iJRASET
International Journal For Research in Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** IV **Month of publication:** April 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.78833

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue IV Apr 2026- Available at www.ijraset.com*

# SECURESIGHT - A Multi-Module Website Security Analyzer

Aromal B.S[1], Ben Abishai Barik[2], Fathima Hana[3], Krishna S[4], Ms. Aiswarya S.S[5]
[1, 2, 3, 4]*Department of Computer Science and Engineering, Rajadhani Institute of Engineering and Technology, Trivandrum, Kerala, India*
[5]*Assistant Professor, Department of Computer Science and Engineering Rajadhani Institute of Engineering and Technology, Trivandrum, Kerala, India*

***Abstract:*** ***The rapid expansion of web technologies has transformed how organizations deliver services in commerce, education, and communication. At the same time, this growth has widened the attack surface for cyber threats, with adversaries exploiting insecure configurations, weak authentication, and malicious scripts. Manual detection of such vulnerabilities is often slow and error-prone, highlighting the need for automated solutions. This paper introduces SecureSight, a modular platform for website security analysis. The system integrates multiple modules— JavaScript malware detection, domain reputation checks, HTTP/TLS validation, and vulnerability scanning—into a unified framework. Built with React.js for the frontend and Node.js/Express.js for the backend, SecureSight executes concurrent scans and consolidates results through a centralized risk evaluation engine. Experimental evaluation demonstrates that the platform effectively identifies common vulnerabilities and provides actionable insights, supporting developers and security professionals in strengthening web application defenses.***
***Keywords: Website Security, Vulnerability Detection, SecureSight, Malware Analysis, Risk Evaluation.***

## I. INTRODUCTION

Web applications are central to today's digital infrastructure, supporting online transactions, information exchange, and customer engagement across industries. Their widespread adoption, however, has also expanded the attack surface for cyber threats. Adversaries frequently exploit vulnerabilities such as SQL injection, cross-site scripting (XSS), insecure HTTP configurations, and malicious script injection to compromise systems and steal data.

Although security practices have advanced, many existing detection tools remain specialized, focusing on narrow aspects of web security. Developers often need to combine multiple utilities to achieve comprehensive coverage, which increases complexity and reduces efficiency.

To address these limitations, SecureSight was developed as a unified platform for automated website security analysis. The system integrates multiple detection techniques into a modular architecture, enabling scalable assessments and delivering results through an interactive dashboard.

## II. RELATED WORK

Web security research has produced a variety of approaches to vulnerability detection. Traditional scanners rely on signature-based methods, which identify known malicious patterns in website code. While effective for common threats, these methods often fail to detect zero-day exploits or novel attack techniques.

Yu et al. introduced M-BERT, a modified BERT model for malicious URL classification. Their approach improved detection accuracy by analyzing URL structures semantically, though its high computational requirements limit lightweight deployment. Alnabulsi et al. proposed the Gathering Multiple Signatures Approach (GMSA), which detects SQL injection, XSS, shell injection, and file inclusion attacks. Despite strong precision, its dependence on predefined signatures reduces effectiveness against evolving threats.

Cigoj and Blažič developed an automated WCMS vulnerability discovery tool that fingerprints content management systems such as WordPress and Joomla, mapping them against known vulnerabilities. Their work revealed widespread unpatched sites but was restricted to CVE-based detection. Patel et al. presented a Web Vulnerability Scanner (WVS) framework that integrates multiple scanning techniques, emphasizing usability and scalability, though interpretation of results still required expert knowledge.

Hu et al. explored JavaScript malware detection using obfuscation analysis and string reconstruction, which improved identification of hidden malicious scripts but raised challenges in transparency for end users.

©IJRASET: All Rights are Reserved 

In practice, widely used tools such as OWASP ZAP, Google Safe Browsing, VirusTotal, and Mozilla Observatory each provide specialized capabilities— penetration testing, domain reputation checks, malware aggregation, and header validation respectively. However, these operate independently, requiring users to combine multiple platforms for comprehensive analysis.

