INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 3)
Available online at: www.ijariit.com

Implementation of DevSecOps using Open-Source tools

Rahul B. S.
rbs9711@gmail.com
SJB Institute of Technology, Bengaluru, Karnataka

Prajwal Kharvi
paju476@gmail.com
SJB Institute of Technology, Bengaluru, Karnataka

Manu M. N.
manu2me1@gmail.com
SJB Institute of Technology, Bengaluru, Karnataka

ABSTRACT

DevSecOps is nothing but involving next level security to the DevOps process since Security is the key priority given to any application or data irrespective of any organization. DevSecOps not only provide security but also provides the same speed and agility which DevOps tends to provide any data or Application. DevSecOps will also tend to keep integrating, developing and updating of the latest security methods that tend to give adequate, proficient and productive results in a secured manner. Therefore, to demonstrate the DevSecOps we intend to use the open source tools which can be freely downloaded and used to demonstrate the data security. In order to run the frequent process of integrating and testing for the security of data, we require more resource and time respectively. To overcome this problem here we implementing and integrating security as part of the pipeline process and hence can achieve faster response time and fend off attacks before-hand and accordingly create a secure environment and more protected system.

Keywords— *DevOps, DevSecOps*

1. INTRODUCTION

DevSecOps abbreviates to Development security operations, which contains the process of how data can be transferred securely and fend off attacks like data breach and another tampering of data. The importance of DevSecOps is to transform the development cycle into a more secure environment. Traditionally DevSecOps acts as a bridge between operation, development, and integration. There is a considerably abundant requirement of data security for online platforms like Clouds and repositories like Git which is usually accessed by public irrespective of their identity. Hence, implantation of DevSecOps to the open source tools in order to secure the data pulled or pushed on to the repository can be useful in tremendous ways, where fixing code level is considered more robust and achieve certain advantages if done correctly.

However, DevSecOps brought a very new approach to integrating security process into development and operations and hence the fast-paced process chain can be formed. Relooking for security is an obstacle of time and hurdle in the era of continuous production cycles. The notion of DevSecOps appeared like a blooming solution to this problem and ensuring security-integration across the Development Operation workflow without disturbing the time and efficiency pace of DevOps processes. This process also reinforces the developers to understand the perks of data security where he can easily identify what kind of vulnerability has been found.

2. RELATED WORK

By embedding Security within the CI/CD pipeline is very much useful to collect a huge number of bugs and data vulnerabilities and to store several of reports on the defects and faults found in the source code downloaded from the repository. Here basically we are going to configure several static and dynamic tools which are used for checking the code for defects in the pipeline itself instead of dedicating a separate phase after deployment of source code for security check where the programmer has to again scan for faults and redesign the code from the scratch if major. Here static code scanning undergoes scanning without running the code i.e., in the non-runtime environment and checks for code smells, flaws and other bugs which can be found on the surface of the source code. Whereas the dynamic testing undergoes security scanning by running the code i.e., in runtime environment which has undergone static testing and hence detects for any further bugs and vulnerabilities once after deployment of code. Once both static and dynamic tests are undergone, a result is displayed showing a total number of faults and the same result is forwarded via email for further corrections.

3. PROPOSED WORK

The above figure shows the proposed DevSecOps system of our project. Here we, The Solution combines the process of development, testing, security analysis and deployment. The user should go to the UI and define the SCM repository location of the code to be scanned for security issues. The solution will automatically download the source code from the SCM repository and perform the security scan at a defined schedule. The project would perform static security analysis and generates the security report. Similarly, dynamic security analysis and generates the security report. This report can be stored as a structured data file and then pushed to an issue tracking system.

