Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Integrated Framework for Securing Web Applications: Machine Learning-Driven XSS Detection and Network-Level Threat Mitigation

Dr. Niharika Prasanna Kumar[1*], Mr.Dhanesh Ramesh[2], Mr Harshith Raju P V[3], Mr Kishore Srinivasan[4], Mr. Sumukh G Mahendrakar [5]

[1*]*RV Institute of Technology and Management, JP Nagar, Bengaluru - 560076*
E-mail: niharikaresearch7@gmail.com
[2]*RV Institute of Technology and Management, JP Nagar, Bengaluru - 560076*
E-mail: dhaneshrameshofficial@gmail.com
[3]*RV Institute of Technology and Management, JP Nagar, Bengaluru - 560076*
E-mail: harshrajupv@gmail.com
[4]*RV Institute of Technology and Management, JP Nagar, Bengaluru - 560076*
E-mail: kishoresrinivasan05@gmail.com
[5]*RV Institute of Technology and Management, JP Nagar, Bengaluru - 560076*
E-mail: sumukhmahendrakar@gmail.com

ARTICLE INFO

Received: 15 Nov 2024

Revised: 21 Dec 2024

Accepted: 15 Jan 2025

ABSTRACT

This paper presents a comprehensive framework designed to address the growing need for securing web-based platforms in today's digital age, where the internet is integral to everyday life. The increasing complexity of cyber threats necessitates advanced solutions, prompting the development of a robust framework that focuses on detecting and mitigating vulnerabilities within web applications. Specifically, this framework targets Cross-Site Scripting (XSS) vulnerabilities and inadequate HTTP header configurations, along with providing protection against SQL injection attacks. The proposed approach leverages state-of-the-art machine learning (ML) algorithms to enable proactive threat detection, enhancing the capability of organizations to identify and neutralize XSS attacks effectively. Furthermore, the framework incorporates real-time network protection mechanisms, exemplified by the integration of the pfSense firewall, to mitigate threats at the network level preemptively. This holistic approach to web security not only reinforces organizational resilience but also ensures compliance with regulatory standards and best practices, thereby reducing the risk of non-compliance and enhancing stakeholder trust. Overall, this framework represents a significant advancement in fortifying the security posture of web-based systems, enabling organizations to navigate the evolving threat landscape confidently and protect critical services and sensitive information.

Keywords: Cross-Site-Scripting, SQL-Map, Network, Machine-Learning, Cyber-Security, Firewall, Integrity

1. INTRODUCTION

In the rapidly evolving digital landscape, the proliferation of internet-driven services has transformed business operations and interpersonal interactions, creating both opportunities and vulnerabilities. The increasing complexity of web applications has led to a surge in cyber threats, including Cross-Site Scripting (XSS), SQL injections, and inadequate HTTP header protections, which pose significant risks to sensitive data and service continuity. Traditional cybersecurity methods, often reliant on manual processes, are proving inadequate against the sophisticated nature of these threats, resulting in security gaps that malicious actors can exploit.

The motivation for this paper stems from the urgent need to address these escalating cyber threats through innovative and automated solutions. As organizations face mounting pressure to protect their digital assets, the integration of advanced technologies such as machine learning and real-time network protection becomes essential. By developing a strategic framework for the automated detection and mitigation of web security vulnerabilities, this paper aims to

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

empower organizations to enhance their security posture and maintain user trust in their data protection capabilities.

The problem statement highlights the challenges organizations encounter in effectively detecting and mitigating web security vulnerabilities. Despite advancements in technology, many organizations still rely on slow and error-prone manual processes. This paper seeks to explore how organizations can implement robust frameworks that autonomously identify and mitigate vulnerabilities, thereby strengthening the security of web-based systems and safeguarding vital assets in an increasingly dynamic cyber threat landscape.

2. RELATED WORK

2.1 Fundamentals

In the intricate world of cybersecurity, firewall rule sets serve as essential protectors, carefully managing network traffic to safeguard valuable data from various threats. This literature review explores the complexities of managing these critical rule sets. It draws insights from academic research, industry reports, and technical guides to illuminate effective methodologies, innovative technologies, and best practices for designing, implementing, and optimizing these rules. A key focus is on prioritizing rules to ensure that crucial traffic flows smoothly while maintaining security, as well as managing the complexity that can arise from dense rule sets, which may hinder system performance if not handled properly.[1-2]

Additionally, the review highlights the importance of anomaly detection in spotting suspicious activities before they escalate into serious security breaches. It also examines various optimization strategies that can enhance both operational efficiency and security. The transformative potential of artificial intelligence and machine learning is discussed, showcasing how these technologies can revolutionize the management of firewall rules by automating processes and adapting to new threats with remarkable accuracy.

As the cybersecurity landscape continues to evolve, this review aims to provide actionable insights and identify gaps in current research, ultimately guiding professionals through the challenges of securing modern networks against an ever-changing array of threats. By integrating diverse perspectives, it emphasizes the multifaceted nature of managing firewall rule sets and presents innovative solutions to help cybersecurity experts navigate this complex terrain effectively.[3-4]

2.2 XSS

XSS (Cross-Site Scripting) tools are vital in identifying, exploiting, and mitigating XSS vulnerabilities in web applications. These tools offer functionalities tailored to different stages of the attack process, from payload generation to analysis and reporting.[5-6]

