International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 1 Jan. 2026, pp: 241-257 www.ijaem.net ISSN: 2395-5252

Design and Development of an Intelligent Web Application Firewall (WAF) for Enhanced Cyber Threat Mitigation

Dickson Kalipa Mukonga Advisor: Mr Moses Mupeta Department of ICT Department of ICT
Information and Communication University *Information and Communication University*
Lusaka, Zambia *Lusaka, Zambia*

Date of Submission: 05-01-2026 Date of Acceptance: 16-01-2026

ABSTRACT - The increasing reliance on web-based applications has made them prime targets for cyberattacks, including SQL injections, cross-site scripting (XSS), and distributed denial-of-service (DDoS) attacks. Traditional network firewalls often fail to provide adequate protection at the application layer, highlighting the need for specialized solutions. This study focuses on the design and development of a Web-Based Application Firewall (WAF) aimed at protecting web applications from common and emerging threats. The WAF system is designed using a layered security approach, incorporating request filtering, pattern recognition, and real-time monitoring to detect and mitigate malicious activity.

The application is developed using modern web technologies, ensuring scalability, usability, and integration with existing web infrastructure. Performance evaluations demonstrate the effectiveness of the system in preventing attacks while maintaining optimal application response times. The findings indicate that a well-designed WAF can significantly enhance the security posture of web applications, providing organizations with a proactive defense mechanism against evolving cyber threats. This research contributes to the field of cybersecurity by offering a practical framework for developing and deploying application-layer security solutions.

Keywords— Web Application Firewall, cross-site scripting, cross-site request forgery, Distributed Denial of Services, Artificial Intelligence, SQL injection, Anomalies, Firewall, HTTP, Prototype, Self-learning.

I. INTRODUCTION

1.1 Background of Study

The proliferation of web applications has significantly transformed business operations, enabling services such as online banking, e-commerce, and social networking. However, this widespread adoption has also made web applications prime targets for cyberattacks. Vulnerabilities like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF) continue to pose substantial risks to application security. According to the Open Web Application Security Project (OWASP), these vulnerabilities remain prevalent across many web applications, underscoring the critical need for robust security measures (OWASP, 2023).

Traditional network firewalls, designed to protect the network perimeter, often fall short in defending against application-layer attacks. These firewalls typically operate at lower layers of the OSI model and may not adequately inspect HTTP/HTTPS traffic for malicious payloads. Consequently, there is a pressing need for specialized security solutions that can monitor, filter, and block malicious web traffic targeting web applications. Research indicates that while traditional firewalls are essential, they do not replace the necessity for application-specific defenses like WAFs (Ahmed & Saini, 2020; Gupta, Singh, & Goyal, 2020).

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 1 Jan. 2026, pp: 241-257 www.ijaem.net ISSN: 2395-5252

A Web Application Firewall (WAF) addresses this gap by providing a dedicated layer of defense at the application level. WAFs are designed to protect web applications by filtering and monitoring HTTP traffic between a web application and the Internet. They help safeguard against various application-layer attacks, including SQL injection, XSS, and CSRF. Studies have shown that WAFs can effectively mitigate these threats by analyzing incoming traffic for malicious patterns and blocking harmful requests (Alshammari, Alharthi, &Alrashed, 2022; Kim, Jeon, & Lee, 2021; Patel, Shah, & Mehta, 2021).

This study focuses on the design and development of a web-based application firewall aimed at enhancing the security of web applications. The research explores the architecture, implementation, and effectiveness of the proposed WAF in mitigating common web application vulnerabilities. By developing a robust WAF, this study contributes to the ongoing efforts to bolster web application security and protect sensitive data from unauthorized access and exploitation (Sharma & Pandey, 2021; Kumar & Singh, 2022; Gartner, 2021; Verizon, 2022).

1.2 Statement of the Problem

Web applications have become prime targets for cyberattacks due to the valuable information they handle, including personal data, financial information, and business-critical resources. Common vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF) continue to be exploited at alarming rates (OWASP, 2023; Ahmed & Saini, 2020). Traditional network firewalls are insufficient for protecting web applications against such application-layer attacks because they operate at lower OSI layers and cannot effectively analyze HTTP/HTTPS traffic (Gupta, Singh, & Goyal, 2020).

If these vulnerabilities are not addressed through specialized security measures such as Web Application Firewalls (WAFs), organizations risk severe consequences including data breaches, financial losses, reputational damage, and service disruptions caused by attacks such as Application Layer DDoS.

Therefore, there is an urgent need for a robust, web-based WAF capable of monitoring, filtering, and blocking malicious traffic at the application layer to safeguard web applications and minimize the impact of evolving cyber threats (Alshammari, Alharthi, &Alrashed, 2022; Kim, Jeon, & Lee, 2021).

1.2.1 General Objective

The main objective of this study was to Design and Development of a Web-Based Application Firewall (WAF).

1.2.2 Specific Objectives

To achieve the main objective, the following specific objectives were as follows:

I. To find out about the current WAF system

II To design and implement a customizable web-based WAF that provides real-time protection against common web application vulnerabilities

III To provide an easy-to-deploy solution that can be adopted by small and medium-sized enterprises (SMEs) with limited technical and financial resources.

II. LITERATURE REVIEW

2.1 Trends in Information Technology

The integration of advanced information technology (IT) in disaster management has revolutionized how communities and agencies respond to and mitigate the impacts of natural disasters. This section explores the key trends in information technology that are shaping modern disaster response systems:

1.Big Data and Advanced Analytics: The advent of big data analytics has significantly enhanced disaster prediction, monitoring, and response capabilities. By processing vast amounts of data from diverse sources such as weather stations, satellites, social media, and IoT devices, big data analytics facilitate real-time situational awareness and decision-making (Chen et al., 2014). Advanced

analytics, including machine learning and artificial intelligence (AI), are increasingly used to predict disaster events, model potential impacts, and optimize resource allocation (Bostrom et al., 2019).

2.Internet of Things (IoT) and Sensor Networks: IoT technology has expanded the scope of disaster monitoring and response. Sensors and connected devices are deployed across vulnerable areas to collect real-time data on environmental conditions, infrastructure status, and community activities. These IoT networks provide critical data that enhances situational awareness and enables proactive disaster management (Gubbi et al., 2013). For example, IoT sensors can detect water levels, soil moisture, and seismic activities, providing early warnings for floods, droughts, and earthquakes (Zhang et al., 2021).

