Enhancing Web Application Security Using Machine Learning Based WAFs

Ms. Kusumakumari Daram[1*]
Research Scholar
Department of Computer Science and Engineering B.E.S.T. Innovation University, Andhra Pradesh.

Dr.P.Senthilkumar[2], Professor,
Department of Computer Science and Engineering
Shadan Women`s College of Engineering and Technology Khairatabad, Hyderabad.
psenthilkumarshadan@gmail.com,
psenthilexcel@gmail.com.

Abstract— Web applications face an ever changing world of cyber threats, thus strong security methods are critical for protecting sensitive data and guaranteeing service continuity. Traditional Web Application Firewalls (WAFs), which rely on static rule sets and signature-based detection, frequently fail to recognize novel or sophisticated attacks and generate a significant number of false positives. Recent advances in machine learning have revolutionized WAF capabilities, allowing for adaptive, intelligent threat detection and response. ML-based WAFs analyze massive quantities of web traffic in real time, using statistical models and neural networks to distinguish legitimate and malicious requests with high accuracy. These technologies can reduce false positives by up to 90% and achieve detection rates of over 95% for complex threats such as SQL injection, cross-site scripting (XSS),the model quickly finds rules using a rule service-based method, and then resolves conflicts using an action constraint technique. The rule merging procedure is then applied to a set of rules that contain no service-related anomalies.

Keywords- Machine Learning,Web Application Firewall, SQL Injection,Cross-site Scripting and rule service.

I. INTRODUCTION

In today's digitally connected world, web apps are the foundation of modern company operations, providing services ranging from e-commerce and banking to healthcare and social media[1] However, as people's reliance on web-based systems has grown, so has their vulnerability to cybercrime. SQL Injection, Cross-Site Scripting (XSS), and Distributed Denial-of-Service (DDoS) assaults are becoming more widespread, causing data breaches, service interruptions, and significant financial losses.To reduce these dangers, Web Application Firewalls (WAFs) have emerged as an important line of protection. Traditional WAFs use predefined rules or signature-based detection to identify and block malicious HTTP traffic. While effective in some cases, these systems frequently struggle to react to quickly changing attack patterns and zero-day vulnerabilities.Their static nature frequently results in high false-positive rates and limited effectiveness against sophisticated or obfuscated threats.

Recent advances in Artificial Intelligence (AI) and Machine Learning (ML) have opened new horizons in the field of cybersecurity, particularly in web application protection[2]. Machine

Learning-Based Web Application Firewalls represent a significant evolution in threat detection and response. E-commerce is increasingly being used by businesses to boost revenue, as web apps become more prevalent [3]. Web applications are prone to errors, making them a lucrative target for attackers. The frequency of security incidents involving online applications is steadily growing [4]. To address the inadequacies of traditional network firewalls, a variety of countermeasures exist, including Web Application Firewalls (WAFs), which have recenrequest tly been added to network architecture. WAFs can mitigate boken access control vulnerabilities, such as those that lead to forced browsing, by maintaining a rigorous flow. Using WAFs for tight request flow enforcement can be problematic due to their poor relationship between settings and application implementation. This approach protects applications from generic attacks without addressing specific problems within the program. Enforcing a WAF policy on incoming requests does not necessarily guard against application-specific implementation flaws.By leveraging data-driven algorithms, these intelligent WAFs can identify complex patterns of malicious behavior, learn from new attack vectors, and adapt in real-time to defend against both known and unknown threats.the integration of machine learning techniques in WAF architectures, studying how they boost detection accuracy, minimize operational overhead, and provide dynamic security solutions for modern web environments.

