# SAT: Integrated Multi-agent Blackbox Security Assessment Tool using Machine Learning

Jahanzeb Shahid*, Zia Muhammad†, Zafar Iqbal†, Muhammad Sohaib Khan*, Yousef Amer§, Weisheng Si‡
*Department of Information Security, National University of Sciences and Technology (NUST), Pakistan
†Department of Cybersecurity, Faculty of Computing and AI, Air University Islamabad, Pakistan
§School of Engineering, University of South Australia, §Department of Engineering, Khalifa University, UAE
‡School of Computer, Data and Mathematical Sciences, Western Sydney University, Australia
Email:*{jahanzeb.ncsael, sohaib.niazi}@mcs.edu.pk, †{zia.muhammad, zafar.iqbal}@mail.au.edu.pk
§Yousef.Amer@unisa.edu.au, §yousef.amer@ku.ac.ae †W.Si@WesternSydney.edu.au

2022 2nd International Conference on Artificial Intelligence (ICAI) | 978-1-6654-6896-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICAI55435.2022.9773750

***Abstract*—The widespread adoption of eCommerce, iBanking, and eGovernment institutions has resulted in an exponential rise in the use of web applications. Due to a large number of users, web applications have become a prime target of cybercriminals who want to steal Personally Identifiable Information (PII) and disrupt business activities. Hence, there is a dire need to audit the websites and ensure information security. In this regard, several web vulnerability scanners are employed for vulnerability assessment of web applications but attacks are still increasing day by day. Therefore, a considerable amount of research has been carried out to measure the effectiveness and limitations of the publicly available web scanners. It is identified that most of the publicly available scanners possess weaknesses and do not generate desired results. In this paper, the evaluation of publicly available web vulnerability scanners is performed against the top ten OWASP[1] vulnerabilities and their performance is measured on the precision of their results. Based on these results, we proposed an Integrated Multi-Agent Blackbox Security Assessment Tool (SAT) for the security assessment of web applications. Research has proved that the vulnerabilities assessment results of the SAT are more extensive and accurate.**

***Index Terms*—Web Applications, Web Security, Web Vulnerability Scanner, OWASP, DVWA, Pentesting, Security Testing.**

## I. Introduction

The internet plays a significant role in our lives. Nowadays, people interact with each other with the help of customized and user-friendly web applications developed using different languages, platforms, and scripting environments. During website development, some seamlessly benign mistakes are left behind without developer intention and these are unintentional most of the time. Afterward, when these websites are deployed and made publically available then attackers take advantage by exploiting these kinds of vulnerabilities. The heterogeneous nature and dynamic environment of web applications make it difficult for a developer to fully secure and timely update their website against emerging threats and attacks. That is why there is a need for security assessment tools and vulnerability scanners. A security assessment tool scan website against known vulnerabilities and provides a reflection of a website that can be helpful for developers and pentesters in order to fix holes before a website comes online. This in-house assessment is important to identify safety, completeness, successes, and quality of web applications before deployment on servers.

[1]**OWASP®** The Open Web Application Security Project (OWASP) is an online community that produces comprehensive articles, documentation, methodologies, and tools in the arena of web and mobile security.

TABLE I
List of Abbreviations

| Abbreviation | Description |
|---|---|
| OWASP | Open Web Application Security Project |
| VAPT | Vulnerability Assessment and Penetration Testing |
| WAVSEP | Application Vulnerability Security Evaluation Project |
| WIVET | Web-Input Vector Extractor Teaser |
| DAST | Dynamic Application Security Testing |
| NIST | National Institute of Standards and Technology |
| STLC | Software Testing LifeCyce |
| DVWA | Damn Vulnerable Web App |
| AI | Artificial Intelligence |
| AIAT | Artificial Intelligence Applications Testing |
| ANN | Artificial neural networks |
| IoT | Internet of Things |
| FOSS | Free and Open-Source Software |
| ML | Machine Learning |
| APTs | Advanced persistent threat |
| PII | Personal Identifiable Information |
| API | Application Programming Interface |
| MCVE | Common Vulnerabilities and Exposures |
| CTI | Cyber Threat Intelligence |
| VPNs | Virtual private network |
| SAT | Security Assessment Tool |
| SQL | Structured Query Language |
| ZAP | Zed Attack Proxy |
| JSON | JavaScript Object Notation |
| OAuth | Open Authentication |
| DDoS | Distributed Denial of Service |
| HEC | Higher Education Commission |
| NCSAEL | National Cyber Security Auditing and Evaluation Lab |

According to the ”Barracuda application security Report”, on vulnerabilities in corporate information systems [1], it is presented that nearly 72% of organizations suffered at least one security breach from an application vulnerability. Similarly, loopholes during the development process cause serious threats to web applications. Changes in configuration settings are adequate to affix only 17% of vulnerabilities, most of those have a low severity level. On average, 33 vulnerabilities are found in each web application, among 6 of those are of high severity. Moreover, in the time of 2022, number of severe vulnerabilities per web application have grown 3 times as compared to 2017 [2] due to more threats.Moreover, White-Hat Security Organization [3] claims that, 86% of scanned

web apps have on an average 56% vulnerabilities in a web application. Among these, at least one is classified as a critical level. Another research conducted by Symantec [4] proves that 76% of websites possess at least one serious vulnerability, and 20% of servers consist of critical vulnerabilities. This was analyzed by executed almost 1400 scans. Although, several web vulnerability scanners are publicly available and several firms are legalized in order to perform auditing work. But web application attacks are non-stop and increasing exponentially as most publicly scanners are unable to cater all the possible vulnerabilities resides in a web application [13].