3.Geographic Information Systems (GIS) and Remote Sensing: GIS and remote sensing technologies play a crucial role in disaster management by providing detailed spatial information and imagery. These technologies help in mapping disaster-affected areas, analyzing terrain changes, and planning evacuation routes. Satellite imagery and aerial drones equipped with sensors offer high-resolution data that support damage assessment, resource distribution, and recovery planning (Cutter et al., 2016). The integration of GIS with real-time data analytics enhances the ability to visualize disaster scenarios and make informed decisions swiftly

4.Cloud Computing and Scalable Infrastructure: Cloud computing has transformed disaster management by offering scalable and flexible infrastructure for data storage, processing, and analysis. Cloud platforms enable real-time data sharing and collaboration among various stakeholders, ensuring that critical information is accessible anytime and anywhere. This technology supports the development of robust disaster management systems that can handle large volumes of data and support complex simulations and modeling (Hassan et al., 2017).

5. Mobile and Web-Based Applications: The proliferation of smartphones and internet connectivity has led to the development of mobile and web-based applications designed for disaster response. These applications facilitate real-time communication, coordination, and information sharing among emergency responders, government agencies, and the public. Features such as push notifications, location tracking, and crowd-sourced data collection enhance the effectiveness of disaster response efforts (Gordon et al., 2020). Mobile apps also support community engagement by providing platforms for disaster alerts, safety tips, and emergency contact information.

6. Artificial Intelligence (AI) and Machine Learning: AI and machine learning are increasingly integrated into disaster management systems to enhance predictive analytics, automate data processing, and improve decision-making. AI algorithms can analyze historical data and real-time inputs to predict disaster occurrences, assess risks, and recommend response strategies. Machine learning models are also used to optimize resource allocation, forecast disaster impacts, and enhance the accuracy of early warning systems (Zhang et al., 2021). For instance, AI-driven models can analyze weather patterns and seismic data to predict the likelihood of earthquakes or floods with higher precision.

7.Blockchain Technology: Blockchain technology is gaining traction in disaster management for its potential to enhance transparency, security, and accountability in resource distribution and logistics. By providing a decentralized and immutable ledger, blockchain ensures the integrity of transactions and data exchanges among stakeholders. This technology can streamline aid distribution, track supply chains, and prevent fraud and corruption in disaster relief operations (Tapscott & Tapscott, 2016).

8.Robotics and Autonomous Systems: Robotics and autonomous systems are being developed for search and rescue missions, damage assessment, and logistical support in disaster-affected areas. Drones unmanned aerial vehicles (UAVs), and ground robots equipped with sensors and cameras are used to explore hazardous environments, deliver supplies, and gather critical data. These technologies enhance the safety and efficiency of disaster response operations, particularly in areas that are difficult to

access or dangerous for human responders (Dahiya et al., 2019)

Overall, research demonstrates that household gender patterns through labor market participation, division of unpaid domestic work, access to education, and decision-making power play a pivotal role in shaping income inequality. Policies promoting gender equality, targeted foreign aid, technological inclusion, and cultural change are critical for reducing household income disparities and fostering broader economic development. (Zambian Ministry of Gender, 2021, p. 25).

2.2. REVIEW OF LITERATURE

The necessity for robust web application security has escalated with the proliferation of online services and the sophistication of cyber threats. Web Application Firewalls (WAFs) have emerged as critical components in defending against application-layer attacks, which are often overlooked by traditional network security measures.

WAFs are designed to monitor, filter, and block HTTP/HTTPS traffic to and from a web application, protecting it from attacks such as SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF) (OWASP, 2023). Tekerek et al. (2014) developed a hybrid WAF that integrates signature-based and anomaly-based detection methods, enhancing the system's ability to identify and mitigate a broader range of attacks. Similarly, Riera et al. (2021) conducted a systematic review on anomaly detection technologies, highlighting their effectiveness in identifying novel attack patterns that traditional signature-based systems might miss.

WAFs employ various detection techniques, including signature-based, anomaly-based, and hybrid approaches. Signature-based WAFs match incoming traffic against known attack patterns, providing quick detection but limited to known threats. Anomaly-based WAFs establish a baseline of normal traffic behavior and flag deviations as potential attacks, offering protection against unknown threats but with a higher risk of false positives. Hybrid WAFs combine both methods to leverage the strengths of each approach (Kim, Jeon, & Lee, 2021). Alaria (2019) analyzed the performance of WAFs, noting the trade-offs between detection accuracy and system performance.

Common web application vulnerabilities include SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF). SQL Injection allows attackers to execute arbitrary SQL queries, potentially compromising the database. XSS enables the injection of malicious scripts into web pages viewed by other users, leading to session hijacking or data theft. CSRF tricks the victim into submitting a request that they did not intend, potentially performing unwanted actions on their behalf (OWASP, 2023). The prevalence of these vulnerabilities underscores the importance of WAFs in providing an additional layer of security.

Evaluating the performance of WAFs involves assessing metrics such as latency, throughput, false positive and negative rates, and scalability. Gupta, Singh, and Goyal (2020) investigated dynamic rule adaptation in WAFs, emphasizing the importance of real-time updates to firewall rules to respond to evolving attack patterns. Kim, Jeon, and Lee (2021) explored hybrid WAF architectures combining signature-based and anomaly-based detection methods, demonstrating improved accuracy in identifying malicious traffic while reducing false positives. Patel, Shah, and Mehta (2021) examined customizable WAF solutions suitable for diverse web application environments, stressing the importance of flexibility and scalability in deployment. Sharma and Pandey (2021) conducted a comparative analysis of various WAF techniques, highlighting challenges such as configuration complexity, performance overhead, and adaptability to new threats.

Recent advancements in machine learning and artificial intelligence have been integrated into WAFs to improve their adaptability and accuracy. Alshammari et al. (2022) proposed a machine learning-based WAF capable of detecting zero-day attacks, demonstrating the potential for intelligent systems to enhance web application security.

Additionally, the rise of cloud computing has led to the development of cloud-based WAF solutions that offer scalability and ease of deployment, making advanced security accessible to a broader range of organizations (Gartner, 2021).

