A Literature Review on Web Application Security

S.Swetha[1], Dr.S.Selvi [2]

[1]Pre-Final year student, [2]Associate Professor, Department of Computer Science and Engineering, Government College of Engineering, Bargur, Krishnagiri, Tamilnadu, India. Email-s.selvi@gcebargur.ac.in

Abstract - In a rapidly evolving virtual panorama, the vulnerability of web applications to various cyber threats needs proactive measures. Web application security is crucial because it protects confidential information, prevents unauthorized access, and safeguards users from diverse online threats. Neglecting security can lead to statistics breaches, identification theft, and compromise of private information. This paper evaluates diverse web application assault detection mechanisms and how resistant they are to various attacking techniques.

Keywords- Crawling, Cross-site Scripting, SQL Injection, Vulnerability Detection, Web Vulnerability Scanners.

I. INTRODUCTION

Web application security is an essential truth in safeguarding online structures from potential threats and vulnerabilities. Vulnerability is a protection risk that permits an attacker to compromise the device's security. Vulnerability is classified as Logical Vulnerability and Technical Vulnerability. The logical vulnerability represents glitches that are probably available within the web application due to bad structures used in code. Technical vulnerability is taken into consideration as a common vulnerability and an extensive variety of tools and sources are to be in deal with it. SQL Injection, Cross-site Scripting, Cross-site Request Forgery, and many others., are examples of technical vulnerabilities. Vulnerability assessment and Penetration testing (VAPT) is a unique and extraordinary protection approach that may be used to get rid of each technical and logical vulnerability from any of the layers. Its major aim is to audit the special layers of the system via external professionals. VAPT is a part of the software program checking out technique which especially focuses on security. So VAPT may be carried out with traditional software testing strategies which include Black box testing, White box testing, and grey box testing [1].

Comparing the web application security dangers primarily based on the research from leading practices which might be followed as an application security standard that covers around 80-90% of all common assaults and threats. To prevent attacks Open web application security top ten list is considered as standard for Vulnerability evaluation. Run time Application Self-Protection (RASP) is a technique that executes on a server and kicks in while an application is running. It is designed to discover attacks on an application in real time. Fig .1 shows the block diagram of RASP security layer. whilst an application starts to run, it can protect it from untrusted access or behavior via reading both the application's behavior and the context of that conduct [2].

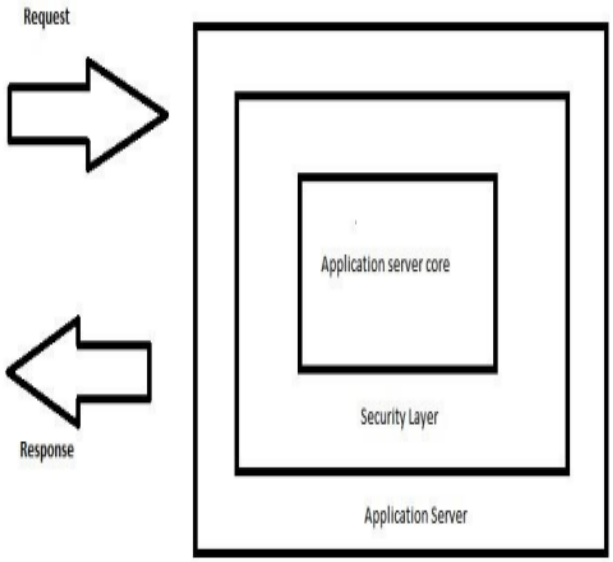

Figure 1 Block diagram of the RASP security layer

The Open Web Application Security Project (OWASP) report listed the top 10 maximum common web application vulnerabilities and the injection type is currently ranked first. Moreover, in line with a Nation of the Internet report, injection attacks are the pinnacle hazard, accounting for almost -two-thirds of all assaults in 2019. OWASP pinnacle 10 is likewise called an "awareness report". It affords a great statistical evaluation of the maximum excessive vulnerabilities in web applications [18]. SQL injection and cross-site monitoring are the most common vulnerabilities discovered inside the web application safety [9,17]. Table 1 lists the ten most critical web security applications which are given below.

