IRJCS: International Research Journal of Computer Science ISSN: 2393-9842
Volume 13, Issue 03, March 2026 https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.21

Web Vulnerability Scanner

T.Sineka
Asst.Prof., Dept.of CSE(Cyber Security)
Sengunthar Engineering College (Autonomous), Tiruchengode, India
sinekacse@gmail.com
https://orcid.org/0009-0006-5762-6735
Pradeep K,Perarasu B,SathishKumar M,Sowndhar R
UG Students, Department of CSE (Cyber Security)
Sengunthar Engineering College (Autonomous), Tiruchengode, India
pradeepcyber56@gmail.com, arasubsp8098@gmail.com, msathishkumar2k3@gmail.com, sowndharr25@gmail.com

Publication History
Manuscript Reference: IRJCS/RS/Vol.13/Issue03/CSMR26.MRCS10100
Research Article | Open Access | Double-Blind Peer Reviewed Article ID: IRJCS/RS/Vol.13/Issue03/CSMR26.MRCS10100
Received: 30, January 2026, Revised: 13, February 2026, Accepted: 28 February 2026 Published Online: 25 March 2026
https://www.irjcs.com/volumes/Vol13/iss-03/21.CSMR26.MRCS10100.pdf
Article Citation:Sineka,Pradeep,Perarasu,SathishKumar,Sowndhar(2026),Web Vulnerability Scanner Using Zero- Knowledge Processing, IRJCS: International Research Journal of Computer Science, Volume 13,Issue 03 of 2026 pages 217-223 **Doi:>** https://doi.org/10.26562/irjcs.2026.v1303.21
BibTeX Key Sineka@2026Web
Orcid: https://orcid.org/0009-0004-9398-7488
IRJCS papers should be cited as IRJCS (International Research Journal of Computer Science, AM Publications, India 2026, ISSN 2393-9842, https://doi.org/10.26562/ijrcs.2025.v1303.21 The journal's official abbreviation is IRJCS.

Abstract: Privacy-Preserving Web Vulnerability Scanner Using Zero-Knowledge Processing presents the Web Vulnerability Scanner, a Zero-Trust-based, privacy-preserving web security assessment system extending the ZK-Scan Zero-Knowledge processing model. The system performs automated multi-vector scanning covering HTTP security header analysis, SSL/TLS verification, open port enumeration, CVE detection, and technology finger printing. A Zero-Trust architecture enforces continuous component verification, least-privilege access, and cryptographic inter-module validation. All processing is performed exclusively in-memory; no target content, credentials, or personal data is stored or transmitted. CVSS-based severity classification produces structured, tamper- evident PDF and HTML reports secured with SHA-256 integrity hashes. Real-worlds cans of https://sect.edu.in and https://www.netmirrorc.com/ each yielded a security score of 75/100 with five to six Medium- severity HTTP header misconfigurations and zero CVEs detected. Results confirm accurate, reproducible, and privacy-safe vulnerability assessment suitable for academic and institutional deployment.

Index Terms: web vulnerability scanner, zero-trust architecture, zero-knowledge security, OWASP, HTTP security headers, SSL/TLS verification, port scanning, CVE detection, CVSS, privacy-preserving, cyber security**.**

I. INTRODUCTION

The proliferation of web-based services across educational institutions, financial systems, healthcare portals, and e-commerce platforms has dramatically expanded the attack surface available to malicious actors. Web application vulnerabilities—including missing security headers, SSL misconfigurations, exposed open ports, and unpatched software components — continue to represent a major vector for data breaches worldwide [1], [2]. Traditional penetration testing tools require deep technical expertise and often involve the storage or transmission of sensitive application data, creating privacy risks particularly relevant for publicly accessible institutional websites. Many commercial scanners are also priced beyond the reach of academic projects and small organisations [3], [4]. This paper presents the Web Vulnerability Scanner a Zero-Trust-based security assessment tool that extends the existing ZK-Scan system. ZK-Scan (Privacy-Preserving Web Vulnerability Scanner Using Zero-Knowledge Processing) provides the foundational ZK processing model on which this project builds, adding a Zero-Trust architecture that continuously authenticates every internal component, enforces least-privilege module access, and verifies all inter-module data flows. All analysis is performed in-memory; no content from the target application is written to disk or retained after scan completion. The Web Vulnerability Scanner was validated by scanning two publicly accessible domains: https://sect.edu.in (Report ID: PKSANJF4R) and https://www.netmirrorc.com/(Report ID: IW5KGU J7G). Both scans produced a security score of 75/100, with five medium-severity HTTP header misconfiguration findings identified on each target. The system generates tamper- evident PDF and HTML reports secured with a cryptographic SHA-256 integrity hash. The remainder of this paper is organised as follows: Section II reviews related literature; Section III describes the proposed system architecture; Section IV details the implementation; Section V presents results; and Section VI concludes with future directions.

