERENCES

1. S. Partovian et. al., “Analysis of Log Files to Enable Smart-Troubleshooting in Industry 4.0: A Systematic Mapping Study,” IEEE Access, vol. 12, pp. 147640–147660, 2024.
2. T. Marlaithong et. al., “A Log Parsing Framework for ALICE O2 Facilities,” IEEE Access, vol. 11, pp. 69439–69457, 2023.
3. M. Raeiszadeh et. al., “ALogSCAN: A Self-Supervised Dual Network for Adaptive and Timely Log Anomaly Detection in Clouds,” IEEE Transactions on Machine Learning in Communications and Networking, 2025.
4. A. Horváth et. Al., “Anomaly Detection Algorithms for Real-Time Log Data Analysis at Scale,” IEEE Access, vol. 13, 2025.
5. A. Alzamil et. al., “DualBERT: Fusing Symbolic and Temporal Dynamics for High-Precision Log Anomaly Detection,” IEEE Access, vol. 14, 2026.
6. Y. Aboudrar et. al., “AI-Driven Firewall Log Analysis: Enhancing Threat Detection with Deep Learning Techniques,” International Journal of Advanced Computer Science and Applications, vol. 16, no. 7, pp. 808–815, 2025.
7. M. Schufrin et. al., “Visual Firewall Log Analysis – At the Border between Analytical and Appealing,” IEEE Visualization Workshop, 2018.
8. J. K. Lee et. al., “Traffic and Overhead Analysis of Applied Pre-filtering ACL Firewall on HPC Service Network,” Journal of Communications and Networks, vol. 23, no. 3, pp. 192–200, 2021.
9. L. Durante et. al., “A Formal Model and Technique to Redistribute the Packet Filtering Load in Multiple Firewall Networks,” IEEE Transactions on Information Forensics and Security, vol. 16, pp. 2637–2648, 2021.
10. G. Black et. al., “Descriptor: Firewall Attack Detections and Extractions (FADE),” IEEE Data Descriptions, 2025.