Grenze International Journal of Engineering and Technology, June Issue

Practical Exploitation and Security Assessment of OWASP Top 10 Web Vulnerabilities

A. Neela Madheswari[1], Adwaith Anilkumar[2], M. Akash[3], Akshay Ajay[4] and Arjun Balagopalan[5]
[1]Professor, CSE Department, Mahendra Engineering College, Namakkal, India
Email: neela.madheswari@gmail.com
[2-5]UG Scholar, Cyber Security Department, Mahendra Engineering College, Namakkal, India
Email: adhwaithea@gmail.com, vijayakash451@gmail.com, ajayakshay283@gmail.com, arjunbalagopalan4510@gmail.com

***Abstract*— The rapid expansion of web applications has increased exposure to critical security risks, necessitating advanced automated assessment methodologies. This paper presents a full-stack security platform designed for Practical Exploitation and Security Assessment of OWASP Top 10 Web Vulnerabilities. Developed using the Django 4.2 framework, the system employs a defense-in-depth architecture, integrating Argon2 password hashing, AES-256-GCM encryption for sensitive data at rest, and HMAC-SHA256 for API request integrity. We detail modular scanning engines designed to detect and exploit vulnerabilities including SQL injection, XSS, and SSRF. Furthermore, the tool incorporates an automated CVSS v3.1 scoring engine for standardized risk quantification. Experimental results demonstrate the platform's effectiveness in providing real-time vulnerability management and a tamper- evident audit trail. Finally, we explore future research into Post- Quantum Cryptography and AI-driven triage to enhance defensive resilience.**

***Index Terms*— OWASP top 10, Pentesting, Full-stack security, CVSS scoring, HMAC integrity and automated scanning.**

I. INTRODUCTION

For today's modern digital ecosystems, web technologies are important for enabling ubiquitous computing, cloud services and AI integration. There are various range of applications based on web such as single page applications, responsive web development, progressive web apps, server-less architecture, and mobile-based design, and many more There are various versions of Web starting from Web 1.0, where only the static information is displayed and using browser, the user can view the web pages, Web 2.0, where user centric approach and the support of collaborative environment is provided, Web 3.0, where intelligence is added to the web sites, Web 4.0, where applications are made easy and they can share information easily that links various sources such as social medias, web sites, blogs, and video sharing web sites [1]. People start using web sites daily and at the same time, the role of web security plays an important topic. In one of the largest airlines, in Air India, an attack was performed and the user's credit card and sensitive information was captured [2].
Traditional vulnerability scanning tools focus primarily on automated detection of security weaknesses but often provide limited insight into the practical exploitation of these vulnerabilities.

Grenze ID: 01.GIJET.12.2.543_1

Understanding how vulnerabilities can be exploited in real-world scenarios is essential for security analysts and developers in order to design effective mitigation strategies. Therefore, combining automated vulnerability assessment with controlled exploitation techniques can provide a deeper understanding of potential attack vectors and their impact on web applications.

This research presents a framework for the practical exploitation and security assessment of web vulnerabilities based on the OWASP Top 10. The proposed system integrates automated scanning mechanisms, controlled exploitation modules, and vulnerability risk evaluation using standardized scoring techniques. The primary objective of this work is to develop a modular secure platform that supports vulnerability detection, exploitation analysis, and risk prioritization for web based applications. The system enables security professionals and researchers to evaluate web application security in a controlled environment and provides insights that can assist in improving defensive mechanisms. By integrating vulnerability scanning, risk scoring, and role- based access management, the platform contributes to a structured approach for web security testing and assessment.

The remainder of this paper is organized as follows. Section II discusses related work in the area of web application security assessment tools and vulnerability analysis techniques. Section III describes the architecture and design of the proposed system. Section IV explains the methodology used for vulnerability detection and exploitation. Section V presents vulnerability detection techniques; section VI describes the experimental setup and evaluation results. Finally, Section VII concludes the research work and outlines future research directions.

