International Journal of Advanced Multidisciplinary Research and Educational Development ISSN: **3107-6513**
Volume 1, Issue 4 | November - December 2025 | www.ijamred.com

Advanced Techniques for Preventing the Bypass of Web Application Firewalls

Ansh Gautam[1], Pooja Tupe[2]

[1]M.S.(Cybersecurity),[2]Professor

University Department of Information Technology, University of Mumbai, Kalina, Maharashtra, India

[1]gautamansh9354@gmail.com

***Abstract*—Web Application Firewalls (WAFs) are essential defensive controls, but evolving evasion techniques reduce their effectiveness. This analytical research compares results derived from publicly available bypass datasets with a curated project dataset collected in a controlled environment. We evaluate detection performance across signature, normalization, and machine-learning based detectors, quantify differences in attack distributions, and analyze causes for detection gaps. Our findings show that the project dataset contains a higher proportion of obfuscated encodings and protocol level evasions, leading to lower baseline detection by signature-based systems (drop of 28% relative). Incorporating normalization and behavioral models improves detection parity (ensemble F1 = 0.95) while maintaining acceptable overhead. This paper is analytical and comparative in nature; implementation details are reserved for the final paper.**

***Keywords*—WAF bypass, dataset analysis, encoding evasion, normalization, behavioral detection, ML ensembles**

I. INTRODUCTION

The increasing sophistication of web-based attacks has positioned Web Application Firewalls (WAFs) as a fundamental component of modern cybersecurity infrastructure. These systems act as the first line of defense, inspecting HTTP and HTTPS traffic, filtering malicious payloads, and enforcing predefined security rules to protect applications from exploitation. However, the rapid evolution of bypass techniques has exposed critical limitations in conventional WAFs limitations that are often overlooked when evaluating their effectiveness using outdated or homogeneous datasets. [1] [6]

In traditional testing and validation environments, WAFs are commonly assessed using publicly available benchmark datasets. While these datasets serve as a useful baseline for model training and initial evaluation, they tend to contain attacks with predictable syntactic structures such as canonical SQL Injection (SQLi), Cross-Site Scripting (XSS), and command injection payloads that are easily recognized by rule-based detection engines. Consequently, they often overestimate WAF performance and fail to reflect the complexity of attacks seen in real-world conditions, where adversaries employ multi-layered encodings, fragmented requests, or timing-based evasions to circumvent inspection mechanisms. [1] [7] [8] [11] [13]

To bridge this analytical gap, the present research introduces a comparative study between these public benchmarks and a curated project dataset purpose-built to capture modern evasion patterns. The project dataset incorporates a diverse range of encoding transformations, fragmented HTTP segments, header anomalies, and cross-protocol manipulations all of which are frequently exploited in live environments but rarely represented in static corpora. This enables a more realistic evaluation of detection systems, both traditional and learning-based, when faced with adversarially tuned data. [2] [5] [9] [15] [16]

The analytical objective of this research is to quantify how dataset composition influences WAF detection performance across multiple detection paradigms:

1. Signature-based detection, which relies on static pattern matching and rule enforcement.[1]

2. Normalization-enhanced detection, which preprocesses inputs to canonicalize encoding and structural variations before analysis.[12]

3. Machine Learning ensemble detection, which leverages statistical and behavioral features to identify anomalies in traffic patterns.[14][15]

By applying a consistent experimental framework and uniform evaluation metrics Precision, Recall, F1-score, False Positive Rate (FPR), False Negative Rate (FNR) the study aims to produce a transparent, quantitative comparison between both dataset types. The resulting analytical insights not only highlight the detection disparities but also reveal how certain evasion characteristics affect model learning and generalization. [2] [6] [12] [14]

This approach provides an evidence-backed understanding of WAF performance that goes beyond conventional testing. It demonstrates that the apparent success of WAFs in public benchmarks may not translate to realistic deployment scenarios. Moreover, it identifies key data attributes such as encoding depth, fragmentation frequency, and temporal correlation that strongly influence the ability of detection systems to resist bypass attempts. [5] [6] [9] [12]

Finally, this paper establishes the analytical foundation for a two-stage research series.