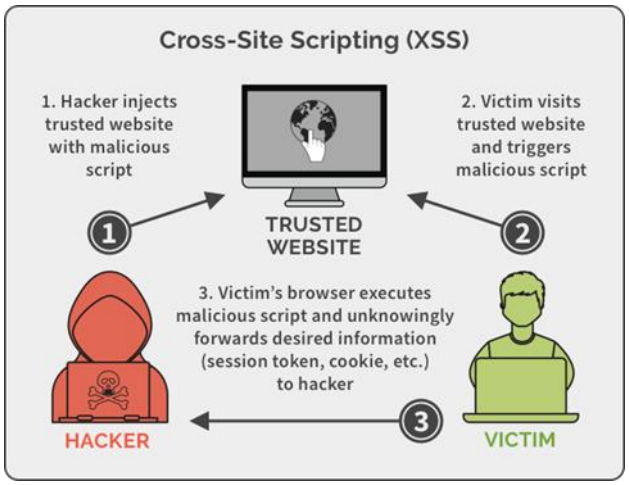

Fig 1. Overview of XSS

Fig 1. describes the high-level methodology of how an XSS is carried out. Tools for payload generation facilitate the creation of malicious scripts that exploit XSS vulnerabilities. These tools often include features for crafting various

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

types of payloads, such as simple script tags, event handlers, and encoded payloads to evade detection by security controls. Payload generators streamline the process of constructing XSS payloads, enabling attackers to quickly test the effectiveness of their injection techniques

DOM manipulation tools focus on dynamically altering the Document Object Model (DOM) of a web page to execute XSS attacks. These techniques are particularly effective for bypassing client-side security controls and executing XSS payloads in scenarios where traditional injection methods may fail.

Browser exploitation tools are designed to identify and exploit vulnerabilities in web browsers to facilitate XSS attacks. These tools often target specific browser versions and configurations, leveraging known security weaknesses to execute arbitrary code within the browser environment. By exploiting browser vulnerabilities, attackers can bypass client-side security mechanisms and execute XSS payloads successfully, even in scenarios where the web application itself is well-defended against injection attacks.

Reporting and analysis tools play a crucial role in identifying and mitigating XSS vulnerabilities within web applications. These tools include vulnerability scanners, web application firewalls, and security platforms that offer comprehensive reporting capabilities and in-depth analysis of detected vulnerabilities. By identifying XSS vulnerabilities proactively and providing actionable insights for remediation, reporting and analysis tools help organizations strengthen their web application security posture and mitigate the risk of XSS attacks.

2.3 Machine Learning

Machine learning (ML) is a branch of artificial intelligence that focuses on creating algorithms capable of learning from data and making predictions or decisions. In the context of XSS (Cross-Site Scripting) detection, ML algorithms are used to automatically identify patterns indicative of XSS attacks within web application traffic or user inputs. These models can analyze various features, such as the structure of input parameters, typical user behavior, or unusual content patterns, to distinguish between legitimate and potentially malicious activities.[7-8]

Random forests, a popular ML technique, is particularly effective for XSS detection due to their ability to handle high-dimensional data and manage missing values efficiently. These algorithms can be trained on labeled datasets containing examples of both benign and malicious input to learn the characteristics of XSS attacks. Once trained, the ML model can classify new input instances as either safe or potentially vulnerable, assisting in real-time XSS prevention.

The random forest algorithm works by constructing multiple decision trees during training and outputs the class that is the mode of the classes (classification) or the mean prediction (regression) of the individual trees. Each tree in the forest is built from a random subset of the training data and a random subset of the features, which helps reduce overfitting and enhances the model's generalization capability. This random sampling introduces diversity among the trees, ensuring that no single tree dominates the prediction. The algorithm is highly robust to noise and can handle large datasets with higher dimensionality effectively. Additionally, the random forest algorithm provides estimates of feature importance, which can be valuable for understanding the influence of different features on the prediction outcomes.

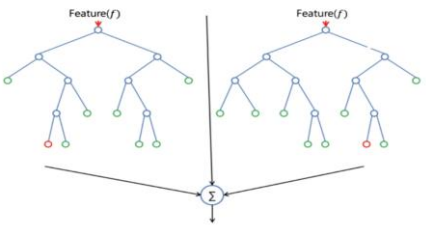

Fig 2. Random forest methodology

In XSS detection, as shown in Fig 2. random forests can be trained to recognize complex patterns indicative of XSS attacks by analyzing various input features. These features may include the length and structure of user input, the presence of special characters, context within the web application, or unusual content distribution. The ensemble

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

learning approach of random forests allows them to effectively capture both global and local patterns related to XSS, making them a robust choice for real-time detection in dynamic web environments. By leveraging the collective wisdom of multiple decision trees, random forests provide accurate and scalable protection against XSS vulnerabilities.[9]

One significant advantage of random forests is their ability to be used for both classification and regression problems, which constitute the majority of current ML systems. The main difference between a decision tree and a random forest lies in their approach to making predictions and handling the risk of overfitting.

In a decision tree model, the algorithm learns from the training data by formulating a set of rules based on the features and labels provided. These rules are used to make predictions about unseen data. For instance, if predicting whether a person will click on an online advertisement based on past click history and relevant features, a decision tree will generate rules to make that prediction.[10]

On the other hand, a random forest model is an ensemble of decision trees. It randomly selects subsets of observations and features from the training data and builds multiple decision trees. Each tree in the forest makes its own predictions, and the final prediction is obtained by averaging or voting over the predictions of all trees. One key advantage of using a random forest is its ability to mitigate overfitting, especially when dealing with deep decision trees.

Overfitting occurs when a model learns to capture noise in the training data rather than the underlying patterns, leading to poor performance on unseen data. Random forests address this issue by creating random subsets of features for each tree and building smaller, less complex trees.

However, it's important to note that random forests may not always prevent overfitting, particularly in cases where the dataset is noisy or contains irrelevant features. Additionally, building multiple trees and combining their predictions can increase computational complexity and slow down the training process, especially when dealing with a large number of trees. Despite these potential drawbacks, random forests are widely used and effective ML algorithms, particularly for classification and regression tasks.