## III. PROPOSED SYSTEM

SecureSight is a modular web security analyzer built on a full-stack architecture. Users submit a website URL through a React.js interface, which triggers backend modules implemented in Node.js/Express.js. Each module performs specialized checks:

*1)* Detecting obfuscated or malicious JavaScript
*2)* Assessing domain credibility via WHOIS data
*3)* Validating HTTP headers and TLS configurations
*4)* Identifying vulnerabilities such as SQL injection and XSS

Results are aggregated by a centralized risk evaluation engine, which assigns severity scores and generates a unified security report. The modular design ensures scalability and allows new modules to be added without disrupting existing functionality.

## IV. SYSTEM ARCHITECTURE

The SecureSight platform follows a client-server architecture designed for scalability and efficient security analysis.

The frontend is developed using React.js, which provides a responsive and interactive user interface for submitting URLs and displaying analysis results. The backend is implemented using Node.js and Express.js, which coordinate the execution of multiple security modules.

Communication between the frontend and backend occurs through RESTful APIs, enabling seamless data exchange and asynchronous processing.

To improve performance, SecureSight employs concurrent execution, where all security modules run simultaneously using asynchronous programming techniques such as Promise.all() and async/await. This approach significantly reduces scanning time and allows the system to generate results more efficiently.

The architecture also includes a centralized risk evaluation engine, which processes results from each module and generates a consolidated security report displayed on the dashboard

## V. SECURITY MODULES

### *A. JavaScript Malware Detection*

The JavaScript Malware Detection module analyzes website scripts to identify suspicious or malicious code patterns. The system retrieves webpage HTML using **Axios** and parses the scripts using **Cheerio**.

This module detects:

- Obfuscated JavaScript code (Base64 or hexadecimal encoding)
- Suspicious dynamic functions such as eval() or document.write()
- Hidden or compressed script blocks used to bypass security filters

These patterns are commonly associated with malware distribution and phishing attacks

### *B. URL Reputation Analysis*

The URL Reputation Analysis module evaluates the trustworthiness of a domain by retrieving domain registration information through WHOIS queries.

This module examines:

- Domain creation date
- Registrar information
- Top-level domain reputation

Newly registered domains or domains using suspicious extensions are flagged as potential phishing risks.

### *C. HTTP Header and TLS Validation*

The HTTP Header and TLS Validation module performs a server configuration audit to ensure that the website follows modern security standards.

The module checks for important security headers including:

- Content-Security-Policy
- Strict-Transport-Security
- X-Frame-Options

Additionally, it verifies the validity of the TLS certificate and ensures that the website enforces secure HTTPS connections.

*D. Vulnerability Scanning*

The Vulnerability Scanning module analyzes website content to identify potential security weaknesses such as:

- SQL Injection indicators
- Cross-Site Scripting (XSS) patterns
- Unsecured form submissions
- Exposed API keys or sensitive data

This module inspects HTML forms and input fields to identify potential entry points that attackers may exploit.

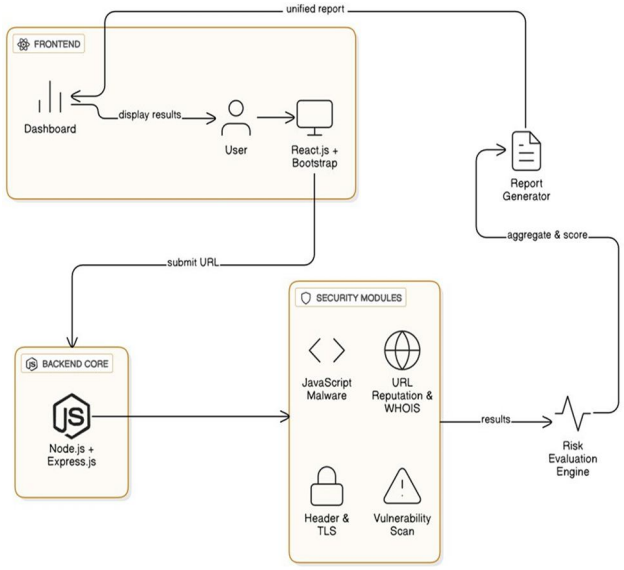



Figure 1 System Architecture Daigram

The SecureSight architecture consists of three major layers: the User Interface Layer, Backend Processing Layer, and Security Analysis Layer. The user interacts with the system through a web-based interface developed using React.js. When a URL is submitted, the request is sent to the backend server implemented using Node.js and Express.js.