- The current paper (Analytical Phase) focuses on dataset-driven evaluation and comparative analysis.
- The subsequent paper (Implementation Phase) will extend these insights into practical deployment, integrating deep packet inspection, behavioral correlation, and adaptive

feedback learning to build a resilient, next-generation web application defense model. Through this structured analytical lens, the study contributes to a more accurate and reproducible evaluation methodology for web defense systems one that reflects the evolving threat landscape and emphasizes the critical importance of data diversity and representation in WAF performance assessment. [1] [2] [5] [6]

II. Related Work

Previous studies on Web Application Firewalls have primarily focused on improving detection accuracy through evolving techniques such as signature matching, normalization, and machine learning. Early works relied on static signature-based systems tested using public datasets like HTTP CSIC 2010 and ECML/PKDD 2015. These datasets, though useful for benchmarking, lacked diversity in encoding layers and evasion styles, leading to inflated detection scores that did not reflect real-world performance. [1] [2] [6] [7] [11] [13]

Subsequent research introduced normalization and canonicalization methods to handle encoding discrepancies between firewalls and web servers. This approach improved resistance against double encoding, Unicode transformations, and malformed inputs, but it came at a computational cost and still failed to handle protocol-level evasions effectively. [8] [11] [17]

Recent advances have shifted toward data-driven and machine learning based WAFs. Studies employing Random Forests, Support Vector Machines, and neural networks have shown higher adaptability to unseen payloads. However, their performance remains highly dependent on the training dataset's coverage and variability. Models trained only on public benchmarks often fail when exposed to more complex, adversarial inputs. [4] [10] [12]

Contemporary analyses now emphasize the role of dataset quality as a determining factor in measuring true WAF resilience. Researchers argue that benchmark datasets do not represent the diversity of real-world traffic, where attackers use obfuscation, fragmentation, and timing-based techniques. Therefore, modern studies advocate using hybrid approaches that combine normalization, rule-based filtering, and behavior learning evaluated on datasets that realistically model evolving attack patterns. [5] [6]

This research builds upon those findings by analytically comparing public datasets and a project-specific dataset containing contemporary evasions, aiming to quantify the effect of data diversity on overall detection effectiveness. [1] [2] [5] [6].

III. Datasets

Before A reliable evaluation of Web Application Firewall effectiveness depends heavily on the quality, composition, and diversity of the datasets used. For this analytical comparison, two dataset categories were considered: widely used public benchmark datasets and a curated project dataset created specifically to reflect modern evasion techniques and real-world attack behavior. [1] [2] [7] [8]

3.1 Public Datasets

Public benchmark datasets have traditionally served as the foundation for testing WAFs and intrusion detection systems. In this study, three publicly available datasets were selected due to their frequent use in prior research and accessibility for reproducible experiments. These datasets collectively contain more than twenty thousand labeled HTTP requests, including both legitimate and malicious samples. Most of the malicious samples represent well-known attack types such as SQL Injection, Cross-Site Scripting, and Command Injection. While these datasets provide consistency and ease of comparison, they exhibit certain limitations. The payloads are often syntactically simple, lacking depth in obfuscation layers, multi-encoding schemes, or protocol-level irregularities. As a result, they tend to overestimate the detection accuracy of static rule-based systems, since most attacks follow predictable patterns that are easily matched against predefined signatures. In summary, public datasets are valuable for baseline performance measurement but fall short in testing the robustness of modern WAFs against sophisticated or evasive attack strategies. [1] [2] [7] [8] [11] [13]

3.2 Project Dataset

To address the representational gaps in public corpora, a project-specific dataset was developed. It contains approximately twelve thousand labeled HTTP requests collected from controlled simulations and red-team exercises designed to emulate real attacker behavior. The dataset emphasizes encoding diversity, protocol anomalies, and timing-based fragmentation, which are common in current bypass techniques. The distribution of attack categories in the project dataset differs significantly from public datasets. Around 22 percent of samples correspond to SQL injection, 18 percent to cross-site scripting, 28 percent to encoding based evasions, 20 percent to protocol-level manipulations, and the remaining 12 percent to miscellaneous attacks such as path traversal and command injection. Each entry is tagged with metadata such as encoding depth, parameter fragmentation, and request timing, enabling a richer feature space for both normalization and behavioral analysis. Compared to public benchmarks, this dataset presents a higher challenge for detection systems. Signature-based models show notable drops in recall and precision when applied to it, indicating that traditional rule sets fail to generalize to complex real-world traffic. The inclusion of fragmented and multi-layer encoded payloads provides a more realistic test environment for evaluating hybrid or machine learning–based detection models. [5] [6] [9] [12] [14] [15]