2.4 Security Header Protocols

Security header protocols are crucial components of web security frameworks, providing mechanisms to mitigate common vulnerabilities and safeguard against various attacks. Content Security Policy (CSP) allows website administrators to define and enforce a whitelist of trusted sources from which content such as scripts, stylesheets, and images can be executed. By specifying approved domains for loading resources, CSP mitigates the risks associated with Cross-Site Scripting (XSS) attacks. Implementation involves adding CSP directives to HTTP headers or within HTML meta tags, detailing allowed content sources and enforcement policies. CSP offers granular control over content execution, enabling administrators to specify policies for different content types and providing options for reporting violations. Properly configured CSP headers can significantly enhance web security by reducing the attack surface for XSS vulnerabilities and thwarting malicious script injections.[11]

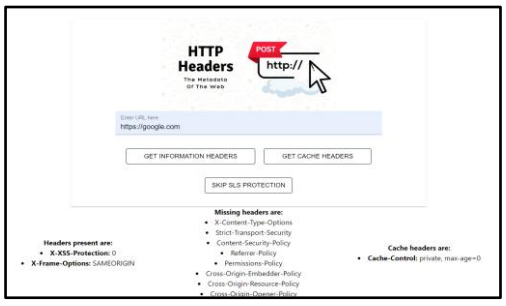

Fig 3. Discovering the Presence of HTTP Headers

Fig 3. Refers to different types of headers. Among them, is the Cross-Origin Resource Sharing (CORS) that regulates access to resources on a web server from external domains, preventing unauthorized cross-origin requests that could compromise sensitive data or expose vulnerabilities. CORS works by adding HTTP headers to responses, indicating

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

which external origins are permitted to access the server's resources. Implementation involves configuring CORS policies on the server side to specify allowed origins, HTTP methods, and headers. By enforcing the Same-Origin Policy (SOP), CORS helps protect against Cross-Site Request Forgery (CSRF) and unauthorized data access from malicious third-party domains. Properly configured CORS policies strike a balance between accessibility and security, enabling legitimate cross-origin interactions while safeguarding against potential exploitation.

HTTP Strict Transport Security (HSTS) is a crucial security mechanism designed to enforce secure communications between web browsers and servers. By instructing web browsers to only establish connections over secure HTTPS, HSTS mitigates the risk of various cyber threats, including man-in-the-middle (MITM) attacks and protocol downgrade attacks. HSTS is implemented through an HTTP response header sent by the server. This header directs the browser to enforce HTTPS for all subsequent requests to the specified domain. Once a browser receives the HSTS header, it will automatically upgrade any HTTP requests to HTTPS, ensuring that users are always connected securely, even if they initially enter an insecure URL. By adopting HSTS, websites can significantly bolster their security posture, ensuring that user data remains confidential and intact during online interactions, thereby mitigating the risk of various cyber threats, including MITM and protocol downgrade attacks.

3. PROBLEM STATEMENT

In today's interconnected digital environment, the proliferation of internet-driven services has reshaped how businesses operate and how people connect globally. However, this extensive connectivity has also exposed web-based platforms to a myriad of security vulnerabilities, ranging from Cross-Site Scripting (XSS) to SQL injections and inadequate HTTP header protections. These vulnerabilities pose significant risks to enterprises, potentially compromising sensitive data, eroding user trust, and even disrupting essential services.

Despite advances in cybersecurity technology, organizations struggle to effectively detect and mitigate these web security vulnerabilities. Manual processes for identifying vulnerabilities are slow and error-prone, leaving organizations vulnerable to exploitation by cyber attackers. Traditional security measures often lack the agility needed to keep up with evolving threats, leaving gaps that malicious actors can exploit.

Addressing this challenge requires a strategic approach that leverages cutting-edge technologies like machine learning, advanced scanning techniques, and real-time network protection. By automating the detection and mitigation of vulnerabilities, organizations can bolster their defenses, respond to threats swiftly, and safeguard critical web services.

In essence, the key question at hand is: How can organizations develop and implement a robust framework to autonomously identify and mitigate web security vulnerabilities, using advanced technologies, to strengthen the security of web-based systems and protect vital assets amidst a dynamic cyber threat landscape?

4. METHODOLOGY

The methodology employed in developing the proposed framework for enhancing web security is outlined. This framework integrates various software, hardware, and functional components to create a robust and scalable system capable of detecting and mitigating cyber threats effectively. By detailing the system requirements, both software and hardware, along with the necessary functional and non-functional criteria, a comprehensive guide for implementing this security framework is provided. Additionally, the section discusses the mathematical equations relevant to the machine learning algorithms utilized in this proposed system, highlighting their significance in the detection and prevention of security vulnerabilities.[12]

4.1 Requirements

The proposed framework for enhancing web security requires specific software and system components to ensure compatibility, performance, and functionality. Essential software includes major operating systems (Windows, Linux, macOS), web servers (Apache or Nginx), DBMS (MySQL or PostgreSQL), and programming proficiency in Python, JavaScript, and SQL. Machine learning libraries (scikit-learn, TensorFlow, PyTorch) are necessary for threat detection, and tools like SQLMap are required for penetration testing and network security.

System requirements encompass a multi-core processor, adequate RAM, sufficient storage, a high-speed network

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

interface, and a GUI for administration. Hardware needs involve multi-core processors with parallel processing support, ample RAM, high-speed SSDs, multiple Gigabit Ethernet interfaces, and redundancy features like RAID configurations.

Client hardware includes devices such as desktops, laptops, tablets, and smartphones with modern web browsers. Networking infrastructure needs high-speed Ethernet switches, enterprise-grade routers, and hardware firewalls. Scalability considerations include a modular architecture, load balancing mechanisms, and distributed storage solutions. Adhering to these requirements ensures robust, scalable, and effective web security.