II. LITERATURE SURVEY

Web Application Firewalls (WAFs) are security systems that monitor, filter, and analyze HTTP traffic sent between a client and a web server. Traditional WAFs are typically rule-based or signature-based systems that detect and block harmful inputs by matching predefined patterns of known attack signatures, regular expressions, and protocol anomalies. These firewalls protect against well-documented attacks including SQL Injection, Cross-Site Scripting (XSS), and Remote Code Execution, which are among the OWASP Top 10. When a request matches a rule connected with a known attack, the WAF blocks it or alerts the system administrator.Anomaly detection uses models of expected user and application behavior to identify suspicious activities. This approach complements abuse detection, which involves matching attack specifications against audited events to detect modeled attacks [5]. One fundamental assumption behind anomaly detection is that attack patterns differ from regular behavior. Furthermore, anomaly detection presupposes that this 'difference' is quantifiable. Many strategies for analyzing various data streams have been proposed on the factor of these assumptions, exclude data mining for system aggregation, statistical investigation for audit files, and chronological sequence analysis for operational system calls. Web apps are server-side applications accessed using capillary web users. over the HyperText Transport Protocol (HTTP)[6]. A web application allows users to

navigate with a click links or URLs in their web browser and provide input parameters using online forms. A URL refers to a server-based application that executes using the input given by users parameters. Therefore output of the program, oftenly in HTML, is returned to the browser for interaction of the user. Since HTTP is a unsettled, application-level request/response protocol that has been used on the World Wide Web from 1990 [7]. As protocol is unsettled, meaning each request is refined individually with no cognition of prior ones. To support user sessions in web applications, session management must be added to the stateless HTTP layer. Web requests can be embedded in a user session using cookies, URL rewriting, or hidden form fields [8]. The Common online applications nowadays rely on an implicit in model or technology for evolution and deployment. Many popular technologies, including JSP/Servlets,PHP and ASP.NET supports managing user sessions.

I. Servlet-based web applications

The J2EE specification [9] includes Java Servlet technology, which provides techniques for extending web server capability and admittance existing business concern systems [10]. The JAVA servlet container's basic functionality is to handle incoming web requests and process them using servlets. A instrumentation converts incoming HTTP pursuit into object-oriented form and validates to see if a servlet is registered to handle the request. If there is a match, the request is processed by the corresponding servlet. Filters are used to process data in requests and answers. Filters include access control, output transformations (e.g., XSLT), logging, and auditing. Servlets and electrical device are stateless components that handle individualist pursuance. Servlets can store and retrieve session-specific data from a shared repository (HttpSession).

II. Web application firewalls and web flaws.

Web applications are vulnerable to assaults due to insufficient network security [11]. Networking firewall, like stateful packets filters, provide access to online applications by permitting TCP port no.80 traffic; nonetheless, web applications helps frequently exploited at the application layer. Onslaught use architectural flaws in application logic, as well as vulnerabilities in the HTTP protocol, browser, and web server technology. A network firewall only restricts access to a web server, independent of the type of request or data sent to it. The Open Web Applications Security Project (OWASP) identified the ten most serious web application vulnerabilities in their OWASP Top 10 . This study focuses on broken access control vulnerabilities, notably those that enable coercive surfing [12]. Forceful surfing is the habit of accessing web pages (URLs) without consideration for their context while using an application session. By ignoring the planned application flow, you risk getting unauthorized access to resources or encountering unexpected behavior [13]. WAFs are frequently used inline between the browser and the server (see figure 1)

to provide real-time access control based on application-level data including the URL, credentials, input parameters, and user session history. A WAF's access decisions might be based on either a positive or negative security paradigm. This criterion describes a technique in which a WAF monitors individual user sessions and maintains track of both previously visited URLs and those that can be accessed at any moment [14].

III. Detection Models of Anomaly

The anomaly perception conceptualization looks at HTTP pursuance logged by common web servers such as Apache [15]. The study focuses on GET requests with parameters for transferring data to server-side programs or active documents. GET and POST/HEAD queries do not contain header information. A variety of models are used to identify anomalous things in a set of input requests (Ur) for a program.. A model assesses a query attribute's string length or its overall feature, such as the presence or absence of a specific attribute. A profile is used to match each model to a program's quality or attributes.A model distributes probabilities to queries and properties. This probability number represents the possibility of a specific feature value happening inside a given profile. The notion is that feature values with low likelihood (abnormal values) indicate a potential attack.he model outputs (query likelihood and properties) determine whether the query is presented as a potential attack or normal. The choice is made by calculating anomaly scores for each query attribute and the query as a whole. If one or more anomaly scores (for the query or one of its attributes) surpass the detection threshold set during the training phase (see below), the entire query is considered anomalous. This prevents attackers from hiding a single harmful attribute in a query with multiple 'regular' properties. Anomaly ratings for queries and characteristics are based on the likelihood values returned by linked models. Equation 1 shows the weighted sum used to calculate the anomaly score value. In this equation, wm represents model m's weight, and pm is its probability value.