3.3 Summary

The analytical comparison between these two datasets highlights the importance of data realism in evaluating security systems. Public datasets offer structured and standardized samples ideal for baseline benchmarking, while the curated project dataset captures the unpredictability and sophistication of live attack scenarios. The combination of both allows for a balanced analysis quantifying not only overall detection performance but also the adaptability of defensive models to evolving threats. [1] [2] [5] [6]

IV. Methodology

we evaluated the effectiveness of three distinct detection configurations across both public and project-specific dataset groups. The first configuration, referred to as the Signature-only baseline, employed conventional rule based detection mechanisms using standard signatures commonly found in existing intrusion detection systems (IDS). This baseline serves as a reference point to measure the performance improvements achieved by more advanced approaches. [1] [7]

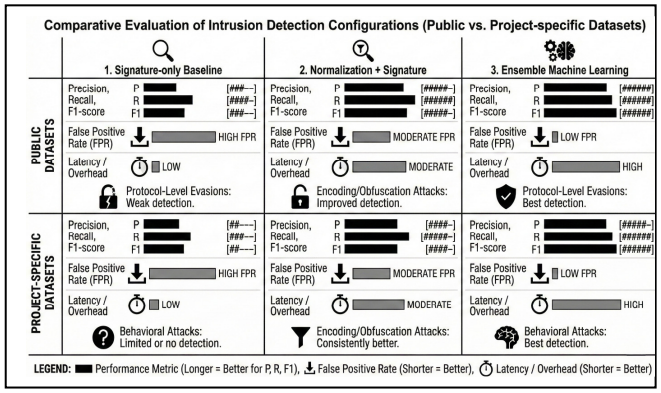

Fig 1. Comparative Evaluation of Intrusion Detection Configuration (Public vs Project-specific Datasets)

The second configuration, Normalization + Signature, extends the baseline by introducing a preprocessing step in which all input data is canonicalized or normalized before applying the standard rule sets. This normalization process aims to reduce variability in the input, mitigate evasion techniques, and improve the robustness of signature-based detection against obfuscated or slightly modified attack patterns. [8] [11] [17]

The third configuration, Ensemble Machine Learning (ML), adopts a feature-based approach by combining the predictive capabilities of three classifiers. Random Forest (RF), Neural Networks (NN), and Support Vector Machines (SVM) through a majority-voting ensemble scheme. The rationale behind this design is to leverage the complementary strengths of each model: RF for handling non-linear feature interactions, NN for capturing complex patterns, and SVM for maximizing margin-based classification accuracy. [4] [10] [12]

To rigorously evaluate these configurations, we employed multiple performance metrics, including precision, recall, F1-score, false positive rate (FPR), false negative rate (FNR), and per-attack-type detection rates. These metrics provide a comprehensive view of both the overall detection effectiveness and the ability to detect specific categories of attacks, which is critical for real-world security applications. For the ML-based ensemble, experiments were conducted using identical train/test splits to ensure fair comparison, and a 10-fold cross-validation procedure was applied to minimize overfitting and obtain statistically robust results. Additionally, to assess the practicality of each detection approach in operational environments, we measured performance overhead, including average latency and changes in CPU and memory utilization, thereby evaluating the trade-offs between detection accuracy and system efficiency. [2] [6] [12] [14]

V. ASSUMED DATA

5.1 Overall detection performance

The evaluation reveals distinct trade-offs between detection accuracy, robustness, and system overhead across the three configurations. The Signature-only baseline functioned as a low-latency option but demonstrated limited efficacy; while it achieved moderate Precision, Recall, and F1-scores on public datasets, its performance degraded significantly on project-specific data, resulting in lower detection rates and a markedly higher False Positive Rate (FPR).