 DOI : 10.35291/2454-9150.2024.0009

Table 1 The Ten Most Critical Web Application Security Risks

NO	RISK
1	A1-Injection
2	A2-broken Authentication and session management
3	A3 –Cross-Site Scripting (XSS)
4	A4 – Broken Access Control
5	A5 – Security Misconfiguration
6	A6 – Sensitive Data Exposure
7	A7 – Insufficient Attack Protection
8	A8 – Cross-Site Request Forgery (CSRF)
9	A9 – Using Components with Known Vulnerabilities
10	A10 – Under-protected APIs

An automated web vulnerability scanner to find efficient result techniques depends upon the imitation of SQLI, XSS, CSRF, and LFI/RFI vulnerabilities payload. consequently, the possibility of analyzing is constrained only to HTTP responses obtained from the application server that runs established web applications. It consists of web crawling, AEP's (software access points) detection and extraction, attacking, evaluation, and report generation. An open-source web vulnerability scanner that uses the black box method to carry out crawling and scanning for websites, to successfully stumble on the presence of exploitable web vulnerabilities. This device is unbiased of a database of known vulnerabilities rather exceptional, underlying properties of application-level vulnerabilities are exploited to successfully discover affected programs [3].

Also, Symantec's internet security threat report (ISTR) of 2019 indicated 59% growth in web program threats, and that 1 in 10 URLs analysed in 2018 became recognized as malicious (Symantec, 2019). As a result, web application protection has grown to be more and more tough in today's environment. Web Application Security Consortium (WASC) added the WASC classification to enable clear and organized web security vulnerabilities and attacks. The task was created to make terminology for web-related protection elements accessible for web developers, security professionals, software publishers, and listeners [4].

Web assessment tools are categorized into two categories: open-source tools and proprietary tools. There are a few limitations and capabilities associated with these classes. Open-source tools are free to download, but the builders additionally have access to their source codes. consequently, builders can customize their functions and features in line with their necessities. there may be a huge network linked with this open-supply technology, in which a developer can locate assistance to personalize open-source tools. Then again, proprietary or paid tools are not freely to be given. However, their beta versions or trial versions are freely available online. Downloading a tribulation version gives very limited features of these tools. In addition, their source codes aren't free to be had, so those tools cannot be customized [5].

Varieties of popular attacks on websites and web programs can be listed as SQL injection (SQLi), cross-site Scripting (XSS), Command injection (CMDi), path traversal, defacements, and DoS/DDoS. Embedding security into a software development lifecycle (DLC) encompasses a set of various strategies and checks at exceptional stages, e.g. Static software security testing (SAST) at an early level of DLC, and Dynamic application security testing (DAST) at testing and operation tiers. SAST scans source code like a white box testing from the internal out, at the same time as DAST implements black box testing of the runtime behavior at the same time as executing it from the outside in. Comprehensive application protection solutions are incredibly suitable to maximize the coverage of ever-evolving cyberattacks. web application vulnerability scanners (WAVS) as "automatic tools that are used to scan web programs and come across net vulnerabilities, additionally called black-box vulnerability scanners". Furthermore, they may be often called point-and-shoot (PaS) penetration testing tools that test web applications robotically [6].

Web application vulnerability scanners (WAVS) include three modules: crawling module, attacking module, and analysis module [7]. The Crawling is made with the aid of a factor referred to as Crawler. This explores the web application to get a better and perceive the web pages, the related input vectors such as fields of input of the HTML forms, request parameters GET and POST, and cookies. Except, the crawler creates an indexed listing of all crawled pages. The detection of the presence of web vulnerabilities depends basically on the first-rate of the Crawler. The fuzzing is made through an element known as fuzzier, this element generates potentially vulnerable values to cause a vulnerability for each entry and vulnerability kind for which the web software vulnerability scanner assessments. The attacking module places the outcomes acquired through the fuzzing phase under evaluation to discover the presence of the vulnerabilities and provide feedback to the alternative modules [8].