The backend server coordinates the execution of multiple security modules including JavaScript Malware Detection, URL Reputation Analysis, HTTP Header and TLS Validation, and Vulnerability Scanning. These modules operate concurrently to analyze different aspects of the website. The results from each module are sent to the Risk Evaluation Engine, which aggregates the findings and generates a comprehensive security report.

## VI. IMPLEMENTATION

SecureSight is implemented using a full-stack web development environment.

The frontend interface is developed using React.js, providing a component-based architecture for efficient UI rendering. The dashboard displays security analysis results using interactive charts and status indicators.

The backend server is built using Node.js with Express.js, which manages concurrent requests and orchestrates the execution of security modules.

The Axios library is used to retrieve website content and HTTP headers, while Cheerio parses HTML documents to analyze scripts and webpage structure.

Each module operates as an independent service, ensuring modularity and easier maintenance. The results generated by the modules are aggregated by the Risk Evaluation Engine, which calculates weighted risk scores based on vulnerability severity.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

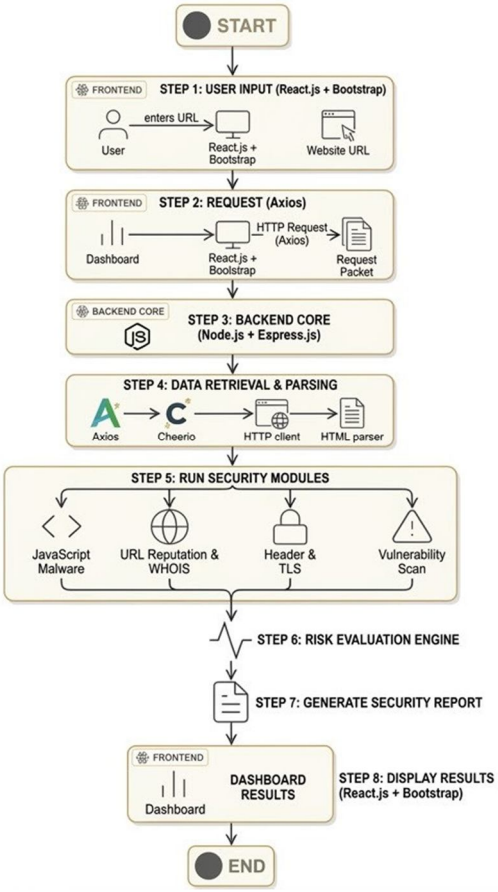


Figure 2 SecureSight Workflow Diagram

The workflow of SecureSight begins when the user submits a website URL through the web interface. The request is forwarded to the backend server where the system retrieves the HTML content and HTTP headers of the target website.

The retrieved data is processed by different security analysis modules running concurrently. Each module performs specific security checks such as malware detection, header validation, and vulnerability analysis. The results generated by these modules are then evaluated by the Risk Evaluation Engine, which assigns a risk score based on the severity of detected vulnerabilities. Finally, the results are sent back to the frontend dashboard where users can view the security status and recommended remediation actions.

## VII. RESULTS AND DISCUSSION

Testing across multiple websites demonstrated SecureSight's ability to detect suspicious scripts, missing headers, and insecure configurations. The dashboard presents findings with risk scores and color-coded indicators, enabling users to quickly assess security posture. Websites were categorized into low, medium, and high risk levels based on severity.

