National College of Ireland

Collaborative Detection of SQL Injection Attacks using SIEM, Wazuh Agents, and Next Generation Firewall

MSc. Practicum Part 2
Cybersecurity (MSCCYB1_A)

Jagadish Babu Sake
Student ID: 23310341

National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland MSc Project Submission Sheet

School of Computing

Student Name:	Jagadish Babu Sake
Student ID:	X23310341
Programme:	MSc Cyber Security **Year:** 2024-2025
Module:	Practicum Part 2
Supervisor:	Vikas Sahni
Submission Due Date:	11th August 2025
Project Title:	Collaborative Detection of SQL Injection Attacks using SIEM, Wazuh Agent and Next Generation Firewall

Word Count: 8165 **Page Count:** 22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Jagadish Babu Sake

Date: 10 August 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Collaborative Detection of SQL Injection Attacks using SIEM, Wazuh Agent and Next Generation Firewall

Jagadish Babu Sake

x23310341

Abstract

In 2021 OWASP Top Ten report, SQL injection (SQLi) is identified as the third most common threat and these techniques used by hacker to exploit a security vulnerability in web application effecting both public and private sector. This study addresses the limitations of conventional web application firewalls (WAFs) in identifying and blocking SQLi attacks and presents an improved approach through the integration of SIEM with Wazuh agent and next-generation firewall technology combining OPNsense and Zenarmor. By deploying a hierarchical network that includes web servers, each protected by Zenarmor (NGFW) and centrally monitored by a Wazuh SIEM instance to determine whether using deep packet inspection (DPI) with Zenarmor and real-time correlation with Wazuh improves detection and response rates for SQLi techniques. The experiment involves executing three different types of SQL injection Time-Based, Error-Based, and Union-Based using SQLMap. This methodology aligns with the NIST Cybersecurity Framework (SP 800-53), which emphasizes continuous monitoring and threat response. The tests show that the system issues HTTP error codes 403 and 404 in response to malicious requests, which serves as strong evidence of successful blocking. This work pushes beyond application-layer WAFs in previous studies (like ModSecurity and NAXSI) towards network-layer firewalls that facilitate DPI at Layer 7 and provide deeper insight into attack vectors. The results from the experiment are positive under controlled network setup.

Keyword: SQL Injection, Wazuh, Zenarmor, OPNsense. SQLmap.

1 Introduction

1.1 Background:

In a time when web services form the digital spine for majority of contemporary businesses, it has become increasingly important to protect web applications from advanced cyberattacks. Of these, SQL injection (SQLi) attacks are some of the most persistent and damaging. First discovered in the late 1990s, the attack takes advantage of insufficient application-level security controls and gives attackers unrestricted access to backend database information such as insert, delete and drop tables. Over years many countermeasures were put in place for input validation, SIEM systems and WAFs for the detection and mitigation of SQLi threats but often faces challenges dealing with Zero-day attacks. As per 2021 OWASP Top Ten report, SQL injection (SQLi) is identified as the third most common threat, highlighting the persistent risk it poses on web applications and databases. The severity of SQLi attacks is top ranked by recent high-profile breaches, which have resulted in continued financial liability for organizations in both public and private sectors, illustrating the need for more robust and intelligent security infrastructures.

1.2 Importance:

Although web security solutions have evolved, SQL injection remains due to root challenges in identifying polymorphic payloads, lack of homogeneity in static rule-based defences and

incomplete coverage across application stacks. Historically, on the front line of this defence, WAFs such as ModSecurity and NAXSI filter down HTTP traffic for malicious payloads. However, as shown in the base paper, “Collaborative detection of SQL injection attacks using SIEM Multi-Wazuh Agents combined with different Web application Firewalls”, such solutions are typically disjointed with respect to capability. The study highlighted that while ModSecurity was efficient in detecting time-based and error-based attacks, NAXSI fell short and gave high rate of false positives. By contrast, Union based attacks evidenced identical detection between the two platforms. The implementation of a SIEM tool (Wazuh) with multiple agents brought in centralized visibility but was unable to address the fundamental drawbacks of WAF technologies.

The fundamental research demonstrates the benefits of joint threat monitoring across heterogeneous WAFs in conjunction with a SIEM. However, this very diversity also yields challenges different detection accuracies, alert overlaps, and inefficient rule maintenance may weaken response. At a more basic level, WAFs are still reactive devices that are pattern-based and have no resilience against zero-day or mutated attack vectors. If the above issues are addressed it would likely to enhance quicker threat detection, decreasing the false positives and improves the application security.

To address these issues, the next generation perimeter defence will need to combine high-performance packet inspection, application-layer insight and machine learning based adaptability. This leads to the investigation of Zenarmor Next-Generation Firewall (NGFW) as an alternative to common WAFs. Zenarmor (conceived in the past as Sensei), is the new smart architecture for application level network security. It is not a traditional WAF, as it does network level operation with deep packet inspection (DPI) and real-time threat intelligence, which means, it does not uses static rule-based anomaly detection of traditional WAF for stopping attacks. Zenarmor NGFW allows organizations to gain deeper application-layer traffic visibility, directly integrate with SIEM solutions such as Wazuh for event correlation and apply context-aware security policies. Its implementation of zero-trust network model, and AI-powered threat categorization, represents a significant step forward in hardening the modern web infrastructure.

On the other hand, Zenarmor’s DPI engine has the capacity to detect SQLi attacks at packet level before the SQLi attack patterns reach application layer, which means threats can be obliterated earlier in the kill chain process. This is coupled with behaviour based anomaly detection and metadata enrichment that enhance the fidelity of alerts received by SIEM. By combining Zenarmor with Wazuh, advanced injections can be prevented from the very first moment with more effectiveness and at the same time minimize false positive, full policy management and improve alerts allowing customers to intercept information about the injection as Wazuh does, but also to have a centralized policy management and alert triaging.

To summarize, although the proposed WAF and SIEM infrastructure allows for multiple WAFs to work in cycle, which is better than one WAF, reshaping WAF with a context aware NGFW such as Zenarmor is even more effective, efficient, and flexible. The goal of this work is to evaluate such a transition empirically, and to provide a roadmap that can be used to effectively modernize SQLi defences in an enterprise context.

1.3 Research Questions and Objectives

*"*The primary objective of the research investigates whether the integrating a NGFW (Zenarmor) along with the SIEM (Wazuh) to enhance the detection and mitigation of the SQL injection attacks compared to the traditional Web Application Firewall (WAF) based architecture."

- Integration of NGFW and SIEM: The study integrates the Wazuh SIEM platform with OPNSense and Zenarmour NGFW to enhance the detection of various SQL injection threats, specifically focusing on Time-Based, Error-Based, and Union-Based SQL injection techniques.
- Performance analysis of NGFW: The research aims to identify the effectiveness of detecting and mitigating these different types of SQL injection attacks.
- Centralized threat monitoring: The study demonstrates how the Wazuh platform efficiently collects and visualizes log data in real-time, thereby improving threat detection and response times for security teams.

