Implementation And Performance Analysis Development Security Operations (DevSecOps) using Static Analysis and Security Testing (SAST)

1st Wedy Freddy Santoso
Departement of Computer
Politeknik Caltex Riau
Pekanbaru, Indonesia
wedy20s2tk@mahasiswa.pcr.ac.id

2nd Dadang Syarif Sihabuan Sahid
Departement of Computer
Politeknik Caltex Riau
Pekanbaru, Indonesia
dangsyarif@gmail.com

***Abstract*—DevSecOps solves the problem by integrating the security of development operations through various development life cycles. benefits, implementation and challenges during the process. in addition to many documented web hacks. For the scope of work reported that the focus is on two widely used digital library systems: DSpace and Greenstone, in performing Static Application Security Testing (SAST) in addition to more traditional port scanning. Weaknesses were found and details how to make improvements to both systems to make them more secure. can ensure by considering more broadly on the forms of security problems found, to assist the development of software architecture in the future.**

Keywords—Development, operations Security, Architecture, Application.

I. INTRODUCTION

Every individual and organization to understand the importance of cyber security. In 2019, for example, it is estimated that globally a business becomes a victim of ransomware attacks every 14 seconds and is expected to increase to every 11 seconds by 2021 [1]. Regarding personal data breaches, in the same year, cyber security firm UpGuard reported on the discovery of a staggering 550 million Facebook records, totaling 146 GB in size, which had been left arbitrarily exposed on Amazon Cloud Services by Cultura Colectiva, a third-party vendor to the media giant. social [2].

Cybersecurity has become increasingly popular in recent years. Computers and other forms of electronic devices have undoubtedly made many daily tasks easier to complete, but the prevalence of digital platforms that make this possible also increases the risk, with hackers constantly trying to exploit flaws in a system and tamper with personal information, extorting money, subsequently engage in other malicious acts [3].

Cyber security is a field of ICT that is responsible for the protection of information assets, through Protection against threats that harm information, stored and transported by interconnected information systems [3], If cyber security is carried out after development is complete, the system is built in an insecure manner by bugs that are hard to fix. However, when security teams share knowledge and provide tools for team development and operations, the latter can modify systems and applications accordingly [4].

DevSecOps is about breaking security, further passing knowledge to different teams, and ensuring that security is implemented at the right level and at the right time [2]. [5]. DevSecOps can be defined as an approach to improve and accelerate the delivery of business value by making dev and ops team collaboration effective.

II. RELATED RESEARCH

A. Development Security Operations (DevSeeOps)

Previous research on DevSecOps, revealed that culture, automation, measurement and sharing (CAMS) are important factors to consider, in a fashion similar to DevOps. So, organizations cannot simply buy or lease their services to DevOps, and the same goes for DevSecOps. In fact, culture has been recognized as an important part of both, but DevSecOps emphasizes the importance of creating security [8].

B. Static Analysis and Security Testing (SAST)

Several studies have also shown that better ratios for identifying right and wrong can be obtained by combining different types of methods to take advantage of different synergies. This shows how a combination of methods can reduce the identification of positive (true vulnerabilities detected) and negative (true vulnerabilities not found). The analyzed work concluded that any security vulnerabilities included in the AST tool report, including manual reviews, should be verified. positive identification is actually harmless and can be corrected in security analysis. However, negative identification is more difficult to find if the previous method does not have the ability to detect it, causing a real danger. these methods include the use of static white box security analysis (SAST), dynamic black box security analysis (DAST), or ar interactive white box security analysis (IAST). Manual analysis requires highly specialized staff and time. To perform a web application security analysis, using any method, it is necessary to cover the entire attack that accesses

parts and layers of the application and use methods to automate security analysis as much as possible [9].

TABLE I. SAST VULNERABILITIES CATEGORIES

CWE	Detections
CWE-79 XSS persistent	44
CWE-79 XSS reflected	244
CWE-94 dangerous file inclusion	20
CWE-676 dangerous function	2
CWE-95 code injection	5
CWE-91 JSON injection	2
CWE-321 hardcoded encryption key	1
CWE-601 open redirect	16
CWE-259 hardcoded password	2
CWE-22 path manipulation	48
CWE-359 privacy violation	17
CWE-89 SQL injection	1
CWE-497 system information leak	1
Total	**403**

III. THEORETICAL BACKGROUND

A. *Development Security Operations (DevSecOps)*

DevSecOps is relatively new to the field of information security. The fundamental idea aligns with the concept of having security as an integral part of software development principles, processes and methodologies. The DevOps model is rapidly being adopted by the technology industry to support the need to develop and release core business systems and applications to customers in a much faster and reliable manner than the software development life cycle (SDLC) model traditionally followed. The security industry has adapted to the DevOps demand by introducing relevant processes in the form of DevSecOps principles and methods without affecting the original intent of DevOps. The author will review how security processes can be effectively embedded in a DevOps model to improve the success of IT projects within an organization. However, this article is not meant to review how to adopt DevOps to its advantage over traditional approaches[9].