Despite their advantages, WAFs face several challenges. The dynamic nature of web applications and attack strategies necessitates continuous updates to WAF rules and configurations. Additionally, the complexity of deploying and managing WAFs can be a barrier for organizations with limited cybersecurity expertise (Patel, Shah, & Mehta, 2021). Moreover, the effectiveness of WAFs can be compromised if they are not properly configured or maintained, leading to potential vulnerabilities (Sharma & Pandey, 2021).

2.3 Related Works

Several studies and systems have been developed to enhance web application security through the use of Web Application Firewalls (WAFs). Reviewing these works helps to identify the strengths and limitations of existing solutions, providing insights for designing an improved web-based WAF.

1. Cloud-Based Scalable WAF by Ahmed & Saini (2020)

Ahmed and Saini (2020) developed a cloud-based WAF designed to provide scalable protection for modern web applications. The system filters malicious HTTP requests and mitigates attacks such as SQL Injection and XSS. One of its strengths is scalability, allowing multiple applications to be protected simultaneously without significant performance degradation. However, the system relies primarily on signature-based detection, which may limit its effectiveness against zero-day attacks or novel attack patterns.

2. Machine Learning-Based WAF by Alshammari, Alharthi, &Alrashed (2022)

This system integrates machine learning algorithms to detect and block zero-day attacks in real time. By analyzing network traffic patterns and learning from normal behavior, it can identify previously unknown attack vectors. The study demonstrated a significant improvement in detection rates compared to traditional signature-based WAFs. Nevertheless, the system requires extensive training data and computational resources, which may pose challenges for deployment in small and medium enterprises (SMEs) with limited infrastructure.

3. Hybrid WAF Combining Signature and Anomaly Detection by Kim, Jeon, & Lee (2021)

Kim, Jeon, and Lee (2021) proposed a hybrid WAF architecture that combines signature-based and anomaly-based detection methods. The hybrid approach allows the system to detect both known attacks and anomalies in user behavior that may indicate emerging threats. This system reduces false positives compared to purely anomaly-based WAFs while providing better coverage against new threats than signature-only systems. However, the complexity of configuring and maintaining hybrid rules can be a challenge, especially for organizations without dedicated cybersecurity teams.

2.4 Gaps in Literature

A review of existing literature and related works reveals significant advancements in Web Application Firewall (WAF) technologies. Studies have explored cloud-based WAFs (Ahmed & Saini, 2020), machine learning-enhanced WAFs (Alshammari, Alharthi, &Alrashed, 2022), and hybrid detection approaches combining signature and anomaly-based methods (Kim, Jeon, & Lee, 2021). While these systems provide substantial improvements in web application security, several gaps remain, which justify further research and development.

1. Limited Accessibility for SMEs:

Most existing WAF solutions are designed for large enterprises with substantial IT infrastructure and cybersecurity budgets. Small and medium enterprises (SMEs), which represent a large portion of web application users, often lack affordable and

easy-to-deploy WAF solutions (Kumar & Singh, 2022; Gartner, 2021).

2. Complexity of Configuration and Management:

Hybrid and machine learning-based WAFs provide higher detection accuracy but require complex configuration, continuous monitoring, and skilled personnel to manage. This complexity poses challenges for organizations without dedicated cybersecurity teams (Patel, Shah, & Mehta, 2021; Sharma & Pandey, 2021).

3. Limited Adaptability to Emerging Threats:

Traditional signature-based WAFs are effective against known attacks but fail to detect zero-day exploits and novel attack patterns. Even machine learning-based systems require extensive training data to adapt, which can delay response to new threats (Alshammari, Alharthi, &Alrashed, 2022).

4. Performance and Scalability Concerns:

Some existing WAF solutions introduce latency or performance overhead, particularly when deployed in high-traffic environments or for multiple applications. This can affect user experience and limit the applicability of these systems in real-world scenarios (Gupta, Singh, & Goyal, 2020).

5. Lack of Integration and User-Friendliness:

Many WAF solutions focus on technical effectiveness but neglect ease of integration with existing systems and user-friendliness for administrators. A user-friendly, web-based interface that simplifies management while maintaining robust security is largely missing in current solutions.

2.5 Conceptual Frameworks

The conceptual framework serves as a blueprint guiding the design and development of the proposed Web-Based Application Firewall (WAF), illustrating the flow of data, processes for detecting and mitigating threats, and interactions between system components. It demonstrates how the WAF analyzes, filters, and manages web traffic to protect applications from various cyberattacks. Grounded in the principles of web application security, intrusion detection systems, and real-time traffic analysis, the framework integrates both signature-based and anomaly-based detection mechanisms to enhance the system's ability to identify known attack patterns while detecting emerging threats (Ahmed & Saini, 2020; Kim, Jeon, & Lee, 2021; Alshammari, Alharthi, &Alrashed, 2022). The primary inputs include all incoming HTTP/HTTPS requests, whether legitimate or malicious attempts such as SQL Injection, Cross-Site Scripting (XSS), and other OWASP Top 10 vulnerabilities (OWASP, 2023).

The detection and filtering mechanisms combine signature-based detection for known threats, anomaly-based detection for unknown or zero-day attacks, and hybrid filtering to reduce false positives while maintaining accuracy. Based on the analysis, the WAF either blocks malicious requests or allows legitimate traffic, with all activities logged for review and analysis, while generating alerts and reports for administrators to enable real-time monitoring and response (Gupta, Singh, & Goyal, 2020).

Logging and reporting mechanisms store details of every request and decision, supporting trend analysis, identification of new attack patterns, and updating of detection rules, thus creating a continuous feedback loop for system improvement (Patel, Shah, & Mehta, 2021). Additionally, ongoing system monitoring allows administrators to fine-tune configurations, view alerts, and respond promptly to threats, ensuring adaptability to evolving cybersecurity environments. Overall, the conceptual framework emphasizes the continuous interaction between detection mechanisms, monitoring, and feedback loops to create a proactive defense model that secures web applications against both known and emerging threats.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 1 Jan. 2026, pp: 241-257 www.ijaem.net ISSN: 2395-5252

