



ByteShield: A Custom Layer-7 Web Application Firewall Using Shannon Entropy and Tokenization for Anomaly and Attack Detection

Shubham Mishra¹, Rishabh Manale², and Rahul Prajapati³

Department of Computer Engineering

Smt. Indira Gandhi College of Engineering, Navi Mumbai, India

ABSTRACT :

While web applications now underpin commerce, education, and public services, they remain exposed to injection, scripting, and abuse attacks that conventional defenses struggle to withstand. Traditional Layer-7 web application firewalls (WAFs) depend upon preconfigured signatures and regular expressions, which encoded, obfuscated, and nested payloads can circumvent. Contemporary research counteracts these limitations through hand-configured rule engines, anomaly detection, machine learning, deep learning, entropy-based analysis, lexical tokenization, behavioural profiling, and risk scoring. This paper presents a methodology-focused review of 50 sources documenting the progression of web application defense from static signature matching toward multi-signal, explainable detection. We examine how request normalization, attack signatures, payload statistics, token structure, client behaviour, mitigation policies, and security telemetry have been implemented and evaluated across the reviewed literature. Rather than comparatively ranking detection algorithms in isolation, this review analyzes candidate methods according to their defensive functions and architectural responsibilities. Our synthesis indicates that rule matching, anomaly detection, entropy analysis, tokenization, and rate control constitute mature individual research domains, yet their integration into one transparent, risk-driven inspection pipeline remains limited. We formulate this gap as methodological fragmentation. To resolve it, we derive methodological requirements for the proposed ByteShield architecture, encompassing a reverse-proxy gateway, request normalization, signature rules, Shannon entropy, lexical tokenization, behavioural analysis, weighted risk scoring, graduated mitigation, and a monitoring dashboard, establishing an unambiguous, traceable path from literature analysis to system design.

Index Terms— web application firewall, reverse proxy, SQL injection, cross-site scripting, Shannon entropy, tokenization, anomaly detection, behavioural analysis, risk scoring.

INTRODUCTION

The rapid expansion of web-delivered services has transformed the HTTP request into the principal interface between anonymous users and business data. Banking portals, learning platforms, healthcare systems, and public APIs accept untrusted input from arbitrary clients, and every parameter, header, and request body becomes a potential carrier for malicious payloads. Foundational research on web vulnerabilities and attack classification demonstrates that SQL injection and cross-site scripting (XSS) remain highly persistent, well-documented threats, prioritized in the OWASP Top Ten [1]–[5]. However, deploying application-layer filtering does not by itself guarantee full protection, nor does it ensure that administrators can understand why single requests were rejected. Consequently, studies have progressed beyond edge filtering toward layered inspection and adaptive decision-making. Early Layer-7 defenses relied upon signature collections, regular expressions, reverse proxies, and policy rules, as in ModSecurity-style deployments, while subsequent proxy-based and adaptive WAF architectures introduced central monitoring and multi-threat coverage [6]–[17]. Anomaly-based detection advanced this paradigm by modelling normal request attributes and flagging statistical deviations. More recently, machine-learning and deep-learning classifiers have been applied to HTTP traffic to acquire payload representations without manual feature engineering [18]–[27]. Although learned detectors provide flexibility, they introduce challenges concerning training-data availability, explainability, adversarial evasion, and runtime cost. To mitigate these limitations, the literature increasingly examines information-theoretic and statistical measurements, including Shannon entropy, n-gram profiles, and payload byte distributions, which reveal unusual randomness or structural anomaly without depending upon exact signatures [28]–[37]. In parallel, research concerning evasion, obfuscation, and tokenization demonstrates that encoded payloads routinely bypass static filters, whereas lexical analysis, input normalization, and rate-based mitigation can restore visibility and constrain abuse [38]–[47]. Furthermore, protocol specifications governing HTTP messages and URI encoding establish that canonical request forms are a precondition for reliable inspection [48], [49]. These converging developments justify a methodology-focused synthesis rather than a conventional technology survey. Instead of cataloging the popularity of individual detection algorithms, this review investigates how disparate methods across the literature can be integrated into one reverse-proxy pipeline that permits, rate-limits, blocks, or temporarily blacklists incoming traffic. Recent surveys similarly document growth across rule-based, machine-learning, and hybrid WAF approaches, while emphasizing persistent bottlenecks in obfuscation resistance, false-positive control, explainability, throughput, and repeatable evaluation. Work by [50] and related IEEE reviews further reinforces the necessity of examining these defensive capabilities within an integrated framework.

LITERATURE REVIEW

Web Application Attack Landscape

Studies [1]–[5] establish the foundational context in which Layer-7 defenses operate. Risk rankings such as the OWASP Top Ten, with systematic classifications of SQL injection and surveys of XSS detection, describe how adversaries embed payloads within URLs, headers, parameters, and request bodies. Although these investigations validate the severity of injection and scripting threats, their primary analytical emphasis concerns vulnerability taxonomies and defensive techniques rather than a continuous, explainable inspection pipeline. The literature highlights a fundamental baseline requirement: an effective firewall methodology must inspect every request component before reaching the backend instead of assuming that code-level remediation alone satisfies organizational security obligations.