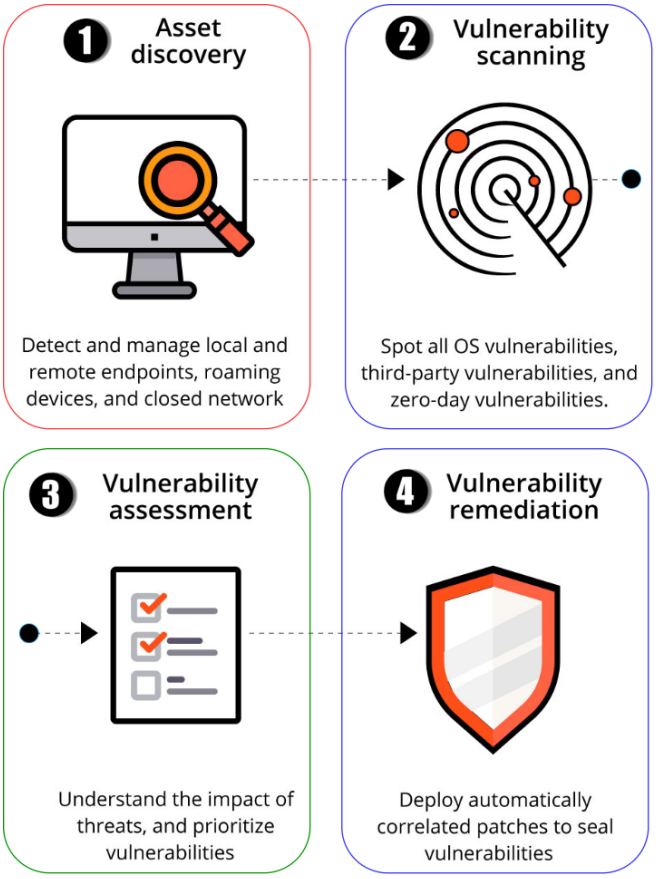


Fig. 1. Working of a vulnerability scanner.

In this paper, a detailed analysis of web vulnerabilities detection tools is presented against a well-defined OWASP. Fig. 1 present a quick overview for working of a vulnerability scanner. As visible, there are four parts of a scanning phase i.e. asset discovery, vulnerability scanning, assets discovery, and remediation. Similarly, on a broader level, all Web assessment tools are classified into two categories as open-source tools and commercial tools. At first, tools were run against DVWA (Damn Vulnerable Web Application) [5]. DVWA is a potentially vulnerable web app that is developed to test tools and professionals' skills in a legal environment. Subsequently, on these results, we proposed and developed its framework SAT for web applications scanning. Subsequently, SAT is employed for vulnerability scanning of the DVWA. In the end, it's identified that results are more accurate as compared to other publicly available tools.

**organization of the paper:** The paper is organized in the following sections: Section II provides a quick overview of existing literature in our domain. Section III introduces a comparison matrix for the evaluation of existing scanners. Section IV defines our assessment test bench and capability of existing tools. Section V provides our proposed framework and its high level architecture, while the conclusion and future work are presented in section VI.

## II. SIGNIFICANT EFFORTS OVER THE PERIOD OF TIME

In this section, we provide a brief overview of reviewed research contribution, benchmarks, and available tools. The details of these are included in the ensuing paragraphs.

Shahid, Waleed Bin. [6] proposed a research model to analyze the output of the generated by scanner and take an intelligent decision. The researcher performed rigorous testing on a dataset to classify specific vulnerabilities.Their research was mainly focused on the top OWASP vulnerabilities namely; CSS, SQLi, Remote code execution, and File inclusion. Their model's outcomes were based on the improvement of performance parameters like false alarm and accuracy for detection of four mentioned vulnerabilities. The scope of their work is very limited, which is addressing only four vulnerabilities on OWASP's list. Similarly Gupta, Aakanshi [7] surveyed the literature to determine the major techniques for Vulnerability Assessment and Penetration Testing (VAPT). They discussed different tools with their features like W3af, Havij, Fimap, Metasploit, Acunetix, and Nexpose. However, they didn't perform any comparative analysis against a benchmark.