There are two most common security vulnerabilities nowadays such as SQL injection and cross-web site scripting. A safety assessment of the application defense center, which had more than 250 e-trade applications, online banking, and corporate websites came up with a statement that there are 85% of web applications are at risk of assaults. The common use of SQL injection attacks is to abuse internet pages that permit users to input data into form fields for database queries. Cross-site scripting (XSS) is likewise a serious trouble of web applications that may be used by an attacker. The attacker can insert the malicious script in web software via any external aid [10].

 DOI : 10.35291/2454-9150.2024.0009

The methods and techniques to guard the web programs can vary from administrational to technical, from prevention to safety, and from coding level to tracking level. several online assets and organizations exist in recent times that regularly update their websites with the recent knowledge or information of threats, assaults, or vulnerabilities in the internet era. OWASP and WASP are examples of such nonprofit organizations [11].

Client-side security, client-side vulnerabilities, and client-side attacks are terms used in cybersecurity to explain security incidents and breaches that arise at the customer's (or users') computer system in place of the company's (at the server side) or anywhere in between. Server-side attacks try to hack and breach the facts and programs on a server. The suggested strategy for client-side servers encompasses SQL injection assault, URL injection attack, meditated XSS and server-side includes broken authentication, security setup, and sensitive data exposure [12].

Security protection techniques available on the server side are AJAX protection mechanism, Input validation, Security of client's program code, SOAP filtering and WSDL strengthening mechanism, Authentication, authorization, and development mechanism of the security program. Security protection techniques available are HTTPS protocol, SRTP protocol, and RTMPS protocol [13].

WA firewalls (WAFs) are the common front-end security mechanism for web-based applications that are continuously below assault. There are two classes of WAF open supply and industrial WAFS. Examples of open-source solutions that may be used to install a firewall to prevent web applications are AQTRONIX WebKnight and ModSecurity. Examples of business solutions that may be used to install a firewall to prevent internet programs are dotDefender, Imperva SecureSphere, and Barracuda [14]. 60% of companies said they have a WAF (web application Firewall) in a few states of deployment [15].

This paper is organized as follows. Section 2 describes the web application security scanner criteria. Section 3 presents the Evaluation metrics. Real-life Applications are given in Section 4. The summary of the paper is concluded in Section 5.

II. WEB APPLICATION SECURITY SCANNER EVALUATION CRITERIA (WASSEC)

The web application security Consortium evolved WASSEC, a seller-neutral record to help safety professionals examine web application scanners and pick out the most suitable tool. the following list describes the capabilities that ought to be considered while evaluating web software safety scanners:

- Protocol aid: The scanner should support all communication protocols that are regularly used by web applications. moreover, proxy talents, together with the hypertext transfer protocol (HTTP) and Socks proxies, must be supported.
- Authentication: The scanner should be able to maintain all authentication strategies typically used in a web application.
- Session Control: During a security test, a scanner must maintain a valid session with the application.
- Crawling: The scanner ought to have a function that can move slowly web software thoroughly primarily based on the user-described configuration.
- Parsing: To obtain information approximately, the capability and layout of the scanned web application, the scanner ought to be capable of parsing the maximum widely used web technology.
- Testing: The scanner used to be capable of coming across security vulnerabilities and architectural flaws in an internet application. It used to additionally provide the consumer with configuration alternatives to customize an experiment.
- Command and Control: The scanner ought to have command and manipulation functions that enhance the user experience. As an example, it schedules scans, pauses and restarts them, and schedules numerous scans simultaneously.
- Reporting: After each scan, a scanner should be able to produce a custom report [7,16].

