Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

SIEGEWALL

Adithya Krishna R
Department of Computer Science and Engineering (Cyber Security)
Vimal Jyothi Engineering College, Chemperi, Kannur

Arjun K
Department of Computer Science and Engineering (Cyber Security)
Vimal Jyothi Engineering College
Chemperi, Kannur

Athul P V
Department of Computer Science and Engineering (Cyber Security)
Vimal Jyothi Engineering College
Chemperi, Kannur

Ms. Subhaga K
Assistant Professor
Department of Computer Science
Vimal Jyothi Engineering College
Chemperi, Kannur

Abstract - **As web applications continue to expand in scale and complexity, they increasingly become targets for sophisticated cyber threats that traditional security measures often struggle to mitigate effectively. SiegeWall introduces a robust security solution by providing a Software-as-a-Service (SaaS) based Web Application Firewall (WAF) designed to safeguard web applications and their services from malicious attacks. Operating as a reverse proxy, SiegeWall sits between the client and the web server, continuously monitoring and filtering all incoming requests before they reach the application. When suspicious or malicious traffic is detected, the system blocks the request in real time, preventing any potential damage to the server or application.SiegeWall maintains detailed logs of detected threats, including source IP addresses and attack patterns, enabling effective threat tracking, analysis, and monitoring. IP addresses identified as malicious are automatically and permanently blocked, reducing the likelihood of repeated attacks. Additionally, the platform incorporates adaptive learning mechanisms that allow it to evolve alongside emerging threats and new attack vectors. By combining real-time traffic inspection, automated threat blocking, and intelligent monitoring, SiegeWall delivers a resilient and scalable security layer. Its user-friendly design and seamless integration require minimal configuration, making it suitable for both individual developers and enterprise-level organizations seeking reliable web application protection.**

Index Terms—**Operating System Security, Behavioral Analysis, Zero-Day Defense, Local AI, Privacy-Preserving Computing, Malware Detection, Proactive Defense**

I. INTRODUCTION

The rapid evolution of web technologies and digital services has necessitated a fundamental shift in how web applications are secured and managed. Modern computing environments are no longer limited to standalone systems; they have expanded into cloud platforms, distributed architectures, and large-scale online services. However, as web applications become more interconnected and accessible, they also become highly vulnerable to sophisticated cyber threats such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and distributed denial-of-service (DDoS) attacks. Traditional security mechanisms, which primarily rely on static rule-based firewalls and signature detection, often fail to pro- vide adequate protection against evolving and unknown threats.

Significant research has been conducted to enhance web ap- plication security using intelligent and adaptive techniques. Maheshwari et al. proposed an adaptive Web Application Firewall capable of detecting multiple types of web attacks through dynamic traffic analysis, improving detection accuracy and reducing reliance on manual rule updates. Similarly, Kumar et al. introduced a machine learning-based WAF designed for real-time threat detection by analyzing traffic behavior patterns, demonstrating improved adaptability compared to traditional systems. Further advancements were made by Durmuşkaya et al., who utilized machine learning models to classify web traffic and enhance detection performance, particularly against injection- based attacks.

Expanding on these approaches, Dawadi et al. explored the integration of deep learning techniques into WAF systems, demonstrating improved capability in detecting complex and evolving attack patterns with reduced false positives. Additionally, survey-based studies have highlighted the limitations of existing WAF solutions, including challenges in handling zero-day attacks, maintaining high detection accuracy, and ensuring scalability in cloud-based environments. These studies emphasize the need for more adaptive and intelligent security frameworks that can respond to dynamic threat landscapes.

The Gap While existing solutions provide effective mechanisms for detecting known attacks and improving accuracy through machine learning, there is still a lack of a unified system that integrates real-time monitoring, rule-based filtering, and behavior-based anomaly detection within a scalable cloud-based architecture. Most traditional systems remain reactive in nature and are not fully capable of handling zero-day vulnerabilities or providing comprehensive visibility into web traffic. This gap highlights the need for a robust, intelligent,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

and scalable Web Application Firewall that can proactively detect, analyze, and mitigate modern web-based threats while ensuring high performance and usability.

