Jurnal Mantik
Volume 4, Number 2, Augustus 2020, pp. 1138-1144 E-ISSN 2685-4236
https://iocscience.org/ejournal/index.php/mantik

Automatic Scanner Tools Analysis As A Website Penetration Testing

Fathurrahmad[1], Ester[2]

Informatics Management, Faculty of Computer Science
AMIK Indonesia, JL. Teuku Nyak Arif, Simpang Mesra, Jeulingke, Syiah Kuala, Banda Aceh City, Aceh

Email: fathurrahmad@amikindonesia.ac.id *

ARTICLE INFO

Article history:
Received: 12/01/2020
Revised: 22/09/2020
Accepted: 01/08/2020

Keywords:
Analysis Tools, Automatic scanner tools · Cybersecurity · Web vulnerabilities

ABSTRACT

Many studies have produced and compared the accuracy of penetration testing tools in looking for vulnerabilities in web applications. In previous studies, comparative comparisons were made about the accuracy of dynamic analysis tools for website vulnerabilities. The purpose of this research is to obtain all information on an AMIK Indonesia website and to analyze it by collecting information on execution time such as network resource usage, attacks carried out, warnings, and vulnerability results sourced from analysis tool reports. The tool used is using Subgraph Vega and OWASP Zap and looking at the comparison of the tool reports. In this way, it can be seen whether the testing carried out by the tool is efficient and meets two important aspects of the analysis tool so that the penetration testing tool can find all vulnerabilities in the web application, and can also report detected vulnerabilities. When viewed from the results of the analysis obtained, SubGraph Vega has fewer Number of vulnerabilities than OWASP ZAP which has more. Meanwhile, other types of attacks were not found to be the same from the second tool.

1. Introduction

The use of websites in everyday life has become normal today [1]. Information sent and stored on a website can be very sensitive, so security is very important in making website development [2,3]. Hence, mandatory security requirements are included in the steps of the website development cycle [4,5]. In the last step in website development, testing is necessary to check for vulnerabilities and failures of a website, before it is at the final stage and is ready for use. Thus the chance of attack will be significantly reduced [6].

Currently, there are various tools for reviewing errors in the coding of an application and website vulnerabilities [7]. The selection of the right tools has an impact on the process of improving website security [8] because its effectiveness will depend on how vulnerable the application is in the application environment. These tools are in accordance with their capabilities and features, where testing is based on the Blackbox and Whitebox methods provided by the tools [9]. The black box method has a testing technique that focuses on information with use case stages and partitions the input domain from in-depth testing coverage [9]. Different testing with the white box method is a technique that examines the system as a whole and reaches a stage that may not yet be accessible using the black box method [9, 10]. Testing is carried out in order to find and correct potential security breaches that could be vulnerabilities that an attacker can exploit in a web application. Analysis can be done automatically or manually using special tools. The downside of web application security testing is the time because it takes a long time to check the entire application code analysis, something that is determined by the number of lines of code and its complexity [11, 12].

The purpose of this study is to obtain all information on an AMIK Indonesia website and to analyze it by collecting information on execution time such as network resource usage, attacks carried out, warnings, and vulnerability results sourced from analysis tools reports. The tools used are using Subgraph Vega and OWASP Zap and see the comparison of the reports on these tools.

2. Domain theory and previous studies

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

In previous studies, several comparisons have been made regarding the accuracy of dynamic analysis tools for website vulnerabilities. Figure 1 shows a method that is normally used to check the accuracy of a scanner. The use of the tools Subgraph Vega [13, 14] and OWASP ZAP [15], both free and widely used for comparison and analysis of a web vulnerability [16, 17]. These tools are particularly appropriate for evaluating vulnerabilities in web applications.