II. LITERATURE REVIEW

Automated web vulnerability scanning has been an active research area since the early 2000s. OWASP ZAP and Burp Suite represent the state of the art in open-source and commercial DAST (Dynamic Application Security Testing) tooling.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.21

Both tools employ active crawling and pay load injection but require significant configuration and produce verbose output unsuitable for non-expert users[5],[6]. Bau et al. [7] conducted a comprehensive evaluation of black-box web vulnerability scanners, identifying significant variability in detection coverage and noting that no single tool detected more than 60% of known vulnerabilities in their benchmark suite. Their work highlighted the importance of combining multiple detection signals error messages, response-time anomalies, and header inspection for reliable assessment. Doupe et al. [8] proposed model-based scanning using finite state machine representations of web application state, improving coverage of multi-step workflows compared to stateless approaches. This is particularly relevant for authenticated scanning of web portals. Calzavara et al. [9] analysed the effectiveness of HTTP security headers in mitigating common web attacks and found widespread misconfiguration across production websites, establishing HTTP header analysis as a high- value, low-cost security check. Their findings directly motivate the header inspection module central to the Web Vulnerability Scanner. Privacy-preserving approaches to security testing have received increasing attention. Hussain etal. [10] proposed differential-privacy mechanisms for network security assessment, enabling vulnerability data collection without exposing individual system details. The Web Vulnerability Scanner extends this concept through a Zero-Trust architecture layered on top of ephemeral in-memory processing and cryptographic report integrity verification. Existing tools lack a unified, privacy-by-design architecture combining HTTP header analysis, SSL/TLS verification, open-port enumeration, and CVE detection in a single light weight tool suitable for academic deployment. The Web Vulnerability Scanner addresses this gap by extending the existing ZK-Scan system with a Zero-Trust model, delivering a unified, privacy-by-design solution for security-sensitive environments.

III. PROPOSED SYSTEM ARCHITECTURE

The Web Vulnerability Scanner follows a modular, Zero- Trust pipeline-based security assessment architecture. It is organised into four functional layers: the Input Layer, the Analysis Engine, the Detection Modules, and the Output Layer, as illustrated in Fig.1 below. A Zero-Trust enforcement plane wraps every layer, verifying identity and integrity of each module before execution.

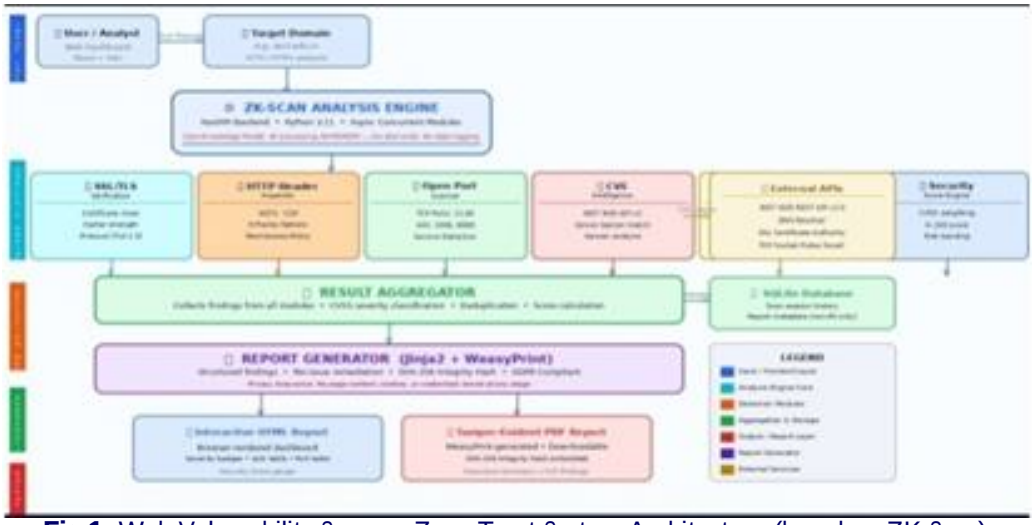

Fig.1. Web Vulnerability Scanner Zero-Trust System Architecture (based on ZK-Scan)