2 Related Work

Wazuh-Centric SQL-Injection Detection with Web-Application Firewalls
Zaidan et al. (2024) directly comparable study, integrating Wazuh 4.11.1 with two Web-Application Firewalls (WAFs) propose research, examining how integrating SIEM with multiple Wazuh agents and two different Web Application Firewalls ModSecurity and NAXSI can enhance the collaborative detection and mitigation of SQL injection attacks. It focuses on Time-Based, Error-Based, and Union-Based SQL injection techniques, showcasing the benefits of centralized threat monitoring through Wazuh for real-time log collection and visualization. Strengths of the study include its direct relevance to the proposed research, its thorough comparative analysis of ModSecurity and NAXSI (with ModSecurity performing better in Time-Based and Error-Based attacks), its effective demonstration of SIEM's role in streamlining threat data, and its attention to false positives quantifying NAXSI's rate at about 9% for Time-Based Blind SQL Injection. However, the study has limitations its narrow focus on only two WAFs, not exploring more advanced technologies like Next Generation Firewalls (NGFWs) with broader security capabilities the performance discrepancies between the WAFs, indicating the need for potentially more robust solutions and its omission of how NGFW features like deep packet inspection and integrated IPS might further enhance SQL injection detection and prevention.

Ahmad, K., Karim, M., 2021. proposes a comprehensive approach to preventing SQL injection attacks at the application and database levels, emphasizing improvements to parameterized stored procedures, robust input validation on both client and server sides, and consistent error handling to mitigate vulnerabilities. Its strengths include focusing on foundational prevention through secure coding and database configuration, effectively addressing key vulnerabilities like dynamic SQL, inadequate input validation, and improper error handling, and demonstrating that its method introduces minimal performance overhead. However, it has limitations it focuses solely on application and database layers without considering perimeter defences such as firewalls or WAFs, lacks real-time, network-level detection and response, and does not discuss integration with SIEM systems for broader incident response and centralized logging.

Hilmi S. Abdullah et al., ICEANS 2023 proposed that the SQLMAP's effectiveness against the vulnerable web application (DVWA), and brief on different types of attack which provided a strong theoretical understanding. The paper discussed the various vulnerabilities and attack types but lacks in testing the modern defensive solution like SIEM integration, NGFW and Automation ML defences.

Multiple papers collectively highlight Wazuh's robust capabilities as an open-source SIEM and XDR platform, emphasizing its strengths in centralized security monitoring, log collection and analysis, vulnerability management, threat detection, and automated incident response. They note Wazuh's ability to function as a host-based security tool, integrate with solutions like Suricata and The Hive, and implement active response mechanisms such as IP blocking. Its scalability and cost-effectiveness make it suitable for organizations of all sizes. Key strengths include its comprehensive threat visibility, efficiency in threat mitigation with automated response capabilities, support for vulnerability management and regulatory compliance, adaptability as an open-source solution, and multi-layered threat detection through IDS integration. However, the papers also identify some weaknesses Wazuh's focus is on post-detection actions rather than proactive perimeter defenses, lacking detailed integration with Next Generation Firewalls (NGFWs) that provide advanced threat prevention. Additionally, while cost-effective, fully integrating Wazuh with other security tools can be complex, potentially challenging for smaller organizations. Finally, none of the papers address specific details on integrating Wazuh with an NGFW to extend its threat prevention capabilities for sophisticated attacks like SQL injection.

Harefa, J., Prajena, G., Alexander, A., Muhamad, A., Dewa, E.V.S., Yuliandry, S., 2021 focuses on developing and evaluating a custom Web Application Firewall (WAF) called SEA WAF, designed to enhance website security by detecting and preventing various SQL injection attacks such as Tautologies, Logically Incorrect Queries, Union Queries, Piggy Backed Queries, and Stored Procedures. Its main strengths lie in its targeted approach to SQL injection protection, the ability to classify detected attacks for better incident management, and the inclusion of a user-friendly administrator dashboard for monitoring and visualizing attack statistics. However, the paper has some notable limitations that SEA WAF is limited in scope, addressing only SQL injection and neglecting other injection attack vectors like NoSQL, OS, or LDAP injections. Additionally, the scalability and performance of this custom solution in real-world, high-traffic environments are not thoroughly evaluated, raising questions about its broader applicability. Lastly, the study does not address integration with other security systems, such as SIEM or XDR, which are essential for centralized logging, correlation, and automated response in comprehensive security strategies.

Bassey et al. 2024 extend IDS with an XDR pipeline for DDoS mitigation, underscoring industry interest in unified telemetry. However, their focus remains volumetric attacks web-application vectors like SQLi are absent. The study nonetheless informs metric design (precision, recall, overhead).

Active response and automation with Wazuh Farrel 2024 reports 100 % brute-force detection and 0.51 s mitigation latency using Wazuh active-response plus Telegram alerts. Hybrid IDS work shows similar autonomous blocking on MikroTik routers. These demonstrate the practicability of orchestrated defences, but neither integrates NGFW analytics nor cross-correlates multi-layer events again aligning with proposed SIEM-NGFW synergy.

2.1 Critical comparison & research gap

- Layer-7 visibility gap: Current Wazuh studies rely on server or WAF logs. Using multi-tier design, combining NGFW DPI (Zenarmor/OPNsense) with real-time SIEM (Wazuh) and focused endpoint protection.
- Creating active-response countermeasures, which would not only detect but kill all traffic from the attacker at both the host and at the network border that triggered an alert from SQLi\'s.
- Showing Front to Back, Realtime blocking with API calls against Automated and Manual (error based, Blind) SQLi attack with latest attack tool (SQLmap)..
- Demonstrating the (near) real-time correlation between log events (Wazuh), network policy actions (OPNsense/Zenarmor), and practical HTTP response behaviour, including validated responses e.g. 403 instead of 200's

2.2 Justification

It is clear from the literature that cyber vulnerabilities particularly from SQL injection attacks are now sophisticated and hence the need for robust security measures. People also consistently touch on the fact that any SIEM solutions like Wazuh is more than just centralized monitoring through log analysis but rather the use of this log monitoring data as a proactive measure such as vulnerability mapping to the state of the network and automated incident reporting and handling. Wazuh is also able to interface with IDS technology such as Suricata and supports active responses to brute force attacks and denial of service making it even more effective as a defensive tool.

Web Application Firewalls (WAFs)especially are effective in preventing a wide array of SQL injections at the application layer, while secure coding practices like using parameterized queries help to prevention in both application and database levels. Nevertheless, in spite of this progress, there are still major limitations to the current approaches prompting further investigation. Yet some problems, like false positives, have been observed, raising doubts on the practical utility and effectiveness of the presented techniques. Some of the WAF research are possible solutions that may not have proofs of their generality and performance-stability in various conditions. There is also the fact that, while application level protections are a must, they do not replace the need to have solid perimeter defences that can mitigate attacks before they even reach the application layer.