II. Overview of SIEGEWALL

SiegeWall is a comprehensive cloud-based Web Application Firewall (WAF) designed to bridge the gap between modern web security requirements and scalable application deployment. Unlike traditional firewalls that rely on static rule- based and reactive security mechanisms responding only after an attack is identified, SiegeWall is built on a proactive defense architecture. It is specifically engineered for modern web envi- ronments where applications are exposed to sophisticated threats such as SQL injection, cross-site scripting (XSS), zero-day at- tacks, and distributed denial-of-service (DDoS), which often bypass conventional security systems.

The system operates as a reverse proxy positioned between the client and the web server, enabling real-time monitoring and filtering of all incoming HTTP/HTTPS requests. SiegeWall in- tegrates both rule-based filtering and machine learning-based behavior analysis using the K-Nearest Neighbors (KNN) algo- rithm to detect malicious activities. By analyzing request pat- terns, payload structures, and traffic behavior, the system can identify both known and previously unseen threats. Addition- ally, SiegeWall provides a centralized dashboard for monitoring traffic, analyzing logs, and managing security rules. By com- bining intelligent threat detection, automated response mech- anisms, and scalable cloud deployment, SiegeWall serves as a unified, secure, and efficient platform for protecting modern web applications.

A. *KEY FEATURES*

1) **Reverse Proxy Security Module (Core Protection Layer):**
   - Intercepts all incoming HTTP/HTTPS requests before they reach the web application, ensuring complete traffic inspection.
   - Filters malicious requests in real time, preventing unau- thorized access and attack execution.
2) **Intelligent Threat Detection Engine:**
   - Combines rule-based filtering with machine learning- based behavior analysis using the KNN algorithm.
   - Detects both known attacks and zero-day style anomalies by analyzing request patterns and traffic behavior.
3) **Automatic IP Blocking Mechanism:**
   - Identifies repeated malicious activity from specific IP addresses and blocks them automatically.
   - Prevents further attack attempts and reduces system exposure to recurring threats.
4) **Centralized Monitoring and Logging System:**
   - Maintains detailed logs of all incoming requests, includ- ing source IP, request type, timestamps, and detected threats.
   - Enables administrators to analyze attack patterns and improve security strategies.
5) **Administrator Dashboard and Visualization:**
   - Provides a real-time dashboard displaying traffic statis- tics, alerts, and attack logs.
   - Offers visualization tools for monitoring system activity and responding quickly to threats.
6) **Scalable Cloud-Based Architecture:**
   - Supports deployment across multiple web applications in distributed and cloud environments.
   - Ensures high availability, performance, and scalability for handling large volumes of web traffic.

The SiegeWall framework incorporates several key components to ensure robust security, real-time threat detection, and efficient web application protection.

III. PROPOSED SYSTEM AND DESIGN

The proposed system, SiegeWall, redefines traditional web ap- plication security by integrating intelligent threat detection into a scalable cloud-based Web Application Firewall architecture. Its core innovation lies in combining rule-based filtering with behavior-based anomaly detection using machine learning, en- abling proactive identification and mitigation of both known and unknown threats. Operating as a reverse proxy, the system con- tinuously analyzes incoming HTTP/HTTPS requests in real time, ensuring that malicious traffic is blocked before it reaches the web application.

The system is composed of three core components:

- **WAF Filtering Engine:**This module serves as the primary defense mechanism by applying predefined security rules to detect and block common web attacks such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and path traversal. It ensures fast and efficient filtering of known threats based on established attack signatures.
- **Behavior Analysis Module:** This component incorporates machine learning techniques, specifically the K-Nearest Neighbors (KNN) algorithm, to analyze traffic patterns and request behavior in real time. It identifies anomalies and suspicious activities, enabling the system to detect previously unseen or zero-day style attacks beyond traditional rule-based approaches.
- **Reverse Proxy and Monitoring System:** Acting as an intermediary between clients and the web server, this module routes all incoming traffic through the WAF for inspection. It also maintains detailed logs of requests, including source IP, timestamps, and detected threats, and provides a centralized dashboard for real-time monitoring and analysis.

Overall, the SiegeWall system stands as a robust and scalable solution for modern web application security, strengthening de- fense mechanisms while maintaining system performance and usability. By shifting from reactive filtering to proactive and intelligent threat detection, it establishes

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

a reliable and efficient environment capable of mitigating evolving cyber threats.