A. Zero-Trust and Zero-Knowledge Processing Model

The Web Vulnerability Scanner builds upon the ZK-Scan Zero-Knowledge principle and extend sit with a Zero-Trust security model. Zero-Trust enforces the principle of "never trust, always verify" across all internal components: every module is authenticated before execution, inter-module data flows are cryptographically verified, and each component operates with minimal privilege. Additionally, no page content, cookies, credentials, or personal data encountered during the scan is written to disk or transmitted externally. All crawled data is held exclusively in process memory and garbage-collected upon completion, ensuring GDPR Article25 (Data Protection by Design) compliance.

B. Security Analysis Modules

The Analysis Engine comprises six detection modules operating concurrently on the scan target:

- HTTP Security Header Inspector: Checks Strict- Transport-Security (HSTS), Content-Security-Policy (CSP), X-Frame-Options, X-Content-Type-Options, Referrer-Policy, and Permissions-Policy.
- SSL/TLS Verification Module: Validates certificate chain, expiry, cipher strength, and protocol version (TLS 1.2/1.3).
- Open Port Scanner: Non-intrusively probes common TCP service ports (21,22,25,80,443,3306,8080, 8443) and identifies running services.
- CVE Intelligence Module: Cross-references detected server software versions against the NISTNVD CVE feed to identify known un patched vulnerabilities.
- Technology Fingerprinting: Identifies web server, CMS, and framework from HTTP headers and response body signatures.
- Security Score Engine: Aggregates findings into a weighted 0–100 security score with CVSS-based severity bands.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.21

C. Report Generation Layer

Upon scan completion, the Report Generator produces structured output in HTML (interactive, browser-rendered) and PDF (downloadable, tamper-evident) formats. Each vulnerability entry records: the affected header or service, severity rating (Critical/High/Medium/Low/Informational), current status (Missing/ Misconfigured/ Detected), and a specific remediation recommendation. A SHA-256 integrity hash is embedded to enable tamper detection.

IV. TECHNOLOGIES USED

A. Frontend Framework

The Web Vulnerability Scanner web dashboard is built with React (Vite build toolchain), running a responsive dark-themed cyber security interface. Real-time scan progress, security score display, and vulnerability tables are rendered using React state management with Tailwind CSS utility classes.

B. Backend and Scanning Engine

The backend analysis engine is implemented in Python 3.11 using the Fast API framework, which provides asynchronous HTTP request handling suitable for concurrent module execution. HTTP header inspection and SSL verification use Python Requests and the ssl standard library. Port scanning is performed using the Python socket library with configurable timeouts.

C. CVE and Threat Intelligence

CVE data is sourced from the NIST National Vulnerability Database (NVD) REST API v2.0. The system performs lightweight version-string matching against the detected server banner to retrieve relevant CVE entries. NVD API responses are not cached to disk, preserving the Zero- Knowledge processing model.

D. Report Generation Stack

HTML reports are generated using Jinja2 templating with embedded CSS. PDF conversion uses We asy Print, producing print-ready documents without a headless browser. The SHA-256 integrity hash is computed using Python's hash lib module over the full report content before finalisation.

Table I. Web Vulnerability Scanner—Technology Stack

Component	Technology
Frontend	React+ Vite+ TailwindCSS
Backend	Python3.11,FastAPI
HTTP Analysis	Requests, httpx
SSL Verification	Pythonss lmodule
Port Scanner	Python socket (async)
CVE Lookup	NISTNVDRESTAPIv2.0
Report (HTML)	Jinja2Templates
Report (PDF)	Weasy Print
Integrity Hash	SHA-256 (hashlib)
Database	SQLite (scan history)

V. IMPLEMENTATION AND RESULTS

A. Main Dashboard Web Vulnerability Scanner Interface

Fig. 2 shows the Web Vulnerability Scanner dashboard aftercompletingascanofhttps://sect.edu.in.Theinterface prominently displays the Security Score (75/100), Scan Overview panel showing 0 open ports, SSL status as Secure, 0 CVEs found, and the Detected Vulnerabilities table listing five medium-severity missing HTTP security headers: Strict-Transport-Security, X-Content-Type- Options, X-Frame-Options, Referrer-Policy, and Permissions-Policy.