Balume and Weisheng Si [8] compared the performances of open-source vulnerability scanners like Arachni and OWASP Zed Attack Proxy (ZAP) against two well-known benchmarks as OWASP and Web Application Vulnerability Security Evaluation Project (WAVSEP). They make some valuable recommendations to perform web application assessments. Their work is novel and its scope can be extended by enriching it with more publicly available tools. Moreover, Araújo [9] assessed different tools such as Paros, Wapiti, Skipfish, Nikto, Wfuzz, NetSparker, and HP web inspect. The author has evaluated the effectiveness, flaws, and limitation of the aforesaid tools. They tested these tools against the most severe attacks namely SQLI and XSS attacks. The scope of their work is limited as it explored only two vulnerabilities. Finaly, Fang, et al. [10] presented a framework that could categorize fingerprints of automated tools and analyzed the payload features of different web assessment tools by using a deep learning algorithm. This research is performed on five tools such as AppScan, AWVS, NetSparker, Vega, and W3af. Their proposed framework's claimed accuracy is 94.6%.

All of these research works are comprehensive efforts, but none of them evaluated publicly available tools against all top 10 vulnerabilities. However, in this paper, a total of 11 publicly available web application scanners have been compared

TABLE II
TOTAL NUMBER OF VULNERABILITIES CLASSIFIED AND DETECTED BY COMMERCIAL AND OPEN-SOURCE SCANNERS

| Vul | Commercial Scanners | | | | | Open-Source Scanners | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acunetix | HP WebInspect | NetSparker | APPSCAN | Nessus | ZAP | Nikto | W3af | Wapiti | Arachni | BurpSuite |
| A1 | 115 | 67 | 60 | 58 | 54 | 26 | 18 | 22 | 5 | 28 | 29 |
| A2 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| A3 | 145 | 90 | 113 | 88 | 29 | 11 | 18 | 31 | 42 | 15 | 90 |
| A4 | 2 | 2 | 1 | 0 | 1 | 11 | 0 | 0 | 1 | 1 | 1 |
| A5 | 51 | 30 | 15 | 25 | 5 | 8 | 4 | 11 | 18 | 17 | 12 |
| A6 | 23 | 27 | 25 | 21 | 15 | 0 | 10 | 13 | 2 | 27 | 27 |
| A7 | 165 | 180 | 140 | 130 | 100 | 743 | 545 | 411 | 223 | 180 | 179 |
| A8 | 133 | 52 | 115 | 95 | 49 | 13 | 8 | 15 | 10 | 8 | 55 |
| A9 | 12 | 15 | 11 | 10 | 15 | 4 | 6 | 8 | 0 | 13 | 6 |
| A10 | 65 | 95 | 55 | 45 | 52 | 36 | 43 | 26 | 39 | 25 | 52 |

against OWASP's top 10 vulnerabilities. This makes it a one-stop guide for the security assessment of web applications.

## III. PREREQUISITES FOR EVALUATION

This is mandatory section for through understanding of paper. It provides details about our selection criteria, evaluation benchmark, and list of tools we are going to evaluate. First of all, each of the selected tool will run on DVMA and will be scored on its ability to find vulnerabilities available in OWASP. Details of these are covered in ensuring paragraphs.

### A. OWASP Top 10 Vulnerabilities

The research is conducted using the Black box testing approach where web applications are audited without prior knowledge of the internal structure and source code. Overall testing is done against the OWASP standard. This is because the OWASP Top 10 [11] is a frequently updated benchmark focusing on web application security on the top ten critical risks. This benchmark standard is composed of a team of security experts from all over the world. OWASP Top 10 is also known as "Awareness Document", it provides extensive statistical analysis of the most severe vulnerabilities in web applications such as Broken Object Level Authorization (A1), Broken Authentication (A2), Excessive Data Exposure (A3), Lack of Resources and Rate Limiting (A4), Broken Function Level Authorization (A5), Mass Assignment (A6), Security Misconfiguration (A7), Injection (A8), Improper Assets Management (A9), and Insufficient Logging and Monitoring (A10).

### B. Damn Vulnerable Web Application (DVWA)

All the selected tools are tested against DVWA. The DVWA is selected because most of the OWSAP Top 10 are present in it such as Command Injection, Brute Force, CSRF, File Upload, File Inclusion, Insecure Captcha, SQL Injection blind, SQL injection, CSS, CSS-DOM, and CSS reflected attacks.

### C. Scanners Selection

There are several aspects involved in selecting scanners for this research work [12]. We selected tools according to the five properties. i.e. (1) Number of protocols/ algorithm supported, (2) Input delivery methods, (3) Vulnerabilities detection capability in web with low False Positive (FP) rate, (4) Easily accessible and regularly updated, (5) Tool is either freely available or offers a trial version. Moreover, we acknowledge that no paid version is taken or compared in any part of our research. All selected tools are either open source or free.

Based on these listed properties, a total of 11 scanners are shortlisted and they are divided into Open source and commercial scanners as can be seen in Table. II. There are 5 commercial scanners while 6 are selected from open-source scanners. A brief overview of these scanners is provided in the following paragraphs for ease of readers' understanding.

