DESIGN AND EVALUATION OF A HYBRID WEB APPLICATION FIREWALL FOR MODERN WEB THREATS

Pankaj Kumar Singh
Cyber Ssecurity Analyst
University of Mumbai

Abstract: Web applications are now used in almost every important digital activity, from online banking and shopping to education, healthcare, and government services. Because these applications handle sensitive information and support critical operations, they have become attractive targets for attackers. Many attacks today are designed to exploit weaknesses in the way web applications accept, process, and respond to user input. Common threats such as SQL injection, cross-site scripting, cross-site request forgery, malicious bot activity, and response-based information leakage continue to cause serious security problems. Traditional defenses are often not enough to stop these attacks because they usually depend on fixed rules or simple filtering techniques.

This research proposes a hybrid Web Application Firewall (WAF) that works as a reverse proxy in front of the backend application. The firewall inspects both incoming requests and outgoing responses so that harmful traffic can be blocked before it reaches the server and sensitive information can also be removed before it reaches the user. The proposed system combines signature-based detection, anomaly-based analysis, response sanitization, centralized logging, and feedback-driven policy updates. This makes the system more flexible than a static firewall because it can respond to both known attacks and unusual behavior that may indicate a new threat.

The main idea of this paper is to show that a WAF should not only block attacks, but also learn from them. By collecting logs, analyzing suspicious traffic, and improving rules over time, the firewall becomes more adaptive and more useful in real-world environments. The research also evaluates the system in terms of security effectiveness, false positive rate, response latency, and monitoring usefulness. This makes the project suitable as a research paper because it combines design, implementation, and evaluation in one complete study.

Keywords: Web Application Firewall, Reverse Proxy, Hybrid Detection, Signature-Based Detection, Anomaly Detection, SQL Injection, Cross-Site Scripting

INTRODUCTION

The internet has changed the way people use services, but it has also changed the way attackers work. In earlier years, attacks were often limited to network-level threats such as port scanning or direct server abuse. Today, most of the damage happens at the application layer, where attackers try to exploit flaws in forms, search fields, login pages, file uploads, and API endpoints. Since web applications interact directly with users and databases, even a small weakness can have serious consequences. This is why application-layer protection has become a major concern in cybersecurity.

A Web Application Firewall is one of the most important tools used to protect web applications. It sits between the user and the application and examines HTTP or HTTPS traffic before allowing it to proceed. Unlike a traditional network firewall, which mainly checks IP addresses, ports, and protocol rules, a WAF understands the structure and content of web requests. It can inspect URLs, parameters, headers, cookies, and even response data. This allows it to detect attacks that might look normal at the network level but are dangerous at the application level.

The project you are working on uses a reverse proxy model, which is a very practical and effective way to deploy a WAF. In this model, all traffic first goes to the WAF and then is forwarded to the backend server only if it is considered safe. This setup gives the firewall a central position in the communication flow, making it easier to inspect traffic, hide backend server details, and enforce security policies consistently. It also helps in monitoring and logging because every request passes through the same checkpoint.

What makes this project especially suitable for research is that it does more than just block suspicious traffic. It also includes anomaly detection, response filtering, logging, dashboard-based monitoring, and feedback-driven improvement. That means the system is not static. It can evolve based on what it learns from traffic patterns and attack behavior. This is the kind of feature that gives a project academic depth and makes it interesting as a research contribution.

Need of the Study

The need for this study comes from the fact that web attacks are becoming more advanced and harder to detect. A simple rule-based firewall can stop obvious attacks, but attackers do not always use obvious methods. They may encode payloads, break

malicious input into smaller parts, hide their intent inside normal-looking requests, or use automated tools to generate large numbers of variations. Because of this, a firewall that depends only on static signatures may fail to identify many dangerous requests. This creates a clear need for a more intelligent defense system.

Another reason for this study is the increasing complexity of modern web applications. Today's applications are not limited to simple pages; they include login systems, APIs, dashboards, file upload features, chat systems, and dynamic interfaces. Each of these components creates additional opportunities for attackers. A basic firewall may not understand the difference between normal application behavior and suspicious application behavior. Therefore, a WAF must be able to examine traffic in context and adapt to different request patterns.