4.2 Algorithm

Machine learning algorithms are crucial in the proposed web security framework, particularly for detecting and mitigating threats like Cross-Site Scripting (XSS). Key algorithms include the Random Forest Classifier and Logistic Regression, both underpinned by fundamental mathematical principles. The Random Forest Classifier is an ensemble learning method that builds multiple decision trees during training. For classification, it predicts the class that appears most frequently among the individual tree predictions. The prediction equation is

$$y\,i = mode\,(\ \ 1\,,\, y\,i\,2\,,\, \ldots\,,\, y\,i\,n\,) \qquad (1)$$

where $y\,i$ is the predicted class for an instance, $y\,i\,j$ is the class predicted by the j-th tree, and n is the total number of trees. Logistic Regression is used for binary classification, modeling the probability that an input belongs to a particular class using the sigmoid function, given by

$$\sigma\,(\,) = 1/(1 + e - z) \qquad (2)$$

where z is a linear combination of input features and their weights, e is the base of the natural logarithm, and $\sigma\,(z)$ represents the probability of the output being in the positive class. Understanding these equations aids in the effective implementation and interpretation of machine learning algorithms within the security framework.[13]

4.3 Assumptions

The development of the web security framework is based on several assumptions to streamline its scope, but acknowledging potential discrepancies with real-world conditions is essential. Key assumptions include:

Controlled Environment: The framework is assumed to operate in a controlled testing environment with access to representative datasets and network traffic. However, real-world networks may be more complex and variable, requiring robust validation and adaptation strategies.

Machine Learning Performance: It is assumed that machine learning algorithms like Random Forest will perform consistently across different datasets and threat scenarios. In reality, their efficacy can vary based on data quality, feature selection, and model hyperparameters, necessitating thorough experimentation and validation.

Tool Reliability: Tools like SQLMap and are assumed to provide consistent functionality for detecting and mitigating vulnerabilities. In practice, these tools might face limitations or unforeseen challenges in detecting complex vulnerabilities or adapting to evolving threats, requiring ongoing monitoring and updates.

End-User Proficiency: The framework assumes end-users are proficient in interpreting and responding to security alerts and recommendations. However, end-user awareness and understanding of security concepts can vary, making effective communication strategies and user training programs essential for informed decision-making.

Acknowledging these assumptions and potential gaps ensures the framework is adaptable and effective in real-world applications.[14]

4.4 Procedure

The methodology for developing and implementing the web security framework follows a structured approach to ensure robust and reliable security solutions. This approach includes clear workflows, comprehensive documentation, and iterative refinement based on feedback and evaluation.

Flowcharts are utilized to visually represent the proposed system's workflow and the functionality of key components,

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

such as. These visual aids outline the step-by-step processes involved in development and implementation, helping to identify potential bottlenecks and areas for improvement. This facilitates a smoother progression of the paper.

Comprehensive documentation is emphasized, with detailed records of each step, decision, and adjustment. This documentation ensures transparency and reproducibility, aiding in tracking progress and serving as a valuable resource for troubleshooting and future enhancements. By systematically capturing insights and outcomes, the team can make informed decisions and continuously refine the security mechanisms based on real-world performance and feedback. The process of maintaining thorough documentation involves regular updates to capture the latest changes and decisions, ensuring that no critical detail is overlooked. Each team member contributes to the documentation, fostering a collaborative environment where collective knowledge is shared and preserved.

This approach also helps in onboarding new team members, as they can quickly get up to speed by reviewing the documented history and rationale behind previous actions. Additionally, well-documented procedures facilitate compliance with industry standards and regulatory requirements, demonstrating a commitment to best practices in security management. By investing in meticulous documentation, the team lays a strong foundation for continuous improvement and resilience against evolving threats.[15]

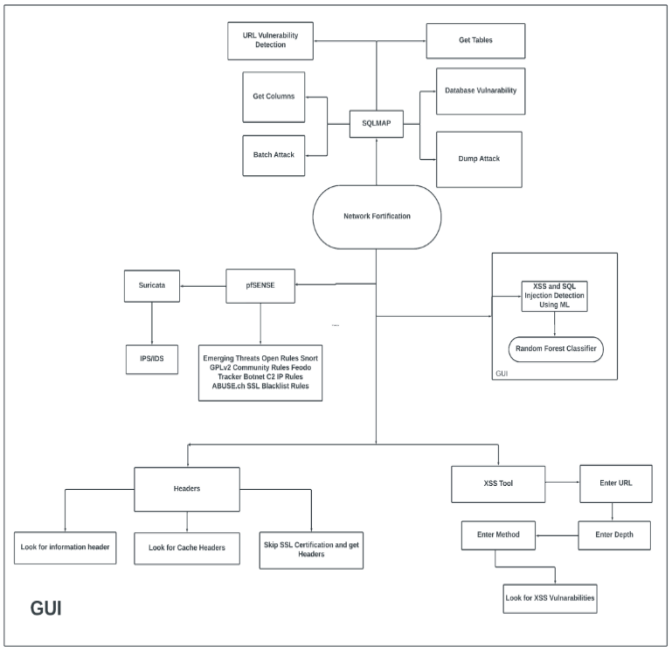

Fig 4. Overall Methodology of Network Fortification

Fig 4 describes the overall methodology of Network Fortification. This methodology ensures that each phase of the system is meticulously planned and executed, leading to the development of a robust and reliable web security framework. The systematic approach and detailed documentation help in maintaining clarity and improving the paper based on continuous evaluation.