#### *Commercial Scanners*

*1. Acunetix WVS:* It is a web vulnerability checker. It is capable of analyzing web applications by exploiting XSS, SQL injections, Host Header Injection, and over 3000 other web vulnerabilities [14].

*2. HP WebInspect:* It is a powerful tool that can launch more than 3000 attacks on web systems. [15].

*3. NetSparker:* A Web scanner designed to discover web vulnerabilities such as XSS and SQL Injection. [16].

*4. AppScan:* This tool provides centralized control with several additional functionalities such as advanced application scanning, remediation capabilities, enterprise application security status metrics along seamless integration with AppScan Standard. The tool also provides user's administration. [17].

*5. Nessus:* It can detect multiple security vulnerabilities in the OS of targeted hosts [18], software patches, and services. Moreover, it has hundreds of plugins that can be employed for detailed and customized scans.

#### *Open-Source Scanners*

*1. ZAP:* The OWASP Zed Attack Proxy (ZAP) is a popular web security assessment tool. It is entirely made available as an open-source project is maintained by OWSAP [19].

*2. Nikto:* It can execute and scan for multiple items including malicious files/ programs, and outdated versions in both software libraries and web servers. It performs scans on configuration files of web servers such as multiple index files, server fingerprinting, and HTTP calls settings [20].

*3. W3af:* W3af (Web Application Attack and Audit Framework) is used to detect and exploit web application vulnerabilities. It is available both as a command line and graphical package. This toolkit is also called the "Metasploit of the Web". It detects vulnerabilities using black-box technique [21].

*4. Wapiti:* It is a generic command-line tool that automates the audit process [22]. It requires minimum user interaction. It can detect vulnerabilities like Injection, CSS, Command Execution Attacks, CRLF Injection, and File Disclosure.

*5. Arachni:* This tool is based on modular and Ruby languages. It trains itself by learning from HTTP (Hypertext Transfer Protocol) responses received during the scanning and testing during assessment [23].

*6. BurpSuite:* It is a Java-based popular penetration testing toolkit [24]. It helps to verify attack vectors and detects vulnerabilities that are directly affecting web applications such as authentication, injection, and security misconfigurations.

## IV. Strength of Security Assessment Tools

During this research, the quality of web application scanners is analyzed based on the following three characteristics and results are discussed based on these three parameters.

1) Quality of Crawling (QoC): Crawling is a process that identifies pages of a web application that is vulnerable to a certain attack. The number of pages crawled by a scanner determines its quality of crawling.
2) Quality of Fuzzing (QoF): The quality of fuzzing depends on the number of inputs that a fuzzer enters to find a certain vulnerability. These inputs must be capable to exploit application vulnerabilities.
3) Quality of Analyzing (QoA): It is the responsibility of the analyzer to analyze the Fuzzer results. It should detect with fewer False Positives (FPs).

### A. Comparative Analysis

Both commercial and open-source tools are tested against OWSAP Top 10 vulnerabilities. The total number of web vulnerabilities classified and detected by tools are listed in Table. II. It can be identified that the vulnerability detection rate of Acunetix is higher in the category of CSS, File Injection, Sensitive Data Exposure, CSRF, and Broken Authentication vulnerabilities. Moreover, This can be seen that OWASP-ZAP has a lower detection rate in several categories like CSS, File Injection, and Insecure Communication but is highly capable to detect security configurations.

### B. Classification of Vulnerabilities

Selected tools are analyzed and their acquired results have been categorized according to the severity level of identified vulnerabilities. These vulnerabilities are categorized into four categories namely High, Medium, Low, and Informational level vulnerabilities. The severity level of vulnerabilities identified by each tool is visualized in Fig. 3. Starting from the left-hand side, this can be seen that Acunetix detected 255 high-level vulnerabilities, 200 medium-level vulnerabilities, 103 low vulnerabilities, and 275 informational vulnerabilities.

On the other hand, Fig. 4 shows the vulnerabilities detection capabilities of open-source tools. This can be seen that OWASP-ZAP is ranked 1st and Burp Suite stood last in the detection of low severity vulnerabilities. Starting from the left-hand side, this can be seen that Burp Suite detected 120 high-level vulnerabilities, 8 medium-level vulnerabilities, 20 low-level vulnerabilities, and 420 informational vulnerabilities.

### C. Evaluation of Achieved Results

In the black-box evaluation of vulnerable web applications, there is a higher possibility that the scanner could generate FP's results. This is because, the scanners mostly rely on the information from service banners and signature matching checks, which leads to false detection. Hence, it is important to verify the results by checking their validity and precision manually. We derived mathematical formulas to evaluate results based on precision, accuracy, and recall. Details of these are as follows.