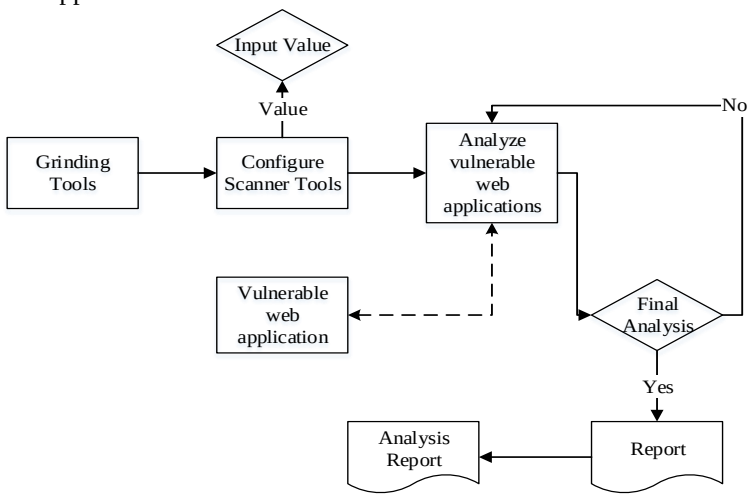

Fig 1. The method used to check the accuracy of a tool

3. System Model

In The system model used consists of two computers, one used as an attacker and the other as a server. Both are connected to a physical network and a network router, as shown in Figure 2. The attack computer used is 8Gb Ram, Inter Core i5 3.2 GHz. To implement tools that can only be used on the Microsoft Windows platform. Whereas on the server computer is Ubuntu Server 14.04 LTS x86, 512 Gb Ram, I CPU, (Virtual). On this computer, 1 virtual machine is installed which is responsible for running the web server. Table 1 shows the characteristics of each component in the system that the researcher uses.

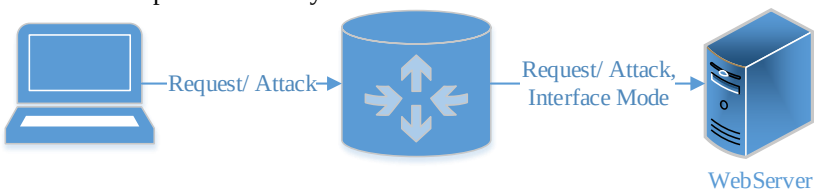

Fig 2. Details of the test environment

Table 1.
Test Environment Features

	Attacker	*Web* Server
OS	Windows 8.1 / 10	Ubuntu Server 14.04 LTS x86
Hardware	8Gb Ram, Inter Core i5 3.2 GHz	512 Gb Ram, I CPU, (Virtual)
Software	OWASP ZAP,	Apache 2, MySQL

3.1. System tools

The tools used in this research are Subgraph Vega and OWASP ZAP. These tools were chosen because many previous researchers used and were known to have good web application vulnerability analysis. These tools have also been used in previous studies [18, 19, 20], which have been carried out to analyze their web applications. All tools used in this study are configured according to the default application default without setting any further special features. However, it is important to note that customizing and using specific plugins have different analytical precision to look for certain types of vulnerabilities.

3.2. Vulnerable web applications

In this study, researchers took samples on the websites of research institutions, which often had dense vulnerabilities and information so that further analysis was needed. There were 10 subdomains, which

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

consisted of 5 as website sub-domains and the others as NS Servers and Mail Servers available at the host domain www.amikindonesia.ac.id.

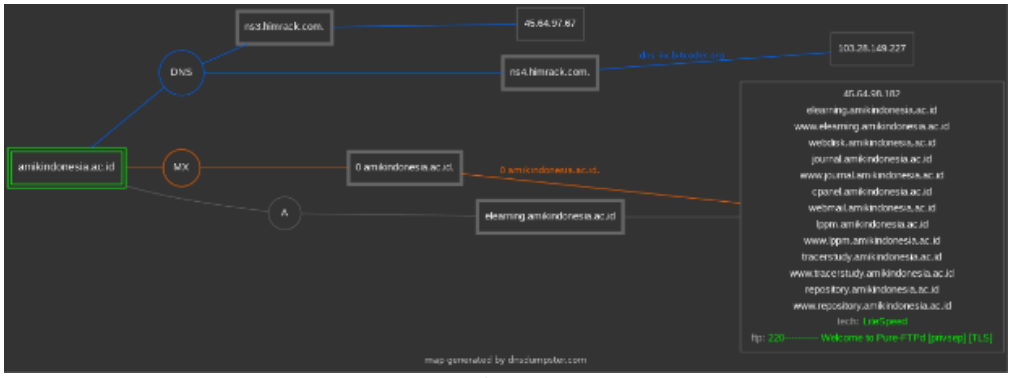

Fig 3. Mapping Results of the AMIK Indonesia Website

3.3. Management tools

In order to produce diagnostics that include information gathering and scanning carried out at this early stage, the researchers determined three types of vulnerability tools, namely, Whois Domain Tools¸ DNSDumpster and Builtwith Tools. So at this stage, the writer will try to do gathering information on the AMIK Indonesia website and the vulnerability analysis becomes appropriate.