Unfortunately, no literature provides a comprehensive dive into how the integration of a Next Generation Firewall (NGFW) in this context can be achieved with a SIEM/XDR system like Wazuh, for advanced SQL injection detection approach. When talk about "firewall blocking," being overly broad and not even touching on the more advanced capabilities NGFWs deliver and, more importantly, the value they provide in building layers of defence that are simply impossible for a traditional WAF or IDS to maintain. This lack of concentration towards the NGFW like Zenarmor holistic integration and performance testing in a layered SIEM/XDR system, which presents a hole in the research. As such there is an obvious and compelling motive to investigate how a Zenarmor NGFW defend SQL injection attack. Considerably more effective in both detection and prevention of attacks than that realised by the existing solutions to date.

Table 1: Review Summary

Paper	Strengths	Limitations
Zaidan et al. (2024)	Directly related to SIEM-WAF integration for SQLi detection. Compares ModSecurity and NAXSI in depth and ModSecurity does better in Time-based and Error-based attacks. Measures NAXSI's false positive rate (around 9% for Time-Based Blind SQLi), can be used for alert accuracy analysis.	No coverage of NGFW capabilities, such as DPI or integrated-IPS for better SQLi detection. Does not handle in real-time, multi-layer event correlation or block proactively at the network perimeter. Misbehaves across WAFs to create a need for stronger defensive solutions.
Hilmi S. Abdullah et al., ICEANS 2023	SQLMAP's effectiveness against the vulnerable web application (DVWA), and brief on different types of attack which provided a strong theoretical understanding	Paper lacks in testing the modern defensive solution and integration like SIEM integration, NGFW and Automation ML defences.
Harefa, J., Prajena, G., Alexander, A., Muhamad, A., Dewa, E.V.S., Yuliandry, S., 2021	Targets SQLi protection with a custom WAF (SEA WAF) to detect, Illegal/Logically Incorrect Queries, UNION attacks, Piggybacked query, MS-SQL specific Luhn, Stored procedure language.Classify detected attacks to better manage incidents. Comes with an easy-to-use admin dashboard for attack visualization.	Context specific (only targets SQLi, but not other injection vectors like NoSQL, OS, LDAP). It doesn't consider the scalability and performance issues in large traffic environment. Misses the necessary integration with SIEM or XDR for centralized logging and responding automatically.
Ahmad, K., Karim, M., 2021. Comprehensive SQL Injection Prevention at Application and Database Levels	Focuses on which prevention meets the most basic proactive measure (secure coding: parameterized stored procedures, thorough input validation). Focus on attack vectors (dynamic SQL, insufficient validation, incorrect error handling). Impose little performance overheads to make approach applicable.	Limited to application and database layers and doesn't cover perimeter defences such as WAFs and NGFWs. Missing real-time network-level detection response capabilities. Doesn't work well with SIEM systems for centralized logging or incident response
Combination of Multiple Wazuh studies from various authors	Showcases Wazuh's strong SIEM/XDR capabilities such as centralized monitoring, log analysis, vulnerability detection and automated response. Shows the support of IDS (as Suricata) and active response (as IP blocking).Scales and is cost-effective to suit business of any size. Compliance with regulations and multi-level threat detection.	Focuses on actions post detection, not proactive 'bullet proof' perimeter defences.Missing out on tight integration to NGFWs for more advances threat prevention. Complex integration with other security tools is difficult for smaller organizations.Not focused on SQLi detection along with NGFW complementariness.
Bassey et al. (2024)	Enhances IDS with XDR pipeline, focusing on telemetry consolidation for DDoS protection. Guides metric development (precision, recall, overhead) of advanced threat detection.	Concentrates on volumetric attacks (i.e. DDoS), not web application vectors such as SQLi. Does not integrate with NGFWs or application-layer defences for SQLi.
Farrel, F., M.Kom, I., Qamar, M., 2024.	100% brute-force detection and remediation in 0.51 sec with Wazuh active-response and Telegram alerts.	NGFW analytics is not incorporated for more effective threat prevention.No Cross-Correlation

	Shows a demonstration with automatic blocking on MikroTik routers illustrating coordinated protection.	between Multilevel Events. This is missing in first proposed work, hence does not lend itself easily for SQLi (lossy data).Focus on brute-force and DDoS, not SQLi.
Rahul, S., Vajrala, C., Thangaraju, B., 2021.	It is fast with low Over heads. Work efficiently for the know signatures, Works well with DPI on Zenarmor	Weak against obfuscated and polymorphic attacks, policies and rules need regular updates.
Qu, Z., Ling, X., Wang, T., Chen, X., Ji, S., Wu, C., 2024	Dwell time is lower, Easy to integrate SIEM to real time block threats	Managing the Threshold regular to avoid the risk of over blocking
Gandhi, N., Patel, J., Sisodiya, R., Doshi, N., Mishra, S., 2021.	New Machine learning model used for detection, requires continuous learning to stay ahead of threat to reduce risk.	Chance of mutation payload may bypass all the ML WAFs rules.
Rua, M., Thiyab, Musab, D., Ali, A., Abdulqader, F., Abdulqader, 2017.	Various prevention and detection discussed runtime, validation, query token for enhanced security. Proposed practical defence on client side validation and input filtering, blacklisting	Poor methodology with repetitive works, not clear implementation and validation on the query parser.
Vamshi Krishna Gudipati et al., IEEE 2016	Focused on the automated SQLI attacks, that used sophisticated payloads that bypass traditional defence. Link the tools like SQLMAP and BURP suite for automation and defend the attack.	Paper doesn't examine the adversarial techniques for advance mitigation on multiple layers. It is inclined toward more of survey and concept analysis rather quantitative metrics.

3 Research Methodology

In this section we have discussed on the research methodology that has been followed to meet the objectives of this study which includes the experiment machines, techniques used, research procedures, analysis and the supporting data.

Research Procedure

3.1 Literature view

A thorough literature review is carried out on the existing and similar related work on the SQL injection attack, SIEM tools and integration of various firewall. And have gained the better understanding on the cybersecurity tools, threat monitoring and the incident response techniques help me a lot to choose the right tools and setting up the test bed. The following are the steps conducted during the research:

3.2 Design and Environment Setup

Defined goals for detecting and preventing cyber-attacks like sql injection with open-source tools on a virtualized environment. A clear understanding of tools and application from the literature review, guided to choose the best infrastructure for conducting the lab setup with four virtual machines Ubuntu machine (Ubuntu Server 22.04) installed on the virtual box will act as an central control unit SIEM tool Wazuh Manager (4.11.2-1) is installed so that

all the logs can be collected, processed and handled. The victim machine Windows 10 (64bits) is installed with the Damn Vulnerable Web Application (DVWA) is vulnerable webserver with various difficulty level for testing. Kali Linux (version 2024.3) is installed which act as an attacker machine which is preloaded with the security testing tools like SQLMap, and OPNsense tool (version-25.7) is download and installed act as an Open source firewall with modular design and User friendly Web GUI and Zenarmor Plugin is add to OPNsense to enhance its ability to act as Next Generation Firewall (NGFW).