1) *Precision of Scanner:* It is the percentage of correctly detected vulnerabilities i.e, If, the accurately detected vulnerabilities are (TP) and falsify detected vulnerabilities are (FP) then, formula for calculation of prescision is given in Eq:1

$$Precision(\%) = \frac{TP}{TP + FP} - (1)$$

For ease of understanding, the precision of two vulnerabilities namely SQLi and CSS has been calculated for all discussed tools in Table. III. Others are not added in this paper but formulas are provided for further calculations. This can be seen that the precision of Acunetix is 100% while IBM Appscan has an accuracy rate of 84%.

TABLE III
Precision Rate of Evaluated Scanners

| Evaluated Scanner | CSS | | SQLI | | Precision |
|---|---|---|---|---|---|
| | TP | FP | TP | FP | |
| Acunetix | 139 | 0 | 66 | 0 | 100% |
| IBM APPSCAN | 6 | 5 | 49 | 0 | 84% |
| Nessus | 200 | 21 | 43 | 5 | 90.88% |
| BurpSuite | 136 | 3 | 62 | 0 | 98.5% |
| Wapiti | 11 | 7 | 4 | 6 | 53% |
| Arachni | 136 | 5 | 60 | 0 | 81.32% |
| WebInspect | 8 | 7 | 11 | 17 | 44.1% |
| Nikto | 9 | 2 | 11 | 6 | 71.4% |
| Netsparker | 136 | 3 | 64 | 0 | 98.5% |
| W3af | 81 | 3 | 19 | 3 | 94.3% |
| OWASP-ZAP | 136 | 0 | 63 | 0 | 100% |

2) *Accuracy of Scanner:* Accuracy is the percentage of the sum of correctly detected true positives (TP) and true negatives (TN) to the summation of all classified instances, i.e, True Positive (TP), False Positive (FP), False Negatives (FN), and True Negatives (TN). The formula to calculate accuracy give in Eq:2. It depicts the closeness of the true value to the predicted value.

$$Accuracy(\%) = \frac{TP + TN}{TP + FP + FN + TN} - (2)$$

3) *Recall:* Recall is the calculation of correctly identified positive instances. Like it is also referred as the ability