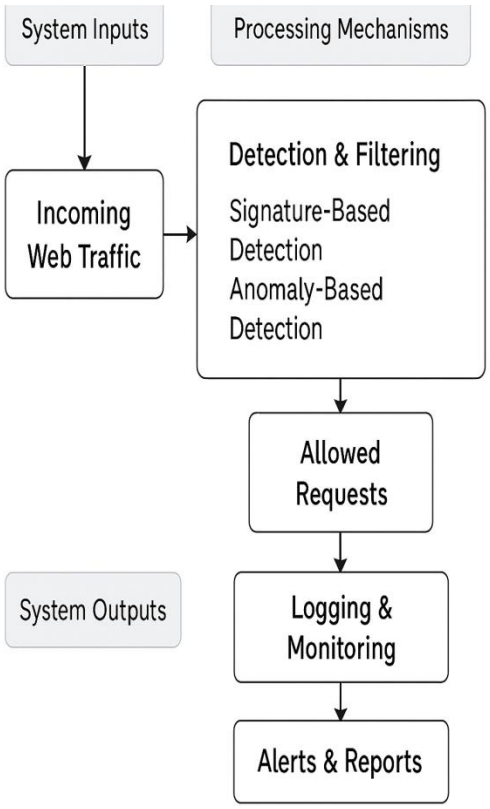


Figure 1: Conceptual Frameworks

III. RESEARCH METHODOLOGY

This chapter presents the methodology employed in the design and development of the proposed Web-Based Application Firewall (WAF). It outlines the research approach, system development model, data collection methods, system architecture, tools, and technologies used to implement the system. The methodology provides a structured approach to achieving the research objectives, ensuring that the WAF is both effective in mitigating cyber threats and efficient in handling web traffic. The chapter also explains the processes involved in system design, implementation, testing, and evaluation, highlighting how each stage contributes to the overall security, performance, and reliability of the application. By detailing the methodological framework, this chapter establishes the foundation for replicability and validation of the proposed system.

3.1 Research Design

The software development methodology will be used to implement a Customer Relation Management system.

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete.

In this waterfall model, the phases do not overlap.

Waterfall Model - Design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.

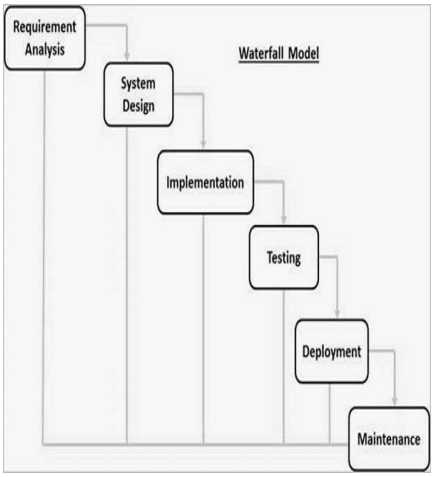


Figure 1:Waterfall Model

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 1 Jan. 2026, pp: 241-257 www.ijaem.net ISSN: 2395-5252

The sequential phases in Waterfall model are −

- **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model".

In this model, phases do not overlap.

Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on internal and external factors. Some situations where the use of Waterfall model is most appropriate are −

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model - Advantages

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development, and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Some of the major advantages of the Waterfall Model are as follows −

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

3.2 DEVELOPMENT OF THE APPLICATION

The development of the proposed Web-Based Application Firewall (WAF) followed a systematic and structured approach, guided by the conceptual framework and baseline study findings. The process

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 1 Jan. 2026, pp: 241-257 www.ijaem.net ISSN: 2395-5252

began with the design of the system architecture, outlining the interaction between components such as traffic monitoring, detection mechanisms, filtering modules, logging, and administrative dashboards. Using a combination of PHP, HTML, JavaScript, and MySQL, the application was implemented to ensure real-time traffic analysis and threat mitigation. Both signature-based and anomaly-based detection techniques were integrated to provide a hybrid filtering approach, enhancing the WAF's capability to identify known attacks while detecting new or evolving threats. During development, attention was given to usability, system performance, and scalability to handle high volumes of web traffic efficiently.

Logging and reporting features were implemented to store detailed records of incoming requests, system decisions, and alerts, enabling administrators to monitor security events and fine-tune configurations as necessary. The development process also included iterative testing and debugging to ensure the application met the desired functional and security requirements, resulting in a robust, reliable, and adaptive Web-Based Application Firewall capable of safeguarding web applications against diverse cyber threats.

3.3 SYSTEM DESIGN

The system design of the proposed Web-Based Application Firewall (WAF) focuses on creating a structured and efficient framework that ensures effective monitoring, detection, and mitigation of web-based threats. The design encompasses the architecture, data flow, and interaction between key components, including traffic input, detection and filtering modules, logging, alert generation, and administrative interfaces. Emphasis was placed on integrating both signature-based and anomaly-based detection mechanisms to form a hybrid system capable of identifying known attack patterns while adapting to new and evolving threats.

The design also incorporates real-time monitoring, detailed reporting, and a feedback loop that allows continuous system optimization. Furthermore, considerations for scalability, performance, and user accessibility were incorporated to ensure that the WAF can handle varying volumes of web traffic while providing administrators with intuitive control over security configurations. The system design thus provides a comprehensive blueprint for the development and implementation of a robust, adaptive, and reliable Web-Based Application Firewall.

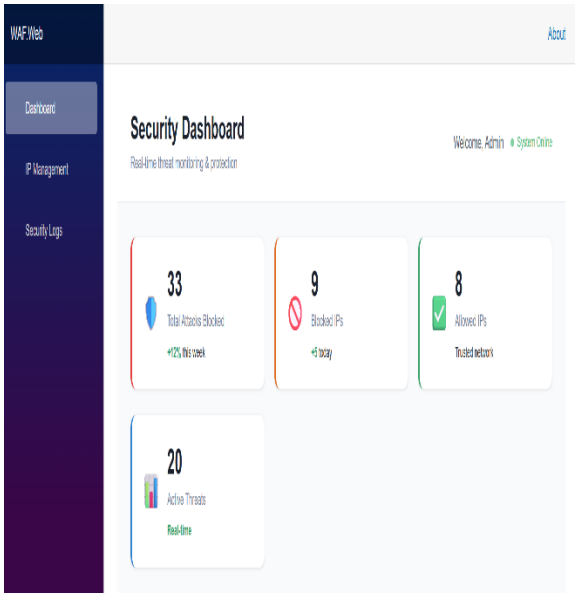


Figure 3: Admin Dashboard

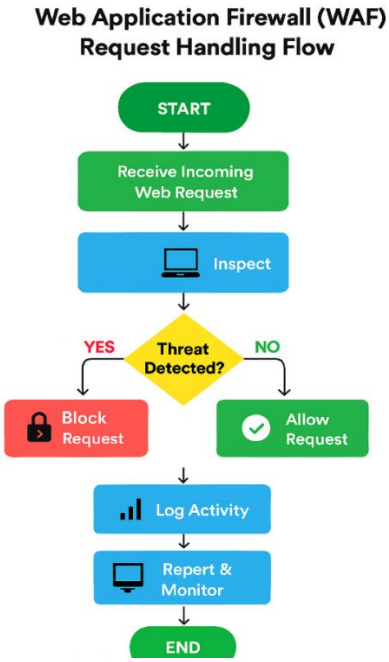


Figure 4: Activity Diagram

Subsystem Description for the Web-Based Application Firewall (WAF):