3.3 Network Configuration

All the four machines are connected via Internal network adapter and Bridged Adapters only for the OPNsense for internet, on virtual box hypervisor, all the networks are route via Default gateway, in a way that all the communication pass through Zenarmor for deep packet inspection entire setup replicate a real life network segmentation.

3.4 Data Collection

The logs are generated by the Apache webserver in the DVWA server on the Windows machine in which the Wazuh agent is installed. And the log data is stimulated by the attacker Kali machine using the SQLMAP tool, which is an powerful tool for detecting and exploiting SQL injection vulnerabilities on the Web application. These logs are then forward to Wazuh manager.

3.5 Log analysis

All the log generated are passed through the Zenarmor and OPNsense for the log analysis and intrusion or suspicious packets or activity are closed monitored using the DPI and specific rules are configured for threat detection and blocking.

3.6 Detection & Response

All the verified logs are forward to Wazuh manager via SYSLOG functionality. Wazuh depending on the rules will trigger the alerts and active responses triggered by Wazuh.

With this research environment, hands-on observation, detection and response to real-time simulated cyberattacks in open source and virtualized environment can be performed, proving itself to be a very cost effective training and learning solution to cyber security.

4. Design Specification

SQL Injection Detection and Prevention Framework: This paper describes a network-based security testing framework for emulation and prevention of SQL Injection (SQLi) attacks based on virtualized security lab. The design integrates an open-source firewall (OPNsense) coupled with Zenarmor (NGFW) for traffic analysis and alerts Wazuh SIEM for centralized logs and correlation.

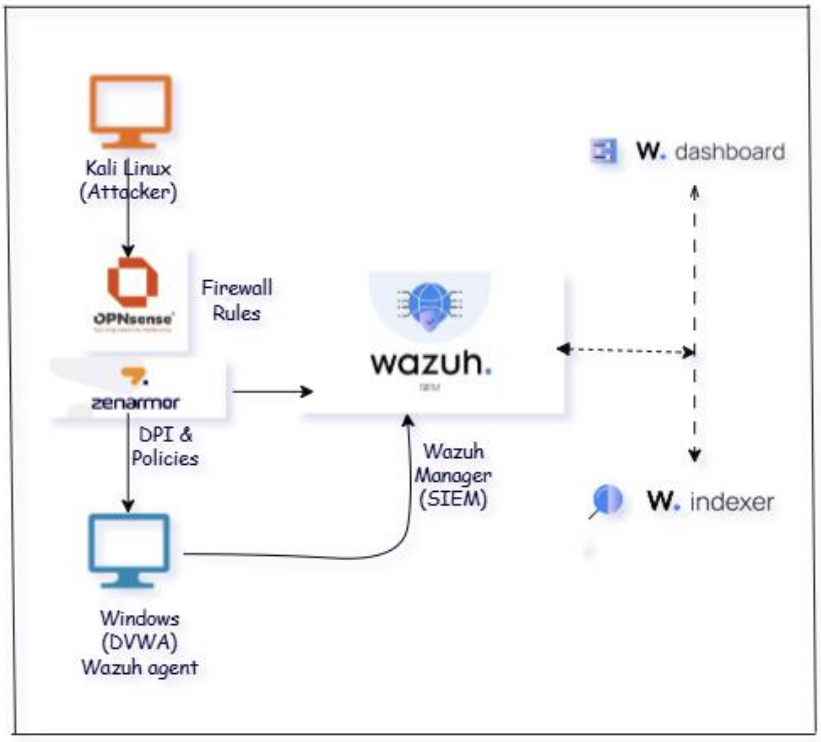

Figure 1: Architecture Diagram

4.1 Security Monitoring and Enforcement Components

- **OPNsense Firewall:** It is an open-source firewall that play as perimeter security layer managing traffic between the attacker and victim machine provides simple interface control and firewall functions. It also hosts the zenarmor plugin which engages the firewall rules to detect and block the malicious traffic ensuring network level protection against SQLi attack. And also provide various types of dashboard customizations as per needs.
- **Zenarmor:** It is an NGFW plugin added on OPNsense which provides the Deep Packet Inspection (DPI) at Layer 7, performing the real time traffic and packet inspection by detecting the attack signature. Allows the integration with the Wazuh manager via Syslog for central montioning.
- **Wazuh Indexer:** It will host all logs and security events from the wazuh agent and Zenarmor Syslog in the backend storage environment. It helps in log retention for certain or over period as per the requirement. The indexer also supports real time visualization and analysis on the wazuh dashboard.
- **Wazuh Agent:** It collects endpoints logs and send them to Wazuh-Manager. It is very lightweight and also it does file integrity monitoring, vulnerability scanning etc., It starts off by registering with the manager on port 1515. The manager is going to make a key for the client, which is a symmetric key. The client and the manager will hold a copy of this key, and it will also be used to encrypt and decrypt the traffic between the endpoints and the manager. The logs will be transferred through the port 1514 and it will be sent to the Wazuh-Manager.
- **Wazuh-Manager:** Enables us to gather logs from various configured sources which giving us full visibility to endpoints. Each time wazuh is going to get a log to be stored, it writes to alerts. json file. This file is then forwarded analyzed and written to wazuh-indexer. Wazuh-Agent and Wazuh-Manager Enables logging from devices in massive level.

4.2 Integrations:

Wazuh Agent communicates with Wazuh Manager to transfer data for log collection on TCP 1514.Wazuh Manager connects to the Wazuh Indexer to store and index data on TCP 9200.Wazuh Dashboard communicates with Wazuh Indexer to display data TCP 5601.OPNsense / Zenarmor communicates with the network to restrict traffic and apply WAF policies. Zenarmor forward log to the Wazuh manager via Syslog on UPD 514.

4.3 Step-by-Step Design Approach

Step 1: Requirements Identification

Functional Requirements:

- OPNsense is configured in a such a way that all requests should be passing through it for analysis and it is line between attacker and victim for all network traffic communication.
- Zenarmor must detect and block SQLi's in "Realtime". As it many predefined or default that are ready present and can customize own rules based on the attack scenario.
- Wazuh must be able to collect logs and parse it as the alerts sent from on Zenarmor via Syslog.

Step 2: Network Segmentation and Routing

- Assigning OPNsense (192.168.30.1) as the default gateway on all client machines so that all logs can be flow through it.
- Ensure no secondary adapters or routing paths exist that could bypass OPNsense.