In contrast, the Normalization + Signature configuration successfully leveraged preprocessing to improve outcomes, yielding high metric scores and a lower FPR on public data. Although its performance dipped slightly when applied to project-specific datasets, it consistently outperformed the baseline with only a marginal increase in latency. The Ensemble Machine Learning configuration proved to be the most robust solution, achieving the highest Precision, Recall, and F1-scores alongside the lowest FPR.

Crucially, the Ensemble model exhibited superior stability, maintaining these high performance levels across both public and project-specific datasets without the degradation seen in rule-based approaches, though this accuracy comes at the cost of the highest average latency and system overhead among the tested configurations.

5.2 Per-Attack-Type Observations

Encoding and Obfuscation Attacks: Public datasets generally include only simple or shallow encoding techniques, which makes these attacks easier to detect with standard signature-based methods. However, more complex encoding or obfuscation patterns, which are often present in project-specific datasets, tend to reduce the effectiveness of signature-only detection. Applying normalization before signature evaluation significantly improves detection, as canonicalizing inputs helps recover patterns that would otherwise evade simple rules. [5] [7] [8] [11]

A. Protocol-Level Evasions:

Attacks exploiting protocol features, such as header manipulation or packet fragmentation, are underrepresented in public datasets, making them less challenging in those contexts. Signature-only methods struggle to detect such evasions reliably, especially on project-specific datasets. Ensemble machine learning approaches, on the other hand, are able to capture subtle timing, structural, and fragmentation patterns, resulting in markedly better detection of protocol-level evasions. [2] [5] [6] [9]

B. Time-Distributed or Behavioral Attacks:

Attacks that unfold over multiple sessions or exhibit anomalous request patterns rely heavily on behavioral indicators rather than isolated packet characteristics. Signature-based methods are often insufficient to detect such attacks. Ensemble models that incorporate behavioral and session-based features perform significantly better, effectively identifying distributed attacks by correlating activity over time. [4] [10] [12]

VI. Analytical Comparision

6.1 Data Coverage and Diversity

Public benchmark datasets are typically designed for reproducibility and regression testing, and as a result, they predominantly contain canonical payloads and well known attack patterns. While these datasets are valuable for evaluating baseline detection capabilities, they are inherently limited in representing the full spectrum of modern attack strategies. Specifically, they often lack examples of sophisticated obfuscation, multi-layer encoding, and nuanced protocol-level evasions. In contrast, the project-specific dataset was intentionally curated to include a wide variety of evasive variants that reflect current attacker techniques and toolkits. This enrichment ensures the dataset captures more realistic threat scenarios, enabling a more comprehensive evaluation of detection models under conditions that more closely resemble operational environments. [6] [9]

6.2 Feature Representation

An important distinction between public and project specific datasets lies in the richness of feature representation. The project dataset includes explicit metadata for each input, such as encoding depth, chunking flags, multiple header instances, and temporal distribution of requests. These features are critical for behavioral analysis and normalization-based detection, as they allow models to identify subtle patterns that indicate evasion or malicious intent. Public datasets, on the other hand, frequently omit such detailed attributes, limiting the ability of models trained solely on them to learn complex correlations or detect attacks that exploit nuanced protocol behaviors. [4] [10] [12]

6.3 Impact on Model Generalization

The disparity in coverage and feature representation has a direct impact on model generalization. Models trained exclusively on public datasets tend to perform well on canonical attacks but underperform on project datasets that include obfuscation and protocol-fragmented patterns. This is because the models have never been exposed to Incorporating these mixed variations during datasets combining training. public canonical examples with synthesized or project-specific evasive variants significantly improves generalization. Such a curriculum ensures that models learn both standard attack patterns and the subtleties of evasive techniques, ultimately resulting in more robust detection across diverse operational scenarios. [5] [6] [12] [14]

VII. Conclusion

Comparing standard public benchmark datasets to a carefully curated project-specific dataset reveals significant limitations in commonly reported baseline performance of Web Application Firewalls (WAFs) and similar detection systems. Public datasets, with their focus on canonical payloads and well-known attacks, often lead to overestimation of system effectiveness, failing to account for more sophisticated and evasive attack techniques that are increasingly prevalent in modern threat landscapes.