B. *Analysis Security Testing*

There are various types of testing techniques that auditors or security analysts can choose from performing web application security analysis, static white box security analysis (SAST), dynamic black box security analysis (DAST) or interactive white box security analysis (IAST) techniques [10]. The OWASP Security Testing Guide Methodology v4.1 [11] suggests that to perform complex web application security analysis it is necessary to be automated as best as possible using static, dynamic and interactive analytical testing tools, including manual checks to find more actual attacks.

C. *Static Analysis Security Testing*

SAST tools will perform a security analysis of the source code of an application program, starting with determining whether an application program is ready for use or not [12]. Apart from these problems, the SAST tool is considered the most important security activity in SSDLC [13].

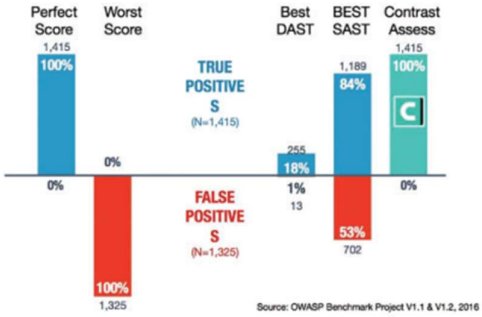

Fig. 1. SSDLC security techniques [15]

The SAST tool analyzes the entire application covering all attacks. Static analysis requires completing a manual audit of the results to discard positive identifications and find negative identifications (much more). However, some assert that different SAST tools have different algorithms by design as Interpretation. Therefore, combining SAST tools can find different types of vulnerabilities and therefore get better combination results [16].

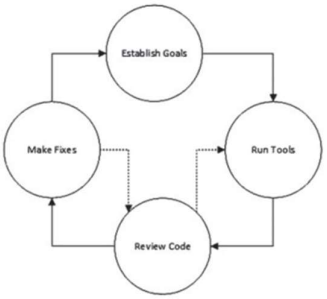

Fig. 2. Static analysis process [17]

IV. THE PROPOSED MODEL

This research is based on increasing security in the development of Information Technology, with the right way to take corrective steps in using various types of tools to detect security vulnerabilities through the following three steps: [3]

a. Find areas where security fails to reach in building reliable monitoring and warning systems.
b. Take measurable action to reduce or eliminate the negative impact of security issues detected on both application platforms.
c. Evaluate the results of each corrective action by comparing the situation before and after each phase.

A. *Monitoring and Warning*

Good monitoring to monitor various types of attacks against all application programs used, in order to have a reliable validation by recording all states that exist in each phase, but also plays an important role in providing clear transparency in the functionality of all security measures implemented. In this phase, we will install a metric scraper on each asset depending on the functional type and security metrics we are looking for that specific asset. Accurately running a system of alerts and rules based on metrics that the monitoring system collects from different assets is also another important thing that we should use to create a

successful DevSecOps file in the exchange of corporate information. [18]

B. Application Protection

Software composition analysis can be applied holistically to ensure that each open source dependency has a compatible license and is free of vulnerabilities. A behavioral by-product of this is that developers feel a sense of ownership over the security of their applications, getting direct feedback on the relative security of the code they write.

Once the code is checked and generated, you can start using security integration testing. Running code in an isolated container sandbox enables automated testing of things like network calls, input validation, and authorization. This test results in rapid feedback, enabling rapid iteration and triage for any identified issues, causing minimal disruption to the overall flow. If things like unexplained network calls or unclean input occur, the test fails, and the pipeline generates actionable feedback in the form of reporting and notification to the relevant team.

This can happen if the AST tool has a large capacity image or has to store data in a container. But the fact is that AST tools do take some time to run, and they can slow down the overall CI/CD pipeline. Interestingly, the CI/CD pipeline was never conceptualized with security in mind, but rather speed and convenience[19]. Here are good steps:

a. Hosting code written in secure and reliable source code, such as GitHub and Git, which can control the version files of the code sent to the associated repository and allow to quickly roll back to a previous version of the code if the code doesn't want to be pushed to the repository.

b. Run a static code analyzer against all newly written code and third-party libraries they use; This scan should also run automatically on each application deployment. As soon as vulnerabilities are detected during application deployment, the deployment work should end immediately and not go to production. The result of this type of failed implementation should be announced to developers and product managers, immediate feedback.

