I.J. Information Technology and Computer Science, **2020, 4, 38-46**
Published Online August 2020 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2020.04.05

Web Vulnerability Finder (WVF): Automated Black- Box Web Vulnerability Scanner

Muhammad Noman Khalid
Department of Computer Sciences, Bahria University Karachi. nomankhalid.bukc@bahria.edu.pk

Muhammad iqbal
Department of Computer Sciences, Bahria University Karachi. & School of Information Sciences & Technology Southwest Jiaotong University, Chengdu, China
E-mail: miqbal.bukc@bahria.edu.pk

Kamran Rasheed
Department of Computer Sciences, Bahria University Karachi. kamranrasheed450@gmail.com

Malik Muneeb Abid
Department of Civil Engineering, International Islamic University, Islamabad
E-mail: muneeb.abid@iiu.edu.pk

Received: 07 February 2020; Accepted: 16 March 2020; Published: 08 August 2020

Abstract: Today the internet has become primary source of communication among people because it holds limitless space and pool of various web applications and resources. The internet allows us to communicate in a fraction of second with another people who is sitting in the other part of the world. At present, the internet has become part of our daily life and its usage is increasing exponentially, therefore it accumulates a number of web applications on daily basis on Web podium. Most of the web applications exist with few weaknesses and that weaknesses give room to several bad buys (hackers) to interrupt that weak part of code in web applications. Recently, SQL Injection, Cross Site Scripting (XSS) and Cross Site Request Forgery (CSRF) seriously threaten the most of the web applications. In this study, we have proposed a black box testing method to detect different web vulnerabilities such as SQL Injection, XSS and CSRF and developed a detection tool i.e. Web Vulnerabilities Finder (WVF) for most of these vulnerabilities. Our proposed method can automatically analyze websites with the aim of finding web vulnerabilities. By applying the tool to some websites, we have found 45 exploitable XSS, SQL Injection 45, Directory Discloser 38 and Local/remote file inclusion 40 vulnerabilities. The experimental results show that our tool can efficiently detect XSS, SQL Injection, Directory Discloser and LFI/RFI vulnerabilities.

Index Terms: WVF, Automated Vulnerability Detection, black-box scanners.

1. Introduction

Website is providing the standard facilities through Internet. Modern web applications are developed with the combination of a back-end and Front-end. Back-end is server-side portion with different programming languages (.Net PHP, Python and Ruby) and a back-end is a client side portion is running on User web browser (which was carried out in JavaScript and CSS/HTML). The two portions frequently interconnect through HTTP or HTTPS protocol by means of Asynchronous XML (AJAX) and JavaScript [1].

The accessibility of web applications is lead them to develop into an integral part of daily life since their availability primarily free through the internet. These applications inherently handle sensitive data and are employed to carry out business-critical activities such as online tax filing, online shopping, online banking and social media accounts. On the other hand it is a common observation that website serve to be a prime target for attackers for the reason of being omnipresent , in-demand and having an incrementing user-base. [2]. With the increased demand of the web applications, bloggers and web service providers are taking more interest in implementing and utilizing the web applications[3]. However, it is important to note that when user gets any bug or finds out any sort of weakness in the website is called vulnerability [4].

There are so many web vulnerabilities such as Cross Site Request Forgery (CSRF), Cross Site Scripting (XSS), and SQL injections listed by Open Web Application Security Project (OWASP) [5]. These vulnerabilities have the ability to exploit the information by representing the considerable website threat. To cope with such challenges, it's important to

integrate the security counter measures in web scanning tools, such as penetration testers to reduce the threats of vulnerabilities. Penetration tester tool helps to detect vulnerabilities in different web application parts by incorporating different potential techniques. Such types of strategies enable the tester to identify and analyze the most of the existing web application vulnerabilities. It is important for the penetration tester to conduct the testing procedure on regular basis, as this technique will help in minimizing and preventing the web application damage. In order to build a successful penetration testing tool, a rich experience of web application development is required [6].