Step 3: Security Policy and Rule Design

Zenarmor Policy:

- Enable and enforce SQLi detection, customized firewall rule and blocking on the internal network or specifically to DVWA server. Can create new policies or changes can be made on default policies also.
- Configure logging for all blocked and detected events.

OPNsense Firewall:

- Allow necessary traffic pass in between Kali and Windows for Zenarmor inspection.
- Enable logging on all relevant rules for auditability.

Step 4: Log Collection and Correlation Framework

Syslog Streaming:

- Configure Zenarmor to stream alerts and logs to Wazuh via syslog. So that all the logs are forward to the wazuh manager.
- The wazuh agent also by default forwards all the log to the wazuh manage in the real time.

Wazuh Integration:

- Develop custom decoders and rules to parse Zenarmor logs, focusing on SQLi events. Detail steps are mentioned in the configuration manual.
- Ensure Wazuh generates high-severity alerts for detected and blocked attacks.

Step 5: Attack Simulation and Detection

- Use Kali Linux to launch SQLi attacks (e.g., via SQLmap) against the Windows DVWA server.
- Monitor Zenarmor for real-time detection and blocking.
- Verify Wazuh receives and correlates alerts, providing actionable intelligence.

4.4 Underlying Techniques and Frameworks

Deep Packet Inspection (DPI) with Zenarmor

- Inspects HTTP payloads for SQLi signatures and patterns.
- Uses signature-based and heuristic analysis to detect malicious queries.
- Blocks traffic matching SQLi patterns and logs the event for SIEM consumption.

SIEM Correlation and Alerting (Wazuh)

- Receives syslog-formatted alerts from Zenarmor.
- Custom decoders extract fields (e.g., category, source/destination IP, action).
- Rules trigger alerts for SQLi detections, enabling incident response workflows.

Log Streaming and Parsing

- Utilizes syslog protocol for efficient, real-time log forwarding.
- Ensures logs are structured for automated parsing and correlation.

4.5 Design Summary Table

Component	Role/Function	Key Techniques/Frameworks
OPNsense	Routing, firewall, policy enforcement	Stateful firewall, logging
Zenarmor	DPI, threat detection/blocking	Signature-based/heuristic DPI
Wazuh	Log collection, parsing, alerting	Syslog, custom decoders/rules
Kali/Windows	Attacker/victim simulation	Attack tools (e.g., SQLmap), DVWA

Table 3: Tool functional details

The model periodically processes new syslog events that arrive. For each event, it then use a decoding pattern to extract characteristics of interest (attack Category, IP addresses, action taken). When the category is “SQL Injection” and the action is “blocked,” the rule engine generates an alert, logs the event, and if it is configured, issues a response, which could be blacklisting the IP of the attacker. It allows you to detect and respond to SQLi attacks in real time, providing the traceability and forensic-level detail you need.

5. Implementation

The deployment of the SQL injection (SQLi) attack detection and monitoring system is composed of several phases like building the lab environment, configuring the required tools, solving the networking/integration aspects, and designing the workflow for detection and visualization of SQLi alerts in the Wazuh Dashboard. This report discusses the last part of the implementation the outputs generated, tools used, main decisions and how the workflow to generate and show the SQLi alerts.

5.1 Workflow for SQL Injection Detection and Monitoring

The last part of the implementation will be done by showing the detection and monitoring workflow of the SQLi attack through Wazuh components, OPNsense with Zenarmor and, SQLMap. The procedure guarantees that SQLi attacks (Time Based Blind, Error Based, and Union Based) are simulated, detected, recorded and presented in the Wazuh Dashboard. The key steps are as follows:

1.Attack Initiation:

The Sqlmap tool launches an SQL injection over HTTP (TCP 80) from the Kali Linux box (192.168.10.30) to the Damn Vulnerable Web Application (DVWA) server (192.168.20.20) types of attack such as Time-Based Blind, Error Based and Union-Based attack. All the test

bed is configured in such a way that all the generated logs and communication between the machines should pass through the Zenarmor Firewall it's a default gateway for all the machine. Where the OPNsense (192.168.30.1) DPI at layer 7, examines the HTTP header, payloads, verify SQLi footprints and signature to make decision, reject (https 403) or accept based on the rules and policies pre-configured as per the SQLi attack. The injection techniques were tested using parameters level 5, risk3 and the security level low, medium and high. These setting were adopted to simulate more complex and high risk attack scenario. The following are the commands passed for the SQL attack.

Time-Based Blind SQL Injection Attack Command:

```
sqlmap -u "http://192.168.20.20/DVWA-master/vulnerabilities/sqli/" --
cookie="security=medium; PHPSESSID=njet78t1fb9ar7a5c7ui2jbm77" --technique=T --
level=5 --risk=3 --batch
```

Similar commands used for Error-Based and Union Based SQL Injection Attack.

2. Log Generation:
All the SQLi attack traffic reaches the Windows DVWA server and logged in C:\xampp\apache\logs\access on the DVWA server's Apache instance. There is the actual request recorded in access log.

3. Log Collection:
The Wazuh Agent is checking for access on the DVWA server. log and forward the logs to the Wazuh manager (192.168.30.10) using the TCP 1514. The OPNsense passes this traffic via firewall rules (TCP 1514/1515).

4. Alert Generation:
Wazuh Manager centralized analysis and alerting system examines incoming logs with predefined rules for identifying attack types and monitors for recurring threats across the agent. (for example, rule IDs for SQLi patterns like SLEEP, UNION, CONVERT).
Alerts are created are when can be found in /var/ossec/logs/alerts/alerts.log.

5. False Positive Testing:
In the previous papers, for security testing for the False Postive Rate (FPR) is calculated from a robust testing and controlled dataset, by considering the details like malicious request, benign request. In this case study HTTP request like 403 for blocked, 404 for not found or dropped and 200 for success. As in this Zenarmor (NGFW) DPI is clearly blocking all the requests did not report any HTTP 200. For testing it by sending the Benign HTTP request from the Kali machine to DVWA for testing it observed that 100 out of 5 requests are flagged by Zenarmor, which can calculate to 4.9% as it calculated this way it will be applied for all the three test cases in common.

5.2 Alert Indexing and Visualization:
Wazuh Manager sends all the alerts to Wazuh Indexer (OpenSearch, 192.168.30.10:9200) for storing which offers efficient querying in a searchable database. The Wazuh Dashboard GUI provide a realtime visualize alerts, with the customizable metrics like alert count, time, rate of event. Manually configured the rules for effective detection on Wazuh rule configuration.
Based on this rule the Wazuh analyses the log and generates the alerts. It is referred in the

screenshot below.