This study is also needed because response security is often ignored. Many security systems focus only on incoming requests and forget that outgoing responses can also leak valuable information. For example, error messages may reveal the structure of a database, internal file paths, or details about the server environment. Attackers often use such information to plan a more targeted attack. A good WAF should therefore inspect responses too, not only requests. This project includes response sanitization for exactly that reason.

Finally, there is a practical need for affordable and manageable security solutions. Many commercial WAF products are expensive or difficult to configure, especially for educational projects, small organizations, or research environments. A reverse proxy WAF built with open-source tools can provide a strong learning and implementation model while keeping costs low. That makes this study useful both academically and practically.

Problem Statement

The main problem addressed in this research is that traditional web security systems are often too rigid to handle modern attacks. Many WAFs depend on prewritten signatures that only work when the attack is already known. If an attacker changes the payload slightly, uses encoding, or hides malicious code inside a normal-looking request, the firewall may fail to detect it. At the same time, purely anomaly-based systems may block too many legitimate requests if they are not trained properly. So the challenge is to build a firewall that is both accurate and flexible.

A second problem is balancing protection and performance. A WAF that inspects every part of every request may become slow, especially under high traffic conditions. On the other hand, if the firewall reduces inspection depth to improve speed, it may miss threats. This trade-off is one of the most important challenges in WAF design. Any useful system must find a practical middle ground where security is strong but latency remains acceptable.

A third problem is visibility. In many security systems, administrators do not get enough information about what is happening in real time. They may know that a request was blocked, but not why it was blocked or whether the same type of attack is increasing over time. Without proper logs, monitoring, and analytics, it becomes difficult to improve the firewall or respond to incidents. This research addresses that issue by making logging and feedback central parts of the design.

Objectives

The first objective of this project is to design a WAF that can inspect HTTP and HTTPS traffic before it reaches the backend application. This ensures that the system acts as a security gate rather than simply as a passive monitor. By placing the WAF in reverse proxy mode, the system gets full visibility into the traffic flow and can make decisions before damage occurs.

The second objective is to combine signature-based and anomaly-based detection in a single system. Signature-based detection is useful for attacks that are already known, such as SQL injection and cross-site scripting. Anomaly-based detection is useful when traffic looks unusual but does not match any known signature. A hybrid model is stronger because it can handle both cases and reduce blind spots.

The third objective is to build a logging and monitoring module that records all important events. This includes blocked requests, allowed requests, suspicious patterns, and response filtering actions. Good logs are important not only for security analysis but also for future improvement of the firewall. They help administrators understand what types of attacks are most common and where the detection rules need tuning.

The fourth objective is to create a feedback mechanism. The firewall should improve over time by learning from history. If it blocks safe traffic by mistake, that feedback should be used to reduce false positives. If a new attack pattern appears, the firewall should be able to incorporate that pattern into its protection logic. This makes the system adaptive rather than fixed.

The fifth objective is to protect the backend server and the user together. The backend server should only receive safe, validated, and policy-compliant traffic. At the same time, the user should not receive internal error details or sensitive system data in the response. This dual protection is one of the main strengths of the proposed design.

Literature Review

Research on Web Application Firewalls has progressed over time from simple filtering to more intelligent and adaptive defense systems. The earliest WAFs were mostly signature-based. They matched incoming requests against known patterns such

as suspicious keywords, script tags, or injection commands. These systems were useful for basic threats, but they were not very flexible. They often failed against new attack forms or malicious input hidden through encoding and obfuscation.

As the need for better visibility and centralized control increased, reverse proxy WAF architectures became more common. In this model, the firewall sits directly in front of the application and handles all client communication. This improves logging, SSL management, policy enforcement, and backend protection. It also makes deployment easier because all traffic passes through one point. Many practical implementations use this architecture because it provides a good balance of control and usability.

More recent work has focused on hybrid and intelligent WAF models. These systems attempt to combine the precision of signature-based detection with the flexibility of behavioral or machine learning-based analysis. This direction is important because modern attacks are often adaptive. A firewall that only checks for known patterns may not be enough. Hybrid systems aim to reduce this weakness by monitoring how requests behave over time and by identifying traffic that looks abnormal compared to normal usage.

Another important theme in the literature is performance. Deep inspection can improve security, but it can also increase processing time and resource usage. This means WAF designers must constantly balance detection strength with deployment practicality. A useful research project should therefore not only build a firewall, but also measure how much overhead it introduces and whether the overhead is acceptable. That is why evaluation is a key part of your project.