This study focuses on vulnerability scanning testers. The main reason for this is that, the end users are well aware with antivirus, malware and web spam software tools, while they are very less conscious about web vulnerability scanning tools. As such, this study aims to highlight the idea of presenting the open-source vulnerability scanner, which has the ability to utilize the technique of black box to answer most of the vulnerability defection taxonomies. This paper makes the contributions shown below:

- We presented automated web vulnerability scanner to detect web vulnerabilities such as SQLi, XSS, LFI/RFI and DD. This is also demonstrate how easy for developer to detect automatically web vulnerabilities and exploit them.
- Four attack modules, which are developed for analysis of web vulnerabilities. Furthermore, a mechanism is presented to automatically derive exploits for discovered vulnerabilities.
- Technical details of experiment are supplied. In order to facilitate the researchers to implemented there black-box scanner and publically available on GitHub.

The remainder of the research is organized in different sections as follows. The next section defines the literature review. Section 3 gives the general methodology of WVF methodologies and research main goals. Comparison is provided in section 4 related to WVF with respect to the open source and commercial vulnerability scanner. Finally, the last section concludes.

2. Litrature Review

A number of organizations such as the threat intelligence software associations and the computer security enterprises are conducting researches in order to find new techniques related to the vulnerabilities. For such a purpose, various potential methods of detection are studied allowing preventing the web application vulnerabilities. The most preferable techniques include the Hybrid analysis, static analysis, dynamic analysis, and machine learning.

There are some other approaches to prevent vulnerabilities on web applications. In order to support penetration tester, another technique that prevent web vulnerabilities is called white box testing. This testing is to check the source code of web application by providing some input to that specific output. One of significance of white-box method is that it can handle as many attack [7, 8]. The penetration testers of white-box has given full access to source code investigate web application vulnerabilities. By the utilization of white box system, it comes to realize that there are numerous downsides for utilizing this procedure. By the use of white box technique, it comes to know that there are many drawbacks for using this technique. Major drawback is for those the possible applications that can be examined are compressed to only those applications that are use the target programming language. There is the issue of false positives and source code of the web application might be inaccessible.

In order to support testers and overcome white box technique shortcomings another methodology is called black box testing. The testers of black box uses no internal knowledge of the web application coding. Nonetheless, many researchers [9, 10] have effectively analyzed and shown constraints and limitations of Black Box scanner to prevent web vulnerability.

In addition, many manual and automated testing tools are used for detecting XSS and that also helps in identifying against SQLI vulnerabilities [11, 12, 13]. However, the burden of high time consumption for the completing the testing a web application for testeris still persists.A lot of work in this direction focused on fuzzing [14], which deals about testing with (semi)-random values. Furthermore, evolutionary algorithms are also employed by some other tools [15]. For instance, the implementation of plugins for corresponding applications and additive effort is required in order carry out the test automation [16].

There are three main categorize of the penetration testing scanners are used specifically for the web applications based on black box testing. These three can be categorized as the open source, commercial, and academic scanner [17, 18]. However, there are few academic scanners that are not used by the user as they are under development and language dependent. Academic scanners are used by different researches for the purpose of introducing their own penetration testing scanners such as the secubat[21], increasing the factors of MySQLinj[20], SQIVS[19], Amnisia[23], and State aware scanner[22],. These scanners are used mainly for the purpose of launching new methodologies in order to cope with the penetration testing.

Different open source scanners such as the wapit, zap, vega, wa3p, and nikito are integrated in the academic scanner providing free accessibility [24]. The access is only provided to the researchers, individuals, and authorized persons. The commercial scanner is a comprised version of both the academic web and open source penetration testing scanner. The

commercial scanner includes the Netsparker, Bugblast, AppScan, and Acunetix. The commercial scanners are not available for normal use and no vendor is allowed to do further customization. Access to commercial scanners are granted only by purchasing the methodologies, algorithms, and architecture. The commercial scanners provide a number of potential advantages as they provide vast aid and exceptional functionalities [24, 25, 26].

The dynamic and the static techniques are integrated in the open source, academic, or the open source scanners. Scanners performing the dynamic strategies are known particularly by the attacks as this way, the server gets a chance to explore and to find different errors and vulnerabilities by utilizing the techniques of targeting the application. In addition, no such target source code is required in order to execute the security results. These types of scanners are considered the best one when they integrate the static approach as they help the penetration tester to discover target application's source code and to identify various potential vulnerabilities. In order to cater such a purpose, different techniques are used such as the taint exploration (fuzzer discovered program faults) and applying the composition [27, 28, 29]. However, WVF is more likely covers the methodologies used by the existing studies [6, 22, 30].