Rule-Based and Adaptive Web Application Firewalls

Studies [6]–[17] show the progression from static filtering toward adaptive protection. Signature rules, reverse proxies, security information and event management (SIEM) integration, normal-request models, and multi-threat architectures have been used to harden web applications. A recurring methodological principle is that detection requires some representation of normal behaviour. This representation may contain parameter lengths, character distributions, request structure, or access history. The limitation is that many systems treat signature matching and anomaly scoring as separate components rather than as signals fused into one risk decision.

Machine and Deep Learning Detectors

Studies [18]–[27] document the transition toward learned detection of web attacks. End-to-end deep networks, word embeddings, character-level convolutional architectures, and classifiers analyze HTTP traffic and payload patterns for SQL injection and XSS. Comparative evaluations indicate that model families exhibit variation in accuracy, training expenditure, and adversarial robustness. Investigations into obfuscated script detection and token-based representations demonstrate that learned models perform better when payloads are first normalized and structured instead of treated as unprocessed byte streams. The literature reveals that an effective detector requires not only a trained classifier, but a transparent, engineered feature pipeline.

Entropy and Information-Theoretic Analysis

Studies [28]–[37] address a critical weakness of exact-match filtering: payloads that are syntactically unusual yet absent from every signature list. Shannon entropy quantifies randomness within character strings, and entropy-based methods have flagged anomalies in network features, payload bytes, and traffic distributions. Such measures require no prior attack signature. Operational systems routinely employ n-gram models, byte-frequency profiles, feature distributions, and sliding-window statistics to enhance sensitivity toward previously unseen attacks. Nevertheless, existing systems typically evaluate entropy at packet or flow level, or apply it to single payload fields; few architectures integrate entropy with tokenization, behavioural history, and weighted decisions. During HTTP inspection, this challenge is worsened because entropy alone remains insufficient: legitimate tokens, encoded identifiers, and compressed values also appear statistically random.

Evasion, Tokenization, and Mitigation Controls

Studies [38]–[47] demonstrate advancements in understanding evasion and response. Adversarial mutation, encoding, case variation, and polymorphic blending have repeatedly been shown to bypass signature and learned filters. Complementary work on lexical analysis emphasizes token structure, grammar-aware inspection, and regular-expression pitfalls. Furthermore, anomaly-detection surveys, intrusion-detection critiques, and denial-of-service taxonomies extend the discussion toward rate control and source management. However, mitigation is largely evaluated as an isolated task. An unexploited methodological opportunity lies in routing combined detection signals into a graduated policy that allows, rate-limits, blocks, or temporarily blacklists sources while logging every decision for later forensic analysis.

REVIEW METHODOLOGY

This review follows a systematic, methodology-oriented analytical protocol. The candidate corpus was assembled through structured scholarly database and citation queries, with IEEE Xplore as the primary source, supplemented by peer-reviewed external venues where specific methodological evidence was required. The synthesis deliberately considers both foundational and more recent literature to reflect the shift from signatures toward hybrid detection. The resulting analytical set comprises 50 sources spanning web vulnerabilities, rule-based WAFs, anomaly detection, machine learning detectors, entropy analysis, evasion research, and mitigation.

Search Strategy

Representative search combinations included: "web firewall" AND "SQL injection"

"web application firewall" AND "XSS detection" "anomaly detection" AND HTTP AND "payload analysis" "Shannon entropy" AND payload

"lexical tokenization" AND "SQL injection" "reverse proxy" AND "attack detection" "rate limiting" AND "IP blacklisting"

"deep learning" AND "web attack"

"WAF evasion" AND "payload obfuscation" "SQL injection" AND XSS AND tokenization "entropy" AND "WAF" AND "anomaly detection"

Inclusion and Exclusion Criteria

Studies were included if they directly presented an empirical or methodological contribution to web application security, WAF design, attack detection, entropy or tokenization analysis, evasion, or mitigation. Publications were excluded if they discussed superficial vendor product overviews, non-refereed commentaries, generic network firewalls lacking application-layer inspection, or unrelated security domains. Relevant non-IEEE publications were retained in the working collection where they directly resolved an identifiable architectural gap; formal adherence to departmental citation guidelines will be coordinated with the project guide.

Data Extraction

Each selected primary source was systematically cataloged across attack context, detection technique, core method, evaluation framework, principal reported results, noted architectural limitations, and methodological implications. The core objective of this extraction was to synthesize recurrent limitations across disparate publications into an operational and testable collection of design requirements. This 50-source analytical matrix directly underpins both the research-gap and method-selection frameworks.