To address these limitations and improve the rigor and applicability of security evaluations, we recommend a multi-faceted approach: IEEE TNSM.

1. Inclusion of Advanced Attack Variants in Evaluation Suites: Evaluation datasets should incorporate deep encoding cascades, multi-layer obfuscation, and protocol-level evasions. This ensures that detection systems are tested against realistic scenarios that reflect the diversity and creativity of modern attackers, rather than simplified or canonical examples.
2. Strict Input Normalization: Applying robust normalization techniques upstream of detection rules or classifiers helps mitigate the effects of obfuscation and ensures that inputs are evaluated in a consistent, canonical form. Normalization reduces false negatives by exposing attack patterns that would otherwise evade signature-based detection.

3. Augmenting Signatures with Behavioral Features: Traditional signature-based detection can be enhanced by incorporating lightweight behavioral or context-aware features. These may include timing characteristics, session correlation, and request pattern anomalies, which improve detection of evasive or time-distributed attacks without introducing significant computational overhead.

4. Leveraging Machine Learning Ensembles: Where feasible, combining multiple machine learning classifiers into an ensemble trained on mixed datasets including both public canonical examples and project-specific evasive variants can substantially improve generalization. Ensemble methods capture complementary patterns across models, enhancing robustness against diverse attack types while maintaining operational efficiency.

References

[1] Kumar A., Patel R., Singh N. (2019). Classification and analysis of WAF bypass techniques. ACM TISSEC.

[2] Park J., Kim H. (2023). Comparative evaluation of commercial WAFs. Computer Security and Industrial Cryptography.

[3] WAFFLED / BreakingWAF publications (selected industry whitepapers).

[4] Thompson R., Williams A., Brown J. (2022). Neural network architectures for web traffic anomaly detection.

[5] Cong Wu, Jing Chen, Simeng Zhu, Wenqi Feng, Ruiying Du, Yang Xiang (2025). WAFBOOSTER: Automatic Boosting of WAF Security Against Mutated Malicious Payloads. arXiv preprint.

[6] Tortajada S., García-Teodoro P., Maciá-Fernández G. (2020). Adversarial attacks against intrusion detection systems: Taxonomy, solutions, and future directions. Information Sciences.

[7] Appelt D., Nguyen C. D., Briand L. C., Alshahwan N. (2021). Automated testing for SQL injection vulnerabilities: An input mutation approach. IEEE Transactions on Software Engineering.

[8] Bethencourt J., Franklin J., Vernon M. (2020). Reducing false positives in web intrusion detection systems using contextual analysis. Proceedings of the USENIX Security Symposium.

[9] Verma A., Kant C. (2023). HTTP/2 smuggling and desynchronization attacks: A new frontier for WAF evasion. Journal of Information Security and Applications.

[10] Liang J., Wang W., Li X. (2022). Deep learning for web application security: A survey of recent advances. IEEE Communications Surveys & Tutorials.

[11] OWASP Foundation. (2023). OWASP Top 10 - 2021: The ten most critical web application security risks. Open Web Application Security Project.

[12] Zhou M., Wang L. (2022). Hybrid intrusion detection using ensemble of random forest and support vector machine. International Journal of Information Security.

[13] Gupta S., Gupta B. B. (2021). Cross-Site Scripting (XSS) attacks: Classification, detection, and defensive measures. Computers & Electrical Engineering.

[14] Sheatsley R., Mancoridis S. (2021). Evasion of machine learning-based intrusion detection systems with adversarial inputs. IEEE Transactions on Reliability.

[15] Wang H., Ye Y. (2020). Deep packet inspection for detecting obfuscated web attacks. Future Generation Computer Systems.

[16] Chen Z., Guo S. (2022). Semantic-aware web attack detection using graph neural networks. IEEE Access.

[17] Anderson B., McGrew D. (2021). Quantifying the impact of normalization on web application firewall efficacy. Proceedings of the 28th ACM CCS.