3. Implementations and Methodologyology

WVF is a new scanning tool and is capable to perform efficient penetration tests on php and .NET based websites to identify most of the web vulnerabilities. This new tool is using more features during scanning process, which makes it robust tool to find the hidden SQLi and XSS vulnerabilities. The architecture has been employed for keeping the design open, aggregated and compatible. This scanner comprises of various parts including crawling, parsing. Towards the end, a report will be produced with an attack part that can be trigged distinctly. When it comes to the competence and performance, this Scanner assures the satisfaction with its capability of dispatching 8 to 10 parallel attacks. Figure 1 summarized the working model of WVF in which the working flow of tool is described.

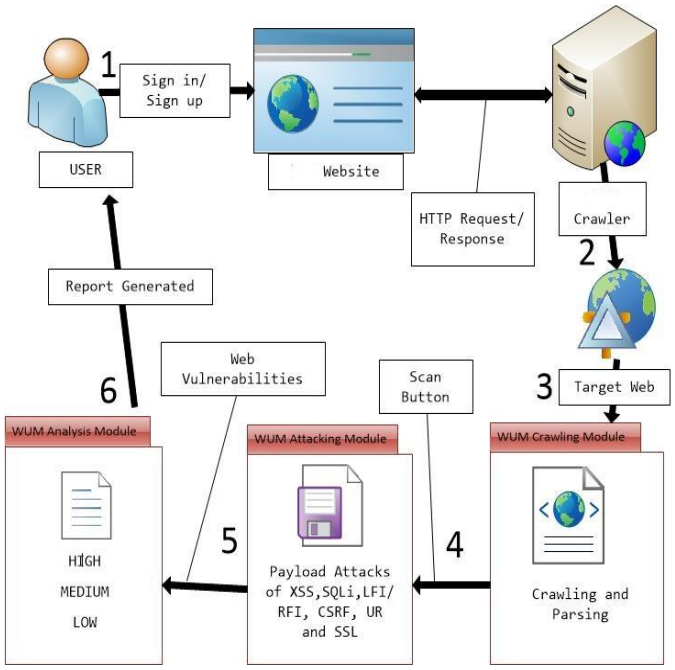

Fig.1. The Architecture of WVF scanner

Using WVF SCANNER, first user need to Sign-up/Sign-in in order to scan entire website and after that user need to put a website URL. Click on scan button so the crawler analyze the whole web application by following all the links on website which is Pages Links URLs, CSS, JS, Images and URLs found in sitemap and robot.txt. Filter out all the pages which are not needed in process of scanning and that are external links. This Scanner will map out the web application structure (URLs) and display entire information about every links. After the crawling module, this Scanner will launch an Attack with different types of Payload and check vulnerabilities on each page that is found in crawling module. This Scanner Analyses each web application links to places input data and subsequently attempts with the combination of inputs. As different vulnerabilities will found so that its represent in 3 different Tabs. In High Tab, vulnerabilities included SQLi, XSS and DD. In the same, way rest of the vulnerabilities shown in medium and low tabs. Each vulnerability will contain details information about vulnerability type, URL, description, Impact, Remedy and References. After scan will be completing the scan, result saved into the PDF file and generated a complete report. To the reported strategies commonly found in other systems [6, 22, 30, and 31].

3.1. Crawling Module

During the dynamic analysis of security, attacks can be prompted just against formerly recognized application Entry point. Hence, it is captious for testing that all the pages that are part of the tested web application are recognized [32]. This can be performed automated and manual. The crawling is to be done from page to page of website. Relatively, the response time of remote web server is slower. This feature is carried out generally in this proceeding; the regular expression is used to produce the set of links to be overlooked. The processing of our web crawler is continued until all the reachable web pages are recognized and interpreted. Moreover, only those pages are hit during automated crawling process, which lies in the base URL. For exercising the crawling constituent, conceptual conclusions were extracted from the existing systems [30, 31, 32, 33].