Analytical Procedure

The literature synthesis followed a four-phase workflow:

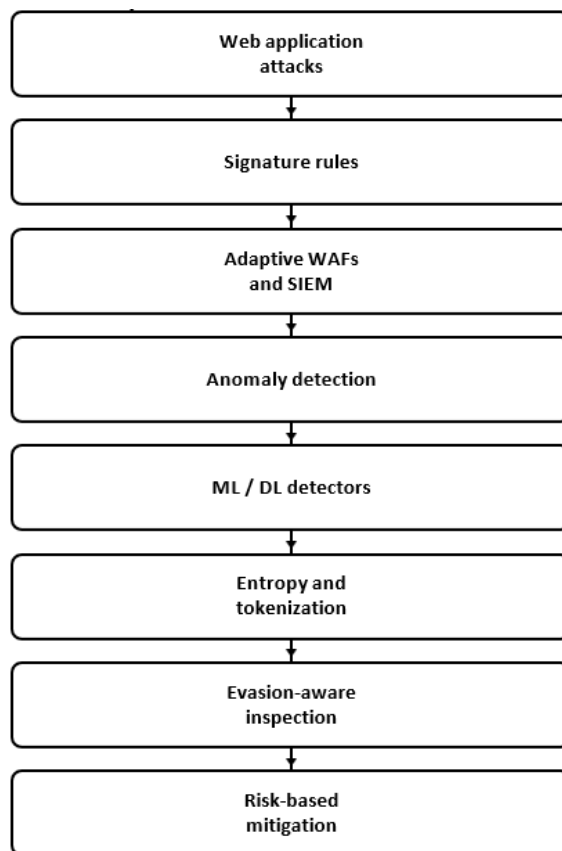
(1) functional categorization of studies by defensive objective and technical role; (2) thematic extraction of cross-cutting detection capabilities and failure points; (3) derivation of concrete methodological requirements addressing these recurrent technical limitations; and (4) systematic comparative mapping of candidate techniques against these requirements. This progression ensures that the proposed architecture is firmly grounded in published findings rather than retrofitted post hoc.

RESULTS AND DISCUSSION

Evolution of the Research Area

The 50 sources reveal a progression rather than a set of unrelated techniques:

This technical progression reflects an evolving defensive challenge: moving from initial signature filtering to adaptive anomaly modelling, obfuscation resistance, explainable scoring, and closed-loop mitigation.



Cross-Study Findings

TABLE I. CROSS-STUDY RESEARCH FINDINGS AND METHODOLOGICAL REQUIREMENTS

Literature Finding	Observed Limitation	Requirement	Methodological Direction
Signatures detect known attacks	Encoded or obfuscated payloads can evade matching	Normalization before inspection	Normalization pipeline
Anomaly models flag deviations	Need representative normal traffic	Multi-signal fused decisions	Scoring
Entropy exposes unusual randomness	Benign encoded values also random	Context-aware thresholds with structural checks	Entropy-token fusion
Tokenization reveals payload structure	Token rules need ongoing maintenance and tuning	Lexical structure analysis	Lexical tokenization engine
ML detectors learn patterns	Opaque decisions and costly labelled data	Explainable scoring	Weighted, thresholded risk scoring
Request frequency exposes abusive clients	Single-signal blocking often causes false positives or missed attacks	Graduated response	Tiered mitigation policy
Evasion mutates payload encodings	Static filters often miss unseen, encoded, mutated variants	Canonical decoding first	Multi-pass decoding
Rate limiting curbs API abuse	Fixed thresholds may block legitimate bursts	Per-client counters with expiry	Redis counters
Logs support audit	Decisions are often not explained	Explainable, structured event records	Dashboard and logging