4. Analysis of results

From the results of the vulnerability scanning on the AMIK Indonesia site, there are vulnerabilities caused by high, medium, and low-level categories as in the explanation of figures 4 to 5 with the following vulnerabilities.

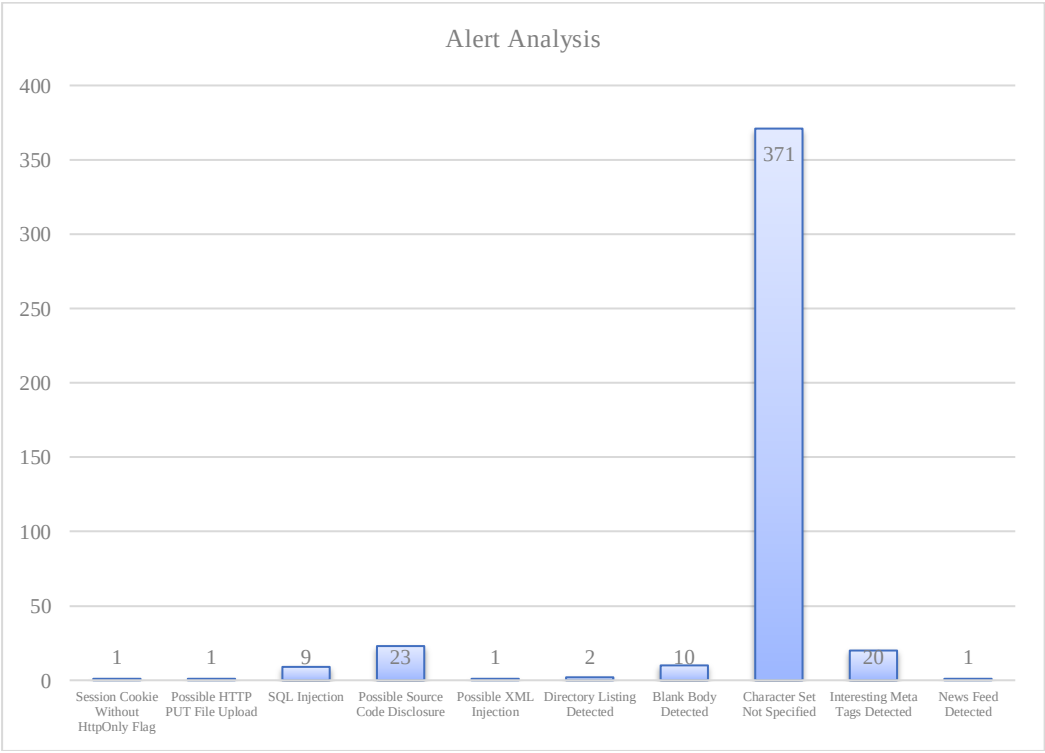

Fig 4. A subgraph of Alert Vega Analysis

Based on the Alert Analysis using Subgraph Vega, it was found that the greatest vulnerability on the AMIK Indonesia website was 462 instances (Figure 4). Based on the Alert Analysis, it was found that the greatest vulnerability on the AMIK Indonesia site with media instances was 1 and the remaining 6 samples were low, and there was no high level. The following is a table of Standby Analysis using OWASP ZAP (Figure 5).