3.2. Attack Module

After the completion of the crawling phase, the processing of the list of target website pages is started and specially, the attack module, checks every page that is found in the crawling session. A set of genuine parameter values is produced for each AEP, and is utilized by various researchers [30, 31, 32, 33] in order to originate HTTP request. Additionally, abundant sets of incorrect parameter values are generated for each AEP. More accurately, such parameter values are produced which disrupt the predefined restraints of parameters. web vulnerabilities is a many parameters/payload to prevent the web vulnerabilities such as sql injection send the query to database and XSS check the input with the JavaScript payload. We employ predefined attack patterns/Payload as parameter outlined in table 1

Table 1. Attack Patterns or Payload of different web vulnerabilities

S. No	Web Vulnerabilities	Parameter
1	SQL injection	‘, ‘1' OR '1' = '1’
2	Cross site scripting XSS	<>, “><imgsrc=x onerror=prompt(1);>,
3	Local/Remote file inclusion LFI/RFI	../../../../../etc/passwd", ../etc/passwd
4	Directory Discloser	index of /=

WVF use malicious parameter values to generate additional HTTP requests. The outcomes of such request are additional HTML pages along with the reference HTML page are used in Phase 3. As conferred in table 1, we produce distinct designs of attack based on the type of parameter value encountered in Phase 2. Typically, for SQLI parameters of text type, such attack patterns are used which requires a single quotation mark. It is momentous to mention that a user of WVF may escalate modern patterns of attack. Hence, in this manner, WVF can be handily protracted to endorse modern and varied web attacks. Each attack type of SQLI owns several characteristics.

The attacks that are tautology-planted, are frequently employed for the purpose of verification bypassing and data derivation, nonetheless, they can further be utilized for searching implantable parameters. WVF bids to accomplish an injectable attribute that is written in the WHERE clause of a SQL query. For example, to bypass a login form, WVF adopts “‘or 1=1 –” attack design. As “--” is the comment operator of SQL and due to the fact, that “1=1” is always appraised to be true.

3.3. Analysis Modules

Analysis Module is the third module. Once the attack module is launched, the analysis module has to depict and interpret the exposures because when an extensive number of websites are scanned in a Scanner, some vulnerabilities false positives are probable. Therefore, Scanner must care requirements to be the confidence value so that false positives are curtailed handily. We carry out the analysis and comparison between other HTML pages and the reference page. Likewise, the analysis of a concluding page is also carried out in order to discover some worthy information in error message, as observed in Table 2.

Table 2. Error Messages of different web vulnerabilities

S. No	Web Vulnerabilities	Parameter
1	SQL injection	server error in '/' application, You have an error in your SQL syntax warning: require()w50..0) warning: ilesize() warning: preg_match() warning: array_merge() warning: mysql_query() warning: mysql_num_rows() Warning: mysql_result() warning: pg_exec() warning: mysql_result() warning: session_start() warning: include(news.php') warning: unknown() warning: getimagesize() is_writable() getimagesize() session_start() mysql_num_rows() mysql_fetch_array() warning: mysql_fetch_assoc() '800a0d5d' MySQL server version for the right syntax to use near
2	Cross site scripting XSS	<>, <
3	Directory Discloser	index of /
4	Local/Remote file inclusion LFI/RFI	user:/root:, root:/bin/bash, bin/false

4. Experiments and Evaluation

In order to preserve wvf vulnerability design flexible and open, we used modular and generic architecture. WVF Scanner is implemented in ASP.net programming langue the Microsoft SQL Server. In previous research [6, 21, 22, 31, 32, 33, 34], add more evaluating web vulnerability scanner through black box testing is honestly difficult. The most important task for end user to discover true positive vulnerabilities by black box scanner. However comparing different black box scanner that discover web vulnerabilities with different approach is nearly impossible. There are some metrics that we are using to evaluate black box scanners. These metrics are as follows

4.1. False Positive or Accuracy

False positive is when scanner mistakenly identify a vulnerability that is not existed. Increase in false positive is a basic concern of user. If false positives are high, each vulnerability must be inspect and report manually by security conscious user. It makes the scanner tool not useful and loose the trust of user on it.

4.2. Code Coverage

Another important metrics to evaluate a black box scanner. In this metric, easily evaluate how black box scanner crawl, attack and fuzzes a web application with different parameters and payload. Black scanner by default cannot find web vulnerabilities however, code of two main phases crawl and attacking has the potential to find web vulnerabilities. As presented in table 1 and 2 how wvf scanner evaluated a web vulnerabilities.

We used these metrics to evaluate a black box scanner however; there are some more important metrics like cost of a scanner, capacity of new vulnerabilities and 0day, ability to create a custom test with payloads, reporting web vulnerabilities and user-friendly scanner. However, wvf more focus on code coverage because efficient code coverage greater chance to discover more vulnerabilities in web application.