TABLE III. SUMMARY OF THE 50 REVIEWED SOURCES

Ref.	Year / Paper / Venue	Attack Context/ Detection Type / Core Method	Main Finding	Limitation / Gap	Methodological Implication
1	2013 Defeating SQL Injection IEEE Computer, vol. 46	Databases Defense survey —	Reviews techniques for preventing and detecting SQL injection in web apps.	Predates modern entropy and token-based inspection.	Prevention baseline
2	2021 OWASP Top Ten Web Application Security Risks 2021 edition	Web applications Risk ranking —	Ranks injection, broken access control, and other critical web risks.	Awareness list, not a detection method.	Threat scope baseline
3	2015 Current State of Research on Cross-Site Scripting (XSS): A Systematic Literature Review Information and Software Technology	Web security XSS review —	Synthesizes XSS defense research.	Review, not an implemented WAF.	XSS requirements
4	2018 A Survey of Detection Methods for XSS Attacks Journal of Network and Computer Applications	Web security XSS detection survey —	Surveys classical and learning-based approaches for detecting XSS attacks.	XSS-focused scope only.	XSS detection coverage
5	2006 A Classification of SQL-Injection Attacks and Countermeasures Proc. ISSSE	SQL injection Attack taxonomy —	Classifies SQL injection attack types and countermeasures.	Predates entropy inspection.	Attack taxonomy
6	2007 Learning DFA Representations of HTTP for Protecting Web Applications Computer Networks	HTTP protocol Learned request model DFA induction	Learns automata for normal HTTP requests.	Generalization and drift unclear.	Normal-request modelling
7	2003 Anomaly Detection of Web-Based Attacks Proc. ACM CCS	Web servers Anomaly detection Statistical request models	Models normal parameter features to flag attacks.	Training data dependence.	Anomaly baseline
8	n/v Improving WAFs to Detect Advanced SQL Injection Attacks IEEE	Layer-7 WAF Advanced SQLi detection WAF improvements	Targets advanced and obfuscated SQL injection.	Details need verification.	WAF obfuscation baseline
9	2010 An Anomaly-Based Approach for Intrusion Detection in Web Traffic Journal of Information Assurance and Security, vol. 5	Web traffic Anomaly detection Learned request models	Proposes anomaly-based intrusion detection for web traffic.	Evaluation scope limited.	Anomaly detection
10	n/v Web Application Firewall Using Proxy and Security Information and Event Management (SIEM) for OWASP Cyber Attack Detection IEEE	Proxy WAF OWASP detection SIEM	Combines proxy filtering with SIEM monitoring.	Details and evaluation need verification.	Logging integration
11	2005 A Learning-Based Approach to the Detection of SQL Attacks DIMVA	Databases SQL attack detection Learned profiles	Learns normal query behaviour.	Training coverage limits accuracy.	Learning baseline

12	2019 Design and Implementation of an Artificial Intelligence-Based Web Application Firewall Model Neural Netw. World	Web applications AI-based WAF Machine learning model	Presents an AI-based WAF design.	Evaluation needs verification.	AI-assisted WAF
13	2005 A Multi-Model Approach to the Detection of Web-Based Attacks Computer Networks	Web servers Multi-model anomaly Statistical models	Combines several models to detect web attacks.	Training dependence.	Multi-signal fusion
14	2010 ModSecurity Handbook 1st ed., Feisty Duck, London, U.K.	Open-source WAF Rule engine Signature rule matching	Documents deployment of rule-based web application firewalls.	Static signatures struggle with encoded or obfuscated payloads.	Rule engine baseline
15	2006 Using Generalization and Characterization Techniques in the Anomaly-Based Detection of Web Attacks NDSS	Web applications Anomaly detection Generalization and characterization	Improves anomaly-based detection of web attacks using characterization.	Training data dependence.	Anomaly refinement
16	n/v Adaptive Web Application Firewall for Multi-Threat Detection IEEE	Adaptive WAF Multi-threat detection —	Adaptive multi-threat WAF.	Details need verification.	Adaptive rules
17	2018 Web Application Attacks Detection Using Machine Learning Techniques ICMLA	Web attacks Anomaly and classification ML techniques	Applies machine learning to detect web application attacks.	Needs verification.	Learned detection baseline for comparison
18	2025 UniEmbed: Word2Vec, USE, and FastText Embeddings with Machine Learning for XSS and SQL Injection Detection	XSS/SQLi Embedding models ML	Targets both XSS and SQL injection detection using text embeddings.	Metadata unverified; learned model, not entropy-based.	Embedding features
19	2016 SQLiGoT: Detecting SQL Injection Attacks Using Graph of Tokens and SVM Computers & Security (Elsevier)	SQL injection Token graphs SVM	Uses token graphs for SQLi.	Learned classifier; no entropy or behaviour.	Token structure
20	2016 Detecting and Removing Web Application Vulnerabilities with Static Analysis and Data Mining IEEE Transactions on Reliability, vol. 65	Web applications Vulnerability detection Static analysis	Combines static analysis with data mining.	Source-code level, not runtime request inspection.	Code-level complement
21	2018 DeepHTTP: Semantics-Structure Model with Attention for Anomalous HTTP Traffic Detection and Pattern Mining arXiv:1810.12751	HTTP traffic Anomaly detection Deep learning	Learns HTTP semantics and structure.	Year and venue unverified.	Payload structure analysis
22	2018 DeepXSS: Cross Site Scripting Detection Based on Deep Learning Proc. ICCAI	XSS Detection Deep learning	Detects XSS with deep learning.	Needs training data; opaque.	Learned XSS detection
23	2009 Obfuscated Malicious JavaScript Detection Using Classification Techniques Proc. MALWARE	JavaScript Obfuscation detection Classifiers	Classifies obfuscated malicious JavaScript, relevant to obfuscated XSS payloads in requests.	Browser-side focus.	Obfuscation-aware inspection signal
24	2012 Automatic Classification of Cross-Site Scripting in Web Pages Using Document-Based and URL-Based Features Proc. ISCC	Web pages XSS classification Document and URL features	Classifies XSS using page and URL features.	Feature engineering required.	URL feature signals