A. SYSTEM ARCHITECTURE

The system architecture of SiegeWall demonstrates how the framework ensures continuous protection through a layered design that separates traffic filtering, behavior analysis, and user interaction. The process is centered on the WAF Filtering Engine, which acts as the core security layer. When a client initiates a request, the reverse proxy intercepts it before it reaches the web application. The request is then passed to the filtering engine, where it is analyzed against predefined security rules to detect known attack patterns such as SQL injection and cross-site scripting (XSS).

Simultaneously, the Behavior Analysis Module operates as an intelligent layer, examining request patterns and traffic behav- ior using the K-Nearest Neighbors (KNN) algorithm to identify anomalies or "zero-day" style threats. The top layer, consisting of the Monitoring and Dashboard System, provides a centralized interface for administrators, ensuring real-time visibility, log analysis, and system control without compromising the core security mechanisms.

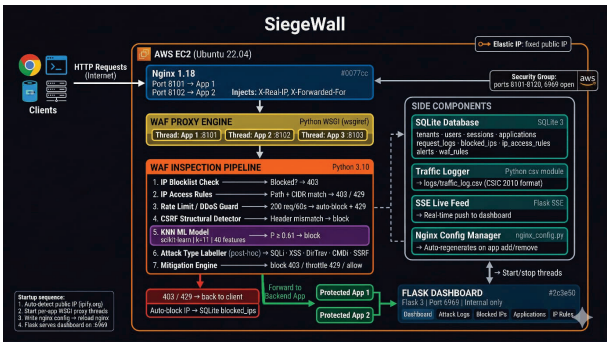

Fig. 1. Architecture Diagram

B. SYSTEM DESIGN

The system's design is further detailed through Use Case and Data Flow Diagrams. The Use Case Diagram shows the interactions between the Administrator and the SiegeWall system. The Administrator initiates actions such as monitoring traffic, analyzing logs, configuring rules, and managing blocked IP addresses. The SiegeWall system processes these requests while enforcing security through the filtering and behavior analysis modules.

The Data Flow Diagrams illustrate the movement of data within the system. The Level 0 DFD shows the main entities: the Client, the SiegeWall system, and the Web Server. The Client sends HTTP/HTTPS requests to the system, which processes them and communicates with the web server based on security decisions.

The Level 0 Data Flow Diagram illustrates the high-level boundary of the SiegeWall system. It depicts the WAF as the central processing entity that mediates communication between the client and the web application.

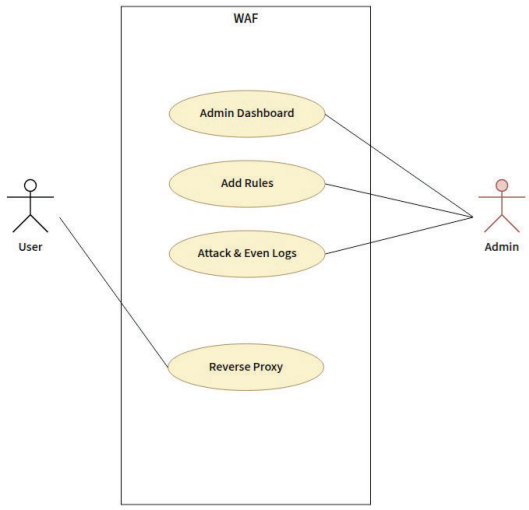

Fig. 2. Use Case Diagram

- Input: The process begins with the Client sending HTTP/ HTTPS requests to the WAF system.
- Process: The SiegeWall system analyzes these requests using rule-based filtering and behavior analysis before deciding whether to allow or block them.
- Feedback: The Web Server returns responses for valid requests, while blocked requests generate alerts and logs, which are then displayed on the dashboard.