$$\text{Anomaly Scores} = \text{Am}\in\text{Model vm} * (1 - \text{rm}) \quad (1)$$

A framework can perform in any two ways: by training or sensing. The preparation process finds normal event attributes and develops criteria for discriminating between regular and aberrant data. This phase is separated into two components. In the first stage, the system generates profiles for each server-side software and its associated attributes[16]. In the second stage, acceptable thresholds are determined. This requires analyzing queries and their attributes against the profiles created in the previous stage. The maximum anomaly score for each program and its attributes is saved. The threshold is then set to a percentage greater than the maximum. The default proportion (in our experiments) is 10%. Modifying this number allows users to strike a balance between false positives and detection accuracy. A configurable parameter controls the length of the training phase, which includes the number of queries and characteristics used to

construct profiles and thresholds. Once the profiles have been established, which means that the models have learned the characteristics of typical occurrences and appropriate thresholds have been calculated.

IV. AI-Based WAFs

The rise of complex and automated attacks, there has been a significant shift toward integrating artificial intelligence into cybersecurity tools, including WAFs. Machine learning and AI offer the ability to detect novel patterns, analyze massive volumes of traffic in real-time, and make decisions without relying solely on predefined rules[17]. AI-based WAFs employ a variety of machine learning techniques to classify traffic, detect anomalies, and continuously learn from new data

These systems are capable of:

Recognizing attack patterns without explicit programming.

Learning user behavior to identify abnormal requests.

Adapting to new threats through continuous training.

Reducing false positives by understanding context and intent.

By incorporating AI, WAFs move from being static gatekeepers to dynamic security agents capable of evolving alongside the threats they are designed to mitigate.

III. Machine Learning Techniques in Web Application Firewalls

Machine learning (ML) is the foundation of modern intelligent WAFs. It helps computers to analyze massive amounts of data, identify trends in traffic behavior, and make autonomous judgments about which requests to grant or deny. Several ML approaches are routinely used in WAF systems, with each providing unique capabilities for identifying and combating web-based assaults.

3.1 Supervised Learning

Supervised learning is a model on a labeled dataset, with each input (e.g., HTTP request data) classified as benignant or malicious. The model learns to correlate input feature like URL patterns, HTTP methods, or parameter values with their labels. The most Common algorithms are Decision Trees, Support Vector Machines (SVM), Random Forests, and Naive Bayes. They are used in Detecting known attacks like SQL Injection or Cross-Site Scripting based on historical data. Its Advantages is High accuracy for known attacks; easy to evaluate and interpret. Its Limitations are requires a large volume of labeled data and struggles with novel or zero-day attacks. For example a supervised model trained on thousands of past attack signatures can accurately classify incoming requests and flag ones resembling known attack vectors.

3.2 Unsupervised Learning

The Unsupervised learning does not require labeled data. Instead, it focuses on identifying anomalies or deviations from normal behavior by analyzing patterns and clustering similar types of traffic. The Common algorithms are K-Means

Clustering, Principal Component Analysis (PCA), and Isolation Forests. Its used in detecting unknown or zero-day attacks by identifying unusual traffic behavior. Its advantages are Effective against novel threats; requires no labeled data. The disadvantages are to produce higher false positives due to ambiguous patterns. For example consider a WAF might learn typical traffic patterns for a login endpoint and flag a sudden spike in failed login attempts from unfamiliar IP ranges.

3.3 The Deep Learning

Deep learning is considered as a subpart of machine learning that uses neural networks with multiple layers to model complex, non-linear relationships in data. It's particularly useful in analyzing unstructured data such as raw HTTP request payloads, headers, or behavioral sequences. The Common models are Long Short-Term Memory (LSTM) networks, Recurrent Neural Networks (RNNs),Convolutional Neural Networks (CNNs). These are used for Detecting advanced threats embedded in request payloads, such as obfuscated scripts or multi-step attacks. The benefits are it can automatically extract features and recognize intricate patterns. Drawbacks are it requires substantial computing power and large datasets for training; less interpretable. Consider the example An LSTM model can analyze sequences of user interactions to detect slow, staged attacks attempting to bypass traditional detection methods.