Ref.	Year / Paper / Venue	Attack Context / Detection Type / Core Method	Main Finding	Limitation / Gap	Methodological Implication
25	2019 Detecting Web Attacks with End-to-End Deep Learning JISA	Web attacks End-to-end learning Deep learning	Reduces manual feature engineering for attacks.	Opaque decisions; needs labelled training data.	Learned feature extraction baseline
26	2017 Anomaly-Based Web Attack Detection: A Deep Learning Approach Proc. ICNCC	HTTP requests Anomaly detection Deep learning	Detects anomalous web attacks via deep learning.	Metadata unverified.	Deep anomaly baseline
27	2018 Web Application Firewall Using Character-Level Convolutional Neural Network IEEE CSPA, Penang, Malaysia	Web application firewall Character-level learning CNN payload classification	Applies character-level CNN inside a WAF.	Opaque model.	Character-level features
28	2004 Anomalous Payload-Based Network Intrusion Detection RAID, Sophia Antipolis	Network payloads Anomaly detection Byte statistics	Models byte frequency distributions of payloads.	Network-layer scope.	Payload statistics
29	2005 Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation Proc. 5th ACM SIGCOMM Internet Measurement Conference (IMC), Berkeley, CA	Network traffic Anomaly detection Maximum entropy estimation of traffic distributions	Uses maximum entropy for traffic anomaly detection.	Packet-level, not HTTP-aware.	Entropy-based baseline
30	2005 Mining Anomalies Using Traffic Feature Distributions Proc. ACM SIGCOMM Conference, Philadelphia, PA, USA	Network traffic Anomaly diagnosis Entropy of feature distributions	Entropy of feature distributions reveals anomalies.	Network-level.	Entropy signal
31	2013 A Close Look on n-Grams in Intrusion Detection Proc. ACM Workshop on Artificial Intelligence and Security	Network payloads n-gram analysis Intrusion detection	Examines how n-grams behave in intrusion detection.	Not HTTP-specific.	Payload token statistics
32	2015 An Entropy-Based Network Anomaly Detection Method Entropy, vol. 17, no. 4	Network flows Entropy-based anomaly detection Entropy traffic feature measures	Reviews entropy for detecting network anomalies.	Network flows, not payloads.	Entropy threshold design

33	1948 A Mathematical Theory of Communication Bell System Technical Journal	Theory Foundation —	Defines entropy as a measure of uncertainty.	Not a security method.	Entropy definition
34	2006 Detecting Unknown Network Attacks Using Language Models DIMVA	Network payloads Unknown attack detection Language models with n-gram features	Detects unknown network attacks using language-model techniques.	Network-level scope.	Payload language statistics
35	2008 An Empirical Evaluation of Entropy-Based Traffic Anomaly Detection Proc. ACM Internet Measurement Conference, Vouliagmeni, Greece	Network traffic Anomaly detection evaluation Entropy-based feature measures	Empirically evaluates entropy-based anomaly detection methods.	Traffic-level, not payload-level.	Entropy lessons
36	2003 Statistical Approaches to DDoS Attack Detection and Response DARPA Information Survivability Conference and Exposition (DISCEX)	DDoS Statistical detection Entropy-based	Applies statistics to DDoS detection.	Flood-focused.	Rate anomaly signals
37	2001 Information-Theoretic Measures for Anomaly Detection IEEE Symposium on Security and Privacy	Network data Information-theoretic measures Entropy and conditional entropy	Uses entropy measures for anomaly detection models.	Needs adaptation to HTTP.	Entropy measure selection
38	2014 Automated Testing for SQL Injection Vulnerabilities: An Input Mutation Approach ISSTA	WAF evasion Adversarial mutation Machine learning guided	Shows WAFs can be evaded by mutated payloads.	Attack-side study.	Evasion-aware testing
39	2009 Anomaly Detection: A Survey ACM Comput. Surv., vol. 41, no. 3	SQL injection Testing Input mutation	Mutates inputs to find flaws.	Testing, not protection.	Mutation test cases
40	2010 Outside the Closed World: On Using Machine Learning for Network Intrusion Detection Proc. IEEE S&P	XSS filters Filter weakness Regular expressions	Shows regex-based XSS filters can be bypassed.	Client-side focus.	Beyond regex matching
41	2004 A Taxonomy of DDoS Attack and DDoS Defense Mechanisms ACM SIGCOMM Comput. Commun. Rev.	XSS Client-side defense Request filtering	Mitigates XSS on the client side.	Client-side only.	Complementary defense
42	2006 Polymorphic Blending Attacks Proc. 15th USENIX Security Symposium, Vancouver, BC	Network IDS Evasion Polymorphic blending	Shows payloads can imitate normal statistics.	Network-level evasion.	Statistical evasion risk
43	2007 Swaddler: An Approach for the Anomaly-Based Detection of State Violations in Web Applications Proc. RAID, Gold Coast, Australia	Intrusion detection Adversarial attacks Taxonomy	Taxonomy of attacks against detectors.	Not web-specific.	Robust multi-signal design
44	2010 Regular Expressions Considered Harmful in Client-Side XSS Filters WWW	Network IDS Machine learning critique —	Cautions against naive machine learning use.	Network IDS focus.	Evaluation design
45	2006 Noxes: A Client-Side Solution for Mitigating Cross-Site Scripting Attacks ACM Symposium on Applied Computing (SAC)	Anomaly detection Survey —	Surveys anomaly detection techniques broadly.	General, not HTTP-specific.	Method selection
46	2013 Adversarial Attacks Against Intrusion Detection Systems: Taxonomy, Solutions and Open Issues Information Sciences	DDoS Taxonomy —	Taxonomy of DDoS attacks and defenses.	Network-level.	Rate control rationale
47	2020 WAF-A-MoLE: Evading Web Application Firewalls Through Adversarial Machine Learning SAC	Web applications Anomaly detection State violation models	Detects state violations.	Needs instrumentation.	Behavioural state signals
48	2005 Uniform Resource Identifier (URI): Generic Syntax RFC 3986, IETF	URI Syntax standard —	Specifies URI syntax and percent-encoding.	Specification, not detection.	Decode before inspect
49	2014 Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing RFC 7230, Internet Engineering Task Force (IETF)	HTTP Message standard —	Specifies HTTP message syntax and routing.	Specification, not detection.	Canonical request parsing
Ref.	Year / Paper / Venue	Attack Context / Detection Type / Core Method	Main Finding	Limitation / Gap	Methodological Implication
50	2025 SQL Injection Detection and Prevention Techniques IEEE ICDSNS	WAF Technique review Rule, ML, hybrid	Reviews SQL injection detection and prevention techniques.	Unverified metadata.	Hybrid detection rationale