WVF scanner along with three other vulnerability scanners are used to evaluate our approach against 50 websites. These web applications are vary in functionality, size and complexity. This sample website applied on W3AF, STATE-AWARE-SCANNER, SECUBAT and WVF as shown in Table 3.

Table 3. Black box Vulnerability Scanner that we compared with WVF

S. No	Web scanner	Language	Version and Description
1	W3AF	Python	Web Pen testing Scanner
2	STATE-AWARE-SCANNER	Python	State Aware Vulnerability detection tool
3	SECUBAT	C#	Website Vulnerability tool
4	WVF	ASP.Net	Web Application Vulnerability scanner

To evaluate WVF scanner and other scanners against well-known vulnerabilities and most of the sample website vulnerable with web vulnerabilities.

- W3af: has BLIND SQLI, SQLI, XXS, OS COMMANDING, EVAL, RFI and LFI fuzzing plugins [24].
- State aware scanner: The scanner used htmlunit technique that issue to HTTP requests and check the response of HTML. They used w3af fuzzing plugin to generate a fuzzing request as w3af but they have added some more payloads [22].
- Secubat: SecuBat is an open web vulnerability scanner proposed by researcher Stefan Kals. They have used generic and modular approach as wvf scanner used. They have also used a multithread crawling module [21].

The results and findings of different scanner are presented in Table 4. As par, result W3AF is found all the vulnerabilities with the total number of 140. Many attempts have been made in order to aimed implemented these scanners. For the other scanner such as STATE-AWARE-SCANNER and SECUBAT is found result respectively as W3AF Found. This table is also shows the total number of web vulnerabilities, mean and accuracy of different scannersthat have compared with the proposed scanner.

Table 4. Result of different tools

Scanner	XSS	SQL.	DD	RFI/LFI	Total vulnera bilities	Mean Of Vulnerabilities Percentile	Accuracy
W3AF	35	30	40	35	140	0. 70	70%
STATE-AWARE-SCANNER	25	30	NA	NA	55	27.5	27.5%
SECUBAT	30	15	30	32	107	0.53.5	53.5%
WVF Scanner	45	45	38	40	150	0.84	84%

As it clearly derived from the above table WVF scanner is found better result as compare to others existing studies. Our steps proceed very much in the same way as existing study. Our technique shows a clearly has an advantage over these scanners with 84% highest accuracy as shown in figure 2.

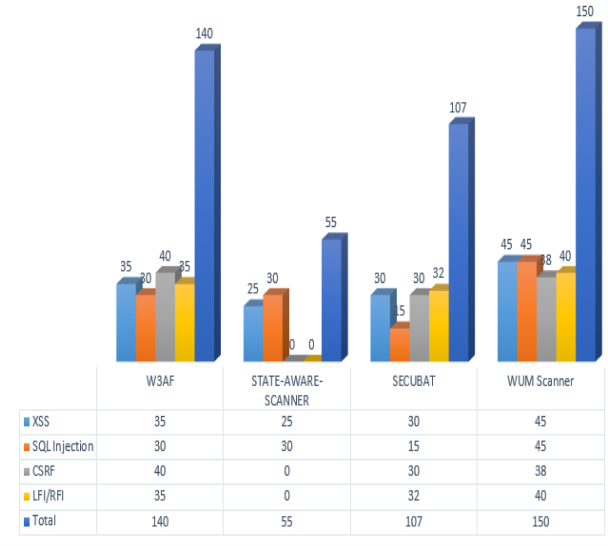

Fig.2. Total number of vulnerabilities detected by each scanner

The table 5 shows the rule and defiantion of paramemter to evaluate the result generated by W3af, state aware, secubat and WVF scanner.

Table 5. Set of Rule/Parameters to evaluate false positive vulnerabilities

S. No	Rule	Definition
1	True Positive (TP)	Scanner detect web vulnerability in reality does exist (Right Prediction)
2	False Positive (FP)	Scanner detect web vulnerability in reality does not exist(Wrong prediction)
3	True Negative (TN)	Scanner not detect web vulnerability
4	Accurate Detection (AD)	Scanner detectaccurately web vulnerability

Table 6 presented a total number of vulnerabilities in sample and false positive, true positive, true negative and accurately detection result for the each scanners.