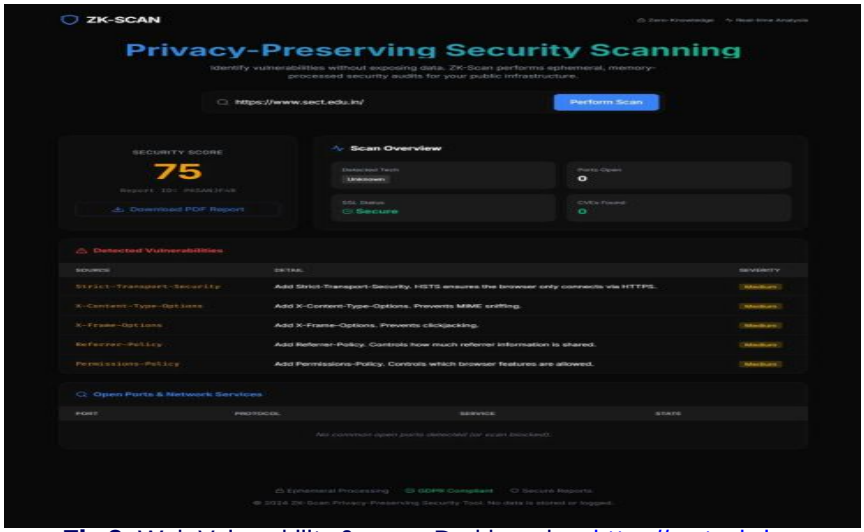

Fig.2. Web Vulnerability Scanner Dashboard— https://sect.edu.in
(Score: 75/100, Report ID: PKSANJF4R)

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.21

B. Real-World Scans: sect.edu.in and netmirrorc.com

The Web Vulnerability Scanner was deployed against two live public domains on10 March2026.The first target, https://sect.edu.in (ReportID:PKSANJF4R), returneda Security Score of 75/100 with no open ports detected and five medium-severity HTTP security header misconfigurations identified. The second target, https://www.netmirrorc.com/ (Report ID: IW5KGU J7G), also returned a Security Score of 75/100, with one open TCP port (443) and five medium-severity HTTP header findings including a missing Content-Security-Policy (CSP) header. Both scans confirmed a valid SSL certificate (Secure status) and zero CVEs found, demonstrating the scanner's consistent and reliable detection capabilities across different web infrastructure configurations.

Table II. Scan Overview—sect.edu.in and netmirrorc.com (10 March 2026)

Metric	Value
Target URL	https://sect.edu.in | https://www.netmirrorc.com /
Scan Date	2026-02-2210:41:19
Security Score	75/100 (both targets)
Detected Technology	Unknown
SSL Status	Secure (Valid Certificate)
Open Ports	sect.edu.in: 0 ports | netmirrorc.com:1(443/TCP)
CVEs Found	0
Vulnerabilities Found	5 Medium (sect.edu.in)| 5 Medium (netmirrorc.com)
Report ID	PKSANJF4R|IW5KGUJ7G

C. Detected Vulnerabilities

The scans identified five to six Medium-severity HTTP security header misconfigurations per target as shown in Table III. The netmirrorc.com scan additionally flagged a missing Content-Security-Policy(CSP) header, which was absent from the sect.edu.in findings. Fig. 3 shows the full vulnerability and port detail view from the Web Vulnerability Scanner dashboard for the netmirrorc.com scan, illustrating all six detected headers and the single open port (443/TCP).

