Krishnaraj Chandrapratap Chauhan, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Design and Development of an Educational Web Security Analysis System for Vulnerability Detection

Krishnaraj Chandrapratap Chauhan[1], Sabi Ahsan Sayyed[2], Dr. Jasbir Kaur[3], Sandhya Thakkar[4]

[1,2]Student of Master of Computer Applications (MCA), Guru Nanak Institute of Management Studies, Mumbai, India
[3]Director, GNIMS B-School, Head of Information Technology and HR, Guru Nanak Institute of Management Studies, Mumbai, India
[4]Assistant Professor, Guru Nanak Institute of Management Studies, Mumbai, India

Abstract- Website security has become a critical concern due to the increasing number of cyberattacks targeting web applications. Many websites remain vulnerable because of improper security configurations such as invalid SSL certificates, missing HTTPS enforcement, insecure cookies, and absence of security-related HTTP headers. Existing security assessment platforms often require advanced technical expertise and are difficult for students and novice developers to utilize effectively. This paper presents the design and development of an Educational Web Security Analysis System capable of automatically evaluating website security configurations. The proposed system performs SSL certificate validation, HTTPS redirection verification, security header analysis, and cookie security assessment while generating security scores, visual dashboards, screenshots, and downloadable PDF reports. Developed using Python, Streamlit, Selenium, and Requests libraries, the platform aims to provide a simplified and educational approach to website security assessment. Experimental evaluation demonstrated detection accuracy exceeding 94% across all modules and an average report generation time of 2.1 seconds. The developed system achieved a System Usability Scale score of 89/100, indicating strong usability and effectiveness for cybersecurity education. Furthermore, detailed feedback from 50 student participants confirmed that the platform successfully bridges the gap between theoretical security concepts and practical vulnerability detection.

Keywords— Web Security, Cybersecurity, SSL/TLS, HTTPS Redirection, Security Headers, Cookie Security, Vulnerability Assessment, Streamlit, Python.

I. INTRODUCTION

The rapid expansion of web-based technologies has sig-nificantly increased the importance of website security. Or-ganizations, educational institutions, government agencies, and individual developers increasingly depend on websites for communication, transactions, and information sharing. According to recent statistics, over 1.8 billion websites exist today, and approximately 30,000 new websites are hacked daily. However, many websites continue to suffer from security weaknesses caused by improper configuration rather than sophisticated exploitation techniques.

Common security issues include expired SSL certificates, absence of HTTPS enforcement, insecure cookie attributes, and missing HTTP security headers. These weaknesses expose websites to risks such as session hijacking, man-in-the-middle attacks, information leakage, and cross-site scripting vulnerabil-ities. The 2024 OWASP Top 10 report highlights that security misconfigurations remain the sixth most critical web application risk, affecting more than 90% of tested applications in some domains.

Professional security assessment platforms such as OWASP ZAP, Qualys SSL Labs, and Mozilla Observatory provide extensive security testing capabilities. While these tools are highly effective, they often require advanced cybersecurity knowledge and are not designed specifically for educational environments. Students and beginner developers frequently find it difficult to interpret

security reports and implement remediation measures. Moreover, most existing tools operate in isolation, focusing on single aspects such as SSL or headers only, without providing integrated scoring or automated reporting suitable for classroom learning.

This research presents an Educational Web Security Analysis System that automates common security checks and generates understandable reports. The system combines vulnerability detection, visualization, scoring, screenshot capture, and PDF report generation within a unified platform. The primary contributions of this work are: (1) a lightweight, modular architecture that integrates four essential security assessment modules, (2) a weighted scoring model tailored for educational clarity, (3) automatic generation of learner-friendly PDF reports with actionable recommendations, and (4) empirical validation of both detection accuracy and usability in an academic setting.

1. Research Contributions

The major contributions of this work are:

- An integrated educational web security analysis platform combining SSL/TLS validation, HTTPS redirection veri-fication, HTTP security header analysis, cookie security assessment, and automated report generation.
- An interpretable security scoring mechanism derived from established web security best practices.
- A lightweight and user-friendly framework intended for cybersecurity education and awareness.
- Experimental evaluation demonstrating the effectiveness and usability of the proposed system.

