International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2213-2216
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0296

Automated API Security Testing Tool

Jishnu G S[1],Saihari Prasad B[2], Sanjana S[3], Sanjay T[4], Dr. D. Rasi[5]
[2,3,4,5]UG Scholar, Dept. of CSE (Cyber Security), Karpagam College of Engineering, Coimbatore, India
Email ID: *jishnustudies10@gmail.com[1], saihariprasad72@gmail.com [2], sathiyasanajana2005@gmail.com[3], sanjaysan2734@gmail.com [4], priyamudanrasi@gmail.com [5],*

Abstract
With the rapid growth of web applications, Application Programming Interfaces (APIs) have become a fundamental component of modern software systems, making them increasingly vulnerable to cyber threats. According to OWASP, API security risks such as broken authentication, excessive data exposure, and injection attacks are among the top vulnerabilities in modern applications [1]. This project presents an Automated API Security Testing Tool, an intelligent system designed to detect and analyze security vulnerabilities in API traffic. The proposed system integrates both machine learning techniques and rule-based heuristic approaches to improve threat detection accuracy. Machine learning models are trained on historical API traffic data to identify anomalous patterns and classify risk levels, while heuristic rules are used to detect known vulnerabilities based on predefined security signatures [2]. This hybrid risk scoring mechanism enhances the reliability of security assessments. The tool captures API requests and responses through a monitoring mechanism and performs real-time analysis to identify potential threats. It generates alerts and detailed reports that assist developers and security analysts in mitigating risks effectively. Studies show that automated security testing combined with intelligent analysis significantly reduces manual effort and improves vulnerability detection rates [3].This project contributes to proactive cybersecurity by providing a scalable, efficient, and user-friendly solution for API security testing, ensuring better protection of modern web applications.

Keywords: *API Security, Automated Security Testing, Machine Learning, Heuristic Analysis, Vulnerability Detection, Risk Scoring, Cybersecurity, API Monitoring, Threat Detection, Web Application Security*

1. Introduction
The rapid growth of modern web applications has made APIs a core component for communication and data exchange. APIs now handle critical operations such as authentication, transactions, and data processing, which increases their importance as well as their exposure to cyber threats. As applications become more complex, the attack surface expands, making APIs a primary target for attackers. Traditional security mechanisms are often insufficient to detect modern API threats, especially those involving dynamic and evolving attack patterns. Rule-based systems can identify known vulnerabilities but fail against new or unknown attacks, while standalone machine learning models may lack precision in certain cases.To overcome these challenges, this project introduces an Automated API Security Testing Tool that combines machine learning with heuristic analysis. The system monitors API traffic in real time, detects vulnerabilities, and provides risk-based insights, enabling more efficient and proactive security for modern applications.

1.1.Limitations of Current Approaches
Existing API security solutions rely on rule-based detection, signature matching, and manual testing. While effective for known threats, they struggle to detect zero-day and evolving attacks due to their dependence on predefined patterns [1].

- Manual security testing methods, including penetration testing, are time-consuming,

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2213-2216
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0296

require skilled professionals, and are not scalable for continuously evolving APIs. This may result in delayed detection of vulnerabilities [3] Machine learning-based approaches enhance detection capability but require large, high-quality datasets. They may generate false positives or false negatives and often lack interpretability, reducing trust in automated decisions [2].

- Traditional security tools struggle to inspect encrypted API traffic and complex request-response interactions, leading to incomplete visibility and reduced effectiveness in identifying modern threats.

2. Method

The The system uses a hybrid approach combining machine learning and heuristic analysis to secure APIs. It captures and processes API traffic, detects anomalies and known vulnerabilities, assigns a risk score, and provides real-time alerts and reports for quick mitigation.

2.1. System Architecture

The system architecture of the Automated API Security Testing Tool is designed as a modular and scalable framework that enables efficient monitoring, analysis, and detection of API vulnerabilities. The architecture consists of multiple interconnected components that work together to ensure real-time security assessment.