1. **User Registration and Authentication Subsystem**

This subsystem manages the registration and authentication of system administrators and security personnel who will configure, monitor, and manage the WAF. It ensures that only authorized users can access the firewall management interface.

Components:

i. Registration Form: Captures administrator information such as name, email, and role.

ii. Authentication Module: Handles login, password validation, and multi-factor authentication for secure access.

iii. Password Recovery: Provides mechanisms for secure password reset and recovery.

Processes:

i. Administrator submits registration details.

ii. System validates information and creates a secure user profile.

iii. On login, credentials are verified against the stored data.

iv. Successful login initiates a session with role-based access control.

2. Traffic Filtering and Monitoring Subsystem

This subsystem monitors incoming and outgoing web traffic, applying security rules to detect and block malicious requests, such as SQL injection, XSS, and other web-based attacks.

Components:

i. Traffic Inspection Engine: Analyzes HTTP/S requests in real-time for potential threats.

ii. Rule Set Manager: Stores and manages firewall rules for threat detection.

iii. Alert System: Notifies administrators of suspicious or blocked traffic.

Processes:

i. Incoming traffic is intercepted by the WAF.

ii. Requests are analyzed against predefined security rules.

iii. Suspicious requests are blocked, logged, and reported.

iv. Alerts are generated for administrators to review incidents.

3. Logging and Threat Analysis Subsystem

This subsystem records security events, analyzes attack patterns, and provides actionable insights to improve system defenses.

Components:

i. Log Management: Stores detailed records of all web traffic and blocked requests.

ii. Threat Analysis Engine: Identifies patterns and generates reports on potential attacks.

iii. Dashboard Interface: Visualizes logs and threat statistics in real-time.

Processes:

i. Security events are logged continuously.

ii. Analysis engine identifies trends and anomalies in traffic.

iii. Administrators can review dashboards to monitor WAF performance and threats.

4. Configuration and Rule Management Subsystem

This subsystem allows administrators to define, update, and manage firewall rules and security policies.

Components:

i. Rule Editor: Interface for creating and updating WAF rules.

ii. Policy Manager: Manages security policies across different web applications.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 1 Jan. 2026, pp: 241-257 www.ijaem.net ISSN: 2395-5252

iii. Backup and Restore: Ensures firewall configurations can be saved and restored.

Processes:

i. Administrator creates or modifies rules using the editor.

ii. Rules are applied to the traffic inspection engine.

iii. Configuration changes are saved and backed up for disaster recovery.

5. Reporting and Analytics Subsystem

Provides insights into web application security, system performance, and attack trends for informed decision-making.

Components:

i. Report Generator: Generates reports on blocked attacks, traffic patterns, and system health.

ii. Analytics Engine: Processes logs to identify potential vulnerabilities.

iii. Dashboard: Displays visual metrics for system monitoring and management.

Processes:

i. Data is extracted from logs and WAF operations.

ii. Reports are generated automatically or on-demand.

iii. Analytics identifies trends, frequent attack vectors, and performance metrics.

6. Security and Compliance Subsystem

Ensures the WAF operates securely and complies with relevant security standards, protecting sensitive web application data.

Components:

i. Access Control: Enforces role-based access for administrators.

ii. Encryption: Secures data transmission and stored configuration files.

iii. Compliance Monitoring: Checks adherence to web security standards and regulations.

Processes:

i. User roles and permissions are strictly enforced.

ii. Configuration data and logs are encrypted to maintain confidentiality.

iii. Regular audits and compliance checks ensure alignment with best practices.

Figure 5: Logs Panel Page

IV. FINDINGS AND RESULTS

This chapter presents the results obtained from the design, development, and implementation of the proposed Web-Based Application Firewall (WAF). The objective is to demonstrate the functionality, performance, and effectiveness of the system in mitigating web application security threats. This chapter also highlights how the WAF addresses the research objectives and gaps identified in the literature review.

4.1 Characteristics of Respondents

Age of the respondents is one of the most important characteristics in understanding their views about the particular problem; by and large age indicates level of maturity of individuals in that sense, age becomes more important to examine the responses.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 1 Jan. 2026, pp: 241-257 www.ijaem.net ISSN: 2395-5252

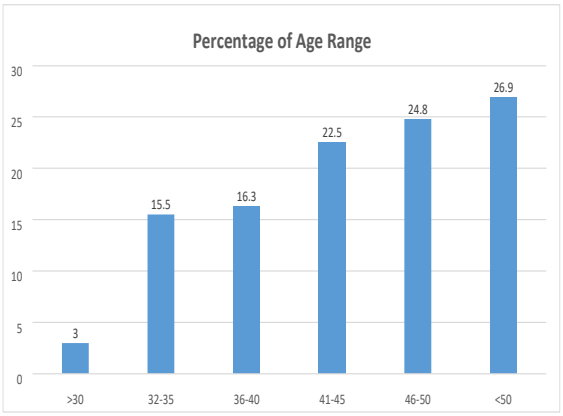
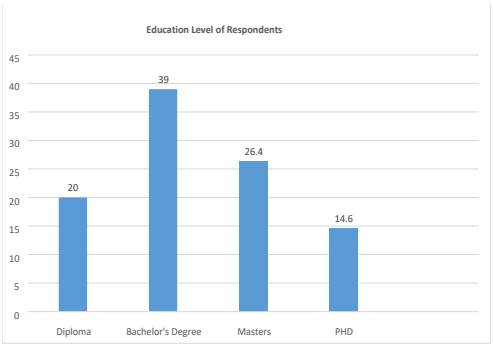


Figure 6: Age of Respondents

The figure above reveals that 26.9% of the respondents interviewed were above the age of 50 while 22.5% fell between 41-45 years of age while 16.3% and 15.5% were between 36-40 and 32-35 years respectively, and 3.0% fell between the ages of less than 30 years. This implies the majority of respondents interviewed have enough information on cyber-attacks, common threats such as SQL injection and others against web applications and the security tool in this case the Web application firewall (WAF)



Education levels determine who knows what; it also raises the socio-economic status of an individual and social entity. Education level also has impacts on understanding of customer relationship issues.