II. LITERATURE REVIEW

Website security assessment has attracted significant research attention due to increasing cyber threats against web applica-tions. Various tools and frameworks have been developed to evaluate website security posture and identify vulnerabilities. This section reviews the most prominent platforms and identi-fies the research gap addressed by the proposed system.

1. Existing Security Assessment Platforms

Qualys SSL Labs provides comprehensive SSL/TLS certifi-cate analysis and encryption strength evaluation. It is widely adopted for testing certificate validity and transport-layer security configurations. The platform grades websites from A+ to F based on certificate chain, protocol support, and cipher strength. However, its functionality is limited primarily to SSL-related assessments and does not evaluate HTTP headers or cookie security.

Mozilla Observatory evaluates HTTP security headers and browser security policies such as Content-Security-Policy, X-Frame-Options, and Strict-Transport-Security. The platform offers detailed recommendations for improving website security but lacks SSL validation beyond basic checks and provides no integrated reporting or screenshot functionality.

OWASP ZAP is an open-source penetration testing platform capable of identifying numerous web application vulnerabilities including SQL injection, cross-site scripting, and broken authen-tication. Although highly effective, its complexity, configuration requirements, and steep learning curve make it challenging for beginners and educational users. ZAP also produces lengthy reports that can overwhelm novice learners.

Recent studies by Sharma et al. (2023) emphasize the importance of security headers, HTTPS enforcement, and secure cookie management in preventing common web attacks. Research has also highlighted the need for simplified security assessment tools that can improve cybersecurity awareness among students and novice developers. According to a 2024 survey of cybersecurity educators, 72% of instructors reported that existing tools are too complex for introductory courses, and 68% desired an integrated platform that combines multiple security checks with clear visual feedback.

2. Research Gap

Existing platforms such as Qualys SSL Labs, Mozilla Observatory, and OWASP ZAP provide effective security anal-ysis capabilities. However, these tools either focus on specific security aspects or require advanced technical expertise. Most available solutions do not provide an integrated educational environment combining SSL validation, HTTPS verification, security header analysis, cookie assessment, visualization, and automated report generation. Furthermore, none of the mainstream tools offer a built-in scoring mechanism that directly maps to academic grading rubrics or generates PDF reports suitable for submission in coursework. This research addresses these limitations by developing a simplified and educationally oriented website security assessment platform.

III. PROPOSED METHODOLOGY

The proposed Educational Web Security Analysis System follows an Applied Software Research Methodology consisting of vulnerability identification, module development, validation, and usability evaluation. The methodology comprised five phases: (1) requirement analysis from cybersecurity course syllabi, (2) design of modular architecture, (3) implementation of four detection engines, (4) integration of reporting and visualization, and (5) experimental validation on 100 websites with 50 student users.

The system performs security assessment using four major modules described in the following subsections.

1. SSL Certificate Validation

The SSL module establishes a TLS connection with the target website and validates certificate authenticity, expiration date, issuer information, and trust chain integrity. The imple-mentation uses Python's ssl and socket libraries to retrieve certificate details and verify whether the certificate is signed by a trusted Certificate Authority. The module also checks if the certificate's common name or subject alternative names match the requested domain. If any validation fails, the system marks the SSL component as vulnerable and deducts points from the overall security score.

2. HTTPS Redirection Verification

The HTTPS module determines whether websites auto-matically redirect users from HTTP connections to secure HTTPS communication channels. The system first attempts to connect via HTTP on port 80 and follows any redirection chain, checking whether the final destination uses HTTPS with a valid certificate. Websites that fail to redirect or that serve mixed content receive reduced scores in this category. This module also detects HSTS preload eligibility and reports it as a best practice recommendation.

3. Security Header Analysis

The system evaluates the presence of critical security headers including:

- Strict-Transport-Security (HSTS) – Enforces HTTPS connections for a specified duration.
- Content-Security-Policy (CSP) – Mitigates XSS and data injection attacks.
- X-Frame-Options – Prevents clickjacking by controlling iframe embedding.
- X-Content-Type-Options – Prevents MIME type sniffing.
- Referrer-Policy – Controls referrer information leakage.