3.4 Reinforcement Learning

Reinforcement learning (RL) refers to teaching an agent to make decisions depending on feedback from its surroundings. In the WAF context, an RL agent can be trained to respond to attacks dynamically by adjusting filters and rules over time based on past outcomes.The case study of Adaptive firewall rule optimization and automatic response to new attack patterns. Its advantages is a continuous learning through interaction; ability to evolve without retraining from scratch. Its drawbacks are More complex to implement; may require careful control to avoid unintended consequences. An example on a WAF using RL can learn to delay or block traffic from a suspicious IP range after observing failed login attempts, then adjust its strategy based on whether the actions reduced the attack surface.

IV. ARCHITECTURE

The segregation of artificial intelligence along with the machine learning into Web Application Firewalls (WAFs) has significant Numerous studies and real-world execution demonstrate the effectiveness of AI-driven WAFs in enhancing web application security.

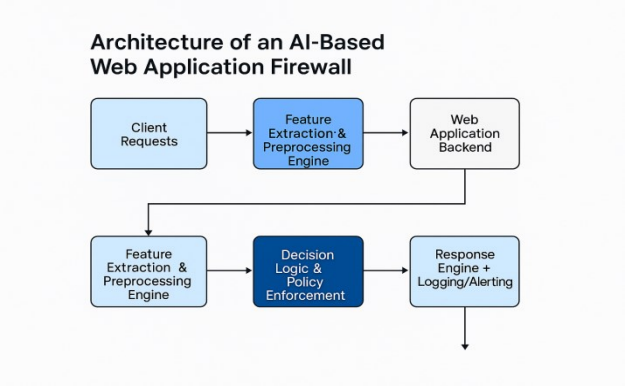

Figure 1 Architecture of an AI-Based Web Application Firewall

In Real-world deployments of AI-powered WAFs offer valuable insight like Cloudflare's AI-enhanced WAF uses pattern recognition, anomaly detection, and threat intelligence feeds to protect millions of sites. Their adaptive engine automatically learns from global traffic trends, enabling real-time protection against new attack vectors. According to Cloudflare, the system has reduced false positive rates by over 50% while improving detection accuracy for zero-day threats simlarly it uses AWS WAF integrates machine learning models via AWS Shield and AWS Firewall Manager[18]. It uses AI to detect SQL injection and XSS attacks, offering customizable protection rules. AWS customers report streamlined management and proactive mitigation of targeted application-layer attacks. The research findings reinforce the viability and effectiveness of AI-based WAFs in both academic and enterprise environments. Their ability to adapt, learn, and scale makes them indispensable tools for modern cybersecurity.

The following figure shows the architecure layers of the WAF in AI.

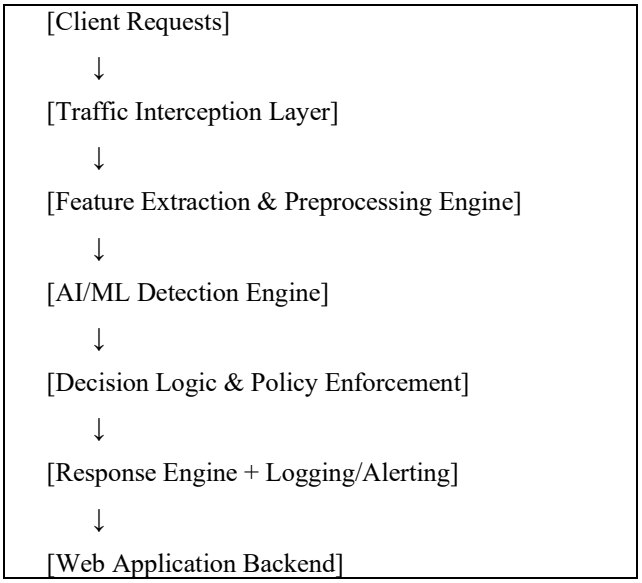

Figure 2. Layers used for WAFs

1. Traffic Interception Layer

Reverse proxy or inline gateway (e.g., NGINX, Envoy, or API Gateway). Captures all HTTP/S requests to and from the application it acts as the first entry point, routing and mirroring requests for analysis.