Figure 1. OWASP Top 10 as per 2025 [8]

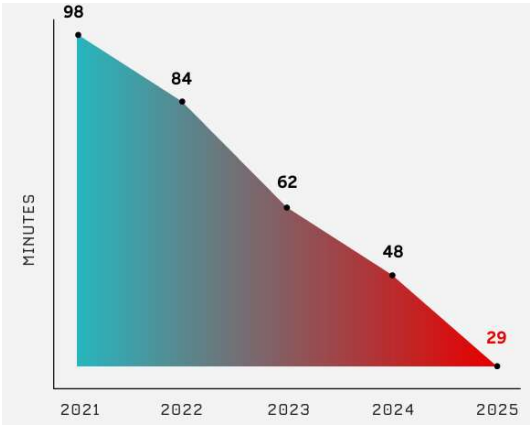

Figure 2. Average eCrime break-out time, January-December 2025 [10]

II. Related Work

Web based systems are distributed in nature and thus security is very important factor. Many security solutions used are firewalls, intrusion detection systems (IDS), intrusion prevention systems (IPS), digital certificates and signatures, cryptographic techniques, and many more [3]. Many web applications or web sites deployed may have various critical vulnerabilities [4]. Some sites are vulnerable to security threats and may cause severe impacts on security risks such as website defacements, data breaches, and other cyber-attacks [5]. In many modern web applications information resources are retrieved through user-specified URLs which lead to harmful destinations and opening opportunities for further exploitation of the target system [6]. A number of vulnerabilities prevailing in web based applications or web based services such as Cross-site Scripting (XSS), Cross Site Request Forgery (CSRF), SQL Injection (SQLi), and many more [7].

As per the 2017 Verizon Data Breach Investigation Report, the more successful breaches resulted from attacks on web applications than any other type of attack. The Open Web Application Security Project (OWASP) focusing on identifying various vulnerabilities and mitigate those attacks caused by those vulnerabilities. OWASP was started during 2003 and the web application vulnerabilities are researched and the top ten of them are listed for every 4 years once. The recent revision is OWASP Top 10 as per the year 2025 [9]. It is shown in figure 1.

Once the attackers gain initial access using these vulnerabilities, their immediate objective is to reduce the break-out time, which is the time required to determine how faster an attacker move laterally from the initial point of compromise to other systems in the network. As per the analysis given in [10], the average eCrime break-out time is given in figure 2.

The rapid expansion of web applications has significantly increased the attack surface available to cybercriminals, making web security a critical research area. Numerous studies have focused on identifying and mitigating vulnerabilities that commonly affect web applications. Among the most widely recognized initiatives in this domain is the Open Web Application Security Project, which publishes the OWASP Top 10, a list of the most critical security risks affecting modern web applications. This list provides a standardized reference for developers and security professionals to understand and mitigate vulnerabilities such as injection attacks, cross-site scripting, and broken access control.

Several automated vulnerability scanning tools have been developed to detect web security flaws. One widely used open-source solution is OWASP ZAP, which performs dynamic application security testing by analyzing HTTP requests and responses to identify potential vulnerabilities. Similarly, Burp Suite provides an integrated platform for web security testing that includes features such as proxy interception, automated scanning, and manual penetration testing capabilities. These tools are widely used by security professionals; however, they often require advanced expertise and may not provide a structured environment for controlled exploitation and experimentation.

In addition to vulnerability scanners, penetration testing frameworks have also been developed to simulate real-world attack scenarios. The Metasploit Framework is one of the most popular exploitation frameworks used by security researchers to validate vulnerabilities through prebuilt exploits and payloads. While such frameworks are effective for demonstrating attack techniques, they generally focus on exploitation rather than integrated vulnerability assessment and risk prioritization.

To evaluate the severity of vulnerabilities, many security tools rely on standardized scoring models such as the Common Vulnerability Scoring System (CVSS). CVSS provides a numerical score that represents the severity of a vulnerability based on exploitability and impact metrics. This scoring mechanism helps organizations prioritize remediation efforts and allocate resources to address high- risk vulnerabilities first.

Despite the availability of numerous security testing tools and frameworks, many existing solutions address vulnerability detection and exploitation separately. Furthermore, several tools lack integrated mechanisms for vulnerability management, risk scoring, and role-based access control within a unified platform. These limitations highlight the need for a comprehensive system that combines automated vulnerability scanning, controlled exploitation, and standardized risk assessment techniques.

The framework proposed in this research aims to address these challenges by providing an integrated platform for the practical exploitation and security assessment of web application vulnerabilities based on the OWASP Top 10. By incorporating automated scanning modules, controlled testing environments, and CVSS-based risk scoring, the system provides a structured approach for analysing web application security and prioritizing vulnerability remediation.

III. System Architecture

The proposed system is designed to provide an integrated platform for the practical exploitation and security assessment of web application vulnerabilities. The architecture follows a modular design that enables efficient vulnerability scanning, controlled exploitation, and risk evaluation of web applications based on the widely recognized OWASP Top 10 security risks. The system architecture consists of several interconnected components including the user interface layer, authentication and access control module, vulnerability scanning engine, exploitation module, risk assessment component, and a secure database layer.