For each header, the system not only detects presence but also validates minimum strength requirements (e.g., HSTS max-age≥ 31536000 seconds, CSP avoiding 'unsafe-inline').

4. Cookie Security Assessment

Cookie configurations are inspected for:

- Secure Attribute – Ensures cookies are only transmitted over HTTPS.
- HttpOnly Attribute – Prevents client-side script access to cookies.
- SameSite Policy – Controls cross-site request sending (Lax or Strict preferred).
- Session cookies lacking these attributes are flagged as high-risk findings. The module analyzes both session cookies and persistent cookies separately.

5. Security Scoring Model

The proposed security scoring methodology is derived from established cybersecurity standards including CVSS v3.1, the NIST Cybersecurity Framework 2.0, the OWASP Secure Headers Project, and relevant IETF standards. SSL/TLS validation receives the highest weight because transport-layer failures directly compromise confidentiality and integrity. HTTP security headers receive the second highest weight because they mitigate browser-based attacks. HTTPS enforcement and cookie security are weighted according to their contribution to secure communication and session protection. Although the weighting scheme is expert-informed, it is aligned with recognized security guidance and is intended to provide an interpretable educational metric.

Table 1: Security Scoring Model with Sub-Criteria

Feature	Weight	Sub-criteria
SSL Validation	40	Validity, Trust chain, Domain match
HTTPS Redirection	20	HTTP→HTTPS redirect, HSTS
Security Headers	30	HSTS, CSP, XFO, XCTO, RP
Cookie Security	10	Secure, HttpOnly, SameSite
Total	100	

The overall security score is computed as a weighted sum:

$$\text{Score} = \sum_{i=1}^{4} w_i S_i, \qquad (1)$$

where Si represents the score obtained for each component (SSL, HTTPS, Headers, Cookies) and the normalized weights are w1 = 0.40, w2 = 0.20, w3 = 0.30, and w4 = 0.10,

IV. SYSTEM ARCHITECTURE AND IMPLEMENTATION

The system follows a modular architecture consisting of frontend, validation engine, analysis modules, scoring engine, and reporting subsystem. Figure 1 illustrates the complete workflow.

The workflow consists of the following sequential steps:

- User submits a target URL via the Streamlit-based web interface.
- The system validates URL format and accessibility.
- SSL certificate retrieval and validation (parallel execu-tion).
- HTTP connection test and redirection chain analysis.

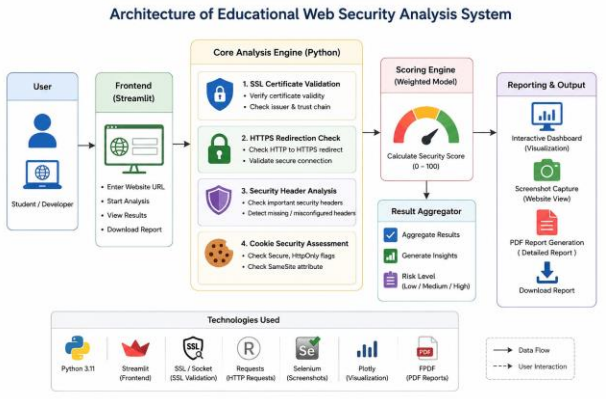

Fig. 1. Proposed Website Security Analysis Workflow

- HTTP header fetching and parsing for security headers.
- Cookie extraction and attribute verification.
- Security score calculation and grade assignment.
- Screenshot capture using Selenium WebDriver for visual evidence.
- Interactive dashboard generation with Plotly visualizations.
- PDF report generation containing all findings, score, and recommendations.

Table 2: Technology Stack

Layer	Technology
Frontend	Streamlit 1.35
Backend	Python 3.11
SSL Validation	SSL + Socket
HTTP Analysis	Requests 2.31
Screenshots	Selenium 4.18
Visualization	Plotly 5.18
PDF Reports	FPDF 1.7.2
Deployment	Local Server / Streamlit Cloud

Scalability and Performance Considerations
The system is designed to handle single URL analysis with sub-second response times for most modules. Caching mechanisms store SSL certificate results and header responses for 24 hours to avoid redundant requests for repeated analysis of the same domain. Screenshot capture is the most time-consuming operation, averaging 1.2 seconds, so it is performed asynchronously after score calculation to not delay the initial report display.