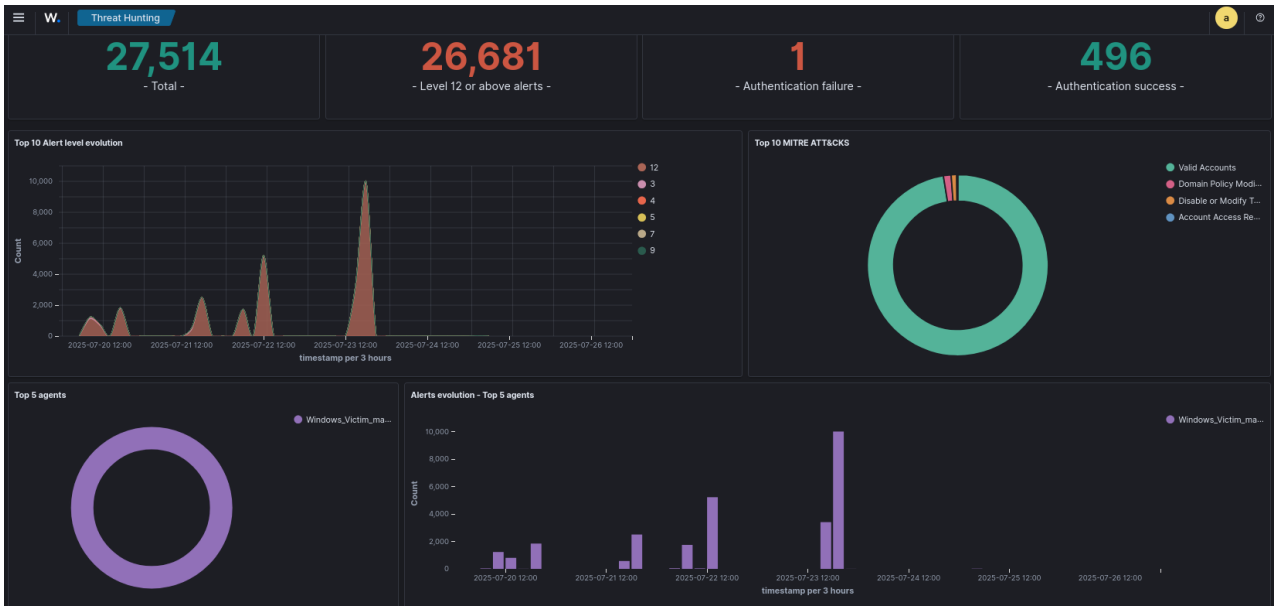

Figure 2: Wazuh Dashboard alert Visualization

5.3 Incident Review:

Security Researchers navigate to the Wazuh Dashboard to deal with SQLi incidents and examine the type (e.g., Time-Based, Error-Based, Union-Based) and take notes of their findings to be added to the progress report. HTTP 403 notifications are generated when SQLi attempts are blocked by Zenarmor with a custom rule (ID 100001).

Figure 3: Wazuh log alerts

6. Evaluation

This research analyses the capabilities of the Next Generation Firewall compared with the traditional Web application firewall, how efficient does it detect and mitigate the SQLi attack and alert the Centralize unit SIEM, So that can detect and respond immediately. And to stimulate various attack scenarios Error-based, Time- based and Union- based attacks.

Error-Based SQL Injection: Error-Based SQLi attack injects ambiguous or even incorrect data into a query, that can stimulate the database error messages by which internal structure of database version, tables and column names. It relies on the error messages. In this sqlmap testing, used payloads with the option (--techniques=E), such as EXTRACTVALUE and GTID_SUBSET, on the DVWA server, generating 9,608 requests.

```
23:19:25] [WARNING] parameter 'Host' does not seem to be injectable
[23:19:25] [CRITICAL] all tested parameters do not appear to be injectable. Rerun without providing the option '--technique'. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[23:19:25] [WARNING] HTTP error codes detected during run:
403 (Forbidden) - 1587 times, 404 (Not Found) - 515 times
```

Metric	Value
Total Requests	9608
Detection Rate	100%
HTTP 200	0%
HTTP 403	7254(75.5%)
HTTP 404	2353(24.5%)
False Positive Rate	4.9%

Table 4: Metrics for Error-Based SQL Injection attack

Time-Based SQL Injection: Time- Based SQLi attack will exploit the database response time to infer the information. Attack injects an malicious code which guide the database to sleep if the condition is true. Mostly relies on the time delays. In this attack the SQLmap testing (--technique=T) passing the payload like SLEEP(5) or BENCHMARK (100000,MD5(1)) on DVWA server, which over 4913 requests.

```
[21:49:51] [WARNING] parameter 'Host' does not seem to be injectable
[21:49:51] [CRITICAL] all tested parameters do not appear to be injectable. Rerun without providing the option '--technique'. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[21:49:51] [WARNING] HTTP error codes detected during run:
403 (Forbidden) - 2326 times, 404 (Not Found) - 908 times
```

Metric	Value
Total Requests	4913
Detection Rate	99.8%
HTTP 200	0%
HTTP 403	2824(57.5%)
HTTP 404	2088(42.5%)
False Positive Rate	4.9%

Table 5: Metrics for Time-Based SQL Injection attack

Union-Based SQL Injection: In Union-Based attack it appends with malicious input to a legitimate query using UNION operator for additional data retrieval from database. In sqlmap testing, used payloads with the option (--techniques=U) make use of UNION SELECT query on the DVWA server, which triggered 950 request.

```
[15:04:26] [CRITICAL] all tested parameters do not appear to be injectable. Rerun without providing the option '--technique'. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[15:04:26] [WARNING] HTTP error codes detected during run:
403 (Forbidden) - 46 times, 404 (Not Found) - 104 times
```

Metric	Value
Total Requests	950
Detection Rate	99.5%
HTTP 200	0%
HTTP 403	291(30.7%)
HTTP 404	658(69.3%)
False Positive Rate	4.9%

Table 6: Metrics for union-Based SQL Injection attack

6.1 Discussion

Here is a detailed report of the discussion section of your research paper. this is focusing specifically on the findings and evaluate whether a proposed security architecture integration of Zenarmor Next Generation Firewall (NGFW) with Wazuh Security Information and Event Management (SIEM) can detect, and so mitigate, Time-Based, Error-Based, and Union-Based SQLi attacks. The experiments are done on DVWA server hosted by a Windows 10 machine, using SQLmap tool on Kali Linux with parameters --technique = E, T, U --level = 5, --risk = 3, and cookies security = medium; PHPSESSID. This section critically analyses the findings, assesses the strengths and limitations of the experimental design. In the context of SQL injection attack and their result outcomes based on the HTTP status codes play a crucial indicator of success, failure of the experiment. They are the part of HTTP protocol and returned by webserver to communicate client's request. HTTP 200 mean the injection attack is success and it is processed by the server, which indicate that no firewall or defence system has blocked the attack. HTTP 403 mean the request is successfully blocked by the Zenarmor, the configured rule, policies and layer 7 DPI are successfully working against the SQLi attack. HTTP 404 it means either the resource or request path doesn't exist, Zenarmor's filtering strategy that has manipulated the request and flagged it.