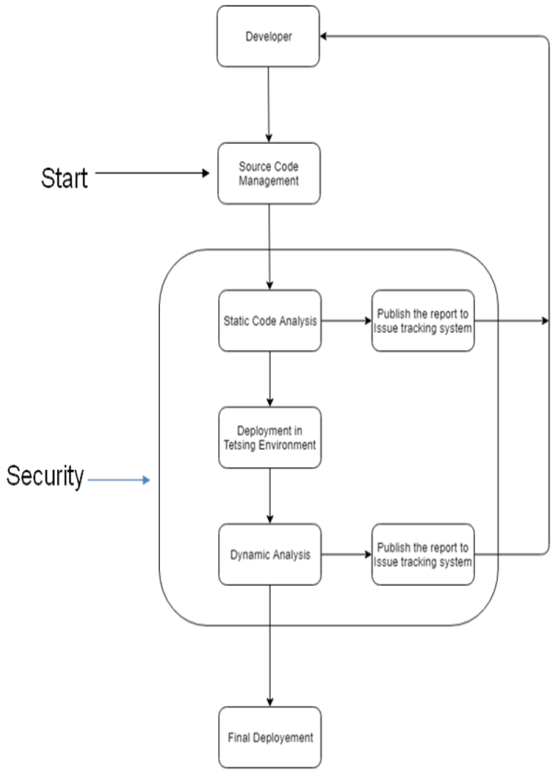

Fig. 1: Architectural diagram of DevSecOps

4. IMPLEMENTATION

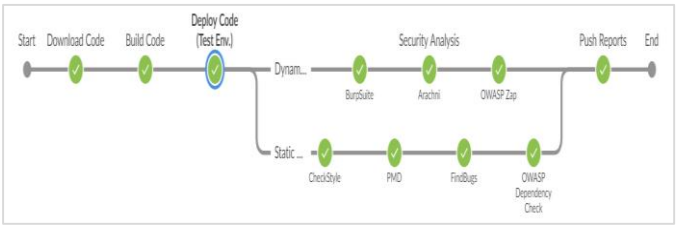

Fig. 2: Pipeline graph

The above pipeline graph shows the various steps undergone by the source code to the final deployment. Jenkins is a server of continuous integration and it checks out the source code from a source code repository and after this it builds code. Jenkins supports various source code management systems. One of the most popular source code management systems id Git. Jenkins downloads the code or module onto the local system.

The programmers then upload or load the block or module of code which they have developed or created into a common repository where it is committed and stored. The piece of additional code is undergone unit testing and can be further used by other colleagues for further enhancement of the code precisely. Use of Git with Jenkins will help the developer to commit code to GitHub repository with consistency. Teams are more likely to commit code changes in organizations more frequently in the continuous integration process, which further leads to better collaboration and software quality. The Software Configuration Management, Build Engineer who will maintain all software repositories which ensure that all incremental changes are synchronized to different code branches, which manage all software build infrastructure and it builds for continuous integration of code changes, and also manage lab infrastructure. We compile the code in the build stage. For code like Java, we can use the maven tool in Jenkins.

In this build phase basically, all the libraries which are used by the code or the dependencies which we call it technically and are downloaded from the internet in order to support the proper functioning of the code.

After completing the build phase, we move on to the testing phase. In this phase, we have various kinds of testing, which includes unit test where we test the chunk/unit of source code for any bugs and if found any, it automatically gets triggered and informs the programmer and emails the reports as well. Here the plugins used in static security testing in the Jenkins are Check style, PMD, and Find Bugs where the check style confronts to checking the style in which the syntax of the code in the programming language is written. While the PMD is used for finding an unused object, methods, variables in the source code. Whereas the Findbugs is used for segregating the bugs found in the code into four categories: scariest, scary, and troubling and of concern respectively. Similarly, the plugins used for dynamic security testing in the Jenkins is the OWASP dependency check where the foremost job of it is to check for all the unknown dependencies which are present along with the ones which are downloaded during the build stage and also checks if they contain publicly disclosed, open-source vulnerabilities hence separate the unknown ones by indicating and highlighting it respectively.

Now comes the release stage where the block of code is let for further corrections and rechecking of errors which are present in the code and hence the programmer corrects the respective code and the fixed code is pushed onto to the production line of the pipeline into the deployment phase.

5. CONCLUSION

DevSecOps the two seemingly opposing goals, “speed of delivery” and “secure code”, are merged into one streamlined/automated process. DevSecOps appeared as a boon ensuring security integration without disturbing the pace of DevOps-driven IT processes. Static and dynamic security check will be performed in each version or addition of the source code. The generated test results security tests will be sent to the issue tracking system for further monitoring and analysis of threats and thereby informing the programmer about any. Therefore, this project validates the building blocks without slowing down the process and without any compromise in security factors.

6. REFERENCES

[1] “Design and implementation of continuous integration scheme based on Jenkins and Ansible” by Wang yiran & Guo Yidong Submitted at InternationalConference on Artificial Intelligence and Big Data (ICAIBD) in May 2018.

[2] “Evaluation of Web Vulnerability Scanners Based on OWASP Benchmark” by Balume Mburano & Weisheng Si Submitted at 26th International Conference on Systems Engineering (ICSEng) in Dec 2018.

[3] Reinforcing DevOps approach with security and risk management: An experience of implementing it in a data center of a Mexican organization, Oswaldo Díaz, Mirna Muñozin 6th International Conference on software process improvement, Oct 2017.

[4] “The Importance of DevSecOps “, Katia Gomes in Northern Illinois University honours program, May 2018.

[5] [ISO/IEC: 27000: 2018], Information technology Security techniques — Information security management systems — Overview and vocabulary.

[6] [ISO/IEC 27002], Information technology — Security techniques — Information security management systems — Code of practice for information security controls