Table 6. Set of Rule/Parameters to evaluate false positive vulnerabilities

Scanner	XSS	SQLi	DD	LFI/RFI	Total FP	Total TN
W3AF Scanner	TN=8	FP=5	TN=10	TN=5	5	23
STATE-AWARE-SCANNER	TN=13	TN=2 FP=04	NA	NA	04	15
SECUBAT Scanner	TN=8	TN=7	FP=5	AD	5	15
WVF Scanner	FP=3	AD	TN=6	FN	3	6

The coverage analysis iscomparing the discover vulnerabilities with total number of vulnerabilities. In our scenario, we have 50 web vulnerabilities of cross site scripting XSS, SQL injection SQLi, local and remote file inclusion LFI/RFI and directory discloser. For this study, we have used label data to find an accurate result of different scanner. The total number of web vulnerabilities with false positive and true negative is presented in the table 7.

The result of wvf scanner compare with existing studies to identify true positive and false negative. In the study, different scanners is found total number of false positive and true negative in the case of WEAF is 13, in the case of State aware scanner is 19, in the case of secubat is 20 and in the case of wvf scanner is 09. Afterward we have added another column of total vulnerabilities as these scanners found total number of vulnerabilities. For this study found much higher values for proposed method with respect to those reported by existing studies.

Table 7. Total false positive and true negative web vulnerabilities

Scanner	XSS	SQLi	DD	LFI/RFI	Total FP TN	Total vulnerabilities	Final Result	Accuracy
W3AF	TN=4	FP=5	TN=2	TN=2	13	140	140-13=127	63.5%
STATE-AWARE-SCANNER	TN=13	TN=2 FP=04	NA	NA	19	55	55-19=36	55.%
SECUBAT	TN=8	TN=7	FP=5	AD	20	107	107-20=87	53.5%
WVF Scanner	FP=3	AD	TN=6	FN	09	150	150-09=141	70.5%

Figure 3 describe the total number of false positive, total number of vulnerabilities, result and accuracy investigated.

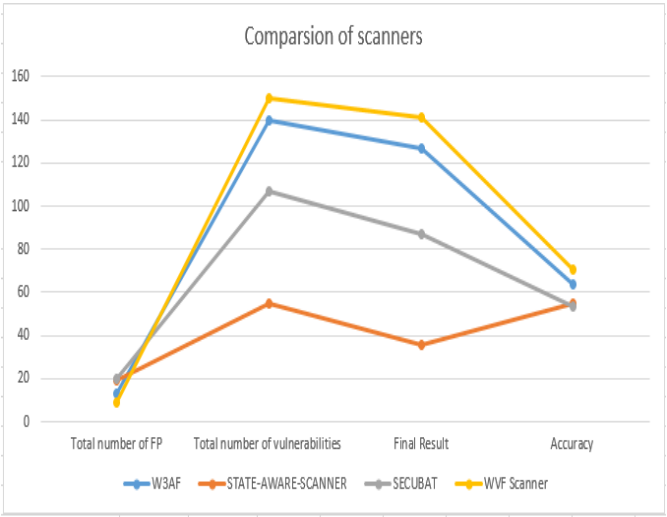

Fig.3. Comparison of scanners of Accuracy

5. Conclusion

In this paper, we have proposed a new approach to analyze the different web vulnerabilities through our WVF tool. We emulated the Cross-site Scripting (XSS), SQL Injection, and Local and remote file inclusion attacks in this research. The results of this study assisted us to develop the rules for vulnerabilities analysis. However, the large percentage of false positives and false negatives were also obtained through WVF. The proposed approach takes web URL as an input to detect web vulnerabilities. In order to evaluate the performance of the wvf tool the experimental work is conducted based on the accuracy (false positive and false negative).The results show that proposed method yields low false negative and false positive for all detection. We have tested our tool on several popular websites. One fifty (150) exploitable XSS, SQL Injection and CSRF vulnerabilities are found, revealing the overlooked security risk of these vulnerabilities. The evaluation result also validates the effectiveness of WVF. For further work, we plan to use variants of attacks to increase detection score of new vulnerabilities the could be parts of ASP pages.

Reference

[1] Pop, Dragos-Paul, and Adam Altar. "Designing an MVC model for rapid web application development." Procedia Engineering 69 (2014): 1172-1179.