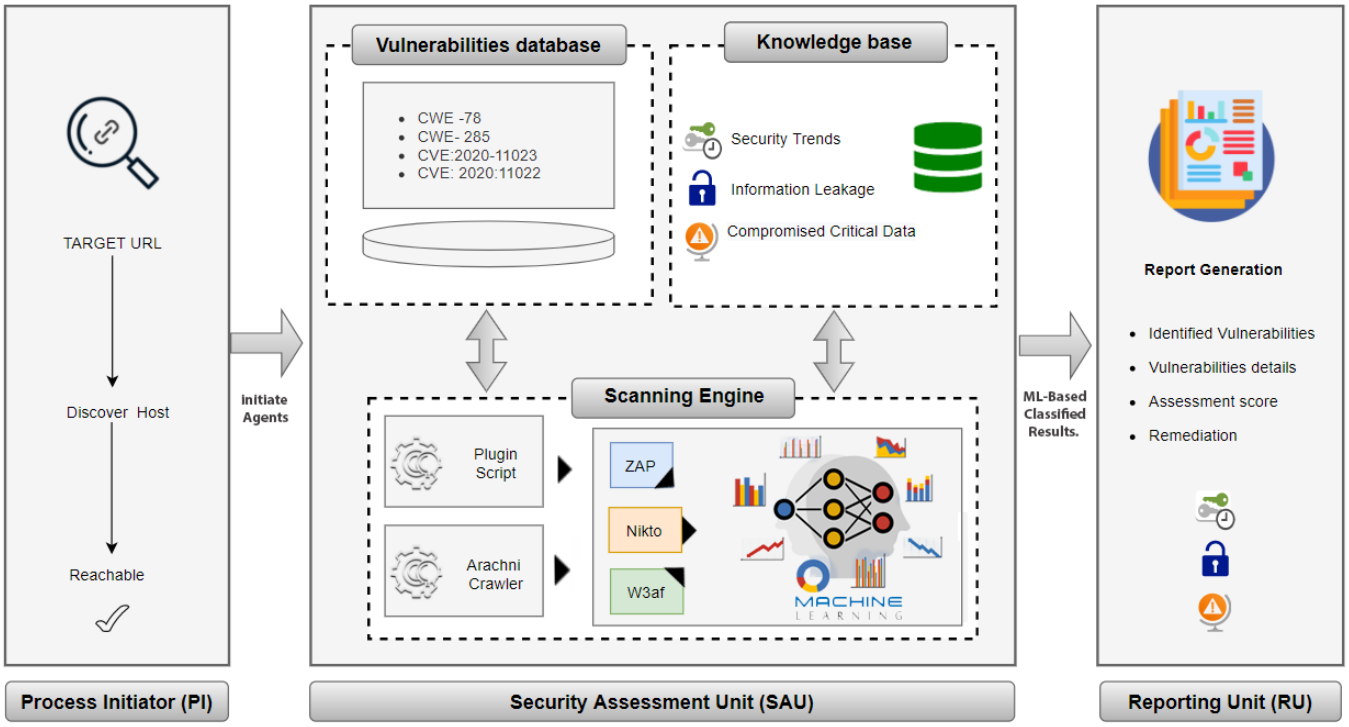


Fig. 2. SAT Framework - Design and Architecture

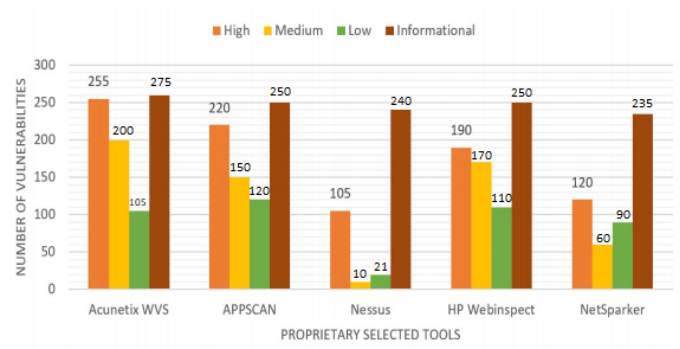


Fig. 3. Commercial Scanners capability to detect severity of vulnerabilities.

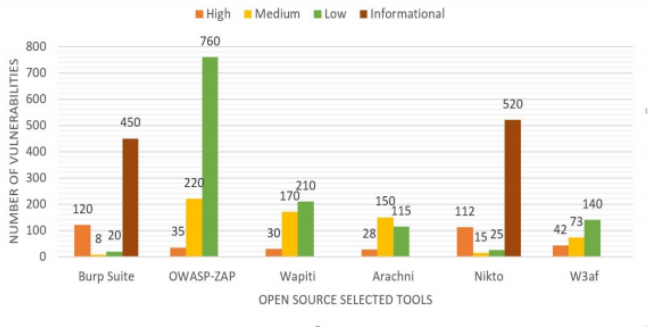


Fig. 4. Open-Source Scanners capability to detect severity of vulnerabilities.

of a scanner for remembering something learned during training. It can be calculated by the formula in Eq:3

$$Recall = \frac{TP}{TP + FN} - (3)$$

### D. Gap Analysis:

All of the discussed tools are comprehensive community efforts but they posses some limitations as well, these limitations are described here and we caterfor these in ou Framework.

- An individual tool cannot detect all the vulnerabilities present in web app. For instance, the Wapiti scanner is unable to detect broken authentication and improper assets management. This can be seen in Table. II.
- Access Control flaws, hardcoded back-door, and identification of multi cover attack is difficult to detect and harder to detect with these tools [25].
- There are no specific goals associated with automated penetration testing tools. Therefore, these automated tools have to check every possible flaw in web app [26].

## V. Proposed Framework

After a thorough analysis of web security assessment tools, It can be seen that each tool possesses different capabilities and no single tool is capable enough to counter all OWASP top 10 vulnerabilities. In this section, we proposed our framework to efficiently counter web scanners' weaknesses. The proposed framework caters to aforesaid challenges and able to detect all of the OWASP top 10 vulnerabilities discussed in III-A.

### A. SAT Framework - Design and Architecture

The proposed framework SAT is depicted in Fig. 2. It has three major components namely Process Initiator (PI), Security Assessment Unit (SAU), and Reporting Unit. Details of these components is provided in the ensuing paragraphs.

**Process Initiator (PI):** User input targeted URL in this module. It is responsible for host discovery and initialization of the scanning process. Unreachable hosts are screened here and further process is terminated.

**Security Assessment Unit (SAU):** This module performs scanning of the input web application. It has three subsections as Scanning Engine, Vulnerabilities database, and Knowledge-base. The scanning engine is in mutual compliance with Zap, Nikto, and W3af. Along with these 3 tools, It uses Arachni Crawler and provides features for cos tom plugin scripts like OAuth, etc., Vulnerabilities database contains a list of all possible vulnerabilities of OWASP top 10, and Knowledge base is an AI (Artificial Intelligence) based analysis engine responsible to identify security trends, Information leakage and highlight compromised critical data of scanned organizations. For comparative analysis and thorough results we shortlisted Decision Tree in combination with Artificial neural network (ANN). Decision tree algorithms use supervised machine learning techniques to solve classification problems. It is capable to identify multiple input classes with higher precision and accuracy. It works like a flow chart to separate data points in multiple categories and branches and correlate similarities to find similarity index for specific inputs. Similarly, ANN is a neural network inspired by the biological neural networks of the brain. It is a combination of connected nodes called artificial neurons that process input into output through multiple layer amendments.

**Reporting Unit (RU):** Reporting unit is responsible for the generation of a detailed report that contains identified vulnerabilities along with their details, Assessment score, and possible remediation.

Afterward, this report can be used for manual analysis and patching process by a security analyst and software developer.