At the top layer of the architecture, users interact with the system through a web-based interface that provides functionalities such as target registration, vulnerability scanning, report generation, and system monitoring. The platform supports multiple user roles such as administrators, penetration testers, security analysts, and viewers. These roles are managed using a role-based access control mechanism to ensure that only authorized users can perform security-sensitive operations.

The authentication and access control module is responsible for managing user identities and enforcing security policies. Secure authentication mechanisms are implemented to protect user credentials and maintain session integrity. This module ensures that users can only access system functionalities that correspond to their assigned roles.

The core functionality of the system resides in the vulnerability scanning engine. This module automatically analyzes web applications to detect security weaknesses associated with the OWASP Top 10 vulnerabilities. The scanning engine performs several security checks, including detection of injection attacks, cross-site scripting vulnerabilities, broken access control issues, and security misconfigurations. By sending carefully crafted

requests to target applications and analysing their responses, the scanning engine can identify potential vulnerabilities that may compromise system security.

Once vulnerabilities are detected, the exploitation module performs controlled validation of the identified weaknesses. This module simulates attack scenarios in a controlled environment to demonstrate how vulnerabilities can be exploited by malicious actors. The controlled exploitation process helps security analysts understand the severity and practical impact of security flaws.

To evaluate the severity of discovered vulnerabilities, the system integrates a risk assessment component based on the Common Vulnerability Scoring System (CVSS). This module calculates risk scores by analysing factors such as exploitability, attack complexity, and potential impact on confidentiality, integrity, and availability. The calculated scores allow security teams to prioritize vulnerabilities and focus on high-risk issues first.

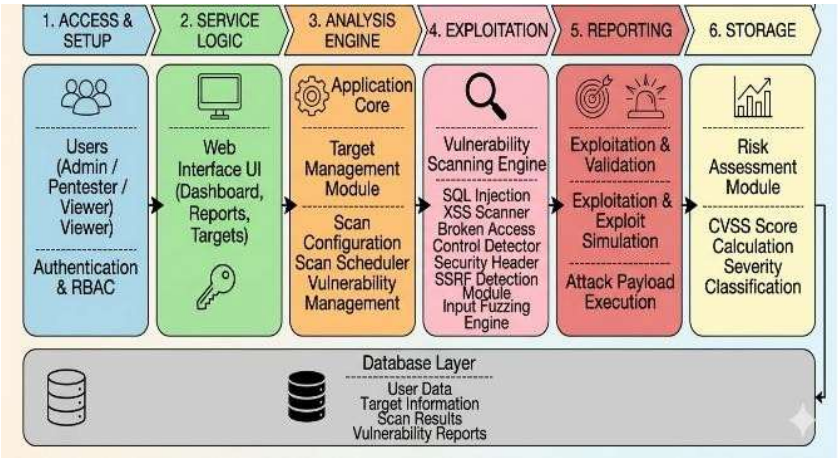

Figure 3. System Architecture of the proposed framework

All scan results, vulnerability details, user information, and system logs are stored within the database layer. Secure storage mechanisms ensure that sensitive information is protected using encryption and access control policies. The database also maintains historical records of scans, enabling users to track vulnerability trends and monitor improvements in web application security over time. The figure 3 specifies the System architecture of the proposed framework.

Overall, the proposed system architecture provides a comprehensive framework for automated vulnerability detection, controlled exploitation, and structured risk evaluation. By integrating these functionalities within a single platform, the system enhances the effectiveness of web application security assessment and provides valuable insights for improving defensive security measures.

IV. SYSTEM METHODOLOGY

The system is developed using three major stages. The major stages are as follows:

A. Requirements and Planning

The various core features that include 8 scan types (SQLi, XSS, SSRF, etc.), are defined. RBAC for 4 user roles, CVSS v3.1 scoring, and cryptographic security (AES-256-GCM, Argon2). Django 4.2 framework, Kali Linux testing platform, and OWASP ZAP integration for comprehensive coverage are used.

B. Architecture Design

Implemented layered defense-in-depth: Presentation using Bootstrap 5 UI, implementing Middleware for observing Security Headers, HMAC, Audit Log, and developing Business Logic using Scanner Engine, CVSS Calculator, and design Data Layer using encrypted SQLite/PostgreSQL. A modular scanning engine with specialized detectors for web vulnerability is designed.