[2] Deepa, G., Thilagam, P. S., Khan, F. A., Praseed, A., Pais, A. R., & Palsetia, N. (2018). Black-box detection of XQuery injection and parameter tampering vulnerabilities in web applications. International Journal of Information Security, 17(1), 105-120.

[3] Khalid, M. N., Farooq, H., Iqbal, M., Alam, M. T., & Rasheed, K. (2018, October). Predicting Web Vulnerabilities in Web Applications Based on Machine Learning. In International Conference on Intelligent Technologies and Applications (pp. 473-484). Springer, Singapore.

[4] Awoleye, Olusesan M., Blessing Ojuloge, and Mathew O. Ilori. "Web application vulnerability assessment and policy direction towards a secure smart government." Government Information Quarterly 31 (2014): S118-S125.

[5] OWASP: Available at http://www.owasp.org/index.php/ Category: OWASP Top Ten Project, 2017.

[6] Bozic, Josip, and Franz Wotawa. "PURITY: a Planning-based secURITY testing tool." Software Quality, Reliability and Security-Companion (QRS-C), 2015 IEEE International Conference on. IEEE, 2015.

[7] DOUPE, A., BOE, B., KRUEGEL, C., AND VIGNA, G. Fear the EAR:Discovering and Mitigating Execution After Redirect Vulnerabilities. InProceedings of the 18th ACM Conference on Computer and CommunicationsSecurity (CCS 2011) (Chicago, IL, October 2011).

[8] JOVANOVIC, N., KRUEGEL, C., AND KIRDA, E. Static analysisfor detecting taint-style vulnerabilities in web applications.Journal of Computer Security 18, 5 (2010), 861–907

[9] Jason Bau, Elie Bursztein, Divij Gupta, John Mitchell, "State of the Art: Automated Black-Box Web Application Vulnerability Testing", 2010

[10] Adam Doup´e, Marco Cova, and Giovanni Vigna, “Why Johnny Can’t Pentest: An Analysis of Black-box Web Vulnerability Scanners”, July 2010

[11] “Burp suite,” http://portswigger.net/burp/, accessed: 2018 -11- 11.

[12] “Zed attack proxy (zap),” https://www.owasp.org/index.php/ OWASP Zed Attack Proxy Project, accessed: 2020-02-10.

[13] Aliero, M. S., Ghani, I., Qureshi, K. N., & Rohani, M. F. A. (2019). An algorithm for detecting SQL injection vulnerability using black-box testing. Journal of Ambient Intelligence and Humanized Computing, 1-18.

[14] “Defensics,” http://www.codenomicon.com/products/defensics/, accessed: 2018-11-10.

[15] F. Duchene, S. Rawat, J.-L. Richier, and R. Groz, “Kameleon- Fuzz: Evolutionary Fuzzing for Black-Box XSS Detection,” in CODASPY. ACM, 2014, pp. 37–48.

[16] B. Garn, I. Kapsalis, D. E. Simos, and S. Winkle8, “On the applicability of combinatorial testing to web application security testing: A case study,” in Proceedings of the 2nd International Workshop on Joining AcadeMiA and Industry Contributions to Testing Automation (JAMAICA’14). ACM, 2014.

[17] N.Antunes and M. Vieira 2010. Benchmarking vulnerability detection Scanners for web services. Paper presented at the Web Services (ICWS), 2010 IEEE International Conference on.

[18] V. Livshits, and M. S. Lam 2005. Finding Security Errors in Java Programs with Static Analysis. In Proceedings of the 14th Usenix Security Symposium, pages 271-286.

[19] Z. Duric2013. A black-box testing Scanner for detecting SQL injection vulnerabilities. Paper presented at the Informatics and Applications (ICIA), 2013 Second International Conference on IEEE.

[20] ALiban, and H. Shadi. 2014. Enhancing Mysql Injector vulnerability checker Scanner (Mysql Injector) using inference binary search algorithm for blind timing-based attack. Paper presented at the Control and System Graduate Research Colloquium (ICSGRC), 2014 IEEE 5th.

[21] S. Kals Kirda E. Kruegel Christopher, and J. Nenad 2006. Secubat: a web vulnerability scanner. Paper presented at the Proceedings of the 15th international conference on World Wide Web.