2. Feature Extraction & Preprocessing Engine

Parses request metadata: IP, headers, query strings, body, cookies, etc. Converts raw HTTP traffic into structured data features (e.g., tokenization, encoding frequency, payload size).It Prepares data suitable for machine learning models.

3. AI/ML Detection Engine

Houses supervised, unsupervised, or deep learning models (e.g., Random Forests, LSTM, CNN) are used for detection of the WAF.It follows certain tasks which is as below.

Tasks:

i. **Anomaly detection**
ii. **Behavior profiling**
iii. **Bot detection**

SQLi/XSS classification

May include multiple models (model ensemble) and also Predicts whether traffic is normal, suspicious, or malicious. In section 4.1 to 4.6 is shows the various threats which are AI based WAF in the Cloud.

4. Decision Logic & Policy Enforcement

Applies security policies based on ML outcomes (e.g., confidence thresholds, user behavior context)Integrates with allow/block/flag rule engine.it helps in determines actions like allow, challenge (e.g., CAPTCHA), rate-limit, or block.

5. Response Engine + Logging & Alerting

Executes appropriate response: allow/block/challenge.Sends alerts to SIEM/SOC systems.Logs flagged requests for retraining and auditing. It Provides real-time responses and feedback loops for continuous learning. Apart from these layers we can also Keep the AI system adaptive and up to date by continuos learning module.

4.1 AI- based WAFs

An AI-powered Web Application Firewalls introduce a range of capabilities that enhance the overall security posture of web utilize. Unlike conventional rule-based systems, AI-based WAFs offer dynamic, context-aware defense mechanisms that evolve alongside emerging threats. Below are the key features that distinguish them from conventional solutions. AI-based WAFs excel in detecting anomalies by establishing a baseline of normal traffic behavior[19]. Using machine learning algorithms, these systems continuously monitor and compare incoming HTTP requests against learned traffic profiles.Example: If a particular user typically logs in once a day from a single location, a sudden series of requests from multiple geographic regions in a short span may be flagged as anomalous. Benefit: Early detection of suspicious activities like credential stuffing, brute-force attacks, or zero-day exploits.

4.2 Behavioral Profiling

Rather than only examining individual requests, AI-enabled WAFs analyze user behavior over time. This profiling helps distinguish between legitimate users, bots, and potential attackers.Example: Repeated access to hidden or non-public URLs, frequent changes in user-agent strings, or automated form submissions may indicate bot activity or reconnaissance.Benefit: Reduces false positives by understanding intent, not just content.

4.3 Continuous Learning and Adaptability

One of the major strengths of AI-based WAFs is their ability to learn from new data. These systems can retrain themselves periodically, refining their understanding of traffic patterns and emerging threats.Example: A WAF may initially flag a new API usage pattern as suspicious but later learn it is a legitimate update from the development team. Benefit: Improves long-term detection accuracy while minimizing the need for manual rule updates.

4.4 Real-Time Threat Detection

AI-based WAFs can process and analyze data in real time, enabling immediate detection and response to attacks as they happen.Example:

During a DDoS attack, the system can recognize the spike in traffic volume and identify malicious sources within seconds, applying rate-limiting or blocking rules on the fly[20].Benefit: Reduces response time and mitigates damage before attacks escalate.

4.5 Zero-Day Attack Mitigation

Traditional WAFs are vulnerable to zero-day attacks because they rely on known signatures. AI-based WAFs, however, identify suspicious behavior even when no prior knowledge of the vulnerability exists.Example: An AI model may flag an unusually structured query or encoded payload that deviates from normal patterns, even if the specific vulnerability is unknown.Benefit: Proactive defense against emerging threats without waiting for security patches or rule updates.