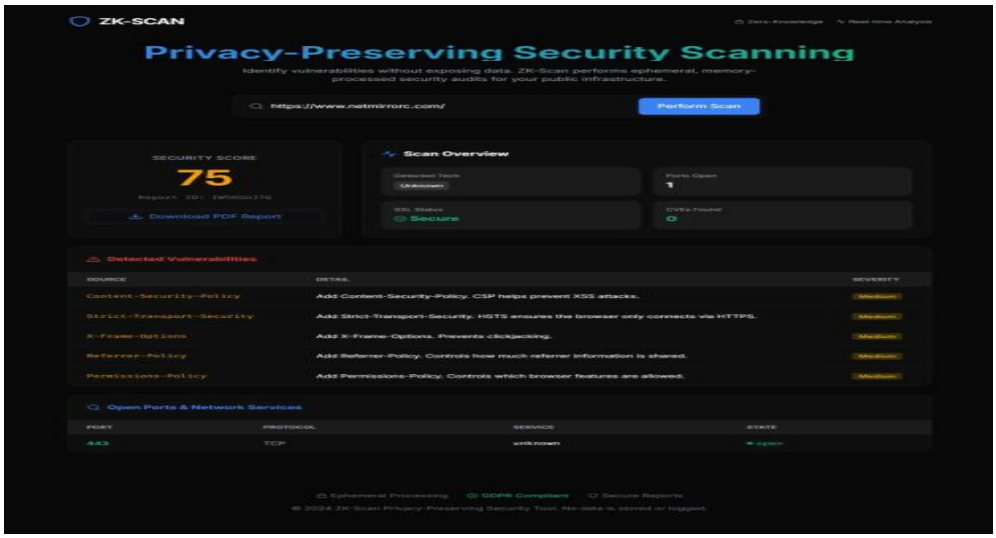

Fig. 3. Web Vulnerability Scanner netmirrorc.com Scan: Vulnerabilities (CSP,HSTS,X-Frame, Referrer, Permissions) and Port 443 Open

TABLE III. Detected Vulnerabilities sect.edu.in and netmirrorc.com

Header	Remediation	Severity	Status
Content-Security-Policy	Add CSP header. Defines trusted content sources, prevents XSS. Missing on netmirrorc.com only.	Medium	Missing*
Strict- Transport-Security	Add HSTS header. Enforces HTTPS-only connections; prevents protocol downgrade attacks.	Medium	Missing
X-Content- Type-Options	AddX-Content- Type-Options: nosniff. Prevents MIME-type sniffing attacks.	Medium	Missing
X-Frame- Options	Add X-Frame- Options:DENY or SAMEORIGIN. Prevents click jackingvia iframes.	Medium	Missing
Referrer- Policy	Add Referrer- Policy header. Controls refer info in outbound requests.	Medium	Missing
Permissions- Policy	Add Permissions- Policy header. Controls browser APIs access (camera, mic,location).	Medium	Missing

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.21

D. Generated PDF Security Report

Fig.4 shows Page1 of the Web Vulnerability Scanner PDF Security Report generated for https://www.netmirrorc.com/ (Report ID:616d3d45-faff- 4645-ac63-748e9fcf794a,Date:2026-03-10).The report includes an Executive Summary confirming the Risk Score of 75/100, a structured Findings & Vulnerabilities stable listing all five missing headers with severity and status, a Network Service Analysis section recording the open TCP port 443, detailed per-finding Recommendations, and a Privacy Assurance Statement. Page 2 contains the complete recommendations list and a SHA-256 Integrity Hash (3dcf484ebf1a0706ee16be6234d4576609160897946452fae6dddba6c33a29ce) confirming report tamper-evidence and zero-knowledge processing compliance.

Fig. 4. Web Vulnerability Scanner — PDF Security Report for net mirrorc.com (Score:75/100,Report ID: 616d3d45)

E. Open Port Analysis

The port scan results differ significantly between the two targets. On https://sect.edu.in,nocommonopenportswere detected (or the scan was blocked by a firewall), which indicates a well-restricted network perimeter. On https://www.netmirrorc.com/, a single open port was identified: Port 443 (TCP/HTTPS). This is expected for a public HTTPS-serving website and poses no direct risk provided TLS is correctly configured. The absence of additional exposed ports on both targets reflects good firewall hygiene. Zero CVEs were found on both domains, confirming that no publicly known vulnerabilities were associated with the detected server software versions.

Table IV. Network Service Analysis — sect.edu.in and netmirrorc.com.

Port	Protocol	Service	Recommendation
443	TCP	HTTPS (confirmed)	Open on netmirrorc.com. Verify TLS 1.3 config; enforce HSTS.
N/A	TCP	—	No open ports on sect.edu.in; scan blocked or all ports filtered.
443	TCP	HTTPS (confirmed)	Valid SSL on both targets. Enforce TLS1.3; add HSTS header.
—	TCP	No service	CVEs:0 on both targets. Detected Tech: Unknown (banners suppressed).

