# Detection of Real Time Web Application Threat Using Machine Learning

Dr. M. Rahimal Beevi[1], Ms. K. Moniga[2]
[1]*Assistant Professor, Department of MCA Mohamed Sathak Engineering College*
[2] *Final MCA, Mohamed Sathak Engineering College, Kilakarai*

***Abstract*—A web vulnerability scanner is an automated security tool designed to identify, analyse, and report vulnerabilities in web applications, websites, and associated services. It works by systematically crawling web pages, mapping application structures, and simulating various attack techniques such as SQL injection, cross-site scripting (XSS), and authentication flaws. By comparing detected behaviours against known vulnerability patterns and security misconfigurations, the scanner helps uncover potential entry points for attackers. Modern web vulnerability scanners often integrate with development pipelines, enabling continuous security assessment and faster remediation. These tools play a critical role in proactive cybersecurity strategies by reducing manual effort, improving coverage, and ensuring compliance with security standards.**

***Index Terms*—Web vulnerability scanner, Web security, Vulnerability assessment, Penetration testing, SQL injection, Cross-site scripting (XSS), Security misconfiguration, Automated scanning, Cybersecurity, Risk management.**

## I. INTRODUCTION

A web vulnerability scanner is a cybersecurity tool designed to automatically detect security weaknesses in websites and web applications. As modern applications handle sensitive data and are constantly exposed to the internet, they become common targets for cyberattacks. A vulnerability scanner helps identify these risks early so they can be fixed before being exploited.

The scanner works by exploring the structure of a web application, identifying input points such as forms and URLs, and then sending test requests to check how the system responds. Based on these responses, it can detect common vulnerabilities like SQL Injection, Cross-Site Scripting, and Cross-Site Request Forgery. These tools are widely used by developers, ethical hackers, and security teams because they automate the process of finding security flaws, saving time and improving accuracy. However, they are not perfect and are usually combined with manual testing for better results.

### 1.1 Organization Profile

Phoenix Softech was established in 2001 in Madurai, with additional branches in Chennai and Coimbatore. The organization focuses on providing IT strategy consulting and technology solutions across healthcare, pharmaceuticals, logistics, and other verticals. Its mission is to provide market access for state-of-the-art Hi-tech products and services in Information Technology.

### 1.2 Methodology

The methodology of a web vulnerability scanner is the step-by-step process it uses to find security issues in a web application. First, the scanner performs discovery, where it explores the website and maps its structure, including pages, links, and input fields. Next, it moves to scanning, sending various test inputs to check for weaknesses. During this phase, it looks for common vulnerabilities like SQL Injection and Cross-Site Scripting

## II. LITERATURE SURVEY

A.H. Shah et al. proposed a framework for detecting SQL injection vulnerabilities in web applications using automated input analysis. Their approach involves simulating user inputs across web forms and

URL parameters and then monitoring the responses to identify abnormal database errors indicative of potential SQLi points. improved detection accuracy compared to traditional manual methods.

Alina Alamichhane and Gopal Karn introduced a hybrid model combining static and dynamic analysis for web vulnerability assessment. Their method leverages Python-based tools to parse web requests, analyse server responses, and detect anomalies such as cross-site scripting (XSS) and insecure deserialization. By integrating both runtime monitoring and static code inspection, the model reduces false positives and enables more precise vulnerability identification.

R. Smith and K. Lee focused on automating XSS detection using Python scripts that inject test payloads into input fields and track reflected content in the DOM. Their study highlights the effectiveness of combining Selenium for dynamic content handling with Requests and Beautiful Soup for parsing and analysing web responses, allowing for efficient identification of stored and reflected XSS vulnerabilities.

M. Chen et al. explored the use of machine learning techniques in web vulnerability detection, particularly leveraging Python's scikit-learn library to classify potentially unsafe inputs. Their work demonstrated that feature extraction from HTTP requests, URL structures, and cookies, followed by supervised learning, can accurately predict SQLi and XSS risks in complex applications.

S. Kumar and P. Sharma proposed a security assessment framework for web applications that incorporates automated vulnerability scanning, log analysis, and risk prioritization. Implemented in Python, their framework allows organizations to systematically identify weaknesses and generate actionable reports, highlighting critical endpoints and misconfigurations that pose the highest threat levels.

## III. EXISTING SYSTEM

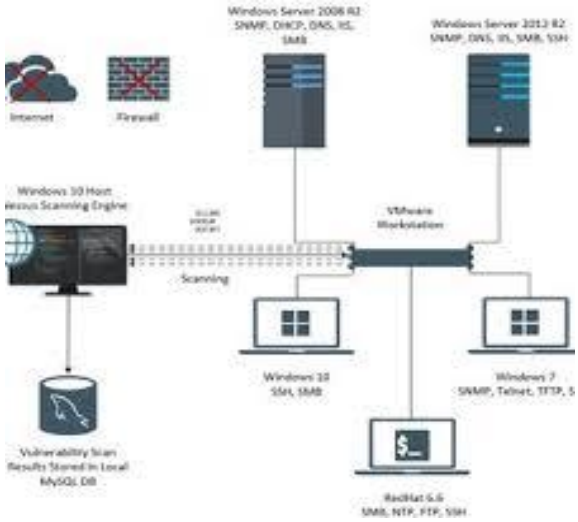
The existing system uses automated tools like OWASP ZAP and Burp Suite to scan web applications. These tools crawl websites, test inputs, and detect common vulnerabilities such as SQL Injection and Cross-Site Scripting. They then generate reports for developers to fix issues.