The modular design also supports future extensions, such as AI-driven threat detection and continuous monitoring, making SecureSight adaptable to evolving cyber threats.

| Website | Malware Detection | Security Headers | SSL Status | Risk Level |
|---|---|---|---|---|
| Site A | None | All Present | Valid | Low |
| Site B | Suspicious Script | Missing HSTS | Valid | Medium |
| Site C | Malware Detected | Missing Headers | Invalid | High |

The experimental evaluation of SecureSight was conducted by testing multiple websites with different security configurations. The system successfully identified missing security headers, suspicious JavaScript patterns, and vulnerabilities in web forms. The risk evaluation engine categorized websites into Low, Medium, and High risk levels based on the severity of detected issues.

## VIII. FUTURE WORK

While *SecureSight* provides an effective platform for automated website security analysis, several enhancements can broaden its scope and impact. One promising direction is the integration of machine learning and AI-based models to detect zero-day vulnerabilities and evolving attack patterns beyond traditional rule-based methods. This would allow the system to adapt dynamically to new threats.

Another extension is real-time monitoring, enabling continuous scanning of websites rather than one-time assessments. This would provide timely alerts when configurations change or new risks emerge, improving proactive defense.

Expanding coverage to API security analysis is also critical, as modern applications rely heavily on APIs for communication. Evaluating endpoints for authentication flaws and data exposure would significantly strengthen overall protection.

Additionally, integration with external threat intelligence databases could improve the accuracy of domain reputation checks by comparing results against global threat feeds. Enhanced data visualization techniques in the dashboard would further help users understand vulnerability trends and prioritize remediation.

Together, these improvements would make SecureSight a more comprehensive, intelligent, and scalable solution for safeguarding web applications in an evolving digital landscape.

## IX. CONCLUSION

This paper presented SecureSight, a modular platform for automated website vulnerability detection. By combining JavaScript malware detection, domain reputation analysis, HTTP/TLS validation, and vulnerability scanning into a single system, SecureSight delivers a comprehensive approach to evaluating web application security.

The system was implemented using modern web technologies, with React.js powering the frontend and Node.js/Express.js managing backend operations. Its modular architecture enables concurrent execution of multiple security checks, improving efficiency and coverage.

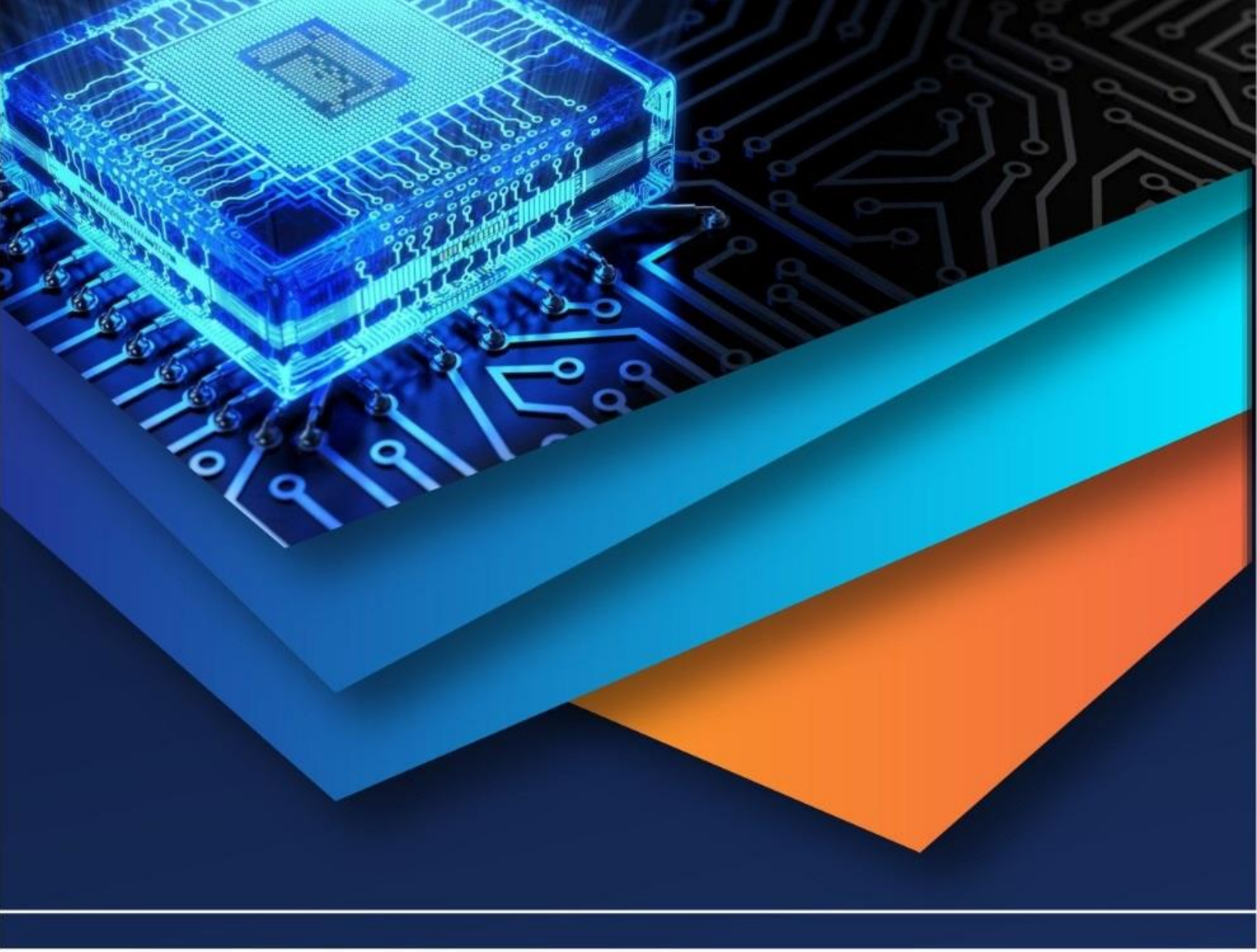
Experimental evaluation confirmed SecureSight's effectiveness in identifying common weaknesses such as malicious scripts, missing headers, and insecure configurations. The interactive dashboard further enhances usability by presenting results in a clear, accessible format.

