WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

QUANTITATIVE ASSESSMENT OF AUTOMATESECURITY VULNERABILITY SCANNERS

Susmitha R[1] , Deepika B[2], Pandeeswari M[3]

Computer science and engineering Sree sowdambika College of engineering chettikuruchi, virudhunagar.

ABSTRACT :

Automated security vulnerability scanners have emerged as vital tools in the realm of cybersecurity, offering rapid and systematic identification of weaknesses in software applications, network infrastructures, and systems. This study aims to conduct a comprehensive quantitative assessment of several widely adopted automated vulnerability scanners, focusing on metrics such as detection rate, false positive and false negative rates, scan duration, system resource usage, and coverage breadth.

In the digital era, web applications serve as critical platforms for businesses and individuals, but their widespread use also makes them prime targets for cyberattacks.

Keywords . Here are some simpler keywords for "Quantitative Assessment of Automated Security Vulnerability Scanners":
Automated security tools,Vulnerability scanning,Security assessment,Scanner performance,Vulnerability detection,Automation in security,Security scanner effectiveness,Vulnerability management,Security risk analysis,Automated testing

INTRODUCTION.

In the digital era, web applications have become an integral component of both business operations and personal services, facilitating communication, commerce, data storage, and more. However, this increasing reliance on web technologies has made them attractive targets for cybercriminals.

2.PROJECT OBJECTIVES

The study aims to analyze and compare the performance of selected automated scanners based on key parameters such as detection accuracy, false positive/negative rates, scanning speed, and coverage of known vulnerabilities.
The specific objectives of the project are as follows:
To evaluate the effectiveness of widely used automated vulnerability scanners in detecting common web application vulnerabilities, including SQL injection, cross-site scripting (XSS), and insecure configurations.
To identify and compare the strengths and weaknesses of open-source and commercial vulnerability scanning tools using standardized test environments and real-world scenarios.

SYSTEM PROPOSAL

1.EXISTING SYSTEM

In the current cybersecurity landscape, organizations increasingly rely on automated vulnerability scanners and web crawlers to detect and manage security flaws in their web applications. These tools form an essential part of vulnerability management programs and are often integrated into continuous security testing pipelines.

2.Disadvantages of the Existing System

Although automated vulnerability scanners and web crawlers are widely used in modern cybersecurity practices, they have several limitations that affect their efficiency and reliability. These disadvantages include:

High Rate of False Positives and False Negatives

Existing scanners may incorrectly flag harmless elements as vulnerabilities (false positives) or fail to detect actual security flaws (false negatives). This undermines trust in the results and requires

manual verification, increasing the workload for security analysts.

3.Proposed System

To overcome the limitations of the existing systems, the proposed system aims to develop an improved framework for automated web vulnerability scanning and crawling, which is more accurate, adaptive, and suitable for modern web applications. The system combines enhanced crawling techniques, advanced vulnerability detection mechanisms, and integration with continuous security pipelines to provide a more reliable and scalable solution.

Key Features of the Proposed System:

Intelligent Crawling Engine

The crawler will be capable of rendering JavaScript-heavy content using headless browsers (e.g., Puppeteer or Selenium), enabling it to fully explore single-page applications (SPAs) and dynamically loaded content.

4.Literature Survey

Title: "Evaluation of Automated Web Vulnerability Scanners: A Comparative Study"

Year: 2020

Methodology:

This study compared multiple automated web vulnerability scanners (open-source and commercial) against a standardized testbed containing known web vulnerabilities. The evaluation measured detection accuracy, false positive/negative rates, scan duration, and OWASP Top 10 coverage. Quantitative metrics were used to analyze tool effectiveness.

Disadvantages:

Many scanners struggled with dynamic JavaScript content, causing incomplete scans. False positives were prevalent, necessitating manual verification, and the testbed lacked real-world complexity, limiting applicability.

2. **Title**: "Web Application Security Testing Using Machine Learning Techniques"

Year: 2021

Methodology:

This research proposed a machine learning-based framework for detecting web vulnerabilities by training classifiers on datasets containing benign and malicious URL requests. The model aimed to improve detection rates beyond traditional signature-based methods.

Disadvantages:

The approach required large, labeled datasets for training, which are often unavailable or imbalanced. Additionally, the model struggled with zero-day attacks and polymorphic threats not represented in the training data.

SYSTEM ARCHITECTURE:

These scanners often rely on web crawlers to systematically discover and analyze URLs within a target application or website to assess it for vulnerabilities like XSS, SQL injection, and misconfigurations.