C. Core Implementation

Scanner modules are built with specific payloads (11 SQLi, 10 XSS), security middleware stack, and vulnerability management lifecycle phases: i) Open, ii) Confirmed and iii) Fixed. Integrated cryptographic stack and RBAC is enforced across all views.

V. VULNERABILITY DETECTION TECHNIQUES

There are two types of security metrics used for measuring security vulnerabilities namely: i) Common Vulnerabilities and Exposures (CVE), and ii) Common Vulnerability Scoring System (CVSS) [11]. The National Institute of Standards and Technology includes National Vulnerability Database, in which it supports CVSS as the metrics with severity ratings score.

The CVSS ratings are mentioned using various versions and the recent ratings is CVSS v4.0 Ratings. As per the CVSS v4.0 ratings, the severity is measured using five different values namely: i) none, specify the severity score as 0, ii) low, specify the range of values from 0.1 to 3.9, iii) medium, specify the range of values from 4.0 to 6.9, iv) high, specify the range of values from 7.0 to 8.9, and v) critical, specify the range of values from 9.0 to 10.0 [12]. Table 1 outlines detection and controlled exploitation techniques for OWASP Top 10:2025 web vulnerabilities, enabling practical security assessment through active scanning and payload execution. Also the CVSS is shown for every vulnerability type with the impact status. Practical Assessment Mode executes payloads in controlled environment, captures exploitation evidence, and generates remediation guidance with CVSS v4.0 risk scoring for comprehensive security validation.

TABLE I. OWASP TOP 10: 2025 EXPLOITATION COVERAGE

OWASP 2025	Exploitation Technique	Payload Execution	CVSS Score
A01: Broken Access Control	IDOR traversal, privilege escalation	Direct object manipulation, path traversal	8.1 High
A02: Security Misconfiguration	Header bypass, XXE injection	Missing HSTS/CSP exploitation, XML entities	7.3 High
A03: Supply Chain	Dependency hijacking	Malicious package injection simulation	Varies
A04: Cryptographic Failures	Padding oracle, key extraction	Chosen ciphertext attacks, side-channel	8.8 High
A05: Injection	SQLi dump, XSS execution	Error/time-blind SQLi, DOM XSS vectors	9.8 Critical
A06: Insecure Design	Business logic bypass	Parameter tampering, race conditions	9.8 Critical
A07: Authentication Failures	Session fixation, credential stuffing	Account takeover simulation	9.8 Critical
A08: Integrity Failures	Deserialization RCE	Gadget chains, unsafe deserialization	8.8 High
A09: Logging Failures	Log poisoning	Audit evasion, log injection	6.5 Medium
A10: Exception Handling	Error-based info disclosure	Stack trace harvesting, error fuzzing	7.5 High

VI. EXPERIMENTAL RESULTS AND OBSERVATIONS

To validate the proposed framework for practical exploitation and security assessment of OWASP Top 10 web vulnerabilities, the developed platform was deployed and tested in a controlled local environment using Kali Linux. The implementation demonstrates the complete workflow of the system, including platform initialization, target registration, vulnerability scanning, result visualization, audit logging, and detailed vulnerability analysis. The platform was executed locally through the Django development server and accessed through the browser at http://127.0.0.1:8080/. The setup script automated dependency installation, database migration, and administrator account initialization. The system was then used to perform practical scans against intentionally vulnerable web targets for demonstration and validation purposes.

The main dashboard provides a centralized overview of the platform's activities. It displays the total number of registered targets the number of scans performed, and a categorized summary of findings based on severity levels such as critical, high, medium, and low. The dashboard also presents recent scan history and a list of registered targets, enabling the user to quickly assess the security posture of the monitored applications.

The target registration interface allows users to add new applications for assessment. During registration, the user can specify the target name, URL, description, and optional authentication credentials for authenticated scanning. This feature enables structured scan management and allows repeated testing of multiple applications under a unified framework.

Once targets are added, they become visible in the dashboard and can be launched for scanning directly through the interface. The registered targets section maintains the target name, URL, and last scanned information, providing a clear operational view of the assessment environment. After launching a scan, the system performs vulnerability detection against the selected target and generates categorized findings. The scan result view shows the scan identifier, target name, scan completion status, scan duration, and a severity distribution summary. It also lists all identified findings with fields such as severity, vulnerability type, title, affected URL, CVSS score, and status. This structured output helps analysts interpret the findings efficiently and prioritize remediation.