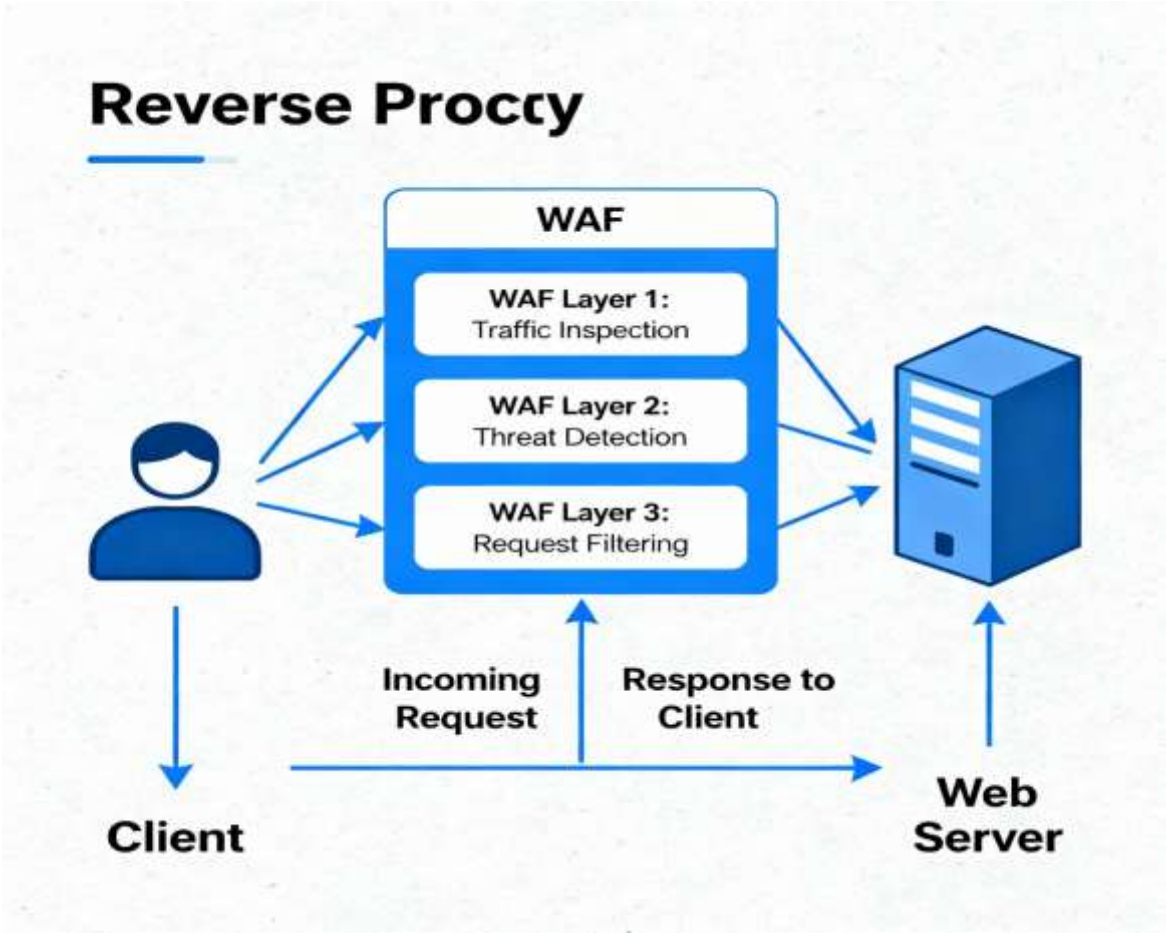
Proposed System

The proposed system is built around a layered reverse proxy architecture. The first layer is the traffic acquisition layer. This layer receives all incoming and outgoing web traffic and ensures that nothing bypasses the WAF. It is the first point of inspection and is responsible for capturing requests in real time. If SSL termination is enabled, this layer can also decrypt traffic for inspection before forwarding it to the next layer.

The second layer is the policy, detection, and storage layer. This is the core of the firewall. Here the system checks requests using signature-based rules and also evaluates behavior using anomaly detection features. It looks at the request method, URL structure, parameter patterns, header consistency, frequency of access, and session behavior. If the request looks malicious, it can be blocked, challenged, or logged for further analysis. All of this data is stored securely so that it can be used for monitoring and future improvement.