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

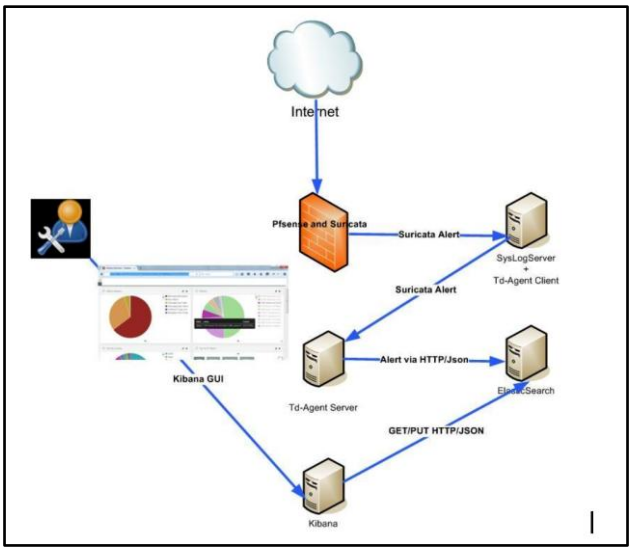

Fig 5. Outlines the working of Suricata

The flowchart in Fig 5. outlines a network security infrastructure starting with Suricata for firewall and intrusion detection. Alerts from these systems are processed by a SysLogServer and Td-Agent Client before being stored in ElasticSearch, with Kibana providing a graphical interface for data analysis.

The FreeBSD operating system, is known for its robustness and flexibility, offering advanced features such as firewalls, VPN services, traffic shaping, and IDS/IPS capabilities. Its web-based GUI allows network administrators to manage security policies, stateful packet filtering, NAT, port forwarding, and VLAN configuration. Its extensibility is enhanced through third-party packages and plugins.[16]

Suricata complements by providing real-time traffic analysis and packet inspection, utilizing signature-based detection, protocol analysis, and anomaly detection to identify security threats. Suricata rules define patterns indicative of threats, enabling swift response.

Integrating Suricata through a package management system and a user-friendly GUI makes it easier for administrators to manage settings, rulesets, and alerts. With React components in the GUI, the platform offers a smooth and interactive user experience. Regular updates and teamwork ensure that the system stays ahead of new threats, giving you a powerful and straightforward tool for keeping your network secure. [17]

The operational implementation of Security Headers within the paper, emphasizing their role in bolstering web application security. The methodology includes header analysis, where HTTP headers like Strict-Transport-Security, X-Frame-Options, Content-Security-Policy, and Referrer-Policy are scrutinized to identify potential security gaps. A Python script, leveraging the Requests library and colorama for enhanced output, evaluates these headers by fetching and analyzing them from a specified URL. It checks for the presence and correctness of key directives, providing warnings and recommendations for remediation. Fig 6 represents the statistical accuracy among other statistics of the model developed.

```
PS D:\FW-Project\WebAppFirewall-with-ML> python script_main.py
Bad samples: 44532
Good samples: 1265974
Baseline Constant negative: 0.966019
------------
Accuracy: 0.999119
Precision: 0.997557
Recall: 0.976098
F1-Score: 0.986711
AUC: 0.999981
PS D:\FW-Project\WebAppFirewall-with-ML>
```

Fig 6. Statistics of ML used for URL Detector

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The core security headers discussed include:

- *Strict-Transport-Security (HSTS)*: Protects against protocol downgrade attacks by enforcing HTTPS connections.
- *X-Frame-Options*: Prevents clickjacking by controlling whether a page can be embedded in a frame on another domain.
- *Content-Security-Policy (CSP)*: Mitigates XSS and data injection attacks by restricting resource loading sources.
- *Referrer-Policy*: Manages referrer information sent with requests to enhance privacy.
- *Permissions-Policy*: Controls browser feature usage to mitigate security risks.
- *Cross-Origin Resource Sharing (CORS):* Controls access to server resources from different origins to prevent CSRF.
- *Feature-Policy:* Restricts specific browser features to mitigate security and privacy risks.
- *Expect-CT*: Ensures certificate transparency to detect mis issued SSL certificates.
- *Public-Key-Pins (HPKP)*: Secures SSL/TLS connections by pinning specific cryptographic keys.
- *Cross-Origin Embedder Policy (COEP) and Cross-Origin Opener Policy (COOP)*: Control cross-origin content embedding and page opening to mitigate security vulnerabilities.

By enforcing these headers, the framework enhances the security posture of web applications, mitigating common attack vectors and safeguarding sensitive data. Fig 7 showsMalicious URL Detection, Fig 8 shows Safe URL Detection.

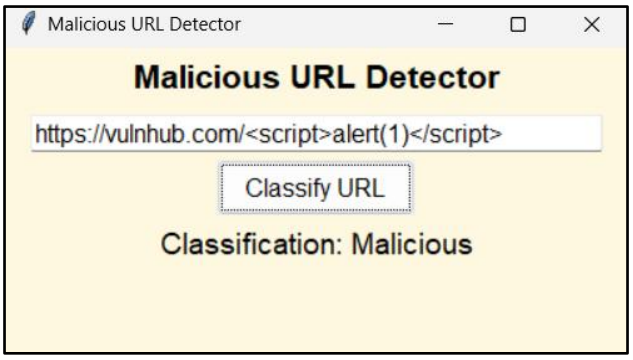

Fig 7. Malicious URL Detection

Fig 8. Safe URL Detection

4.5 Tools Used

The proposed web security framework leverages a range of tools and technologies to enhance detection and mitigation capabilities. Key tools in the proposed system include:

SQLMap: SQLMap is a penetration testing tool specifically designed for detecting and exploiting SQL injection vulnerabilities in web applications. Its automated scanning capabilities, comprehensive detection methods, and exploitation functionalities make it an indispensable tool for identifying and remediating SQL injection

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

vulnerabilities.