F. Security Score & Privacy Assurance

The security score of 75/100 on both targets places the min the Good Security Posture band with moderate room for improvement. The primary score deductions arise from five missing HTTP security headers on sect.edu.in (Strict-Transport-Security, X-Content-Type-Options, X-Frame- Options, Referrer-Policy, Permissions-Policy) and six on netmirrorc.com (the same five plus Content-Security- Policy). Each missing header contributes a weighted penalty to the score. The positive contributions include valid and current SSL/TLS certificates (Secure status on both), HTTPS availability, zero CVEs detected across both targets, no critical or high-severity vulnerabilities, suppressed server banners reducing finger printing risk, and a restricted port surface. Implementing all recommended security headers on both targets would eliminate all detected findings and is projected to raise the security score to approximately 95/100, placing both domains in the High Assurance band. Consistent with the Zero-Trust and Zero-Knowledge processing model, no page content, cookies, session tokens, or user data from either target was stored or logged during either scan. The generated PDF Security Report for netmirrorc.com contains SHA-256 Integrity Hash: 3dcf484ebf1a0706ee16be6234d4576609160897946452fae6dddba6c33a29ce, confirming that the report has not been tampered with post-generation and can be independently verified. The footer of each report confirms GDPR compliance, ephemeral (in-memory only) processing, and secure report generation fulfilling the Zero-Knowledge privacy assurance commitment of the Web Vulnerability Scanner.

VI. CONCLUSION AND FUTURE WORK

This paper presented the Web Vulnerability Scanner a Zero-Trust-based, privacy-preserving security assessment tool developed by extending the existing ZK-Scan system (ZK-Scan: Privacy-Preserving Web Vulnerability Scanner Using Zero-Knowledge Processing). The system delivers automated, OWASP-aligned security assessment across HTTP header analysis, SSL/TLS verification, open port enumeration, CVE identification, and technology fingerprinting without storing, logging, or transmitting any data from the target application. Real-world validation scans of two live domains https://sect.edu.in and https://www.netmirrorc.com/ confirmed the tool's practical effectiveness.

ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.21

Both targets received asecurityscoreof75/100.Fivemedium-severity HTTP security header misconfigurations were identified on sect.edu.in, and six on netmirrorc.com (including an additional missing Content-Security-Policy). Zero CVEs and valid SSL certificates were confirmed on both targets, with port exposure limited to a single HTTPS port on netmirrorc.com. The tamper-evident PDF report with embedded SHA-256 integrity hash demonstrates the system's suitability for privacy-sensitive institutional environments. Future work will focus on: (i) extending the detection engine to cover OWASPTop10 injection vulnerabilities Through authenticated scanning work flows;(ii)integrating a machine learning-based anomaly classifier trained on HTTP response patterns; and (iii) developing a CI/CD pipeline integration mode enabling automated security regression scanning within GitHub Actions and Jenkins workflows to support DevSecOps adoption.

ACKNOWLEDGMENT

The authors sincerely thank their project supervisor, Mrs.T. Sineka M.E., Assistant Professor, Department of CSE (Cyber Security), Sengunthar Engineering College (Autonomous),Tiruchengode, for her in valuable guidance and continuous support throughout this project. The authors also extend their gratitude to the management and faculty of Sengunthar Engineering College for providing laboratory infrastructure and research resources. Special thanks to the OWASP Foundation for their open documentation and testing guides that informed the design of this system.