III. EVALUATION METRICES

- True positives (TPs) are the vulnerabilities detected through a scanner that surely exist within the code.
- False positives (FPs) are vulnerabilities detected through a scanner that does not exist. FPs pose a significant problem to customers. If the FPs are high, the user inspects each suggested vulnerability manually to assess its validity.
- False negatives (FNs) are the vulnerabilities that exist within the code but are not detected by using the scanner.
- Precision is the ratio of successfully detected vulnerabilities to the entire quantity of detected vulnerabilities and is additionally called actual true positive accuracy, positive predictive value, or confidence, which is represented in Eq.1 as follows

$$\text{Precision} = \frac{TP}{TP + FP} \qquad (1)$$

- Recall is the ratio of successfully detected vulnerabilities to the wide variety of total present vulnerabilities also referred to as true positive rate or Sensitivity, which is represented in Eq.2 as follows

$$\text{Recall} = \frac{TP}{TP + FN} \qquad (2)$$

 DOI : 10.35291/2454-9150.2024.0009

- F-degree is the harmonic mean of precision and recall, which is represented in Eq.3 as follows;
-

$$F - Measure = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

It has to be reminded that the more the Precision is, the smaller the wide variety of false positives gets. As a result, the tool turns into more correct in detecting the vulnerability involved. It should additionally be reminded that, the bigger the recall is, the smaller the false negative number turns into. therefore, the tool detects the vulnerability better. Table 2 lists the precision rate of Several Web security scanners [5].

Table 2 represents the precision values of various scanners which are given below.

Table 2 Precision values for various scanners

Evaluated Scanner	CSS		SQLI		Precision
	TP	FN	TP	FN	
Acunetix	139	0	66	0	100%
IBM APPSCAN	6	0	49	0	100%
Nessus	200	0	43	0	100%
BurpSuite	136	1	62	0	99.2%
Wapiti	11	2	4	0	90%
Arachni	136	1	60	0	99.5%
WebInspect	8	0	11	0	100%
Nikto	9	1	11	0	99.5%
Netsparker	136	0	64	0	100%
W3af	81	0	19	0	100%
OWASP-ZAP	136	0	63	0	100%

IV. REAL TIME APPLICATIONS

Web software security has actual-existence applications throughout various industries, ensuring the protection of sensitive records and retaining the integrity of online offerings. a few examples consist of:

Financial Institutions:

• Online Banking safety: protecting user debts and monetary transactions from unauthorized access to or fraudulent activities.

• Payment Gateways: ensuring relaxed transactions and defensive price records in e-commerce systems.

Healthcare:

• Electronic health records (EHR): Securing patient data in online scientific facts structures to maintain privacy and comply with healthcare policies.

• Telemedicine systems: ensuring the confidentiality and integrity of sensitive affected person statistics throughout far-off consultations.

E-commerce:

• Customer Records Protection: Safeguarding purchaser information, along with private and economic information, in online purchasing platforms.

• Transaction Safety: Securing charge techniques and preventing fraudulent activities in e-trade transactions.

Government services:

• Citizen Portals: defensive citizen facts and ensuring comfy access to authority's offerings via online systems.

• E-authorities systems: Securing sensitive information and communications within government internet packages.

Education:

• Student Records Protection: ensuring the security of student statistics and personal facts in online learning systems.

• Academic Portals: protecting access to educational sources and keeping the integrity of academic information.

Technology and Software program development:

• Developer Portals: making sure to secure access to APIs, SDKs, and developer equipment to guard intellectual property and prevent unauthorized access.

• Version Manipulate Systems: Securing source code repositories and preventing unauthorized code adjustments.

Media and enjoyment:

• Streaming services: protective user debts, charge details, and content from unauthorized access on online streaming structures.

• User Authentication: making secure login tactics for consumer accounts on media and entertainment websites.

Critical Infrastructure:

• industrial control structures (ICS): Securing web interfaces and control structures in critical infrastructure sectors which include energy, transportation, and manufacturing.