Explanation of System Architecture

1. **Seed URLs**

Input: Initial list of target URLs provided to the system.

These are starting points (like a homepage or sitemap) where the scanning begins.

2. URL Frontier

Purpose: Manages the queue of URLs to be visited.

URLs are prioritized or scheduled here based on policies (e.g., breadth-first or depth-first crawling).

Component	Security Role
Content Parser	Detects form fields and script inputs for injection testing.
Link Extractor	Ensures no hidden paths are missed (security-through-obscurity bypass).
URL Filter	Keeps scanning within target boundaries to prevent unintentional attacks.
URL Seen Checks	Ensures performance and avoids wasting time on duplicates.
Storage Modules	Help compare old scans vs new scans for change detection (important in CI/CD).

IMPLEMENTATION

MODULES:

User Interface (UI) Module

Allows users to input scan parameters such as target URL, scanner type, and configuration options.

Provides a dashboard for monitoring scan progress and viewing final reports.

Web Crawler Module

Crawls the target web application to discover all accessible pages, links, forms, and input fields.

Supplies the scanning module with a complete list of endpoints to assess.

Scanner Module

Integrates multiple automated vulnerability scanners (e.g., Nikto, OWASP ZAP).

Performs scans on discovered URLs to identify vulnerabilities like SQL injection, XSS, etc.

HARDWARE REQUIREMENTS:

System : Pentium IV 2.4 GHz

Hard Disk : 200 GB

Mouse : Logitech.

Keyboard : 110 keys enhanced

Ram : 4GB

SOFTWARE REQUIREMENTS:

O/S : Windows 10.

Language : PHP, XAMPP, MYSQL

Front End : HTML, CSS, JS

Software used : VS Code

CONCLUSION:

The project "Quantitative Assessment of Automated Security Vulnerability Scanners" successfully demonstrates the importance of evaluating and comparing multiple automated scanners to enhance web application security. By integrating web crawling, automated vulnerability detection, and performance analysis, the system provides a comprehensive framework for identifying security weaknesses across different scanning tools. The comparative approach not only reveals the strengths and limitations of each scanner but also helps organizations make informed decisions when selecting the most effective tool for their security needs.

REFERENCES:

[1] E. Lavens, P. Philippaerts, and W. Joosen, "A Quantitative Assessment of the Detection Performance of Web Vulnerability Scanners," Aug. 23, 2022.

[2] X. Zhou et al., "Comparison of Static Application Security Testing Tools and Large Language Models for Repo-level Vulnerability Detection," Jul. 23, 2024.

[3] M. Esposito, V. Falaschi, and D. Falessi, "An Extensive Comparison of Static Application Security Testing Tools," Mar. 14, 2024.

[4] N. S. Harzevili et al., "A Survey on Automated Software Vulnerability Detection Using Machine Learning and Deep Learning," Jun. 20, 2023.

[5] S. Chakraborty et al., "Deep Learning-Based Vulnerability Detection: Are We There Yet?," Sep. 3, 2020.

[6] "Performance Analysis of Vulnerability Detection Tools and Techniques," 2024.

[7] "Vulnerability Assessment: Analyzing Automated Scanning Techniques for Identifying Security Flaws in Network Infrastructure," Jul. 25, 2024.

[8] "Enhancing Web Application Security through Automated Penetration Testing with Multiple Vulnerability Scanners," 2024.

[9] "A Performance Assessment of Free-to-Use Vulnerability Scanners," 2022.

[10] "Evaluating and Comparing Web Application Security Testing Tools," 2024.

[11] "A Comparative Evaluation of Automated Vulnerability Scans versus Penetration Tests," 2023.

[12] "Benchmarking Vulnerability Assessment Tools for Enhanced Cyber-Physical System Security," 2023.

[13] "A Survey and Comparative Study on Vulnerability Scanning Tools," 2023.

[14] "Performance of Automated Network Vulnerability Scanning at Remediating Security Issues," 2023.

[15] P. R. Kushe, "Comparative Study of Vulnerability Scanning Tools: Nessus vs. Retina," 2017.

[16] "Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks," 2008.

[17] A. Seth et al., "ComparingEffectiveness and Efficiency of IASTand RASP Toolsin a Large Java-based System," 2023.

[18] OWASP Foundation, "OWASP Benchmark Project," Open Web Application Security Project, 2023

[19] CVE, "Common Vulnerabilities and Exposures (CVE) Database," MITRE Corporation, 2023

[20] NIST, "National Vulnerability Database (NVD)," National Institute of Standards and Technology (NIST), 2023