### B. Evaluation

*1) Crawler coverage:* We identify that the effectiveness of a tool depends upon its capability to efficiently crawl various web pages. Similarly, a tool with week QoC will not be able to perform good coverage and result in less efficiency. This is why we analyzed tools against their default and external crawler. Therefore, an approach was chosen to integrate a better crawler component. From our evaluation, it was assessed that Arachni provides better crawling performance as compared to other open-source crawlers like ZAP, W3af, and Wapiti. To check the crawling coverage of a scanner, WIVET (Web-Input Vector Extractor Teaser) version 4 was used. It is a bench-marking project specifically designed to assess crawling coverage. Table. IV compares crawling coverage results of different scanners with their default and WIVET crawler.

Fig. 5 presents a graphical representation of old and new detected vulnerabilities by using an integrated crawler. It shows that results have been significantly improved by using the Arachni crawler for generating URL lists with the OAuth authentication framework. OAuth is an open standard used for access delegation and it is commonly used as a way for Internet users to grant access. In the next step, these URLs are provided as a seed value to other open-source scanners (ZAP, W3af, and Wapiti), and their vulnerabilities, detection have been improved.

TABLE IV
SITE COVERAGE RATE BEFORE AND AFTER USING WIVET CRAWLER

| Scanner | Raw Coverage | Coverage when Seeded with WIVET Crawled Results |
|---|---|---|
| Arachni | 95% | 96.5 |
| OWASP-ZAP | 68% | 93.5% |
| W3af | 22% | 93.5% |
| Wapiti | 49% | 89% |
| Nikto | 41% | 87.5% |

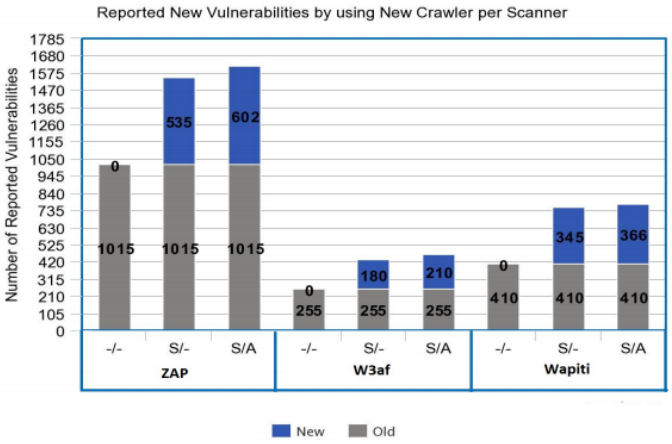


Fig. 5. Reported Vulnerabilities using before and after Arachni Crawler

Table. V depicts the configurations settings for the executed scans. In configuration (-/-) we set the scanner without any seed values i.e. no URL list is provided by using Arachni crawler and no authentication framework is used. Scanners are used with their default settings. In configuration (S/-), seed values i.e. URL's list is provided by using Arachni crawler to other scanner but without using any authentication framework. In configuration (S/A) seed value and authentication framework is used to get better detection results.

TABLE V
OUR CONFIGURATIONS DURING EVALUATION PHASE

| Config. | Scans Executed |
|---|---|
| -/- | Without Seeding and Authentication Configuration |
| S/- | With Seeding and Non-Authentication Configuration |
| S/A | With Seeding and Authentication configuration (using both technical solutions) |

*2) Automated Scripts Generation:* With automated tools, the FP rates are appeared to be high. Some manual analysis is required to confirm the reported vulnerabilities. Even in cases where the size of the application is large, an automated security scan comes handy. However, the result given by the automated tool is not necessarily the conclusion. Manual analysis is often required to confirm the vulnerabilities. Manual techniques are also helpful in finding business logic flaws. Therefore, there is a need to research manual testing techniques that make this process less time-consuming and convenient.

## VI. CONCLUSION

The sole purpose behind the security assessment of web applications aims to timely identification of vulnerabilities of a web application. Developers and pen-testers use various tools to scan web applications and detect every possible vulnerability. In this paper, eleven web application assessment tools are surveyed and evaluated against vulnerable web applications DVWA. The evaluation of publicly available web vulnerability scanners is performed against the top ten OWASP vulnerabilities and their performance is measured on the precision of their results. It is identified that there is not a single scanner that can detect all vulnerabilities present in web applications. Based on these results, we proposed our framework and prototype (SAT) that comprehensively performs security assessment of target web applications to get more extensive and accurate results.

This research can be used for the security assessment of targeted vulnerability assessment tools. Moreover, this research invites security researchers and developers to investigate and counter the latest vulnerabilities and flaws. Users can employ this research for a better selection of commercial and open available free applesauce vulnerability scanners cations in online markets.

## VII. ACKNOWLEDGMENT

This research is supported by the Higher Education Commission (HEC), Pakistan through its initiative of NationalCenter for Cyber Security for the affiliated lab National CyberSecurity Auditing and Evaluation Lab (NCSAEL), Grant No:2(1078)/HEC/ME/2018/707.