The system was able to identify several high-severity issues in the target environment. Among the detected findings were possible SSRF cases, broken access control issues, information exposure via HTTP headers, security misconfigurations, and sensitive file exposure. The platform assigns severity labels and CVSS-based scores to support risk-based prioritization.

A more detailed result page provides fine-grained information about each vulnerability. For example, in the SSRF case, the system records the vulnerability type, CVSS v3.1 score, affected URL, parameter name, payload used, evidence returned by the application, vulnerability description, remediation guidance, and status update option. This detailed evidence-driven approach improves analyst confidence during validation and documentation.

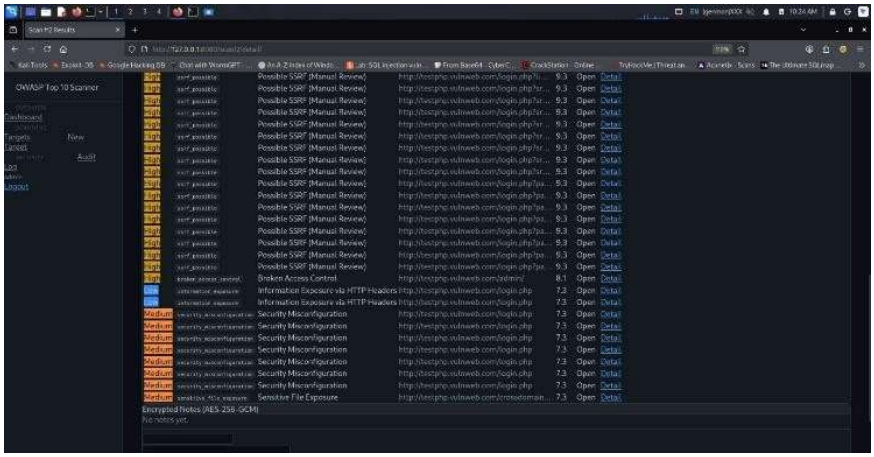

Figure 4. Detailed Scan report showing multiple findings, severity levels, affected URLs, and links to individual vulnerability records

The broader scan detail page displays the full vulnerability list for a specific scan. It includes multiple entries for repeated or parameter-specific findings and serves as the primary view for vulnerability review. This interface supports better traceability by linking each finding to its detailed analysis page.

To support accountability and forensic visibility, the platform includes a real-time security audit log. This module records important user and system actions such as logins, dashboard access, target registration, vulnerability detail access, and other operational events. Each entry includes the timestamp, username, action performed, IP address, and status information. The audit module is especially valuable for administrative oversight and compliance-oriented environments.

The experimental results demonstrate that the proposed platform successfully integrates multiple key capabilities into a single security assessment framework. It supports end-to-end workflow execution, beginning with secure initialization and target onboarding, followed by automated scanning, vulnerability categorization, evidence-based result reporting, and audit logging. The findings shown in the implementation confirm that the system can be used not only for vulnerability discovery but also for practical exploitation analysis and structured remediation planning.

The security assessment was conducted on the intentionally vulnerable web site url: http://testaspnet.vulnweb.com/login.aspx [13]. The identified vulnerabilities were categorized based on severity levels: None, Low, Medium, High, and Critical, following standard security assessment practices. But there is no vulnerability having severity as none. It is shown in figure 5.

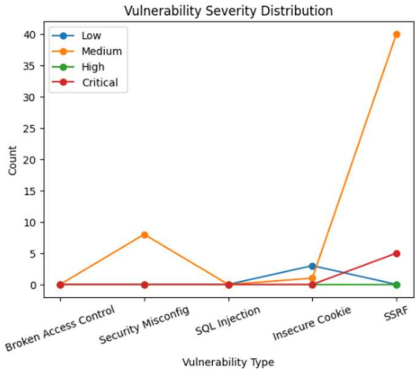

Figure 5. Vulnerability Severity Distribution

A. Security misconfiguration

A total of 8 medium-severity vulnerabilities were identified under security misconfiguration. These issues typically arise due to improper server settings, exposed headers, or unnecessary services being enabled. This vulnerability may lead to information leakage and increase the attack surface.

B. Insecure cookie handling

These vulnerabilities indicate improper handling of cookies, such as missing security flags like HTTPOnly, Secure. It can expose session data to attackers via client-side scripts or network interception.

C. Server Side Request Forgery

This is the most significant finding in the assessment SSRF vulnerabilities can allow attackers to: i) access internal systems, ii) Bypass firewalls, and iii) Interact with cloud metadata services. The presence of critical SSRF vulnerabilities indicates a high risk exposure that requires immediate remediation.