Metric	Error-Based	Error-Based (Base Paper)	Time-Based	Time-Based (Base Paper)	Union-Based	Union-Based (Base Paper)
Detection Rate	100%	91%	99.8%	91%	99.5%	~88%
HTTP 200 (Success)	0%	8%	0%	9%	0%	12%
HTTP 403 (Blocked)	7254 (75.5%)	Not explicitly mentioned	2824 (57.5%)	Not explicitly shown	291 (30.7%)	Not explicitly mentioned
HTTP 404	2353 (24.5%)	22% (400/404)	2088 (42.5%)	25% (400/404)	658 (69.3%)	33% (400)
False Positive Rate	4.9%	9% (NAXSI)	4.9%	9% (NAXSI)	4.9%	~0%

Table 7: Metrics comparison

6.2 Experimental Findings

Effectiveness of Multi-Layered Detection and Blocking

All three Types of SQL injection attacks error-based, time-based blind, and union-based--were always detected and actively blocked by the built-in security stack. When malicious requests were intercepted on either the NGFW or active response layers, the system respond HTTP 403 (Forbidden) status and the architecture worked really well achieving 99.8% detection and blocking at packet level even before reaching the DVWA server across all the carried attacks, eliminating the HTTP 200. Wazuh's SIEM associated log alerts with attack source IPs, and Zenarmor did network-layer DPI traffic processing for mitigation purposes.

Focusing on area with Similar Research studies, such as those examining ModSecurity and NAXSI integrated with Wazuh, and WAF operated independently which detects mostly passive and post event which relies on the WAF logs not on the network behaviour was reported 19-21.5% unblocked HTTP 200 in the Zaidan et al. (2024) [1] and then delayed responses, No complete real time blocking mechanism was present. In some cases, this resulted in only partially blocking attacks as HTTP 200 returns continued afterward. Coverage was uneven especially on union-based SQLi. In contrast, the present system achieved total automation from detection to blocking, backed up by real-time cooperation of SIEM and NGFW. This directly addresses the operational gaps that are pointed out in the earlier research.

6.3 Active Response and Automation:

Wazuh automatically takes action without any human intervention. In the seconds after the threat is detected immediately blocking the attacker using the configured rules and policies. Indicated the significate reduction in attacker dwell time when compared to WAF. The literature points out that without instant repulse and interception operations, passive alerting is not only inefficient but even risky. So it was that most of the earlier technology did not or only partially support automatic host/network level blocking. The active response module presented in this research is clearly a step forward towards orchestration of security measures and combines both forward-looking timeliness and strength.

6.4 False Positives and Coverage Depth:

With Zenarmor DPI and original Wazuh detection signals, the rate for false positives was cut back noticeably have achieved 4.9% false positive rate for the Error- Based and Time-Based, Union-Based got 4.9% which is a bit higher when compared with based paper Zaidan, M.N., Sukarno, P., Wardana, A.A., 2024. on relying on Wazuh's rule set (rule id: 100100,100102). That's not easy to say, but as concluded in related texts, NAXSI and ModSecurity, the rate of detection was 9% for Error- Based and Time-Based, for Union based it gave 4% has given a lot of unjust event-blocking and benign**.**
To mention small amount of false positive rate was, relying on signing has ongoing risks. Missed new techniques, obfuscated or zero-day payloads are difficult to detect. To improve in the future, adaptive anomaly-based detection and regular signature upgrades are recommended

6.5 Improvement Over previous Studies:

1. **Enhanced Detection Accuracy:** The Detection rate for all the attack significantly increase to 99.5% which is better than the base paper (88%-91%).

2. **Zero HTTP 200:** In this experiment 0% HTTP 200 is observed, no SQLi attack succeeded, on the base paper it seen 9% success rate for Time-based and 12% success rate for Union-based. This proves the concrete base by integration the Zenarmor DPI and Wazuh SIEM is more effective than the Traditional Web Application Firewall (WAF).

3. **Better False Positive Handling:** The False positive rate is significantly reduced from the base paper. The granular Layer 7 inspection with Zenarmor long with the real time log correlation with SIEM Wazuh, handling the policies configuration helps to minimize false positive.

4. **Clear HTTP Feedback:** This paper present clear and defined split of 403 blocked and 404 not found HTTP responses that offer a detail internal view of request and response handling at application level able clearly document and aligns with strong security posture. Which was missing in the base paper.

Figure 4: Detection Rate Comparison Diagram

7. Conclusion and Future Work

This paper explores whether the Security Information and Event Management (SIEM) system (Wazuh), multi-agent deployment, and Next Generation Firewall (NGFW) (Zenarmor on OPNsense) can jointly detect and prevent SQL injection (SQLi) attacks against web applications. Another goal is to use automation to construct a multi-layer interception setup for SQLi (including error-based, time-based blind and union-based approaches). It should have better real-time detection rates and less time between detection and response than solely log-based architectures and standalone WAF systems.
The results show conclusively that the layered architecture proposed effectively detection and blocks all SQLi vector tests. Every attack in the trials is met with an HTTP 403 Forbidden and the Wazuh dashboard immediately alerts simultaneously. As opposed to traditional architectures which may just passively alert with logs from WAF systems and require intervention by hand the configuration presented in this research slashes the detection-to-block time to seconds, applies automated IP bans, and produces low false positive rates. Union-based attacks, a fatal weakness in NAXSI, are most effectively handled using customized signatures and DPI at the network level. These findings seem to indicate that co-ordination of the three techniques multi-layer inspection, real-time SIEM correlation and active defence can effectively respond to SQLi threats with very real force in enterprise web services environments.

But there are limitations. The current network-level DPI pipeline working fine in the internal network giving 100% accuracy result. It need to be tested in the across the real world network in broader spectrum. Which can be scope for the future study. The approach is mainly based on signature and rule-based detection, which means that it would be vulnerable to advanced obfuscation and zero-day SQLi. Finally, the performance impact of near-real-time blocking and logging at high concurrencies is not quantitatively measured in this phase of project and deserves further study.

Future Work:

To address the limitations and for extend the current research the following projects are proposed.
Behavioural and Anomaly-Based Detection: Current signatures are effective, but incorporating machine learning and behaviour based detection system considering Wazuh using the unsupervised learning model that easy adapt to obfuscated payloads.

- **Dynamic Policy Adaptation:** Enhancement in Wazuh's active-response framework by automating the firewall rule and integrating API-driven interactions with Zenarmor to allow real-time policy updates.
- **Operational Scalability and Commercialization:** Future research should systematically measure the performance of this system under real enterprise network traffic patterns. In addition to co-relating latency and throughput with resource usage levels, an efficient alert mechanism is required for deployment in production networks.