### 2.2 Purpose of Work

The purpose of a web vulnerability scanner is to automatically detect security weaknesses in web applications before attackers can exploit them. It helps identify common vulnerabilities such as SQL Injection and Cross-Site Scripting, and provides reports so developers can fix the issues. In short, its main purpose is to improve web application security by finding and reducing potential cyber risks.

## IV. PROPOSED SYSTEM

The proposed system of a web vulnerability scanner is an improved security model that automatically detects and analyses vulnerabilities in web applications with higher accuracy and efficiency than traditional scanners. It uses an intelligent crawler and a rule-based or AI-assisted engine to identify security flaws and generate clear reports with suggested fixes. It mainly focuses on detecting common threats such as SQL Injection, Cross-Site Scripting, and Cross-Site Request Forgery, while reducing false positives and improving scan speed.




## V. SYSTEM MODULES

A web vulnerability scanner is usually designed using multiple system modules, where each module performs a specific task in the process of finding and

reporting security issues in a web application. Below is a detailed explanation of each module.

a) Web Application Module (Target System)

This is the actual website or web application that is being tested. It may include login pages, forms, APIs, databases, and user inputs.

The scanner interacts with this module to simulate user behaviour and identify weak points in the system.

b) Crawler Module

The crawler module is responsible for automatically exploring the web application. It works like a search engine bot Collects links, cookies, and parameters. Builds a map of the entire application structure.

c)Request Generator / Attack Simulation Module

This module creates and sends test inputs to the application. It prepares payloads for vulnerabilities like:

SQL Injection

Cross-Site Scripting

Cross-Site Request Forgery

d) Vulnerability Detection Engine

This is the core module of the scanner.

Functions:

Analyses responses from the web application

Detects abnormal behaviour or security flaws

Uses rule-based logic or AI techniques

Matches responses with known attack patterns

It determines whether a vulnerability exists or not.

e) Analysis Module

After detection, this module evaluates the severity and impact of vulnerabilities.

Functions:

Classifies vulnerabilities (Low, Medium, High, Critical)

f). Reporting Module

This module presents the final results in a readable format.

Functions:

Generates detailed vulnerability reports

Shows affected URLs and parameters

Provides proof of vulnerability

Suggests fixes and security recommendations

Displays dashboards for easier understanding

g) Database / Storage Module

This module stores all scan-related data.

Functions:

Saves scan history

Stores detected vulnerabilities

Keeps logs of requests and responses

Helps in future comparison and tracking improvements

## VI. DATAFLOW DIAGRAM

A Data Flow Diagram (DFD) of a Web Vulnerability Scanner shows how data moves between different processes in the system while scanning a web application for security issues.

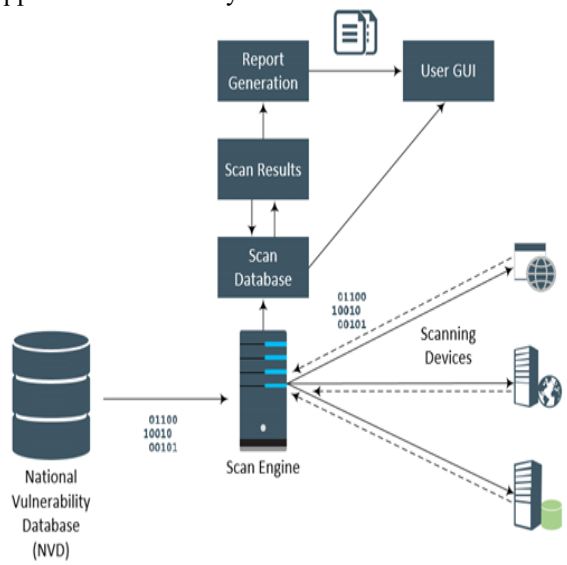



Figure: Data Flow Diagram – Image web vulnerability scanner

## VII. CONCLUSION

A web vulnerability scanner is an essential cybersecurity tool that helps detect security weaknesses in web applications before attackers can exploit them. By automatically crawling websites, testing inputs, and analysing responses, it identifies common vulnerabilities such as SQL injection, cross-site scripting (XSS), authentication flaws, and misconfigurations. While it cannot replace manual security testing or fully detect complex logic-based vulnerabilities, it significantly improves security by providing fast, consistent, and continuous assessment of web applications. When used regularly, especially in development and deployment pipelines, it

strengthens an organization's overall security posture and reduces the risk of data breaches and attacks. In short, web vulnerability scanners act as an early warning system that helps developers and security teams build and maintain safer, more secure web applications.