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

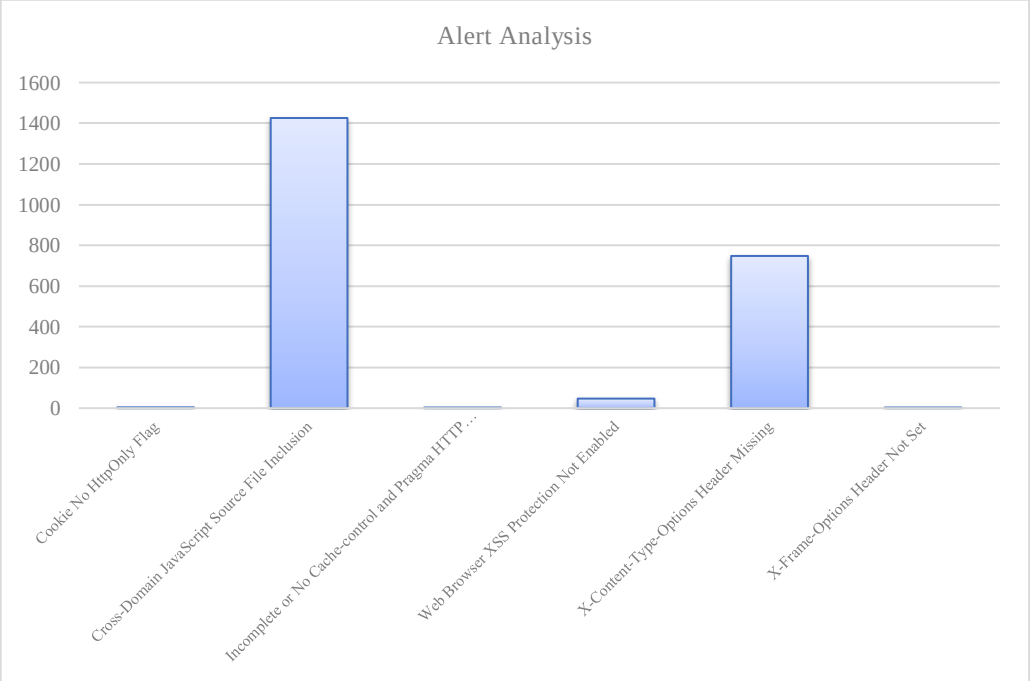

Fig 5. Alert Analysis of the OWASP ZAP

4.1. Time and resources used

The traffic generated by the scanner is aggregated and compared as shown in Figure 6. In this graph, the difference in the amount of traffic generated is different for each. During the Subgraph Vega and OWASP ZAP analysis process, there were no problems with excessive memory load and damage to the scanned website pages. On the website, there is no significant overload or reduction in bandwidth, but the analysis process spent by each tool greatly affects the crawling of the website. In the black box analysis process, the researcher calculates and adds the total number of warnings detected from each tool. This value makes it possible to know the number of attack attempts made by the tool during the analysis process.

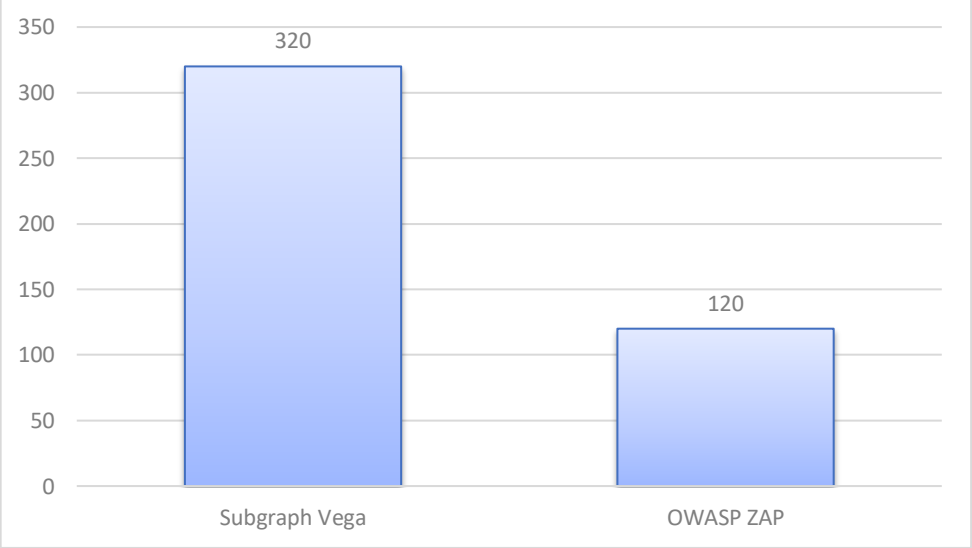

Fig 5. Usage of network resources by each pentest tool (Mb)