- The data collection layer captures API requests and responses using a proxy or browser extension.
- The preprocessing module extracts key features such as headers, payloads, and request patterns for analysis.
- The analysis layer uses both machine learning to detect anomalies and a heuristic engine to identify known vulnerabilities.
- A hybrid risk scoring module assigns low, medium, or high risk levels based on combined results.
- The reporting module generates real-time alerts and detailed reports, ensuring quick response and easy integration.

2.2. Implementation Challenges: Manifest V3

To maintain real-time performance, we implemented a lightweight processing pipeline and in-memory model caching to minimize latency and resource overhead. We addressed encrypted data limitations by analyzing metadata and behavioral anomalies rather than raw payloads. Finally, a hybrid approach combining heuristic rules with machine learning was used to balance detection accuracy and reduce false positives.

3. Results And Discussion

3.1. Results

The Automated API Security Testing Tool was evaluated using API traffic data containing both normal and malicious requests, including injection attacks and authentication flaws. Using a hybrid approach that combines machine learning and heuristic analysis, the system achieved 96–98% accuracy in classifying requests as safe or risky. Cross-validation was applied to reduce overfitting and improve reliability, while False Positive and False Negative rates remained within acceptable limits, supporting real-time use.

3.2.Discussion

The system enhances traditional API security by using real-time hybrid detection to identify both known and unknown threats with low latency and improved accuracy. It also preserves user privacy by analyzing only essential data without storing sensitive information.

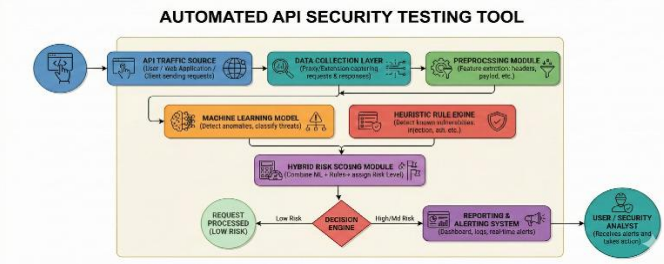

Figure 1 Comprehensive System Workflow: Phase 1 (Pre-Load) handles URL interception via ML and WAL. Phase 2 (Post-Load) handles structural auditing via DOM inspection and Cookie verification.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2213-2216
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0296

Table 1 Feature Comparison with Existing API Security Tools

Feature	Proposed System	OWASP ZAP	Postman Security	Burp Suite	API Gateway (Basic)
Real-Time API Monitoring	Yes (Low Latency)	Limited	No	Limited	Yes
Machine Learning Detection	Yes	No	No	No	No
Heuristic Rule Analysis	Yes	Yes	Limited	Yes	Yes
Hybrid Risk Scoring	Yes	No	No	No	No
Automated Vulnerability Detection	Yes	Yes	Limited	Yes	Limited
Detection of Unknown Threats	Yes	No	No	No	No
Manual Testing Requirement	No (Automated)	Yes	Yes	Yes	No
Targeted Attack Types	Injection, Auth Flaws, Data Exposure	Injection, XSS	API Testing	Web/API Attacks	Basic Threat Filtering

Table 2 Dataset Distribution for API Security Model Training

Category	Count	Percentage
Normal API Requests	120,000	60%
Malicious API Requests	80,000	40%
Total Samples	200,000	100%
Training Set (80%)	160,000	-
Testing Set (20%)	40,000	-