Machine Learning Libraries: The proposed system incorporates machine learning algorithms implemented using libraries such as scikit-learn, TensorFlow, or PyTorch. These libraries provide a rich set of tools and functionalities for training, evaluating, and deploying machine learning models for proactive threat detection and mitigation.

Vulnerability Scanners: Additional vulnerability scanners and security assessment tools may be utilized to complement the capabilities of SQL Map and enhance overall vulnerability detection coverage. These scanners may include open-source or commercial solutions capable of identifying a wide range of security vulnerabilities beyond SQL injection.

Web Application Firewalls (WAFs): While not explicitly mentioned, WAFs may be integrated with the web security framework to provide additional layers of protection against web-based attacks, including XSS, SQL injection, and other common vulnerabilities.

5 INTEGRATION

This section explores the outcomes and discussions of the integrated network fortification system, meticulously examining technical aspects such as GUI integration, the effectiveness of the integrated firewall, machine learning-based threat detection, HTTP header analysis, SQL injection detection, and Cross-Site Scripting (XSS) testing. The GUI integration leverages React and Material-UI to create responsive and visually appealing interface elements, facilitating an intuitive user experience across various devices and screen sizes.

HTTP header analysis functionality allows users to retrieve and examine headers for potential vulnerabilities, enhancing their understanding of web application security posture. The GUI also integrates SQL Map, an open-source penetration testing tool for detecting and exploiting SQL injection flaws, providing comprehensive security testing through various scanning techniques and payloads. For Cross-Site Scripting (XSS) vulnerabilities, the GUI offers testing capabilities to identify and mitigate XSS threats, with detailed reports highlighting detected vulnerabilities and recommended remediation steps.

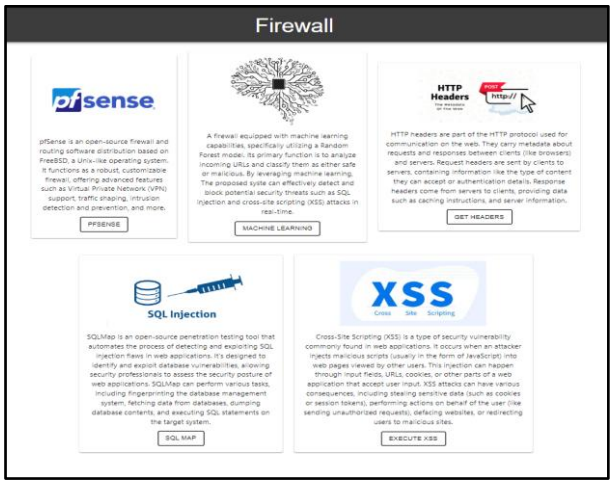

Fig 9. Home GUI

Fig 9. Home GUI component serves as the main entry point, utilizing Axioms for asynchronous requests to ensure efficient communication with the server for real-time updates and data retrieval. The integration also includes an open-source firewall, allowing users to access its management interface seamlessly, thus streamlining firewall policy management and enhancing network security. Machine learning-based threat detection capabilities are incorporated, enabling users to initiate threat analysis and identify patterns indicative of malicious behavior.

This proactive approach ensures comprehensive coverage of potential security threats, enhancing the overall security posture of web applications and contributing to the integrity and trustworthiness of the organization's digital ecosystem. Through this integration, the system provides powerful tools for safeguarding against a variety of cyber threats, enabling efficient monitoring and management of network security.

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

6. RESULT AND ANALYSIS

This section presents the outcomes and discussions from the integrated network fortification system, focusing on key technical aspects including GUI integration, firewall efficacy, machine learning-based threat detection, HTTP header analysis, SQL injection detection, and Cross-Site Scripting (XSS) testing.[18]

The GUI integration, powered by React and Material-UI, ensures a seamless and visually appealing user interface across different devices. The Home component acts as a central hub, offering insights and actions for each integrated security feature, facilitated by efficient backend communication through Axios.

Integration with firewall streamlines management tasks, providing users with straightforward access to configuration and monitoring capabilities, thus bolstering network security measures effectively.

Machine learning-driven threat detection, using algorithms like Random Forest, enhances threat analysis capabilities. It empowers users with real-time insights into potential security risks, facilitating proactive mitigation strategies.

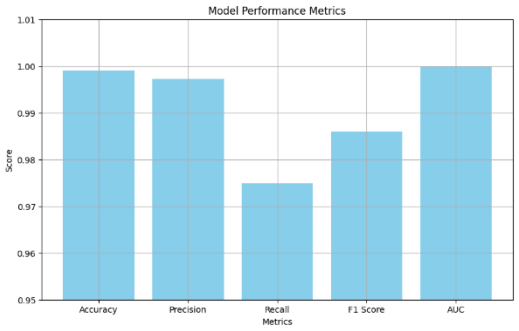

Fig. 10. Model Performance Index

The bar chart in Fig. 10 presents key prformance metrics for the random forest model, including Accuracy, Precision, Recall, F1 Score, and AUC (Area Under the Curve). Each metric provides insights into different aspects of the model's performance. Accuracy and AUC are both notably high, indicating the model's strong overall performance. The Precision is also high, but Recall is relatively lower, suggesting that while the model is good at identifying positives, it might miss some true positives. The F1 Score balances Precision and Recall, showing the trade-off between the two.