4.6 Integration with Threat Intelligence

Some AI-WAFs integrate external threat intelligence feeds, enabling them to correlate observed traffic with known malicious IP addresses, domains, or behavioral indicators. Example: If a request comes from an IP address recently flagged in threat intelligence reports for botnet activity, the WAF can preemptively block or challenge it. Benefit: Enhances contextual decision-making and boosts proactive security measures.

V. IMPLEMENTATION

The research utilizes a mixed-method methodology, including numerical and descriptive informationgathering methods. The research team will use generated traffic and application logs from real-world usage to obtain quantitative measurements. In contrast, cybersecurity expert interviews will serve as the primary source of qualitative information. Statistical analysis with machine learning algorithms helps in processing data to be evaluated for the determining the integrated system's performance level for web application[21].

TABLE: COMPARISON OF TRADITIONAL WAFS AND AI-ENHANCED WAFS

Feature	Traditional WAFs	AI-Enhanced WAFs
Detection Method	Signatures-based	Anomalousness based and behavioral
Ability	Down	Full
Mendacious Positive Charge per unit	Advanced	Lowest
Response Time	Static	Dnamic adjustments
Learning Capability	None	Continuous learning
Threat Intelligence	Limited	Enhanced through data analysis

The WAF received the integration of the trained and validated AI model, which became available for use. By integrating with the WAF, the system gained real-time threat detection abilities because the WAF used AI-driven traffic analysis insights in its operations. The WAF received the training through a configuration point that allowed it to use the AI model for:

1. Using the AI model, the system operated in real time to check traffic patterns; therefore, it could detect irregular behavior, which functioned as indicators of security threats.

2. The WAF automatically adjusted its filtering rules based on the AI-generated recommendations, enabling quick responses to emerging threats.

3. The ability of the Web Application Firewall and Reverse Proxy system modules to protect and optimize websites hosted on the backend web server from hacker attacks will be evaluated. Web cloud (Nextcloud) will be used as a testing tool for websites in each system module that is constructed.

VI. RESULTS

The Tests are divided into two phases, as follows:
The 1st test involves accessing the backend website through an IP reverses proxy.
The 2nd test measures the speeds of accessing web server services, including petitions, transfer, and connections times[22].

1. Backend Web Servers accessing testing

The backend web servers accessing process with the help of IP Address 192.168.41.135 via the Reverses Proxys of IP Address 192.168.41.139. Since the reverses proxys servers only forwards to the backend web servers, the web pages from the backend servers will be shown. The backend website is displayed on the webpage when a user or client visits the IP Reverses Proxys. The path in the URL has to point to 192.168.41.139/nextcloud in order to launch the Nextcloud application. The backend server is where Nextcloud files are stored, and an IP reverse proxy can be used for all storaging, business relationship creation, information upload, and file transfering done[23].

2. Web Servers servicing to testing for accessing

By Applying the benchmark Apache tools, which are run on apache web servers and nginx in order to evaluate them with and without a reverse proxy, accessing to web servers services is examined. Make use of the following settings:

Message: 10000 Concurrent: 1000-10000 #xy –z 1000 –n 10000 http://192.168.41.139/nextcloud (web server nginxs) #xy –z 1000 –n 10000 http://192.168.41.154/nextcloud (web server apaches) A comparing of the instance necessary to use a reverses proxys and will not use a reverse proxy is shown. in Table II and III.

Table II TIME REVERSE REQUEST PROXY

Concurrency	With Reverses Proxys (ms)	Without Reverses Proxys (ms)
100	139.4	62.5
200	286.2	87.2
300	390.3	276.3
400	579.4	381.8
500	659.6	872.7
600	953.7	587.8
700	897.9	684.7
800	1168.2	772.2
900	1224.8	854.2
1000	1542.7	980.2

Average Time per Request	768.76	565.45

Table III TIME REVEERSE TRANSFER PROXYS

Concurrency	Reverses Proxys (Kbyte/sec)	Without Reverses Proxys (Kbyte/sec)
100	438.5	910.35
200	410.05	1061.25
300	445.85	547.52
400	405.67	523.23
500	447.51	282.85
600	376.58	518.25
700	436.48	496.75
800	385.65	492.68
900	459.39	502.37
1000	395.09	494.62
Average Time per Request	771.45	550.28