## VIII. FUTURE ENHANCEMENTS

Web vulnerability scanners are expected to become far more intelligent and deeply integrated into modern software ecosystems. As web applications grow in complexity—especially with APIs, cloud platforms, and AI-driven systems—scanners will evolve from simple detection tools into autonomous security assistant. Future enhancements in web vulnerability scanners are heavily focused on autonomous AI agents, real-time continuous monitoring, and deeper integration into DevSecOps pipelines to address the increasing complexity of modern web applications. The next generation of scanners is shifting from static, rule-based detection to AI-driven, self-directed security agents that can understand context, reduce false positives, and even suggest remediation.

IJERT – International Journal of Engineering Research & Technology

IJERT – International Journal of Engineering Research & Technology

Here are the key future enhancements in web vulnerability scanners based on recent research and industry trends:

1. Agentic AI & Autonomous Scanning

ReAct-style Controllers: Future scanners will utilize LLM-powered (Large Language Model) agents to autonomously plan scanning tasks, invoke relevant tools, and interpret findings without continuous human supervision.

Adaptive Scanning: Rather than fixed rules, agents will "learn" the structure of a specific web application and adapt their scanning strategies in real-time based on intermediate results.

Autonomous Penetration Testing: Scanners will increasingly mimic the behavior of human penetration testers to find complex, chained vulnerabilities.

IJERT – International Journal of Engineering Research & Technology

IJERT – International Journal of Engineering Research & Technology

2. Drastic Reduction of False Positives (AI/ML)

Contextual Understanding: Machine Learning (ML) models will be used to better distinguish between normal behaviour and malicious input, significantly reducing the "noise" of false alarms.

Proof-Based Scanning: Future scanners will prioritize verification. Instead of simply reporting a vulnerability, they will generate concrete evidence of exploitability, minimizing the need for manual verification.

IJERT – International Journal of Engineering Research & Technology

IJERT – International Journal of Engineering Research & Technology

3. Shift-Left and DevOps Integration

CI/CD Pipeline Integration: Scanners will become an integral part of the Software Development Life Cycle (SDLC). They will automatically scan new builds, APIs, and microservices for security issues before they are deployed to production.

Developer-First Tools: Reports will shift from verbose technical jargon to actionable insights that developers can fix directly, minimizing friction in the development process.

IJERT – International Journal of Engineering Research & Technology

IJERT – International Journal of Engineering Research & Technology

4. Continuous Monitoring & Cloud-Native Security

Continuous Vulnerability Management (CVM): Scanners will shift from periodic, point-in-time checks to continuous monitoring of attack surfaces, identifying changes in real-time.

Cloud-Native & API Scanning: As web apps move to containers and microservices, new scanners will specifically target API endpoints and cloud-native configurations (e.g., Kubernetes misconfigurations).

5. Advanced Threat Coverage

Zero-Day Detection: Future scanners will leverage deep learning and heuristic analysis to identify previously unknown (zero-day) vulnerabilities by identifying anomalous behaviors rather than matching known signatures.

Supply Chain Attack Detection: Scanners will detect vulnerabilities in third-party libraries and

dependencies, a major weakness in modern application development.

## REFERENCES


[1] Amouei, M., Rezvani, M., & Fateh, M. (2023). RAT: Reinforcement Learning Driven and Adaptive Testing for Vulnerability Discovery in Web Application Firewalls.

[2] Alaoui, R. L., & Nfaoui, E. H. (2022). Deep Learning for Vulnerability and Attack Detection on Web Applications: A Systematic Literature Review. Future Internet, 14(4), 118. MDPI+1

[3] Bakır, R. (2025). UniEmbed: A Novel Approach to Detect XSS and SQL Injection Attacks Leveraging Multiple Feature Fusion with Machine Learning Techniques. Arabian Journal for Science and Engineering, 50, 15591–15604.

[4] Devi, A., & Kumar, P. (2023). An Experimental Study on Detecting and Mitigating Vulnerabilities in Web Applications. IIETA International Journal of Social Sciences and Education, 14. IIETA

[5] Jeyaboopathiraja, J., & Mithun, N. (2024). Detecting and Removing Web Application Vulnerabilities with SQL Injection Prevention. International Journal of Advanced Research in Computer and Communication Engineering. Peer-reviewed Journal

[6] Kalantari, F., Zaeifi, M., Bao, T., Wang, R., Shoshitaishvili, Y., & Doupé, A. (2022). Context Auditor: Context sensitive Content Injection Mitigation. arXiv. arXiv

[7] Kieźun, A., Guo, P. J., Jayaraman, K., & Ernst, M. D. (2009). Automatic Creation of SQL Injection and Cross Site Scripting Attacks. University of Maryland / MIT. umiacs.umd.edu

[8] Liu, M., Li, K., & Chen, T. (2020). DeepSQLi: Deep Semantic Learning for Testing SQL Injection. arXiv. arXiv

[9] Rathore, D., et al. (2024). Machine Learning for Web Vulnerability Detection. Nanotechnology Perceptions, 20(7). nano-ntp.com

[10] Sambhus, K., & Liu, Y. (2024). Automating SQL Injection and Cross Site Scripting Vulnerability Remediation in Code. Software, 3(1), 28–46. MDPI+1