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

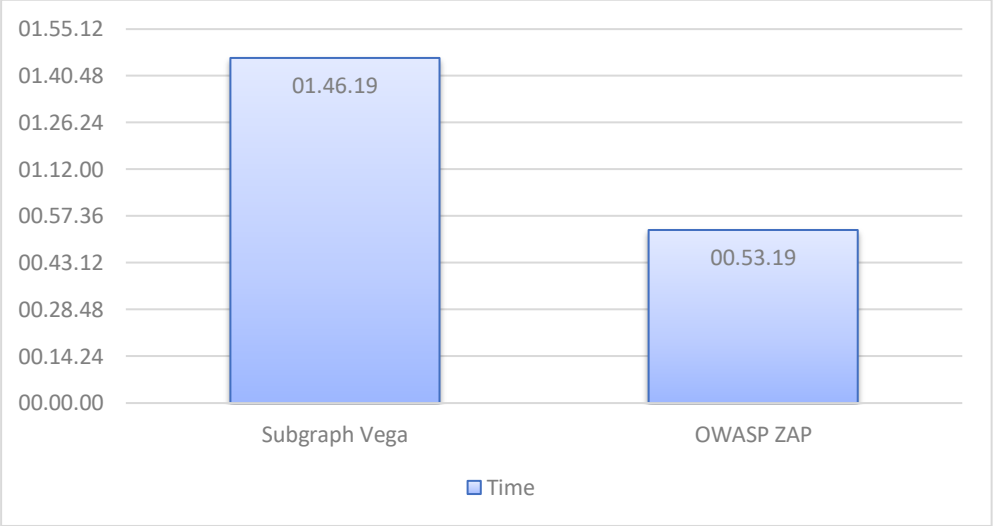

Fig 6. Time used by each tool in each analysis

4.2. Results and Discussion

Subgraph Vega identified there are 3 (three) high-level categories consisting of; Session Cookie Without HttpOnly Flag, Possible HTTP PUT File Upload and SQL Injection. While the medium category consists of 2, namely; Possible Source Code Disclosure and Possible XML Injection. In the low category, there is 1, namely Directory Listing Detected. While in the informational category there are 4 namely; Blank Body Detected, Character Set Not Specified, Interesting Meta Tags Detected, and News Feed Detected.
OWASP ZAP identified 1 medium level category, namely X-Frame-Options Header Not Set, in the low category there were 6 consisting of; Cross-Domain JavaScript Source File Inclusion, Web Browser XSS Protection Not Enabled, X-Content-Type-Options Header Missing, Incomplete or No Cache-control and Pragma HTTP Header Set, X-Content-Type-Options Header Missing, and Cookie No HttpOnly Flag. OWASP ZAP and Subgraph Vega only have similarities in the vulnerability of Session Cookie Without HttpOnly Flag with the number in SubGraph Vega 1, while in OWASP ZAP there are 4, the results of the analysis are shown in table 2 below.

Table 2.
Analysis results using SubGraph Vega and OWASP ZAP

Vulnerability	SubGraph Vega	OWASP ZAP
Session Cookie Without HttpOnly Flag	1	4
Possible HTTP PUT File Upload	1	-
SQL Injection	9	-
Possible Source Code Disclosure	23	-
Possible XML Injection	1	-
Directory Listing Detected	2	-
Blank Body Detected	10	-
Character Set Not Specified	371	-
Interesting Meta Tags Detected	20	-
News Feed Detected	1	-
X-Frame-Options Header Not Set	-	1
Cross-Domain JavaScript Source File Inclusion	-	1426
Web Browser XSS Protection Not Enabled	-	47
X-Content-Type-Options Header Missing	-	2
Incomplete or No Cache-control and Pragma HTTP Header Set	-	1
X-Content-Type-Options Header Missing	-	746

When viewed from the results of the analysis obtained, SubGraph Vega has fewer hits than OWASP ZAP which has more. SubGraph Vega shows that there are Session Cookie Without HttpOnly Flag and Command-line injection attacks on the AMIK Indonesia website, and this is the same as stated by OWASP ZAP, but more vulnerabilities are found. While the types of attacks that were found other than HttpOnly Flag and Command-line injection were not found to be the same from the two tools.

5. Conclusions

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

The approach proposed in this study makes it possible to obtain detailed results that have not been used as reference material for web application development at AMIK Indonesia. The tools used provide evaluation reports about deficiencies and identify vulnerabilities in the web application being tested. Some of the tools used indicate that there is a vulnerability of attack from high to low, but in some of the final reports are not considered vulnerable, although a small number of vulnerabilities exist that can damage web applications.

When viewed from the results of the analysis obtained, SubGraph Vega has fewer vulnerabilities than OWASP ZAP which has more. SubGraph Vega shows that there are Session Cookie Without HttpOnly Flag and Command-line injection attacks on the AMIK Indonesia website, and this is the same as stated by OWASP ZAP, but more vulnerabilities are found. Meanwhile, other types of attacks were not found to be the same for the two tools.