Research Gap

The compiled literature demonstrates that standalone signature-matching engines, anomaly detectors, machine learning classifiers, entropy measures, and rate limiters are individually well-documented. Rather, the central open challenge is methodological fragmentation. Individual studies consistently establish successful techniques for isolated, narrowly delimited detection tasks; however, these functional modules are rarely integrated into a single, coherent, explainable inspection pipeline. Existing literature displays a pronounced scarcity of architectures that simultaneously normalize encoded requests, match known attack rules, quantify payload randomness, scrutinize token structure, profile client behaviour, integrate these signals into a risk score, and implement graduated mitigation.

Consequently, this review identifies the following research gap: while rule-based inspection, anomaly detection, entropy analysis, lexical tokenization, and rate-limiting control have each independently validated their utility, their integration into a single consolidated, normalization-sensitive, risk-quantified, and interpretable Layer-7 inspection methodology remains fragmented and insufficiently evaluated empirically.

Methodological Requirements Derived from the Gap

TABLE II. METHODOLOGICAL REQUIREMENTS DERIVED FROM THE LITERATURE

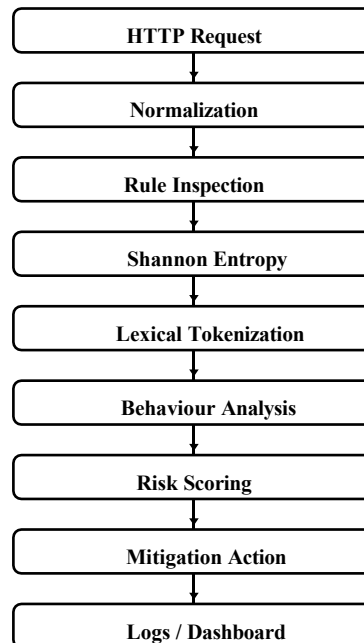
Requirement	Candidate methodological component
Request interception	Reverse-proxy gateway
Reliable attack detection	Signature rules plus request normalization
Unseen payload detection	Shannon entropy and tokens
Behaviour	Per-client frequency profile
Combined signal decisions	Weighted risk scoring
Proportionate mitigation	Threshold-based graduated actions
Explainable decisions	Per-signal contribution records in logs
Real-time monitoring	Dashboard with analytics
Controlled evaluation	Repeatable normal, malicious, and high-volume traffic

Path Toward Method Selection

The combined literature shows that resolving this fragmentation requires a modular, staged inspection pipeline. Clear interception needs a reverse-proxy gateway. Resisting encoded payloads requires input normalization. Known attacks require vetted signature rules. Detecting unknown payloads requires Shannon entropy and lexical tokenization. Repeated abuse requires behavioural frequency analysis. Joining these signals requires weighted risk scoring with thresholds, and graduated response requires rate limiting and temporary blacklisting. Structured logs and a dashboard must expose every decision.

Operational and security evidence must be recorded in structured security logs so administrators can inspect threat type, risk score, source address, endpoint, and action taken. Controlled evaluation must then consistently report detection rate, false-positive rate, false-negative rate, response time, throughput, and mitigation effectiveness.