Overall, SecureSight contributes to democratizing access to professional-grade security analysis. Its scalable design also provides a foundation for future enhancements, including machine learning-based detection, real-time monitoring, and expanded coverage for API security.

## REFERENCES

[1] B. Yu, F. Tang, D. Ergu, R. Zeng, B. Ma, and F. Liu, "Efficient Classification of Malicious URLs: M-BERT—A Modified BERT Variant for Enhanced Semantic Understanding," IEEE Access, vol. 12, pp. 13453-13468, 2024.
[2] H. Alnabulsi, R. Islam, and M. Talukder, "GMSA: Gathering Multiple Signatures Approach to Defend Against Code Injection Attacks," IEEE Access, vol. 6, pp. 77829-77840, 2018.
[3] P. Cigoj and B. J. Blažič, "An Intelligent and Automated WCMS Vulnerability-Discovery Tool: The Current State of the Web," IEEE Access, vol. 7, pp. 175466-175473, 2019.
[4] P. Patel, R. V. Reddy, D. S. Kiran, J. S. S. Harsha, and A. M. P. Reddy, "Enhancing Web Application Security: A Comprehensive Approach with WVS (Web Vulnerability Scanner)," Int. J. Eng. Res. Technol. (IJERT), vol. 13, no. 3, pp. 215-223, Mar. 2024.
[5] L. Hu, S. Sarker, B. Melicher, and A. Starov, "Malicious JavaScript Detection using Obfuscation Analysis and String Reconstruction Techniques," Computers & Security, vol. 149, p. 104152, Feb. 2025.
[6] OWASP Foundation, "OWASP Zed Attack Proxy (ZAP) Project Documentation," OWASP.org. Available: https://owasp.org
[7] Google, "Safe Browsing Transparency Report," Google Transparency Report. Available: https://transparencyreport.google.com/safe-browsing
[8] VirusTotal API Documentation: https://virustotal.com
[9] WHOIS Database, "Domain Registration and Ownership Lookup Service," Whois.domaintools.com. Available: https://whois.domaintools.com
[10] Mozilla Observatory: https://observatory.mozilla.org/

©IJRASET: All Rights are Reserved 



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)