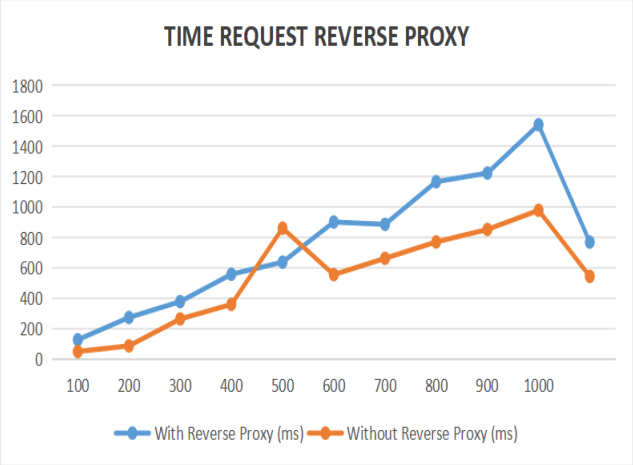

Figure.2 Time Request reverse proxy for the WAFs

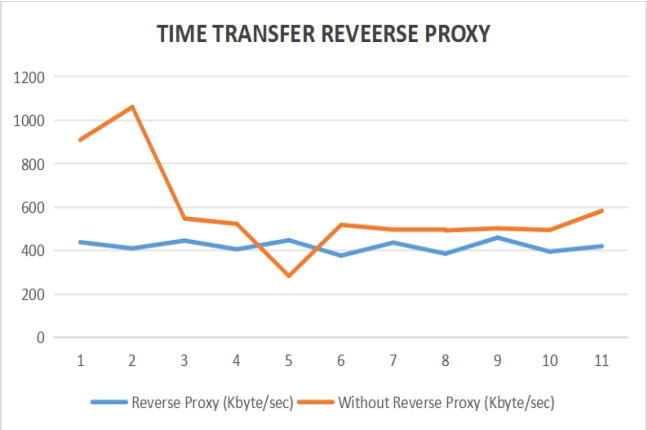

Figure.3 Time Transfer reverse proxy for the WAFs

In figure 2 and 3 shows the time taken for the request for the reverse proxy WAF and time transfer reverse proxy for the WAF in real time respectively. The time value of a petition for accessing to the website (nextcloud) with and without reverses proxys. The transfer time value for accessing the website (nextcloud) with and without Reverses Proxys.

VII. CONCLUSION

The Web Applications Firewalls teamed up with Artificial Intelligence create an effective solution for application security enhancement in the current dynamic threat environment. Current research shows that uniting WAF's traditional technology with artificial intelligence capabilities makes it an excellent system for detecting security threats effectively. Organizations can modify security systems that prevent current and new attack methods in real time by implementing machine learning algorithms.The research findings prove that AI-enhanced WAFs succeed in detecting anomalies while decreasing false positive results, which makes security operations more efficient.

Through their integration, security teams can achieve reduced operational stress, which enables them to dedicate resources to establishing additional organizational defenses.Supported on the findings of research analysis and testing, the application of Reverse Proxy as a web server optimization was successfully carried out by accessing the Reverse IP Proxy and testing the web server access services such as request time, transfer time, and connection time on the Reverse Proxy. The petition of Web Application Firewall as a security for a web server using Mod Safety was successfully proved by administration Incursion testing. WAFs, which use both static and dynamical substantiation, can explicitly ensure the deficiency of definite types of inaccurate behavior in online utilizations. In particular, we guaranteed that if the aggregation of a web application and a WAF policy passes our confirmation procedure, no client/server fundamental physical phenomenon will disrupt the data dependence on mutual sessions state across server-side constituents. These technologies can reduce false positives by up to 90% and achieve detection rates of over 95% for complex threats like as SQL injection, cross-site scripting (XSS),the model quickly finds rules using a rule service-based method, and then resolves conflicts using an action constraint technique. The rule merging procedure is then applied to a set of rules that contain no service-related anomalies.