The resulting methodological chain is:



Proposed Methodological Direction for Future Research

Based on this synthesis, future investigations should prioritize an integrated, multi-signal inspection pipeline instead of isolated filters. The gateway layer must terminate

and forward HTTP/HTTPS communications as a reverse

proxy; the normalization layer must decode URLs, headers, parameters, and bodies into canonical representations; the rule layer must apply maintained signatures addressing SQL injection and XSS; the statistical layer must compute Shannon entropy across suspicious fields; the lexical layer must tokenize payloads and examine token structure; the behavioural layer must monitor request frequency and patterns for every client; and the decision

layer must integrate these heterogeneous signals through weighted risk scoring with configurable thresholds. Mitigation outcomes must directly govern subsequent processing, encompassing permission, rate limiting, blocking, and temporary IP blacklisting. When deployed for API endpoints, the pipeline must preserve persistent per-source counters so that excessive requests are throttled. This architectural direction is corroborated by related literature beyond the core corpus. Surveys of AI-powered and hybrid WAFs characterize a transition toward combining rules with learned models, while emphasizing challenges in explainability and false positives. Entropy-based anomaly investigations similarly underscore the value of distributional measurements for unseen threats. Furthermore, proxy and SIEM-based architectures confirm that centralized logging supports incident analysis and auditing. These complementary findings substantiate the architectural conclusions derived from our 50-source corpus.

CONCLUSION

This methodological review synthesized 50 sources across the domains of web application vulnerabilities, rule-based firewalls, anomaly detection, machine learning detectors, entropy analysis, lexical tokenization, evasion research, and automated mitigation. The evidence reflects considerable maturation across individual capabilities: vulnerability research establishes the persistence of injection and scripting threats; rule-based WAFs furnish a well-proven enforcement point; anomaly and learning methods extend coverage to unseen requests; entropy analysis quantifies anomalous randomness; tokenization exposes payload structure; and rate control constrains malicious traffic. Monitoring dashboards expose logs, threat details, and blocked sources transparently to administrators.

The central impediment identified in this review is not the absence of specific detection algorithms, but rather widespread methodological fragmentation across these components. To advance the state of the art, next-generation Layer-7 defenses must integrate normalization, signature rules, entropy analysis, tokenization, behavioural analysis, and weighted risk scoring into a closed, explainable loop. These conclusions furnish a traceable rationale for the ByteShield design, a reverse-proxy WAF that allows, rate-limits, blocks, or temporarily blacklists requests and records every decision. Future work must empirically validate whether this multi-stage integration improves detection rate, false-positive rate, false-negative rate, response time, throughput, and mitigation effectiveness. No experimental results are reported in this review.

REFERENCES

The following 50 sources form the review reference set. Bibliographic fields could not be checked against publisher records; every entry requires a final author, venue, and page check before submission.

- Shar and Tan, Defeating SQL Injection, IEEE Computer, vol. 46, 2013.
- OWASP Foundation, Inc., OWASP Top Ten Web Application Security Risks, 2021 edition, 2021.
- Hydara, Sultan, Zulzalil, and Admodisastro, Current State of Research on Cross-Site Scripting (XSS): A Systematic Literature Review, Information and Software Technology, 2015.
- Sarmah, Bhattacharyya, and Kalita, A Survey of Detection Methods for XSS Attacks, Journal of Network and Computer Applications, 2018.
- Halfond, Viegas, and Orso, A Classification of SQL-Injection Attacks and Countermeasures, Proc. ISSSE, 2006.
- Ingham et al., Learning DFA Representations of HTTP for Protecting Web Applications, Computer Networks, 2007.
- Kruegel and Vigna, Anomaly Detection of Web-Based Attacks, Proc. ACM CCS, 2003.
- Improving Web Application Firewalls to Detect Advanced SQL Injection Attacks, Xplore:7064617. [verify]
- Torrano-Giménez et al., An Anomaly-Based Approach for Intrusion Detection in Web Traffic, Journal of Information Assurance and Security, vol. 5, 2010.
- Web Application Firewall Using Proxy and Security Information and Event Management (SIEM) for OWASP Cyber Attack Detection, IEEE Xplore document 10346051. [Authors, year, and venue unverified.]
- Valeur et al., A Learning-Based Approach to the Detection of SQL Attacks, DIMVA, 2005.
- Tekerek and Bay, Design and Implementation of an Artificial Intelligence-Based Web Application Firewall Model, Neural Netw. World, 2019.
- Kruegel, Vigna, and Robertson, A Multi-Model Approach to the Detection of Web-Based Attacks, Computer Networks, 2005.
- Ristic, ModSecurity Handbook, 1st ed., Feisty Duck, London, U.K., 2010.
- Robertson et al., Using Generalization and Characterization Techniques in the Anomaly-Based Detection of Web Attacks, NDSS, 2006.
- Adaptive Web Application Firewall for Multi-Threat Detection, IEEE Xplore 10823239. [verify]
- Betarte et al., Web Application Attacks Detection Using Machine Learning Techniques, ICMLA, 2018.
- Bakır, UniEmbed: Word2Vec, USE, and FastText Embeddings with Machine Learning for XSS and SQL Injection Detection, 2025. [Venue and metadata unverified.]
- Kar, Panigrahi, and Sundararajan, SQLiGoT: Detecting SQL Injection Attacks Using Graph of Tokens and SVM, Computers & Security (Elsevier), 2016.
- Medeiros, Neves, and Correia, Detecting and Removing Web Application Vulnerabilities with Static Analysis and Data Mining, IEEE Transactions on Reliability, vol. 65, 2016.