The third layer is the analysis and visualization layer. This layer helps the administrator understand what the firewall is seeing. Instead of only storing raw logs, the system organizes them into readable summaries, dashboards, and reports. This makes it easier to identify attack trends, high-risk periods, repeated attack sources, and detection performance. The monitoring layer is especially important because it turns the WAF from a simple filter into a security intelligence tool.

A major feature of the proposed system is response sanitization. Many attackers learn from application responses, especially when errors or debug messages are exposed. The firewall therefore checks outgoing content and removes sensitive details before the response reaches the user. It can also add or enforce security headers such as CSP, HSTS, X-Frame-Options, and X-Content-Type-Options. This improves browser-side safety and reduces the chance of exploitation.



Methodology

The methodology of this project follows a design, implementation, and evaluation approach. The first step is to build the firewall using reverse proxy tools and detection modules. Nginx is used as the front-facing proxy, ModSecurity is used for rule-based filtering, Python is used for custom logic and automation, and ELK is used for logging and visualization. This gives the system both security and observability.

The second step is to test the firewall under controlled traffic conditions. Normal browsing requests are mixed with malicious payloads so that the system can be observed under realistic conditions. The malicious examples may include SQL injection attempts, XSS payloads, CSRF-like requests, suspicious header manipulation, and repeated automated access. This helps measure whether the WAF can accurately separate safe and dangerous traffic.

The third step is to evaluate the firewall using measurable criteria. The most important metrics are detection rate, false positive rate, response latency, memory usage, CPU use, and log quality. If machine learning is included, precision, recall, F1-score, and ROC-AUC can also be used. These metrics allow the research to show whether the system is effective in both security and performance terms.

The fourth step is improvement through feedback. If the logs show that some legitimate traffic is being blocked, the rules can be adjusted. If repeated attack patterns appear, the signatures or anomaly thresholds can be updated. This feedback loop makes the system more adaptive and practical for longer-term use.