c. For individual developers about failed implementations can be a great source of truth to understand why application implementations fail. Loosely integrating with Jenkins in the job deployment process can fulfill this goal. Having detailed information about the results of running the static code analyzer with new application code in each new deployment will increase the visibility in the background of the deployment pipeline to a decent level

V. CONCLUSION

Using a variety of tools to detect different security vulnerabilities helps developers and organizations to safely release applications, reducing the time and resources that must later be devoted to fixing errors. During the secure software development cycle of an application where vulnerability detection tools help integrate security. Results Correlation between tools with different types is still an aspect that is not too broad. For this reason, it is necessary to develop a methodology or software that allows custom made automatically or semi-automatically for the evaluation and correlation of the results obtained with several different types of tools. It is very important to develop representative tests to perform a series of vulnerability tests included in OWASP and then combine them with SAST tools.

REFERENCES

[1] Issie Lapowsky. 2019. In Latest Facebook Data Exposure, History Repeats Itself. Wired (March 2019).

[2] Issie Lapowsky. 2019. In Latest Facebook Data Exposure, History Repeats Itself. Wired (March 2019).

[3] Matt Powell. 2019. 11 Eye Opening Cyber-Security Statistics for 2019. CPO Magazine (June 2019).

[4] K. Carter, ‘‘Francois Raynaud on DevSecOps,’’ IEEE Softw., vol. 34, no. 5, pp. 93–96, Sep. 2017. doi: 10.1109/MS.2017.3571578.

[5] Pulasthi Perera, Roshali Silva, and Indika Perera. 2017. Improve software quality through practicing DevOps. In 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer), 1-6.DOI:https://doi.org/10.1109/ICTER.2017.8257807

[6] Information Systems Audit and Control Association (ISACA). Accessed: May 20, 2017. [Online]. Available: http://www.isaca.org/KnowledgeCenter/Documents/Glossary/Cybersecurity_Fundamentals_glossary.pdf

[7] V. Mohan and L. B. Othmane, ‘‘SecDevOps: Is it a marketing buzzword?- Mapping research on security in DevOps,’’ in Proc. 11th Int. Conf. Availability, Rel. Secur. (ARES), Salzburg, Austria, Aug./Sep. 2016, pp. 542–547. doi: 10.1109/ARES.2016.92.

[8] Pulasthi Perera, Roshali Silva, and Indika Perera. 2017. Improve software quality through practicing DevOps. In 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer), 1–6. DOI:https://doi.org/10.1109/ICTER.2017.8257807

[9] Higuera, J.B.; Aramburu, C.A.; Higuera, J.-R.B.; Sicilia, M.-A.; Montalvo, J.A.S. Systematic Approach to Malware Analysis (SAMA). Appl. Sci. 2020, 10, 1360.

[10] Jeganathan S, 2019 DevSecOps: A Systemic Approach for Secure Software Development, ISSA Journal

[11] Felderer, M.; Büchler, M.; Johns, M.; Brucker, A.D.; Breu, R.; Pretschner, A. Security Testing: A Survey. In Advances in Computers; Elsevier: Cambridge, MA, USA, 2016.

[12] OWASP Foundation. OWASP Testing Guide, 2020. Available online: https://owasp.org/www-project-websecurity-testing-guide/ (accessed on 9 mei 2021).

[13] Sipser, M. Introduction to the Theory of Computation, 2nd ed.; Thomson Course Technology: Boston, MA, USA, 2006.

[14] Mohino, J.D.V.; Higuera, J.B.; Higuera, J.-R.B.; Montalvo, J.A.S.; Higuera, B.; Mohino, D.V.; Montalvo, J.A.S. The Application of a New Secure Software Development Life Cycle (S-SDLC) with Agile Methodologies. Electronics 2019, 8, 1218

[15] CMS Critic Awards (2020). CMS critic awards, https://www.cmscritic.com/awards/#bestopen-source-cms.

[16] Nunes, P.; Medeiros, I.; Fonseca, J.C.; Neves, N.; Correia, M.; Vieira, M. An empirical study on combining diverse static analysis tools for web security vulnerabilities based on development scenarios. Computing 2018, 101, 161–185.

[17] Chess, B., West, J. (2013). Secure programming with static analysis, USA: Addison-Wesley.

[18] Srinivasu N, On the viability of adaptive paris metro pricing in agent based model for federated clouds, International Journal of Recent Technology and Engineering (2018)

[19] Jyoti B. Kulkarni, Dr. Manna Sheela Rani Chetty, “Depth Map Generation from Stereoscopic Images Using Stereo Matching on GPGPU”, Journal of Advanced Research in dynamical and Control Systems, Volume 9, ISSN 1943- 023X, Special Issue – 02 / 2017.