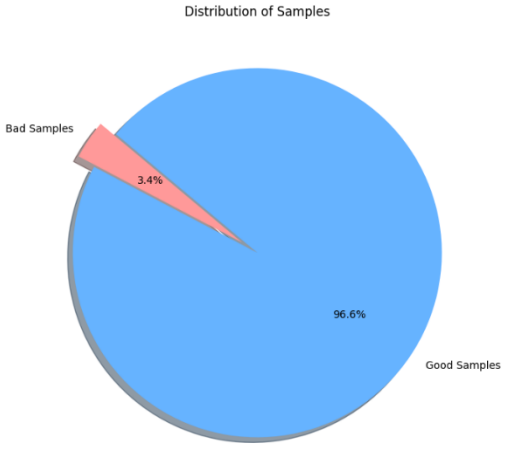

Fig. 11. Distribution of samples

The Fig. 11. illustrates the distribution of good and bad samples within the dataset used by the random forest classifier. The vast majority of samples (96.6%) are classified as "Good Samples," while a smaller fraction (3.4%) is categorized as "Bad Samples." This significant imbalance in the dataset may have influenced the model's

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

performance, particularly in metrics like Recall. Understanding this distribution is crucial for assessing the model's effectiveness and potential bias.

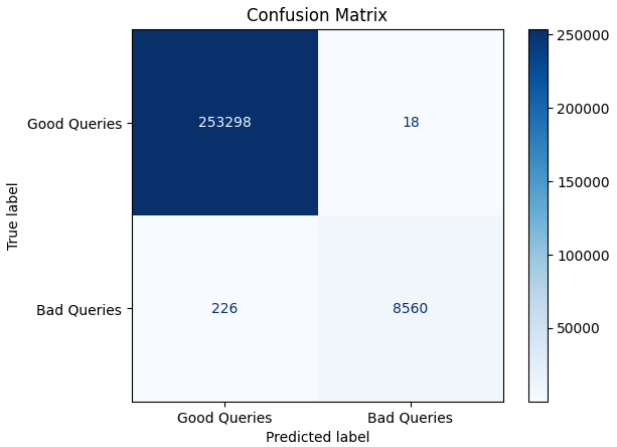

Fig. 12. Confusion Matrix of the Trained Model

The confusion matrix in Fig. 12 provides a detailed breakdown of the random forest model's classification performance. It shows the number of correct predictions for both "Good Queries" and "Bad Queries" as well as the misclassifications. The model correctly identified 253,298 good queries and 8,560 bad queries, with minimal misclassifications (18 false positives and 226 false negatives). This matrix helps evaluate the model's precision, recall, and overall effectiveness in distinguishing between the two classes.

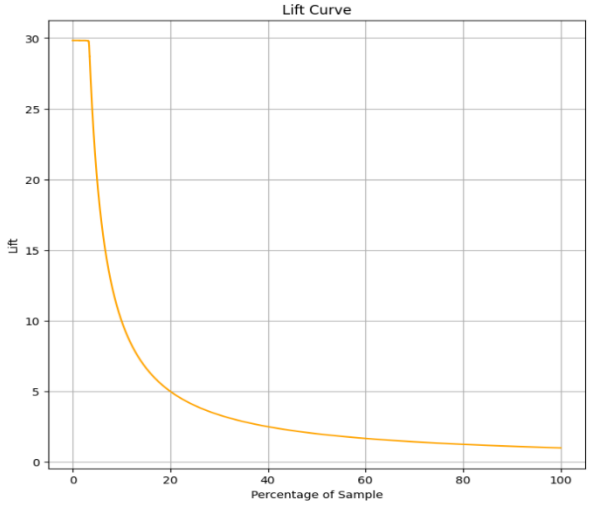

Fig. 13. Comparing the effectiveness to Random Sampling

The lift curve in Fig 13. demonstrates the effectiveness of the random forest model in terms of how well it identifies positive instances relative to a random selection. A steep lift curve at the beginning indicates that the model is much more effective than random chance at identifying the most relevant cases early on. As the curve flattens, the model's advantage diminishes, showing that as more of the population is considered, the model's ability to distinguish between positive and negative instances decreases. This curve is useful for understanding the model's predictive power, particularly in targeted applications like marketing or fraud detection. HTTP header analysis and SQL injection detection tools provide comprehensive assessments of web application vulnerabilities, enabling users to strengthen defenses against common attack vectors.

The XSS testing functionality allows users to simulate and identify XSS vulnerabilities, crucial for preemptively

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

securing web applications. Detailed reports from these tests equip security teams with actionable insights for prompt remediation.

This holistic approach enhances overall system resilience and security posture, enabling organizations to mitigate risks effectively in today's evolving cybersecurity landscape.

7. CONCLUSION AND FUTURE SCOPE:

A comprehensive evaluation of the integrated network fortification system, focusing on its strengths, limitations, and future possibilities. The system has made significant strides in enhancing cybersecurity resilience through integration and automation, but it also faces various technical and operational challenges. These limitations are discussed in detail, along with the broader societal impacts of the system. Future research and development directions are also highlighted to offer a well-rounded perspective on the current status and future opportunities in integrated network fortification.

In an era marked by digital innovation and interconnectedness, the need for robust cybersecurity has never been greater. Integrated network fortification is at the forefront of this effort, providing a holistic approach to defending against an evolving array of cyber threats while fostering societal resilience and digital trust. By combining various security measures into a cohesive system, integrated network fortification proactively addresses vulnerabilities, reducing the risk of breaches and enhancing overall security. This approach not only protects critical infrastructure but also promotes a secure digital environment essential for economic growth.[19]

In conclusion, integrated network fortification represents the collective determination to safeguard the digital future. As the journey of innovation, collaboration, and resilience continues, maintaining a steadfast commitment is crucial to shaping a future where cybersecurity serves as more than just a defense against threats. It should symbolize trust, opportunity, and progress for generations to come. Achieving this vision of a secure digital future requires ongoing efforts to improve security measures, foster global partnerships, and continually adapt to the ever-evolving threat landscape. By pursuing these goals, a digital environment can be created where, security and progress go hand in hand, enabling societies to thrive and advance in an interconnected world.