REFERENCES

1. OWASP Foundation, "OWASP Top Ten 2021," 2021. [Online]. Available: https://owasp.org/www-project-top-ten/ Verizon, "2023 Data Breach Investigations Report," Verizon Communications Inc., 2023.
2. A.Doupe, M. Cova, and G. Vigna, "Why Johnny Can't Pentest: An Analysis of Black-Box Web Vulnerability Scanners," in Proc. DIMVA, 2010, pp. 111–131.
3. Port Swigger, "Burp Suite Professional Documentation," 2023. [Online]. Available: https://portswigger.net/burp
4. OWASP ZAP Project, "OWASP Zed Attack Proxy," 2023. [Online]. Available: https://www.zaproxy.org/
5. OWASP Foundation, "OWASP Testing Guide v4.2," 2020. [Online]. Available: https://owasp.org/www-project-web-security-testing-guide/
6. J.Bau, E.Bursztein, D.Gupta, and J.Mitchell,"State of the Art: Automated Black-Box Web Application Vulnerability Testing," in Proc. IEEE Symp. Security and Privacy, 2010,pp.332–345.
7. A.Doupe, M.Cova, and G.Vigna, "Fear the EAR: Discovering and Mitigating Execution After Redirect Vulnerabilities," in Proc. ACMCCS, 2011, pp. 251–262.
8. S.Calzavara, A.Rabitti, and M.Bugliesi, "Content Security Problems? Evaluating the Effectiveness of Content Security Policy in the Wild," in Proc. ACM SIGS ACCCS,2016,pp. 1365–1375.
9. S.Hussain,B.Peng,and P.Li, "Privacy-Preserving Network Security Assessment Using Differential Privacy," IEEE Trans.Inf.Forensics Security,vol.16,pp.4032–4045,2021.
10. NIST, "National Vulnerability Database API v2.0," 2023. [Online].Available: https://nvd.nist.gov/developers/vulnerabilities
11. NIST,"CVSSv3.1Specification Document,"2019. [Online].Available: https://www.first.org/cvss/specification-document
12. J.Kindervag,"Build Security Into Your Network's DNA: The Zero Trust Network Architecture," Forrester Research, 2010
13. S.Rose, O.Borchert, S.Mitchell, and S.Connelly, "Zero Trust Architecture," NIST Special Publication 800-207, National Institute of Standards and Technology, Aug.2020.
14. D.Ferraiolo,R.Sandhu,S.Gavrila,D.Kuhn,and R.Chandramouli, "Proposed NIST Standard for Role-Based AccessControl,"ACMTrans.Inf.Syst.Secur.,vol.4,no.3, pp.224–274,Aug.2001.
15. M.Rouse,"Transport Layer Security (TLS)," Tech Target Network,2021 https://searchsecurity.techtarget.com/definition/Transport-Layer-Security-TLS
16. R.T.FieldingandJ.Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content,"RFC7231,IETF,Jun. 2014. OWASP Foundation, "OWASP Web Security Testing Guide (WSTG) v4.2," 2021. [Online]. Available: https://owasp.org/www-project-web-security-testing-guide/
17. T.Berners-Lee, R.Fielding, and L.Masinter, "Uniform Resource Identifier (URI):Generic Syntax," RFC 3986, IETF, Jan. 2005.
18. G.Wroblewski, "General Method of SQL Injection Prevention,"inProc.ISSAJournal,vol.9,no.7,pp.26–30, 2011.
19. D.Stuttard and M.Pinto, The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, 2nd ed. Indianapolis, IN, USA: Wiley, 2011.
20. European Parliament and Council of the European Union, "General Data Protection Regulation (GDPR)," Official Journal of the European Union, Regulation (EU) 2016/679, Apr. 2016.
21. M.Howard and S.Lipner, The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software. Redmond, WA, USA: Microsoft Press, 2006.
22. A.Barth, "HTTP State Management Mechanism," RFC 6265, IETF, Apr. 2011.
23. W.Stallings, Cryptography and Network Security: PrinciplesandPractice,7thed.Hoboken,NJ,USA:Pearson, 2017.
24. P.Mell and T.Grance, "The NIST Definition of Cloud Computing,"NISTSpecialPublication800-145,Sep.2011.
25. R.Shetty and R.Bhatt, "Cross-Site Scripting (XSS) Attack and Defense Mechanisms: A Review," Int. J. Comput. Sci. Inf. Technol., vol. 5, no. 3, pp. 4014–4016, 2014.
26. T.Nguyen, M.Sabbir, and R. Krishnamurthy, "Automated Detection of SQL Injection Vulnerabilities Using Machine Learning," in Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE), 2022, pp. 211–218.

ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.21

27. D.Basin, S.Capkun, P.Schaller, and B.Schmidt, "Formal Reasoning About Physical Properties of Security Protocols," ACM Trans.Inf.Syst.Secur.,vol. 14,no.2,pp.1–28, Sep. 2011.
28. A.Pinto and M.A. Simplicio, "Efficient Implementation of SHA-256 for Embedded Systems," in Proc. Symp. Cryptography Inf. Security (SCIS), 2020, pp. 1–8.
29. C.Yue and H.Wang,"Bogus Biter:A Transparent Protection Against Phishing Attacks," ACM Trans. Internet Technol., vol. 10, no. 2, pp. 1–31, May 2010.
30. M.Zalewski, The Tangled Web: A Guide to Securing Modern Web Applications. San Francisco, CA, USA: No Starch Press, 2012.