[22] Doupé, Adam, et al. "Enemy of the State: A State-Aware Black-Box Web Vulnerability Scanner." USENIX Security Symposium. Vol. 14. 2012.

[23] W. G. Halfond, and A Orso, 2005. AMNESIA: Analysis and Monitoring for NEutralizing SQL-Inj ection attacks", Proceedings of the 20thIEEE/ACM international Conference on Automated software engineering, pp. 174-183,2005.

[24] OWSAP Open Web Security Project. Retrieved 29/06/2015, from owasp.org/index.php/Category:VulnerabilitLScanning_ Scanners

[25] M.F Jnena 2013. Modern Approach for WEB Applications Vulnerability Analysis retrieve on 27/812015 fromhttp://library.

iugaza.edu.ps/thesis/1 09553 .pdf 229
[26] Shay Chen 2011. Security Scanner Benchmarking available at http://secScanneraddict. blogspot.my/20 11108/commercial-webapplication- scanner.html
[27] X. Zhang, and Z. Wang. 2010. Notice of Retraction A Static Analysis Scanner for Detecting Web Application Injection Vulnerabilities for ASP Program. Paper presented at the eBusiness and Information System Security (EBISS), 2010 2nd International Conference on
[28] L. Zhang, et al. 2010. D-WAV: A web application vulnerabilities detection Scanner using Characteristics of Web Forms." Software Engineering Advances (ICSEA), 2010 Fifth International Conference on. IEEE, 20 I O.
[29] F. Jose'S. Nuno , V. Marco, and M. Henrique"Analysis of field data on web security vulnerabilities."Dependable and Secure
[30] Kals, Stefan, et al. "Secubat: a web vulnerability scanner." Proceedings of the 15th international conference on World Wide Web. ACM, 2006.
[31] MLA Shahriar, Hossain, and Mohammad Zulkernine. ”Client-side de-tection of cross-site request forgery attacks.” 2010 IEEE 21st Interna-tional Symposium on Software Reliability Engineering. IEEE, 2010.
[32] Goel, Jai Narayan, and B. M. Mehtre. "Vulnerability assessment & penetration testing as a cyber defence technology." Procedia Computer Science 57 (2015): 710-715.
[33] Djuric, Zoran. "A black-box testing tool for detecting SQL injection vulnerabilities." Informatics and Applications (ICIA), 2013 Second International Conference on. IEEE, 2013
[34] Y.-W. Huang, S.-K. Huang, T.-P. Lin, and Ch.-H. Tsai, “Web application security assessment by fault injection and behavior monitoring”, Proceedings of the 12th international conference on World Wide Web, pp. 148-159, 2003

Authors’ Profiles

Muhammad Noman was born in 1991 in Pakistan. He received MS (CS) degree from bahria university Pakistan. His research area information security, machine learning, web security. He is a highly proficient, experienced, and professionally Certified Penetration Tester Engineer and Certified Vulnerability Assessor fromMile2 (USA Awarding Body.

Muhammad Iqbal was born in 1972 in Pakistan. He received B.Sc(Hons) and M.Sc degree in Computer Technology from Sindh University, Pakistan and MS in computer Science from SZABIST, Karachi, Pakistan. Since 2012, he is a PhD student in School of Information Sciences & Technology (SIST), Southwest Jiaotong University, Sichuan, Chengdu, PR China. His research interests are Network Security, Data Mining, Supervised Machine Learning algorithms and high speed data networks.

Malik Muneeb Abid was born in 1987 in Pakistan. He received B.Sc degree in Civil Engineering from U.E.T Taxila, Pakistan and MS degree in Transportation Engineering from NUST, Pakistan. Since 2013, he is a PhD student at School of Transportation and Logistics, Southwest Jiaotong University, Sichuan, Chengdu, PR China. His research interests are Network Robustness, Transportation network modeling and simulation, Data Mining, Supervised Machine Learning algorithms. He is member of IAROR and PEC.

How to cite this paper: Muhammad Noman Khalid, Muhammad iqbal, Kamran Rasheed, Malik Muneeb Abid, "Web Vulnerability Finder (WVF): Automated Black- Box Web Vulnerability Scanner", International Journal of Information Technology and Computer Science(IJITCS), Vol.12, No.4, pp.38-46, 2020. DOI: 10.5815/ijitcs.2020.04.05