The integration of AI and ML algorithms advanced threat detection and response by automating data analysis for real-time threat and anomaly detection. AI-driven predictive security analytics can anticipate and prevent cyber threats, enhancing cybersecurity resilience and proactive risk management. This leads to a more robust and adaptive defense against emerging cyber threats.

The adoption of Zero Trust architecture will revolutionize traditional network security paradigms by emphasizing continuous verification, least privilege access, and micro-segmentation. Integrated security frameworks embracing Zero Trust principles will enforce strict access controls, authenticate users and devices, and monitor network traffic for suspicious activities in real-time. Secure Access Service Edge (SASE) architecture will further streamline network security operations by converging networking and security functionalities, delivering comprehensive security services from a unified cloud-native platform. This approach ensures that every access request is verified, significantly reducing the attack surface and enhancing overall security.

Blockchain technology will enable decentralized identity management solutions, providing secure and tamper-proof identity verification mechanisms. Integrated security frameworks leveraging blockchain-based identity management will enhance authentication, access control, and iden tity federation across distributed networks, reducing the risk of identity theft and unauthorized access.[20]

The future scopes of integrated network fortification are characterized by innovation, collaboration, and adaptation to emerging cybersecurity challenges and opportunities. By embracing AI-driven threat detection, Zero Trust architecture, SASE principles, and blockchain-based security solutions, integrated security frameworks will evolve to meet the evolving needs of organizations and individuals in an increasingly interconnected and digitized world. Proactive adoption of advanced security technologies and best practices will be essential to safeguarding digital assets and infrastructure, ensuring the integrity, confidentiality, and availability of data in the years ahead.

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

REFERENCES

[1] Ucar, E., & Ozhan, E. (2017). The analysis of firewall policy through machine learning and data mining. Wireless Personal Communications, 96, 2891-2909.
[2] Kaushik, K., Khan, A., Kumari, A., Sharma, I., & Dubey, R. (2024). Ethical considerations in AI-based cybersecurity. In Next-generation cybersecurity: AI, ML, and Blockchain (pp. 437-470). Singapore: Springer Nature Singapore.
[3] Liu, A. X. (2008, May). Formal verification of firewall policies. In 2008 IEEE International Conference on Communications (pp. 1494-1498). IEEE.
[4] Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., & Xu, M. (2020). A survey on machine learning techniques for cyber security in the last decade. IEEE access, 8, 222310-222354.
[5] Котенко, И. В., Саенко, И. Б., Лаута, О. С., & Крибель, А. М. (2022). Методика обнаружения аномалий и кибератак на основе интеграции методов фрактального анализа и машинного обучения. Информатика и автоматизация, 21(6), 1328-1358.
[6] Clincy, V., & Shahriar, H. (2018, July). Web application firewall: Network security models and configuration. In 2018 IEEE 42nd annual computer software and applications conference (COMPSAC) (Vol. 1, pp. 835-836). IEEE.
[7] Pingale, S. V., & Sutar, S. R. (2021, November). Analysis of web application firewalls, challenges, and research opportunities. In ICDSMLA 2020: Proceedings of the 2nd International Conference on Data Science, Machine Learning and Applications (pp. 239-248). Singapore: Springer Singapore.
[8] Lee, J. K., Hong, T., & Lee, G. (2024). AI-Based Approach to Firewall Rule Refinement on High-Performance Computing Service Network. Applied Sciences, 14(11), 4373.
[9] Voronkov, A., Martucci, L. A., & Lindskog, S. (2020). Measuring the usability of firewall rule sets. IEEE Access, 8, 27106-27121.
[10] Durante, L., Seno, L., & Valenzano, A. (2021). A formal model and technique to redistribute the packet filtering load in multiple firewall networks. IEEE Transactions on Information Forensics and Security, 16, 2637-2651..
[11] Lee, H., Lee, S., Kim, K., & Kim, H. K. (2023). HSViz-II: Octet Layered Hierarchy Simplified Visualizations for Distributed Firewall Policy Analysis. IEEE Access, 12, 936-948.
[12] Diekmann, C., Hupel, L., Michaelis, J., Haslbeck, M., & Carle, G. (2018). Verified iptables firewall analysis and verification. Journal of automated reasoning, 61, 191-242.
[13] Lo Giudice, F. G., & Ghafir, I. (2023). Firewalls: Types, Policies, Security Issues and Best Practices. Policies, Security Issues and Best Practices (December 15, 2023).
[14] Bringhenti, D., Seno, L., & Valenza, F. (2023). An optimized approach for assisted firewall anomaly resolution. IEEE Access, 11, 119693-119710.
[15] Albin, E., & Rowe, N. C. (2012, March). A realistic experimental comparison of the Suricata and Snort intrusion-detection systems. In 2012 26th International Conference on Advanced Information Networking and Applications Workshops (pp. 122-127). IEEE.
[16] Togay, C., Kasif, A., Catal, C., & Tekinerdogan, B. (2021). A firewall policy anomaly detection framework for reliable network security. IEEE Transactions on Reliability, 71(1), 339-347.
[17] Chomsiri, T., He, X., Nanda, P., & Tan, Z. (2016). Hybrid tree-rule firewall for high speed data transmission. IEEE transactions on cloud computing, 8(4), 1237-1249.
[18] Bringhenti, D., Marchetto, G., Sisto, R., Valenza, F., & Yusupov, J. (2022). Automated firewall configuration in virtual networks. IEEE Transactions on Dependable and Secure Computing, 20(2), 1559-1576.
[19] Lin, Z., & Yao, Z. (2022). Firewall anomaly detection based on double decision tree. Symmetry, 14(12), 2668.
[20] Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017). A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. IEEE internet of things journal, 4(5), 1125-1142.