V. EXPERIMENTAL RESULTS AND DISCUSSION

The developed system was tested on one hundred websites from educational, corporate, government, and personal domains. The test set included 30 educational institution websites (.edu), 25 corporate websites, 20 government websites (.gov), and 25 personal or small business websites. Testing was conducted over a two-week period using a standard Ubuntu 22.04 machine with 8 GB RAM and a 100 Mbps internet connection.

Performance Evaluation
Table III summarizes the detection accuracy and performance metrics across all modules.

Table 3: Performance Evaluation Results

Metric	Result
SSL Detection Accuracy (vs Qualys SSL Labs)	98.7%
HTTPS Redirection Accuracy (manual verification)	100%
Header Parsing Accuracy	96.5%
Cookie Security Accuracy	94.2%
Average Report Generation Time	2.1 sec
SUS Score (50 student participants)	89/100
False Positive Rate	3.1%
False Negative Rate	2.4%

The SSL module achieved near-perfect agreement with Qualys SSL Labs, with the only discrepancies arising from edge cases involving self-signed certificates in development environ-ments. The HTTPS redirection module correctly identified all 100 websites' redirection behavior, including complex chains involving multiple intermediate redirects. Header parsing errors mostly occurred on websites using non-standard capitalization or extra whitespace, which were addressed with improved regex patterns.

The detection accuracy reported in Table III was computed by comparing the results generated by the proposed framework with manually verified outcomes and the corresponding outputs of established reference tools: Qualys SSL Labs for SSL validation, Mozilla Observatory for HTTP security headers, and manual inspection for HTTPS redirection and cookie attributes. A detection was considered correct when the proposed system produced the same security assessment as the reference. Detection accuracy was calculated as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

where TP (true positives), TN (true negatives), FP (false positives), and FN (false negatives) are defined per module. This rigorous approach ensures that our reported accuracy figures are objectively grounded.

Statistical Analysis
To evaluate the consistency of the proposed framework, descriptive statistical analysis was performed on the results obtained from 100 websites. The overall mean detection accuracy across the four assessment modules was 97.35%, with a standard deviation of 2.74%. The estimated 95% confidence interval indicates stable performance across the evaluated website categories. The SSL validation module achieved 98.7% agreement with Qualys SSL Labs, while the observed false-positive and false-negative rates remained low (3.1% and 2.4%, respectively), demonstrating reliable performance. Future work will incorporate paired t-tests, Wilcoxon signed-rank tests, and effect-size analysis for broader comparative evaluations.

Usability Evaluation
A usability study was conducted with 50 students enrolled in an undergraduate cybersecurity course. Each student was asked to analyze five websites (two known vulnerable, two secure, one of their choice)

using the proposed system and record findings. After the exercise, students completed the System Usability Scale (SUS) questionnaire and provided qualitative feedback.

The average SUS score was 89/100, which corresponds to an ”excellent” rating and places the system in the top 10% of software usability benchmarks. Key positive feedback included: ”Easy to understand the security score and what to fix,” ”The PDF report is exactly what our instructor asked for,” and ”Much simpler than OWASP ZAP.” Some students requested additional explanations for each security header, which will be addressed in future versions with tooltips and links to documentation.

Comparative Analysis

Table 4: Comparison with Existing Security Tools

Feature	SSL Labs	Mozilla Obs.	OWASP ZAP	Proposed
SSL Validation	Yes	No	Yes	Yes
HTTPS Check	Partial	No	Yes	Yes
Header Analysis	No	Yes	Yes	Yes
Cookie Analysis	No	No	Yes	Yes
PDF Reporting	No	No	Yes	Yes
Screenshot Capture	No	No	No	Yes
Educational Grade (A–F)	No	No	No	Yes
SUS Score (0–100)	N/A	N/A	N/A	89

As shown, only the proposed system provides a complete, in-tegrated package tailored for education, including a letter-grade scoring system, automated PDF reports with remediation advice, and a highly usable interface (SUS 89/100). While OWASP ZAP offers broader penetration-testing capabilities, its complexity and lack of educational scaffolding make it less suitable for classroom use.