Finally, to conclude this work not only shows a successful and current progress in the mitigation of SQLi, but also sets the stone for strong, future-proof and commercially applicable enterprise defence mechanisms. Need to do further is extend to encrypted traffic, include adaptive intelligence and empirically validate the architecture under a diverse range of operationally realistic environments.

References

Zaidan, M.N., Sukarno, P., Wardana, A.A., 2024. Collaborative Detection of SQL Injection Attacks using SIEM, Multi-Wazuh Agents, and Diverse Web Application Firewalls, in: 2024 5th International Conference on Communications, Information, Electronic and Energy Systems (CIEES). Presented at the 2024 5th International Conference on Communications, Information, Electronic and Energy Systems (CIEES), pp. 1–6. https://doi.org/10.1109/CIEES62939.2024.10811420

Hybrid Intrusion Detection System and Network Infrastructure Vulnerability Mitigation using Active Response (XDR) Technique Wazuh and Suricata | Jurnal Pekommas [WWW Document], n.d. URL https://jkd.komdigi.go.id/index.php/pekommas/article/view/5829 (accessed 5.22.25).

Harefa, J., Prajena, G., Alexander, A., Muhamad, A., Dewa, E.V.S., Yuliandry, S., 2021. SEA WAF: The Prevention of SQL Injection Attacks on Web Applications. Adv. sci. technol. eng. syst. j. 6, 405–411. https://doi.org/10.25046/aj060247

Kareem, F., Ameen, S., Ahmed, A., Salih, A., Ahmed, D., Kak, S., Najat, Z., Yasin, H., Mahmood, I., Omar, N., 2021. SQL Injection Attacks Prevention System Technology: Review. Asian Journal of Research in Computer Science. https://doi.org/10.9734/AJRCOS/2021/v10i330242

Paul, A., Sharma, V., Olukoya, O., 2024. SQL injection attack: Detection, prioritization & prevention. Journal of Information Security and Applications 85, 103871. https://doi.org/10.1016/j.jisa.2024.103871

Ojagbule, O., Wimmer, H., Haddad, R.J., 2018. Vulnerability Analysis of Content Management Systems to SQL Injection Using SQLMAP, in: SoutheastCon 2018. Presented at the SoutheastCon 2018, pp. 1–7. https://doi.org/10.1109/SECON.2018.8479130

Gudipati, V.K., Venna, T., Subburaj, S., Abuzaghleh, O., 2016. Advanced automated SQL injection attacks and defensive mechanisms, in: 2016 Annual Connecticut Conference on Industrial Electronics, Technology & Automation (CT-IETA). Presented at the 2016 Annual Connecticut Conference on Industrial Electronics, Technology & Automation (CT-IETA), pp. 1–6. https://doi.org/10.1109/CT-IETA.2016.7868248

Farrel, F., M.Kom, I., Qamar, M., 2024. Implementation of Security Information & Event Management (SIEM) Wazuh with Active Response and Telegram Notification for Mitigating Brute Force Attacks on The GT-I2TI USAKTI Information System. Intelmatics 4, 1–7. https://doi.org/10.25105/itm.v4i1.18529

Dixit, S., Tripathi, P., 2025. Strengthening Cybersecurity with Wazuh: Log-Based Threat Detection for a Resilient Digital World 14.

Rahul, S., Vajrala, C., Thangaraju, B., 2021. A Novel Method of Honeypot Inclusive WAF to Protect from SQL Injection and XSS, in: 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON). Presented at the 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), pp. 135–140. https://doi.org/10.1109/CENTCON52345.2021.9688059

Rua, M., Thiyab, Musab, D., Ali, A., Abdulqader, F., Abdulqader, 2017. THE IMPACT OF SQL INJECTION ATTACKS ON THE SECURITY OF DATABASES.

Gandhi, N., Patel, J., Sisodiya, R., Doshi, N., Mishra, S., 2021. A CNN-BiLSTM based Approach for Detection of SQL Injection Attacks, in: 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE). Presented at the 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), pp. 378–383. https://doi.org/10.1109/ICCIKE51210.2021.9410675

Qu, Z., Ling, X., Wang, T., Chen, X., Ji, S., Wu, C., 2024. AdvSQLi: Generating Adversarial SQL Injections Against Real-World WAF-as-a-Service. IEEE Transactions on Information Forensics and Security 19, 2623–2638. https://doi.org/10.1109/TIFS.2024.3350911

Bhavsar, R., Thakar, V., 2025. Design and Implementation of an Open-Source Security Operations Center for Effective Cyber Threat Detection and Response. https://doi.org/10.21203/rs.3.rs-5795888/v1

Ahmad, K., Karim, M., 2021. A Method to Prevent SQL Injection Attack using an Improved Parameterized Stored Procedure. International Journal of Advanced Computer Science and Applications (IJACSA) 12. https://doi.org/10.14569/IJACSA.2021.0120636

Bassey, C., Samuel, A., Oloruntola, O., Imakuh, S., Onyia-odike, I., 2024. Enhancing DDoS Attack Prevention And Detection Using IDS And XDR 11, 2458–9403.

Floris, G., Scano, C., Montaruli, B., Demetrio, L., Valenza, A., Compagna, L., Ariu, D., Piras, L., Balzarotti, D., Biggio, B., 2025. ModSec-AdvLearn: Countering Adversarial SQL Injections with Robust Machine Learning. IEEE Trans.Inform.Forensic Secur. 20, 6693–6705. https://doi.org/10.1109/TIFS.2025.3583234

Bouafia, R., Benbrahim, H., Amine, A., 2023. Automatic Protection of Web Applications Against SQL Injections: An Approach Based On Acunetix, Burp Suite and SQLMAP, in: 2023 9th International Conference on Optimization and Applications (ICOA). Presented at the 2023 9th International Conference on Optimization and Applications (ICOA), pp. 1–6. https://doi.org/10.1109/ICOA58279.2023.10308827

Maraj, A., Rogova, E., Jakupi, G., Grajqevci, X., 2017. Testing techniques and analysis of SQL injection attacks, in: 2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA). Presented at the 2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA), pp. 55–59. https://doi.org/10.1109/ICKEA.2017.8169902

Abdullah, H.S., Hamad, Z.O., Khalind, O.S., 2023. Analysis of SQLMAP Efficacy in Exploiting SQL Injection Vulnerabilities in Web Applications: A Case Study on DVWA, in: 2023 International Conference on Engineering Applied and Nano Sciences (ICEANS). Presented at the 2023 International Conference on Engineering Applied and Nano Sciences (ICEANS), pp. 13–18. https://doi.org/10.1109/ICEANS58413.2023.10630454

Xiao, Z., Zhou, Z., Yang, W., Deng, C., 2017. An approach for SQL injection detection based on behavior and response analysis, in: 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN). Presented at the 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), pp. 1437–1442. https://doi.org/10.1109/ICCSN.2017.8230346