- Yu et al., DeepHTTP: Semantics-Structure Model with Attention for Anomalous HTTP Traffic Detection and Pattern Mining, arXiv:1810.12751, 2018. [Project slides cite 2021; verify.]
- Fang et al., DeepXSS: Cross Site Scripting Detection Based on Deep Learning, Proc. ICCAI, 2018.
- Likarish, Jung, and Jo, Obfuscated Malicious JavaScript Detection Using Classification Techniques, Proc. MALWARE, 2009.
- Nunan et al., Automatic Classification of Cross-Site Scripting in Web Pages Using Document-Based and URL-Based Features, Proc. ISCC, 2012.
- Pan et al., Detecting Web Attacks with End-to-End Deep Learning, JISA 2019.
- Liang, Zhao, and Ye, Anomaly-Based Web Attack Detection: A Deep Learning Approach, Proc. ICNCC, 2017. [verify]
- Ito and Iyatomi, Web Application Firewall Using Character-Level Convolutional Neural Network, IEEE CSPA, Penang, Malaysia, 2018.
- Wang and Stolfo, Anomalous Payload-Based Network Intrusion Detection, RAID, Sophia Antipolis, 2004.
- Gu, McCallum, and Towsley, Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation, Proc. 5th ACM SIGCOMM Internet Measurement Conference (IMC), Berkeley, CA, 2005.
- Lakhina, Crovella, and Diot, Mining Anomalies Using Traffic Feature Distributions, Proc. ACM SIGCOMM Conference, Philadelphia, PA, USA, 2005.
- Wressnegger et al., A Close Look on n-Grams in Intrusion Detection, Proc. ACM Workshop on Artificial Intelligence and Security, 2013.
- Berezinski, Jasiul, and Szpyrka, An Entropy-Based Network Anomaly Detection Method, Entropy, vol. 17, no. 4, 2015.
- Shannon, A Mathematical Theory of Communication, Bell System Technical Journal, 1948.
- Rieck and Laskov, Detecting Unknown Network Attacks Using Language Models, DIMVA, 2006.
- Nychis et al., An Empirical Evaluation of Entropy-Based Traffic Anomaly Detection, Proc. ACM Internet Measurement Conference, Vouliagmeni, Greece, 2008.
- Feinstein et al., Statistical Approaches to DDoS Attack Detection and Response, DARPA Information Survivability Conference and Exposition (DISCEX), 2003.
- Lee and Xiang, Information-Theoretic Measures for Anomaly Detection, IEEE Symposium on Security and Privacy, 2001.
- Appelt et al., Automated Testing for SQL Injection Vulnerabilities: An Input Mutation Approach, ISSTA, 2014.
- Chandola et al., Anomaly Detection: A Survey, ACM Comput. Surv., vol. 41, no. 3, 2009.
- Sommer and Paxson, Outside the Closed World: On Using Machine Learning for Network Intrusion Detection, Proc. IEEE S&P, 2010.
- Mirkovic and Reiher, A Taxonomy of DDoS Attack and DDoS Defense Mechanisms, ACM SIGCOMM Comput. Commun. Rev., 2004.
- Fogla et al., Polymorphic Blending Attacks, Proc. 15th USENIX Security Symposium, Vancouver, BC, 2006.
- Cova et al., Swaddler: An Approach for the Anomaly-Based Detection of State Violations in Web Applications, Proc. RAID, Gold Coast, Australia, 2007.
- Bates, Barth, and Jackson, Regular Expressions Considered Harmful in Client-Side XSS Filters, WWW, 2010.
- Kirda et al., Noxes: A Client-Side Solution for Mitigating Cross-Site Scripting Attacks, ACM Symposium on Applied Computing (SAC), 2006.
- Corona, Giacinto, and Roli, Adversarial Attacks Against Intrusion Detection Systems: Taxonomy, Solutions and Open Issues, Information Sciences, 2013.
- Demetrio et al., WAF-A-MoLE: Evading Web Application Firewalls Through Adversarial Machine Learning, SAC, 2020.
- Berners-Lee, Fielding, and Masinter, Uniform Resource Identifier (URI): Generic Syntax, RFC 3986, IETF, 2005.
- Fielding and Reschke, Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing, RFC 7230, Internet Engineering Task Force (IETF), 2014.
- SQL Injection Detection and Prevention Techniques, IEEE ICDSNS, 2025. [Authors, venue name, pages, and DOI unverified; entry taken from project slides.]