It is important to develop rules in the settings for each tool that will allow OWASP ZAP and SubGraph Vega to define specific types of attacks rather than general attacks. These tools can improve accuracy when configured to suit the target application. This can reduce the number of vulnerabilities that are exploited but not reported and the number of false positives. Each tool report contains basic information and helps the developer identify errors in web application development. Stages in testing can save time in developing applications than if it is done manually. From the results obtained, it is considered important to carry out further analysis to use various tools so that they can become the basis for developing security in web applications

Acknowledgment

Thanks to the Directorate General of Research and Development Strengthening Ministry of Research, Technology and Higher Education as the research funder for the Beginner Lecturer Research (PDP) grant for the 2020 Fiscal Year. Furthermore, the LPPM Leaders and their staff and the laboratory research assistant team as well as the lecturers and AMIK Indonesian academics who always provide support so that the implementation of this research is in accordance with the author's expectations.

6. Reference

[1] Iqbal, T. (2019). Menjadi Programmer Handal : HTML, PHP, MySQL, dan CSS. (Vol. 1). KITA Publisher.
[2] Ula, M. (2019). Evaluasi Kinerja Software Web Penetration Testing. TECHSI-Jurnal Teknik Informatika, 11(3), 336-352.
[3] Lee-Geiller, S., & Lee, T. D. (2019). Using government websites to enhance democratic E-governance: A conceptual model for evaluation. Government Information Quarterly, 36(2), 208-225.
[4] Aljawarneh, S. A., Alawneh, A., & Jaradat, R. (2017). Cloud security engineering: Early stages of SDLC. Future Generation Computer Systems, 74, 385-392.
[5] Muñoz, F. R., Vega, E. A. A., & Villalba, L. J. G. (2018). Analyzing the traffic of penetration testing tools with an IDS. The Journal of Supercomputing, 74(12), 6454-6469.
[6] Schwarz, M., Weiser, S., Gruss, D., Maurice, C., & Mangard, S. (2017, July). Malware guard extension: Using SGX to conceal cache attacks. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 3-24). Springer, Cham.
[7] Kaur, A., & Dhiman, G. (2019). A review on search-based tools and techniques to identify bad code smells in object-oriented systems. In Harmony search and nature inspired optimization algorithms (pp. 909-921). Springer, Singapore.
[8] Higgins, D., Balint, T., Liversage, H., & Winters, P. (2018). Investigating the impacts of increased rural land tenure security: A systematic review of the evidence. Journal of rural studies, 61, 34-62.
[9] Wali, M. (2020). Modul Praktikum Rekayasa Perangkat Lunak. (Vol. 1). Ellunar Publisher.
[10] Sivakorn, S., Argyros, G., Pei, K., Keromytis, A. D., & Jana, S. (2017, May). HVLearn: Automated black-box analysis of hostname verification in SSL/TLS implementations. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 521-538). IEEE.
[11] Takanen, A., Demott, J. D., Miller, C., & Kettunen, A. (2018). Fuzzing for software security testing and quality assurance. Artech House.
[12] Parizi, R. M., Dehghantanha, A., Choo, K. K. R., & Singh, A. (2018). Empirical vulnerability analysis of automated smart contracts security testing on blockchains. arXiv preprint arXiv:1809.02702.
[13] Sanaa, M., Pouillot, R., Vega, F. G., Strain, E., & Van Doren, J. M. (2019). GenomeGraphR: a user-friendly open-source web application for foodborne pathogen whole genome sequencing data integration, analysis, and visualization. PloS one, 14(2), e0213039.
[14] Ramsingh, C., & Centonze, P. (2017). Program Analysis For Database Injections. INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY, 16(6), 6977-6986.

[15] Holík, F., & Neradova, S. (2017, May). Vulnerabilities of modern web applications. In 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 1256-1261). IEEE.
[16] Idrissi, S. E., Berbiche, N., Guerouate, F., & Shibi, M. (2017). Performance evaluation of web application security scanners for prevention and protection against vulnerabilities. International Journal of Applied Engineering Research, 12(21), 11068-11076.
[17] Rawat, G. S., & Singh, K. (2019, October). Detection and prevention of vulnerabilities in open source software: An experimental study. In Communication and Computing Systems: Proceedings of the 2nd International Conference on Communication and Computing Systems (ICCCS 2018), December 1-2, 2018, Gurgaon, India (p. 297). CRC Press.
[18] Backman, L. (2018). Why is security still an issue?: A study comparing developers' software security awareness to existing vulnerabilities in software applications.
[19] Fajar, Abdullah, and Setiadi Yazid. "Enumeration and Handling Security Issues of Government Official Web Application." 2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS). IEEE, 2018.
[20] Araújo, P. J., & Paiva, A. C. (2018). Pattern based Web Security Testing. In 6th International Conference on Model-Driven Engineering and Software Development (MODELSWARD).

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.