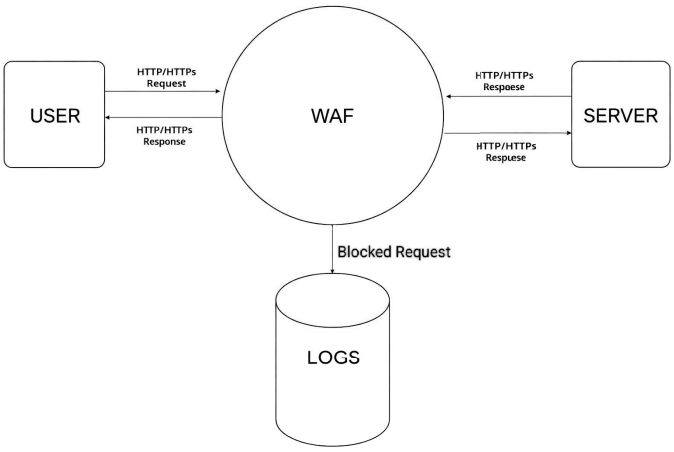

Fig. 3. Data Flow Diagram (Level 0)

The Level 1 Data Flow Diagram expands the central process to detail interactions between specific modules and data storage components.

- Security Analysis: The WAF Filtering Engine is shown as a composite module that performs rule matching, payload inspection, and attack detection, ensuring thorough analysis of incoming requests. The Behavior Analysis

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Module provides additional anomaly detection based on traffic patterns.

- Monitoring and Logging: The system maintains a continuous data flow where all requests and detected threats are stored in the database. The dashboard retrieves this data to provide real-time visualization and alerts.
- System Management: The Administrator interacts with the system through the dashboard to configure rules, manage blocked IPs, and monitor system performance.
- Storage and Communication: The database stores logs, request details, and attack information, while the reverse proxy ensures secure communication between clients and the web server.

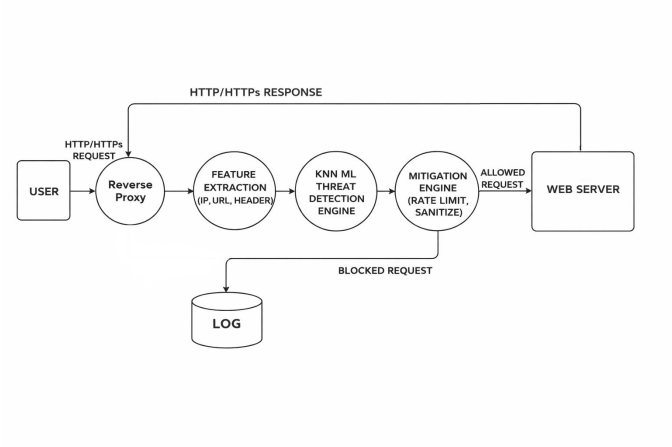

Fig. 4. .Data Flow Diagram (Level 1)

IV. IMPLEMENTATION

The development of the SiegeWall system was executed in a structured, phased approach to ensure the stability of the core security modules before integrating monitoring and user-facing features.

A. *MODULES*

Phase 1: System Design and Architecture The initial phase focused on defining the overall system architecture, including the reverse proxy model and integration of filtering and analysis components. This stage involved designing the workflow for handling HTTP/HTTPS requests and establishing the foundation for scalable cloud-based deployment.

Phase 2: WAF Core and Filtering Engine This critical phase involved the development of the WAF Filtering Engine and reverse proxy mechanism. Using Python and Flask along with Nginx, the system was designed to intercept incoming requests, apply rule-based filtering, and block known web attacks such as SQL injection, cross-site scripting (XSS), and CSRF in real time.

Phase 3: Machine Learning Integration This phase focused on implementing the behavior-based detection module using machine learning techniques:

- Part 1 (Model Integration):The K-Nearest Neighbors (KNN) algorithm was integrated to analyze request patterns and classify traffic as normal or malicious.
- Part 2 (Behavior Analysis): The system was enhanced to monitor request frequency, payload structure, and user behavior to detect anomalies and identify previously unseen threats.

Phase 4: Dashboard and Monitoring System The final phase focused on usability and system management. A centralized dashboard was developed using React and web technologies to provide real-time traffic monitoring, visualization of attack logs, alert notifications, and administrative control over security rules and IP blocking mechanisms.

B. *TOOLS AND TECHNIQUES*

The development of SiegeWall utilizes a combination of modern web technologies and machine learning tech- niques to ensure efficient real-time threat detection and scalable web application security.