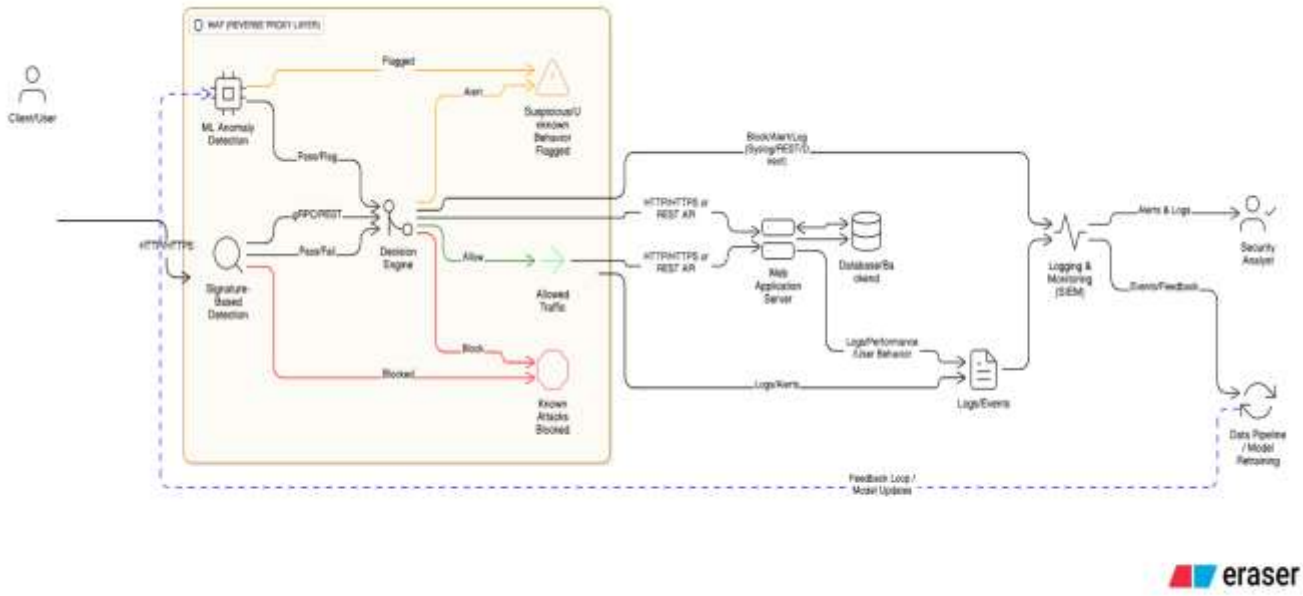
Implementation Details

The request inspection module is the part of the system that makes the first security decision. It checks incoming HTTP methods, path structures, query parameters, cookies, and headers for suspicious content. For example, if a request contains unusual SQL keywords, excessive encoding, or abnormal parameter growth, it may be flagged for blocking. This module is important because it provides the first layer of defense.

The anomaly detection component adds intelligence to the system. Instead of checking only known attack signatures, it examines how traffic behaves. If a user suddenly sends too many requests, accesses unusual paths, or behaves differently from normal session patterns, the system can treat that as suspicious. This is useful for detecting bots, brute-force attempts, and attacks that do not follow standard signatures.

The response sanitization module acts as the final protective layer. It checks outgoing server responses and removes anything that might expose internal details. It also normalizes headers and can replace backend error messages with cleaner generic responses. This prevents attackers from learning too much about the server environment.

The logging module stores both security and operational information. This includes the time of the event, source address, request pattern, detection result, and action taken. Over time, this log data becomes very valuable because it can show trends, attack frequency, and firewall behavior. The visualization layer then turns that data into dashboards and charts that are easy to understand.



Evaluation and Discussion

The expected outcome of this project is that the hybrid firewall will perform better than a simple rule-based firewall. Signature rules should successfully stop known threats, while anomaly analysis should help detect new or unusual attack patterns. This combination should reduce the chance of missed attacks and make the system stronger in real-world conditions.

At the same time, the system will likely introduce some overhead because every request is inspected. That is normal for a WAF. The important question is whether the additional delay is acceptable compared with the security gain. In most applications, a moderate increase in latency is worth the improvement in protection, especially for systems that handle sensitive data.

The discussion should also highlight usability. A security system is only useful if administrators can understand and manage it easily. The logging and dashboard features help with this by showing what is blocked, what is allowed, and how the traffic changes over time. This makes the system more practical and more suitable for real deployment.

Conclusion

This project presents a strong and modern approach to web application protection. It combines reverse proxy deployment, hybrid threat detection, response sanitization, logging, and feedback-driven improvement into one integrated WAF design. That makes it more advanced than a basic firewall and more suitable as a research project.

The main value of the study is that it addresses real security problems while also creating room for evaluation and improvement. Modern web applications need defenses that are both accurate and adaptive, and this project tries to provide exactly that. By combining rule-based filtering with behavioral analysis and monitoring, the system aims to protect applications more effectively against both known and emerging threats.

Overall, this is a good research topic because it is practical, current, and technically meaningful. It has a clear problem, a clear solution, and measurable outcomes.

REFERENCES

1. D. Arnaldy and T. Setia Hati, "Performance Analysis of Reverse Proxy and Web Application Firewall with Telegram Bot as Attack Notification On Web Server," in 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE), Bali, Indonesia, 2020, pp. 1–6.

2. R. A. Muzaki, H. Ritchi, and C. Obrina, "Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall," Padjadjaran University, December 2020.

3. S. Prandl, M. Lazarescu, and S. Pham, "A Study of Web Application Firewall Solutions," in Lecture Notes in Computer Science, vol. 9521, International Conference on Information Systems Security, Dec. 2015, pp. 350–364, doi:10.1007/978-3 319-26961-0_29.

4. 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, Turin, Italy, Jul. 2018, pp. 203–208, doi:10.1109/COMPSAC.2018.00144.

5. L. Desmet, F. Piessens, W. Joosen, and P. Verbaeten, "Bridging the gap between web application firewalls and web applications," Nov. 2006.

6. S. Khan, "Bridging the Gap between Web Application Firewall and Web Applications," International Journal for Research in Applied Science and Engineering Technology, vol. 9, no. 5, pp. 272–276, May 2021, doi:10.22214/ijraset.2021.34087.

7. 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, Turin, Italy, Jul. 2018, pp. 835–836, doi:10.1109/COMPSAC.2018.00144.