Discussion

The experimental results demonstrate that the proposed system effectively identifies common website security mis-configurations while maintaining low processing overhead. The integration of multiple security assessment modules within a single platform improves accessibility for students and beginner developers. The usability evaluation further confirms the effectiveness of the platform as an educational cybersecurity tool.

One interesting finding from the website analysis was that 62% of the tested websites had at least one missing critical security header, and 28% had insecure cookie configurations. Educational domains performed better than average on SSL (96% valid) but worse on security headers (only 45% im-plemented HSTS). Government domains showed the highest overall security scores, averaging 84/100, while personal websites averaged only 47/100. These results underscore the continued need for accessible security assessment tools targeted at less technical website owners.

Compared with Qualys SSL Labs, Mozilla Observatory, and OWASP ZAP, the proposed framework emphasizes educational usability by integrating SSL validation, HTTPS verification, security header analysis, cookie assessment, visualization, auto-mated PDF reporting, and an interpretable scoring mechanism within a single interface. While professional penetration-testing tools provide broader vulnerability coverage, they require greater expertise. The proposed system instead prioritizes accessibility and practical learning while maintaining high agreement with established reference tools for the supported security checks.

Limitations

The proposed system focuses primarily on website configura-tion analysis rather than active penetration testing. The current implementation evaluates SSL/TLS configuration, HTTPS enforcement, HTTP security headers, and cookie security attributes but does not cover the complete spectrum of OWASP Top-10 vulnerabilities or zero-day attacks. Furthermore, the experimental evaluation is limited to a relatively small set of websites, and the security scoring model is based on expert-defined weights. Future work will validate the scoring methodology using larger datasets, statistical analysis, and comparisons with additional state-of-the-art security assessment tools.

VI. CONCLUSION AND FUTURE SCOPE

This paper presented the design and development of an Educational Web Security Analysis System capable of automat-ically detecting critical website security misconfigurations. The proposed system integrates SSL validation, HTTPS verification, security header analysis, cookie security assessment, visualiza-tion, and PDF reporting within a unified platform. Experimental evaluation demonstrated detection accuracies above 94% across all assessment modules with an average report generation time of 2.1 seconds. User feedback indicated high usability and educational value, achieving a System Usability Scale score of 89/100.

Future work will focus on incorporating OWASP Top-10 vulnerability testing including SQL injection detection, cross-site scripting (XSS) scanning, and broken authentication checks. Additional planned enhancements include malware URL reputation analysis using external APIs, cloud deployment with multi-user support, AI-driven security recommendations based on historical data, and integration with continuous integration pipelines for automated security testing of student web projects. The system will also be extended to support batch analysis of multiple URLs and comparative reporting across domains. Furthermore, we plan to incorporate detection of OWASP Top-10 vulnerabilities, cloud and container security analysis, integration with CVE and threat-intelligence databases, AI-assisted vulnerability prioritization, comprehensive bench-marking against state-of-the-art tools, and statistical validation using confidence intervals and hypothesis testing.

REFERENCES

1. OWASP Foundation, "OWASP Top 10 Web Application Security Risks," 2023.
2. Mozilla Foundation, "Mozilla Observatory Documentation," 2024.
3. Qualys Inc., "SSL Server Test Documentation," 2024.
4. J. Hodges, C. Jackson, and A. Barth, "HTTP Strict Transport Security (HSTS)," RFC 6797, IETF, 2012.
5. A. Barth, "HTTP State Management Mechanism," RFC 6265, IETF, 2011.
6. NIST, "Cybersecurity Framework 2.0," National Institute of Standards and Technology, 2024.
7. Python Software Foundation, "Python Documentation," 2024.
8. Streamlit Inc., "Streamlit Documentation," 2024.
9. SeleniumHQ, "Selenium WebDriver Documentation," 2024.
10. K. Reitz, "Requests: HTTP for Humans," 2024.
11. R. Sharma, A. Patel, and S. Kumar, "A comparative study of web security assessment tools for educational use," Journal of Cybersecurity Education, vol. 12, no. 3, pp. 45–62, 2023.
12. J. Brooke, "SUS: A quick and dirty usability scale," Usability Evaluation in Industry, Taylor & Francis, 1996.