- *Programming Languages*: The implementation adopts a multi-language approach to optimize different components of the system. Python serves as the primary language for developing the backend logic, request handling, and machine learning integration due to its simplicity and extensive library support. JavaScript is used for building interactive frontend components, while HTML and CSS are utilized for structuring and designing the user interface. This combination enables efficient system operation and a responsive user experience.
- *Frameworks and Libraries*: The backend of the system is developed using the Flask framework, which facilitates lightweight and efficient handling of HTTP requests and API communication. The behavior-based detection mechanism is implemented using machine learning techniques, specifically the K-Nearest Neighbors (KNN) algorithm, to analyze traffic patterns and identify anomalies. On the frontend, React is used to create a dynamic and interactive dashboard interface, allowing real-time visualization of system activity.
- *Database and Storage*: SQLite is used as the primary database for storing system logs, detected threats, and request details. It provides a lightweight and efficient storage solution without requiring a separate database server, enabling quick data retrieval and analysis.
- *Development and Monitoring Tools*: The development process was carried out using standard development environments such as Visual Studio Code for coding and debugging. Nginx is utilized as the reverse proxy server to handle incoming HTTP/HTTPS traffic and route it through the WAF filtering layer. The system is deployed and tested on both Windows and Linux environments to ensure compatibility, reliability, and consistent performance across platforms.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

V. Results and Discussion

The implementation of SiegeWall successfully achieved its pri- mary objectives of integrating a real-time Web Application Fire- wall with an intelligent behavior-based detection mecha- nism. The system was validated through a series of functional tests designed to evaluate the rule-based filtering engine and the KNN- based anomaly detection model against common evasion tech- niques used in web-based attacks.

Functional Testing and Threat Detection The core security module demonstrated high accuracy in identifying "zero-day" style anomalies that traditional rule-based firewalls often miss. In the prototype testing environment, the system successfully executed the following security checks:

- Attack Filtering: The system correctly flagged mali- cious requests such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and path traver- sal attacks, blocking them before reaching the web server.
- Behavioral Monitoring: The KNN-based detection engine analyzed request patterns such as frequency and payload structure, identifying abnormal traffic and clas- sifying it as suspicious or malicious.
- IP Blocking Mechanism: The system detected repeated malicious activity from specific IP addresses and auto- matically blocked them, preventing further attack at- tempts.

System Performance and Resource Efficiency Performance testing confirmed the system's ability to handle real-time traffic efficiently without introducing significant latency. The reverse proxy architecture maintained stable operation while processing multiple concurrent HTTP/HTTPS requests and supporting scalable cloud-based deployment.

Monitoring and Visualization The administrator dashboard successfully provided real-time visibility into system activity. Functional testing verified that traffic statistics, attack logs, and alerts were displayed accurately, enabling efficient monitoring and quick response to threats.