These applications spotlight the various approaches in which web application protection is essential for safeguarding sensitive statistics and retaining the functionality of online services in diverse sectors.

 DOI : 10.35291/2454-9150.2024.0009

V. CONCLUSION

This paper discussed the major threats in web applications and the fundamental information needed to prevent web exploits. Security threats have become more common nowadays because of emerging technologies in various fields like e-commerce, healthcare, and financial institutions. It also provides various methodologies like Vulnerability Assessment and Penetration Testing (VAPT) and run-time Application Self Protection (RASP) to make a secure web application.

This study elaborates the detection of web application vulnerabilities using scanners. The web application scanning tools are evaluated using measures such as detection rate accuracy, precision, capability to come across special vulnerabilities, and severity level. Web security scanners offer valuable automatic tools for identifying vulnerabilities and strengthening the security posture of web applications. A balance between computerized tools and human expertise is prime to efficiently mitigating dangers and maintaining robust web security in the face of evolving cyber threats.

REFERENCES

[1] A perusal of Web Application Security Approach; Ashikali M Hasan, Divyakant T. Meva, Anil K Roy, Jignesh Doshi (2017).

[2] A Survey on Web Application Security; Danish Mairaj Inamdar (2020).

[3] Web Unique Method (WUM): An Open Source Blackbox Scanner for Detecting Web Vulnerabilities; Muhammad Noman Khalid, Muhammad Iqbal, Muhammad Talha Alam, Vishal Jain, Hira Mirza and Kamran Rasheed (2017).

[4] A systematic review and taxonomy of web applications threats; Yassine Sadqi and Yassine Maleh.

[5] A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions; Jahanzeb Shahid, Muhammad Khurram Hameed, Ibrahim Tariq Javed, Kashif Naseer Qureshi, Moazam Ali and Noel Cresp (2022).

[6] Vulnerabilities Mapping based on OWASP-SANS: A Survey for Static Application Security Testing (SAST); Jinfeng Li (2020).

[7] Evaluation of Black-Box Web Application Security Scanners in Detecting Injection Vulnerabilities Muzun Althunayyan, Neetesh Saxena, Shancang Li, and Prosanta Gope.

[8] Performance Evaluation of Web Application Security Scanners for Prevention and Protection against Vulnerabilities; S. El Idrissi, N. Berbiche, F. Guerouate and M. Sbihi (2017).

[9] Analysis of Effectiveness of Black-Box Web Application Scanners in Detection of Stored SQL Injection and Stored XSS Vulnerabilities; Muhammad Parvez1, Pavol Zavarsky, Nidal Khoury.

[10] A Study on Web Application Vulnerabilities to find an optimal Security Architecture; Dr. C. Amuthadevi, Sparsh Srivastava, Raghav Khatoria, and Varun Sangwan.

[11] The Reality of Applying Security in Web Applications in Academia; Mohamed Al-Ibrahim, Yousef Shams Al-Deen (2014).

[12] A Study on Web Application Security and Detecting Security Vulnerabilities Sandeep Kumar, Renuka Mahajan, Naresh Kumar, Sunil Kumar Khatri.

[13] Research on Security Technology based on WEB Application; Fanxing Kong.

[14] The Reality of Applying Security in Web Applications in Academia; Mohamed Al-Ibrahim, Yousef Shams Al-Deen (2014).

[15] An OWASP Top Ten Driven Survey on Web Application Protection Methods; Ouissem Ben Fredj, Omar Cheikhrouhou, Moez Krichen, Habib Hamam, and Abdelouahid Derhab.

[16] Vieira, Antunes, & Madeira, using web security scanners to detect vulnerabilities in web services. In IEEE/IFIP International Conference Conference on Dependable Systems & ESTORIL (2009).

[17] Research challenges and issues in web security; Parveen Sadotra (2015).

[18] web application security; Dr.Shalu Tandon, Devasnshi Chopra, Ankit Bewal, Sayantika Manna.

 DOI : 10.35291/2454-9150.2024.0009