## REFERENCES

[1] [Online]. Available: https://blog.barracuda.com/2021/05/18/report-the-state-of-application-security-in-2021/ [Accessed: 10- Jun- 2022].

[2] [Online]. Available: https://www.ptsecurity.com/ww-en/analytics/web-application-vulnerabilities-statistics-2019/ [Accessed: 10- Jun-2022]/

[3] Badotra, Sumit, and Amit Sundas. "A systematic review on security of E-commerce systems." International Journal of Applied Science and Engineering 18.2 (2021): 1-19.

[4] Symantec, "Welcome to integrated cyber defense." [Online]. Available: https://www.symantec.com/ [Accessed: 25- Jun- 2022].

[5] DVWA, "Damn vulnerable web application." [Online]. Available: http://www.dvwa.co.uk/ [Accessed: 217- Jun- 2022].

[6] Shahid, Waleed Bin, et al. "An enhanced deep learning based framework for web attacks detection, mitigation and attacker profiling." Journal of Network and Computer Applications 198 (2022): 103270.

[7] Gupta, Aakanshi, et al. "Extracting rules for vulnerabilities detection with static metrics using machine learning." International Journal of System Assurance Engineering and Management 12.1 (2021): 65-76.

[8] B. Mburano and W. Si, "Evaluation of Web Vulnerability Scanners Based on OWASP Benchmark," 2018 26th International Conference on Systems Engineering (ICSEng), 2018, pp. 1-6, doi: 10.1109/ICSENG.2018.8638176.

[9] Araújo, Ricardo, António Pinto, and Pedro Pinto. "A Performance Assessment of Free-to-Use Vulnerability Scanners-Revisited." IFIP International Conference on ICT Systems Security and Privacy Protection. Springer, Cham, 2021.

[10] Y. Fang, X. Long, L. Liu, and C. Huang, "Darkhunter: A fingerprint recognition model for web automated scanners based on cnn," in Proceedings of the 2nd International Conference on Cryptography, Security and Privacy. ACM, 2018, pp. 10–15.

[11] OWASP, "Owasp foundation." [Online]. Available: https://www.owasp.org/index.php/Main_Page [Accessed: 17- Jun- 2022].

[12] Mirza, Samrah, et al. "A Malware Evasion Technique for Auditing Android Anti-Malware Solutions." 2021 IEEE 30th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE). IEEE, 2021.

[13] Fatima, Maheen, et al. "A survey on common criteria (CC) evaluating schemes for security assessment of IT products." PeerJ Computer Science 7 (2021): e701.

[14] [Online]. Available: https://www.acunetix.com/vulnerability-scanner/. [Accessed: 25- Aug- 2021].

[15] "Micro Focus", Micro Focus, 2021. [Online]. Available:https://www.microfocus.com/en-us/products/webinspect-dynamic-analysis-dast/free-trial. [Accessed: 11- Jan- 2022].

[16] "Netsparker — Web Application Security For Enterprise", [Online]. Available: https://www.netsparker.com/. [Accessed: 28- Dec- 2021].

[17] "HCL Software", Hcltechsw.com, 2021. [Online]. Available: https://www.hcltechsw.com/appscan. [Accessed: 28- Dec- 2021].

[18] "Nessus Product Family", Tenable®, 2021. [Online]. Available: https://www.tenable.com/products/nessus. [Accessed: 29- Dec- 2021].

[19] "OWASP ZAP Zed Attack Proxy — OWASP",[Online]. Available: https://owasp.org/www-project-zap/. [Accessed: 30- Dec- 2021].

[20] "Nikto2 — CIRT.net", Cirt.net, 2021. [Online]. Available: https://cirt.net/Nikto2. [Accessed: 5- Jan- 2022].

[21] "w3af - Open Source Web Application Security Scanner", W3af.org, 2021. [Online]. Available: http://w3af.org/. [Accessed: 5- Jan- 2022].

[22] "Wapiti : Open-Source web-application vulnerability scanner in Python for Windows, Linux, BSD, OSX", Wapiti.sourceforge.io, 2021. [Online]. Available: https://wapiti.sourceforge.io/. [Accessed: 5- jan- 2022].

[23] "Arachni - Web Application Security Scanner Framework", Arachni - Web Application Security Scanner Framework, 2021. [Online]. Available: https://www.arachni-scanner.com/. [Accessed: 5- Jan- 2022].

[24] "Burp Suite - Application Testing Software", Portswigger.net, 2021. [Online]. Available: https://portswigger.net/burp. [Accessed: Jan- 2022].

[25] Luo, Yunheng. "SQLi-Fuzzer: A SQL Injection Vulnerability Discovery Framework Based on Machine Learning." 2021 IEEE 21st International Conference on Communication Technology (ICCT). IEEE, 2021.

[26] Peroli, Michele, et al. "MobSTer: A model-based security testing framework for web applications." Software Testing, Verification and Reliability 28.8 (2018): e1685.