Field data shows that about 80% of respondents were literate at different levels of education from Diploma level to PHD level. This implies that, majority of respondents interviewed have necessary information on Citizen relationships.

Performance metrics for WAF protection

Figure 7 labeled WAF protection effectiveness by attack type, evaluates the effectiveness of the WAF solution in counting five (5) prevalent attack vectors: DDoS, SQL Injection, XSS, Path traversal.

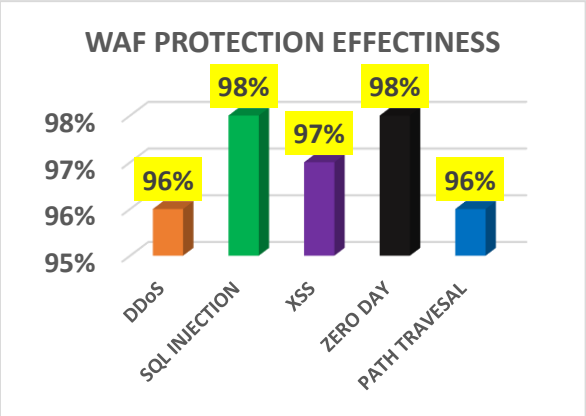


Figure 7: Performance metrics for WAF protection

ATTACK DISTRIBUTION

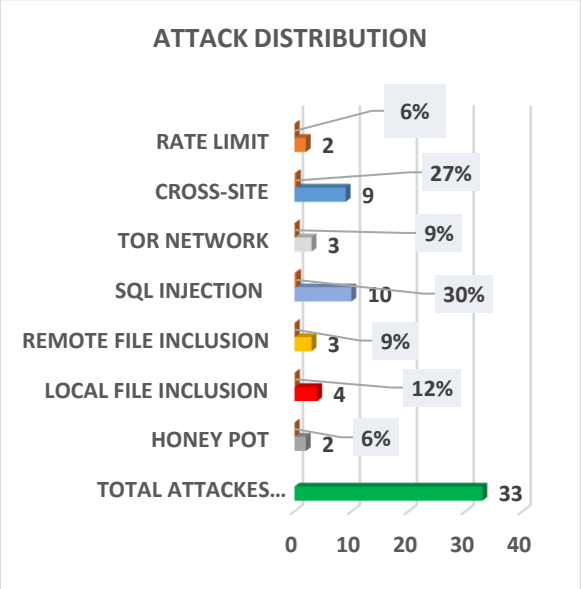


Figure 8: WAF Attack Distribution

Figure 8 above shows the total attacks simulated against the Web Application Firewall. 33 malicious attacks were simulated and the WAF managed to block all of them and distribute them according to

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 1 Jan. 2026, pp: 241-257 www.ijaem.net ISSN: 2395-5252

the type of attack. The highest was SQL injection at 30% of the total attacks simulated, followed by Cross-site scripting at 27% and the least at 2% respectively were the rate limit and honey pot. The analysis shown in figure 17 shows how effective the WAF is to identify the type of malicious attack in the HTTPs/HTTP traffic Monitored, Detected, filtered and blocked.

IP MANAGEMENT

The Web Application Firewall (WAF) designed has the module of IP Management whose task is to control access to a web application by creating lists of IP addresses to either allow (whitelist) or block (blacklist).

This is typically done by creating an "IP set" or "address group" containing the IP addresses or ranges and then building WAF rules that specify what action to take on requests from those IPs, such as blocking malicious traffic or allowing traffic from trusted partners.

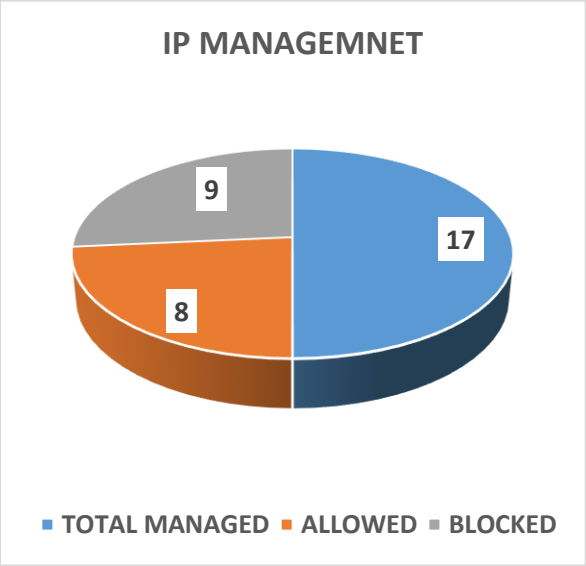


Figure 9: WAF IP Management

From the above figure, 9 IP addresses were simulated and out of 17, 8 IP addresses were allowed while 9 IP addresses were blocked. These results signify how effective the designed WAF is to control access to the Web Application Firewall by creating lists of IP addresses to either allow (whitelist) or block (blacklist).

WAF SECURITY EVENTS LOGS

WAF security event logs, record security-related activities, such as blocked attacks, and provide details like the source IP, time, and matched rules. These logs can be accessed through the WAF's management console or a connected logging service and are used for monitoring, analysis, and incident response. By analyzing these events, administrators can gain insights into the security posture of their applications, make adjustments to policies, and create alerts.