D. Risk Evaluation

The overall risk level of the assessed application can be classified as moderate to high, primarily due to the presence of multiple Server-Side Request Forgery (SSRF) vulnerabilities, including critical instances. While the absence of high-risk issues such as SQL Injection and Broken Access Control indicates relatively strong input validation and access mechanisms, the significant number of medium and critical SSRF findings increases the potential for internal system exploitation. Additionally, security misconfigurations and insecure cookie handling contribute to the overall risk by exposing the application to information leakage and session-related attacks. Immediate remediation of critical SSRF vulnerabilities is strongly recommended to reduce the application's attack surface.

VII. CONCLUSION

The vulnerability assessment reveals that while the application is resilient against common attacks such as SQL Injection and Broken Access Control, it is significantly affected by SSRF and security misconfigurations. Immediate attention is required to mitigate critical SSRF vulnerabilities, followed by improvements in configuration management and secure cookie handling practices. Web vulnerability analysis is very important concept in the current digital world since every firm has its own website or web app through which it extends its services online to all over the world. Hence this work is considered as the proposed system framework design and the vulnerability analysis is done as per the OWASP Top 10 Web vulnerabilities and the findings are shown. Also the CVSS rating is measured for every type of web vulnerability. Using this proposed system, any web site or web app can be tested using the corresponding url and it is easily analyzed the vulnerabilities prevailed in it so that the web site or web app can be mitigated for security attacks.

REFERENCES

[1] Anuj Soni, Sachin Gupta, Navjot Singh Talwandi, "Evolurion of Web Technologies in recent years", Journal of Emerging Technologies and Innovative Research, vol.10, Issue. 9, Sep 2023.

[2] Sitara Anumotu, Kushagra Jha, Amit Balhara, Pronika Chawla, "Security Issues and Vulnerabilities in Web Application", Next Generation of Internet of Things, Springer Nature, Singapore, 2023.

[3] Odiaga Gloria Awuor, "Review of the Security challenges in web-based systems", World Journal of Advanced Engineering Technology and Sciences, 08(02), 204-216, 2023.

[4] Awang, N.F., Manaf, A.A, "Detecting Vulnerabilities in Web Applications Using Automated Black Box and Manual Penetration Testing". In: Awad, A.I., Hassanien, A.E., Baba, K. (eds) Advances in Security of Information and Communication Networks. SecNet 2013. Communications in Computer and Information Science, vol 381. Springer, Berlin, Heidelberg. 2013.

[5] Khare, N., Kadam, S. "Comparative Analysis of Vulnerability and Threats Related to Blogging Platforms", In: Rajagopal, S., Popat, K., Meva, D., Bajeja, S., Mudholkar, P. (eds) Artificial Intelligence Based Smart and Secured Applications. ASCIS 2024. Communications in Computer and Information Science, vol 2429. Springer, Cham, 2024.

[6] Enze Wang, Jianjun Chen, Wei Xie, Chuhan Wang, Yifei Gao, Zhenhua Wang, "Where URLs Become Weapons: Automated Discovery of SSRF Vulnerabilities in Web Applications", IEEE Symposium on Security and Privacy, 2024.

[7] Suliman Alazmi, Daniel Conte De Leon, "A Systematic Literature Review on the Characteristics and Effectiveness of Web Application Vulnerability Scanners", IEEE Access, 2022.

[8] OWASP Top 10, https://op-c.net/blog/owasp-top-10-2025-updates-changes-web-application-security/, accessed during Jan 2026.

[9] Protect your Applications against all OWASP Top 10 Risks, Imperva, https://www.imperva.com/docs/IM_eBook_Ten_OWASP_Threats.pdf, accessed during Jan 2026.
[10] Crowdstrike 2026 Global Threat Report – Year of the Evasive Adversary, https://go.crowdstrike.com/rs/281-OBQ-266/images/Crow dStrike-2026-Global-Threat-Report.pdf, accessed during Jan 2026.
[11] Yi Cheng, Julia Deng, Jason Li, Scott DeLoach, Anoop Singhal, Xinming Ou, "Metrics of Security", https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=917850, accessed during Jan 2026.
[12] Vulnerability metrics, https://nvd.nist.gov/vuln-metrics/cvss#, accessed during Jan 2026.
[13] Acunetix, "Testphp.vulnweb.com – Intentionally Vulnerable Web Application, Available: http://testphp.vulnweb.com, Accessed: 2026.