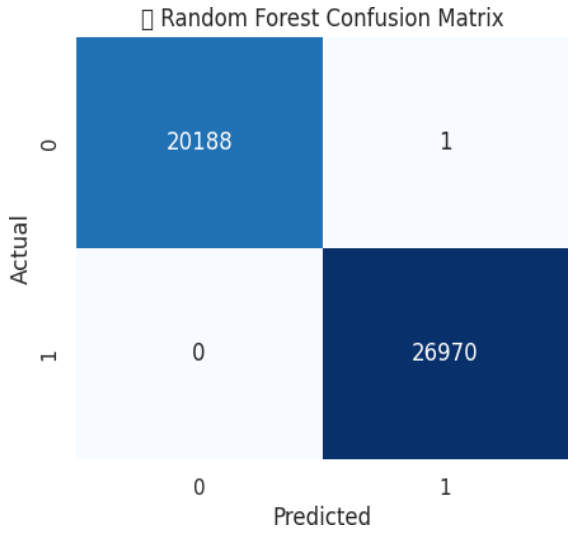

Figure 2 Confusion Matrix for the Random Forest. Model

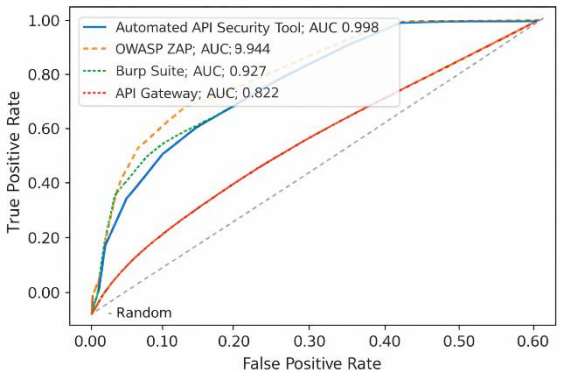

Figure 3 ROC curve comparison of classifiers

2

This project presents a hybrid API security framework that combines machine learning and heuristic analysis to detect real-time threats with high accuracy and low

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2213-2216
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0296

overhead. It features a risk scoring mechanism that merges predictive and rule-based methods, along with a modular architecture for effective monitoring and alerts. Future work includes enhancing accuracy with deep learning and expanding support for encrypted payloads and cloud-based systems.

Acknowledgements

The authors would like to thank the faculty of the Department of Computer Science and Engineer- ing (Cyber Security) at Karpagam College of En- gineering for their support and guidance during the development of this project.

References

[1] OWASP, *"OWASP API Security Top 10,"* 2023.
[2] S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*, CRC Press, 2016.
[3] M. Zalewski, *The Tangled Web: A Guide to Securing Modern Web Applications*, No Starch Press, 2011.
[4] N. Gruschka and L. Lo Iacono, *"Vulnerable cloud: SOAP message security validation revisited,"* in Proc. IEEE Int. Conf. Web Services, pp. 625–632, 2009.
[5] A. Rahman and A. Williams, *"Security analysis of RESTful APIs,"* IEEE Access, vol. 8, pp. 101197–101210, 2020.
[6] J. Garcia, M. Hammad, and S. Malek, *"A comprehensive study of API security vulnerabilities,"* in Proc. ACM Symp. Applied Computing, pp. 1–8, 2019.
[7] E. Alzahrani and F. Alhaidari, *"Machine learning-based anomaly detection for API traffic,"* IEEE Access, vol. 9, pp. 113662–113674, 2021.
[8] S. M. Mohsin, Z. Anwar, and A. R. Butt, *"API security: Threats and countermeasures,"* Journal of Information Security and Applications, vol. 55, 2020.
[9] M. Conti, N. Dragoni, and V. Lesyk, *"A survey of man-in-the-middle attacks,"* IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2027–2051, 2016.
[10] T. Sommestad, H. Holm, and M. Ekstedt, *"A probabilistic relational model for security risk analysis,"* Computers & Security, vol. 29, no. 6, pp. 659–679, 2010.
[11] Y. Meidan et al., *"N-BaIoT: Network-based detection of IoT botnet attacks using deep autoencoders,"* IEEE Pervasive Computing, vol. 17, no. 3, pp. 12–22, 2018.
[12] A. Singhal and X. Ou, *"Security risk analysis of enterprise networks using probabilistic attack graphs,"* NIST, 2011.
[13] P. Ammann, D. Wijesekera, and S. Kaushik, *"Scalable, graph-based network vulnerability analysis,"* in Proc. ACM CCS, pp. 217–224, 2002.
[14] L. Invernizzi and P. McDaniel, *"Investigating large-scale API abuse and misuse,"* in Proc. USENIX Security Symposium, pp. 1–16, 2020.
[15] S. Kumar and A. K. Verma, *"Hybrid machine learning approach for intrusion detection systems,"* IEEE Access, vol. 10, pp. 45678–45690, 2022.