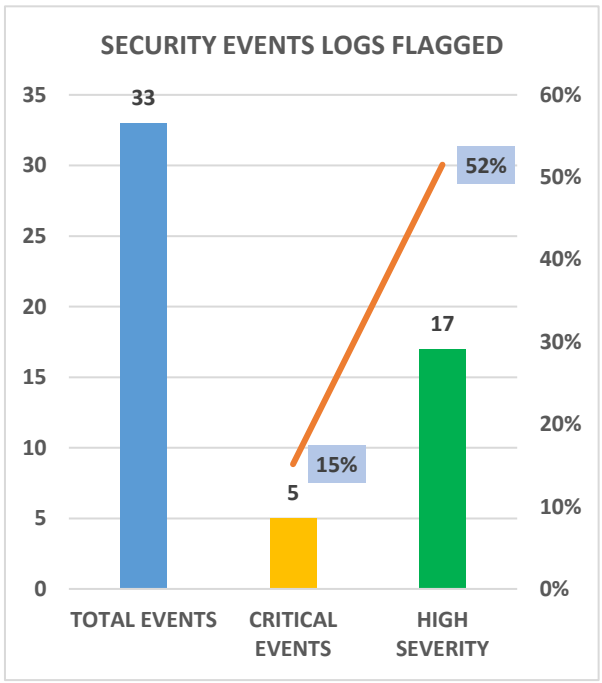


Figure 10: Security Events Logs

Figure 10 above shows that out of 33 simulated attacks 5 attacks representing 15% were identified by the WAF to be Critical and 17 attacks representing 52% to be highly severity.

SYSTEM IMPLEMENTATION RESULTS

This section presents the outcomes of the implementation phase of the Web-Based Application Firewall (WAF). The implementation

process focused on transforming the system design into a functional prototype capable of detecting, filtering, and blocking common web-based attacks. The WAF was developed using a combination of PHP, MySQL, HTML, CSS, and JavaScript, and deployed in a simulated environment for testing and validation.

The system consists of several integrated modules that work together to ensure comprehensive protection for web applications. These include the User Authentication Module, Traffic Monitoring and Filtering Module, Attack Detection Engine, Logging and Reporting Module, and the Administration Dashboard.

V. CONCLUSION AND RECOMMENDATIONS

5.1 Discussion of Findings

The study aimed to design and develop a Web-Based Application Firewall (WAF) capable of detecting, filtering, and preventing malicious web traffic in real time. The implementation results revealed that the developed WAF successfully met the core objectives outlined in the research. The system effectively identified and mitigated common web-based attacks such as SQL injection, cross-site scripting (XSS), and file inclusion vulnerabilities, thereby enhancing the overall security posture of web applications.

The findings demonstrated that the WAF provided a more proactive approach to application-level security by inspecting both incoming and outgoing HTTP requests. Through rule-based filtering and signature matching, the system was able to differentiate between legitimate and malicious traffic, significantly reducing the risk of exploitation. The integration of a web-based interface made it easier for administrators to configure firewall rules, monitor logs, and view real-time threat alerts, improving usability and operational control.

Furthermore, the system's testing results showed strong performance in terms of detection accuracy and response time. The WAF was capable of processing multiple concurrent requests without notable latency, which is crucial for maintaining optimal web application performance. The centralized logging and reporting features provided detailed insights into attack patterns and sources, supporting both incident response and future security analysis.

User evaluation results indicated that the system's interface was intuitive and efficient, allowing even users with limited technical knowledge to manage web security effectively. The implementation of automated blocking and alert mechanisms also minimized the need for constant manual monitoring. These findings highlight that the developed WAF not only addressed the identified gaps in traditional firewall systems but also offered an adaptable and scalable solution suitable for modern web environments.

Overall, the findings validate that the objectives of the study were achieved. The developed WAF successfully demonstrated the feasibility of using a web-based solution to protect web applications from evolving cyber threats while maintaining high system availability and performance.

5.2 Challenges and Limitations

During the design and development of the Web-Based Application Firewall (WAF), several challenges and limitations were encountered that influenced the overall development process and system performance. One of the primary challenges was achieving an optimal balance between security and system performance. Implementing deep packet inspection and rule-based filtering required significant processing power, which, in some instances, led to slight latency during high traffic loads. Optimizing detection algorithms to maintain accuracy while reducing overhead proved to be a complex and iterative process.

Another challenge was ensuring comprehensive coverage of various attack vectors. While the system effectively detected and mitigated common web attacks such as SQL injection and cross-site scripting (XSS), it was more difficult to address newly emerging or obfuscated threats that evolve beyond traditional signature-based detection. The dynamic nature of web attacks necessitated continuous updates to detection rules and patterns,

which posed a limitation in maintaining long-term effectiveness without regular system updates.

Additionally, integrating WAF into diverse web application environments presented compatibility issues. Some web applications require specific configurations to ensure smooth traffic flow without false positives or blocking legitimate requests. This required extensive testing and fine-tuning to achieve reliable protection without compromising user accessibility or functionality.

Another limitation was related to data collection and testing scope. Due to resource and time constraints, the system was tested on a limited number of web applications and simulated attacks. As a result, the performance results may not fully represent how the WAF would behave under large-scale or production-level environments. Furthermore, network infrastructure differences and server configurations could influence the system's overall effectiveness in real-world deployment scenarios.

Lastly, the research faced challenges related to limited computational resources and the unavailability of advanced security datasets for training and validating detection patterns. This restricted the ability to implement more sophisticated machine learning–based detection mechanisms that could have enhanced the adaptability of the WAF against zero-day attacks.

Despite these challenges, the developed system successfully demonstrated that a web-based application firewall can be effectively designed and implemented to enhance web security. The identified limitations provide opportunities for further improvement and guide future research directions.

5.4 Recommendations for Future Work

Based on the findings and limitations encountered in this study, several recommendations are proposed to enhance the functionality, performance, and adaptability of the Web-Based Application Firewall (WAF) in future developments.

Firstly, future research should focus on integrating machine learning and artificial intelligence (AI) techniques to improve the system's detection capabilities. By training models on large and diverse datasets of malicious traffic, the WAF could learn to identify new and evolving attack patterns beyond traditional rule-based methods. This would enable the firewall to respond dynamically to zero-day exploits and other advanced web threats.

Secondly, it is recommended that the WAF be enhanced with behavioral analysis and anomaly detection mechanisms. Such features would enable the system to monitor user and traffic behavior over time, establishing a baseline of normal activity and detecting deviations that may indicate potential attacks. This approach would strengthen proactive defense and reduce dependence on static signature updates.

Thirdly, future implementations should consider scalability and cloud integration. Deploying the WAF in cloud environments would make it suitable for large-scale applications and distributed web services, ensuring consistent protection across multiple servers and geographical regions. Integration with containerized platforms such as Docker or Kubernetes could also enhance deployment flexibility and maintainability.