REFERENCES

[1] E. Armstrong, J. Ball, S. Bodoff, D. B. Carson, I. Evans, D. Green, K. Haase, and E. Jendrock. The J2EE 1.4 Tutorial. Sun Microsystems, Inc., December 2005.

[2] I. Bar-Gad. Web application firewalls protect data. http://www.networkworld.com/news/tech/2002/0603tech.html, March 2005.

[3] Barnett, K. R. M. Leino, and W. Schulte. The Spec# Programming System: An Overview. Lecture Notes in Computer Science, 3362, 2004.

[4] S. W. Boyd and A. D. Keromytis. Sqlrand: Preventing sql injection attacks. In ACNS, pages 292–302, 2004.

[5] L. Burdy, Y. Cheon, D. Cok, M. Ernst, J. Kiniry, G. T. Leavens, K. R. M. Leino, and E. Poll. An overview of JML tools and applications. International Journal on Software Tools for Technology Transfer (STTT), 7(3):212–232, June 2005.

[6] D. R. Cok. ESC/Java2 Implementation Notes.http: //secure.ucd.ie/products/opensource/ESCJava2/ESCTools/docs/Escjava2-ImplementationNotes/Escjava2-ImplementationNotes.pdf.

[7] W. A. S. Consortium. The Web Hacking Incidents Database. http://www.webappsec.org/projects/whid/

[8] L. Desmet, F. Piessens, W. Joosen, and P. Verbaeten. Static Verification of Indirect Data Sharing in Loosely-coupled Component Systems. In Software Composition, volume 4089 of Lecture Notes in Computer Science, pages 34–49. Springer Berlin / Heidelberg, 2006.

[9] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. http://www.ietf.org/rfc/rfc2616.txt, 1999. Request For Comments: 2616 (Category: Standards Track).

[10] K. Golnabi, R. K. Min, L. Khan, and E. Al-Shaer. Analysis of Firewall Policy Rules Using

Data Mining Techniques. In 10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006), April 2006.

[11] V. Haldar, D. Chandra, and M. Franz. Dynamic taint propagation for java. acsac, 0:303–311, 2005.

[12] W. G. J. Halfond and A. Orso. Amnesia: analysis and monitoring for neutralizing sql-injection attacks. In ASE '05: Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering, pages 174–183, New York, NY, USA, 2005. ACM Press.

[13] Y.-W. Huang, F. Yu, C. Hang, C.-H. Tsai, D.-T. Lee, and S.-Y. Kuo. Securing web application code by static analysis and runtime protection. In WWW '04: Proceedings of the 13th international conference on World Wide Web, pages 40–52, New York, NY, USA, 2004. ACM Press.

[14] J2EE platform specification. http://java.sun.com/j2ee/.

[15] B. Jacobs, K. R. M. Leino, F. Piessens, and W. Schulte. Safe concurrency for aggregate objects with invariants. In Proceedings of the Third IEEE International Conference on Software Engineering and Formal Methods, pages 137–146. IEEE Computer Society, 2005.

[16] Karl Forster, Lockstep Systems, Inc. Why Firewalls Fail to Protect Web Sites. http://www.lockstep.
com/products/webagain/why-firewalls-fail.pdf.

[17] KindSoftware. The Extended Static Checker for Java version 2 (ESC/Java2). http: //secure.ucd.ie/products/opensource/ESCJava2/.

[18] G. T. Leavens. The Java Modeling Language (JML). http://www.jmlspecs.org/.

[19] K. R. M. Leino, G. Nelson, and J. B. Saxe. ESC/Java User's Manual.

[20] National Institute of Standards and Technology (NIST). National vulnerability database. http://nvd.nist.gov/statistics.cfm.

[21] A. Nguyen-Tuong, S. Guarnieri, D. Greene, J. Shirley, and D. Evans. Automatically hardening web applications using precise tainting. In SEC, pages 295–308, 2005

[22] J. Offutt, Y. Wu, X. Du, and H. Huang. Bypass testing of web applications. In ISSRE, pages 187–197, 2004.

[23] Open Web Application Security Project (OWASP). Top ten most critical web application vulnerabilities.
http://www.owasp.org/documentation/topten.html,2005.