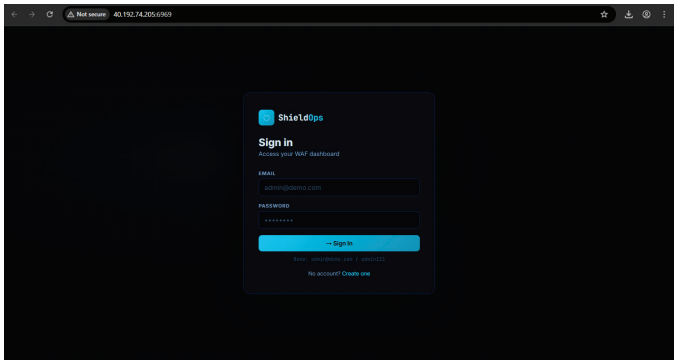

Fig. 5. SeigeWall Login Page

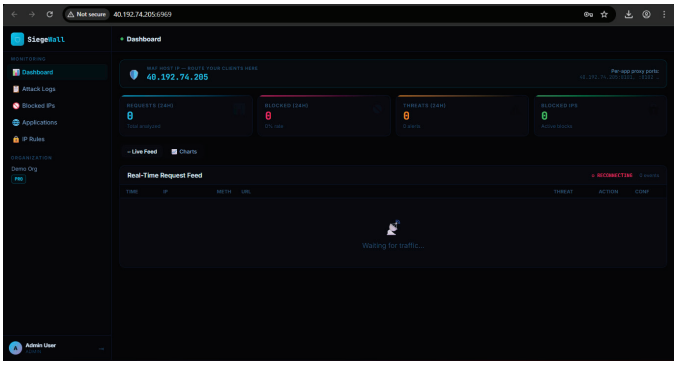

Fig. 6. SeigeWall Dashboard with Real-Time Request Feed

VI. Conclusion and Future Work

A. Conclusion

The SiegeWall system was successfully developed as a cloud-based Web Application Firewall designed to enhance the security of web applications. The system operates as a reverse proxy that filters and monitors all incoming HTTP/HTTPS traffic before it reaches the web server. This approach en- sures that malicious requests are detected and blocked in real time, preventing potential damage to the application. The project demonstrates how a WAF can effectively protect against common web attacks such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and other threats. By combining rule-based filtering with behavior-based analysis using the K-Nearest Neighbors (KNN) algorithm, the system improves its ability to detect both known and unknown attacks. SiegeWall records important details such as source IP addresses, request types, timestamps, and de- tected threats. This information provides valuable insights for analyzing attack patterns and improving security strategies. The automatic IP blocking mechanism further strengthens the system by preventing repeated attacks from malicious sources. The system also includes a user-friendly dashboard that allows administrators to monitor traffic, view alerts, and analyze logs in real time. Overall, SiegeWall provides a scal- able, efficient, and cost-effective solution for enhancing web application security. It highlights the importance of combining real-time monitoring, intelligent filtering, and adaptive analysis in modern cybersecurity systems.

B. Future Work

Future enhancements of SiegeWall can focus on improv- ing its intelligence, scalability, and usability to meet evolv- ing cybersecurity demands. By integrating advanced machine learning and deep learning techniques, the system can achieve higher detection accuracy and identify complex attack patterns more effectively. It can also be scaled to support large-scale enterprise environments with high traffic and distributed appli- cations. Implementing cloud-based multi-region deployment will enhance performance, ensure redundancy, and enable global threat monitoring. Additionally, strengthening the sys- tem to better detect and respond to zero-day vulnerabilities

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

will improve overall security against sophisticated attacks. Finally, upgrading the user interface with advanced analytics, visualization tools, and detailed reporting features will provide deeper insights into system activities and improve user experience.

REFERENCES

[1] M. Maheshwari, A. Nayak, A. Sethy, and S. G, "Adaptive web application firewall for multi-threat detection," *2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS)*, pp. 232–238, 12 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10823239

[2] A. Kumar, J. B. Simha, and R. Agarwal, "Machine learning-based web application firewall for real-time threat detection," pp. 1–8, 11 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10912239

[3] M. E. Durmuşkaya and S. Bayraklı, "Web application firewall based on machine learning models," *PeerJ Computer Science*, vol. 11, p. e2975, 07 2025. [Online]. Available: https://peerj.com/articles/cs-2975/

[4] B. R. Dawadi, B. Adhikari, and D. K. Srivastava, "Deep learning technique-enabled web application firewall for the detection of web attacks," *Sensors*, vol. 23, p. 2073, 02 2023.

[5] M. Hosain, S. A. Shuvo, M. Ogbe, M. S. Jalal Mazumder, Y. Rahman, M. A. Hakim, and A. Pandey, "Web technologies security in the ai era: A survey of cdn-enhanced defenses," *2025 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, pp. 180–186, 11 2025.

[6] A. Razzaq, A. Hur, S. Shahbaz, M. Masood, and H. F. Ahmad, "Critical analysis on web application firewall solutions," in *2013 IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS)*. IEEE, 2013, pp. 1–6.

[7] S. Prandl, M. Lazarescu, and D.-S. Pham, "A study of web application firewall solutions," in *International conference on information systems security*. Springer, 2015, pp. 501–510.

[8] N. Gupta, A. Saikia, and D. Sanghi, "Web application firewall," *Indian Institute of Technology, Kanpur*, vol. 61, p. 62, 2007.

[9] V. Clincy and H. Shahriar, "Web application firewall: Network security models and configuration," in *2018 IEEE 42nd annual computer software and applications conference (COMPSAC)*, vol. 1. IEEE, 2018, pp. 835–836.

[10] A. Shaheed and M. B. Kurdy, "Web application firewall using machine learning and features engineering," *Security and Communication Networks*, vol. 2022, no. 1, p. 5280158, 2022.