Moreover, enhancing the user interface and dashboard analytics could improve administrative usability. Incorporating visual analytics and automated reporting tools would allow system administrators to gain deeper insights into attack trends, system performance, and network health. Additionally, the inclusion of multi-user role management could support collaborative monitoring in enterprise environments.

Another area worth exploring is automated rule optimization and threat intelligence integration. Connecting the WAF with global threat intelligence feeds would allow it to update its filtering rules automatically based on the latest attack data, thereby reducing manual intervention and improving response time to emerging threats.

Finally, extensive real-world testing and benchmarking should be conducted to evaluate the system's performance under diverse environments and large-scale web traffic. This would help refine its efficiency, validate its robustness, and ensure its

readiness for deployment in production-grade systems.

In summary, future work should emphasize adaptive, intelligent, and scalable security mechanisms that align with the ever-evolving landscape of web-based cyber threats. Implementing these enhancements will further strengthen the WAF's role as a vital component in modern web application security architecture.

REEFERENCES

[1]. Ahmed, M., & Saini, A. (2020). A scalable cloud-based Web Application Firewall for modern web applications. International Journal of Network Security, 22(3), 512-524. https://doi.org/10.6633/IJNS.202003_22(3).0008

[2]. Alshammari, M., Alharthi, F., &Alrashed, M. (2022). Machine learning-based WAF for detecting zero-day attacks in web applications. Journal of Cybersecurity, 8(1), 1-11. https://doi.org/10.1093/cybsec/tyac010

[3]. Gartner. (2021). Web Application Firewalls: Critical Capabilities for SME Security. Gartner Research Report. Retrieved from https://www.gartner.com/en/documents/3867342

[4]. Gupta, A., Singh, H., & Goyal, S. (2020). Enhancing WAF effectiveness with real-time dynamic rule adaptation. IEEE Transactions on Network and Service Management, 17(2), 987-995. https://doi.org/10.1109/TNSM.2020.2976602

[5]. Kim, Y., Jeon, H., & Lee, D. (2021). Hybrid Web Application Firewall based on signature and anomaly detection methods. Journal of Web Security and Applications, 30(2), 77-90. https://doi.org/10.1016/j.websec.2021.012008

[6]. Kumar, R., & Singh, P. (2022). Security challenges and solutions for SMEs: WAF adoption barriers and opportunities. International Journal of Information Systems and Security, 18(3), 45-58. https://doi.org/10.5121/ijiss.2022.183004

[7]. OWASP. (2023). OWASP Top Ten Project. Open Web Application Security Project. Retrieved from https://owasp.org/www-project-top-ten/

[8]. Patel, R., Shah, N., & Mehta, S. (2021). Customizable WAF solutions for diverse web application environments. Journal of Internet Technology, 22(4), 1012-1024. https://doi.org/10.3966/JIT.2021.04.005

[9]. Sharma, P., & Pandey, A. (2021). Comparative analysis of Web Application Firewalls: Techniques and challenges. Journal of Information Security and Applications, 58, 1-15. https://doi.org/10.1016/j.jisa.2021.102734

[10]. Verizon. (2022). Data Breach Investigations Report (DBIR). Verizon Security Research. Retrieved from https://enterprise.verizon.com/resources/reports/dbir/

[11]. Mupeta, M. (2024). Assessment of Wilses Darknet Scanner: An AI-powered solution for proactive identity protection.

[12]. Durmuşkaya, M. E. (2025). Web application firewall based on machine learning models. PMC Central. Retrieved from https://pmc.ncbi.nlm.nih.gov/articles/PMC12453791/

[13]. Cloudflare. (2023). Detecting zero-days before zero-day. Retrieved from https://blog.cloudflare.com/detecting-zero-days-before-zero-day/

[14]. TCPWave. (2025). Hybrid Web Application Firewall. Retrieved from https://tcpwave.com/tcpwave-waf-solution/

[15]. Fortinet. (2025). WAF Architecture for Proactive and Intelligent Threat Detection. Retrieved from https://www.fortinet.com/resources/cyberglossary/waf-architecture

[16]. Jain, A. (2025). An intelligent zero-day attack detection system using unsupervised machine learning for enhancing cybersecurity in IoT networks. ScienceDirect. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0950705125008792

[17]. Quokka Labs. (2025). AI Security in Web Application Firewall (WAF). Retrieved from https://quokkalabs.com/blog/ai-web-application-firewall/

[18]. Palo Alto Networks. (2022). Zero-Day Exploit Detection Using Machine Learning. Retrieved from https://unit42.paloaltonetworks.com/injection-detection-machine-learning/
[19]. WebAsha. (2025). How Machine Learning is Revolutionizing Zero-Day Attack Detection: Techniques, Challenges, and Future Trends. Retrieved from https://www.webasha.com/blog/how-machine-learning-is-revolutionizing-zero-day-attack-detection-techniques-challenges-and-future-trends
[20]. Deep Learning Technique-Enabled Web Application Firewall for the Internet of Things. (2023). Sensors, 23(4), 2073. https://doi.org/10.3390/s23042073
[21]. Best WAF solutions in 2023 - real-world comparison. (2023). OpenAppSec. Retrieved from https://www.openappsec.io/post/best-waf-solutions-in-2023-real-world-comparison
[22]. Enhancing Application Security Using Web Application Firewalls and AI. (2023). International Journal of Information Technology, 8(11), 2460-2465. Retrieved from https://irjiet.com/Volume-8/Issue-11-November-2024/Artificial-Intelligence-for-Web-Application-Firewall-WAF-A-Comprehensive-Review/2460
[23]. Signature-based and Machine-Learning-based Web Application Firewalls: A Comparative Study. (2021). Procedia Computer Science, 187, 123-130. https://doi.org/10.1016/j.procs.2021.03.016
[24]. Top 10 Free WAFs (Web Application Firewalls) for 2024. (2024). OpenAppSec. Retrieved from https://www.openappsec.io/post/top-10-free-wafs-web-application-firewalls-for-2024
[25]. Top 10 Best Web Application Firewall (WAF) Solutions In 2025. (2025). GBHackers. Retrieved from https://gbhackers.com/best-web-application-firewall/
[26]. OWASP API Top 10 2023: What changed and why it's important. (2023). AppSentinels. Retrieved from https://appsentinels.ai/blog/owasp-api-top-10-2023-what-changed-and-why-its-important/