Integrating Security with DevSecOps: Techniques and Challenges

Zaheeruddin Ahmed
School of Engineering & IT
Manipal Academy of Higher Education
Dubai, U.A.E
E-mail: zaheeruddin@manipaldubai.com

Shoba. C. Francis
School of Engineering & IT
Manipal Academy of Higher Education
Dubai, U.A.E
E-mail: shoba.francis@manipaldubai.com

***Abstract*— The delivery of software applications is key to the success of many development organizations. The software development process over the period increased its scope and included other areas like operations with core Information Technology. Software project practices with DevOps have demonstrated how to streamline these processes, improve the quality of products with present technologies and speed up the processes. In the project cell, a university students' organization which undertake software development projects and services implemented the DevOps methodology for student projects. This model showed results with effective completion of projects. However, what was missing was robust and secure applications, increasing concern and risk involved with insecure products that brought our attention to the implementation and integration of security model into our present development process. The goal of this article is to describe the integration of DevSecOps practice to our ongoing projects that will avoid insecure practices during development.**

Keywords— DevOps, DevOpsSec, Software Development

I. INTRODUCTION

The software development methodology field is already growing, and any development team can't oversee its significance. In the entire process, it's difficult to manage all the areas interrelated. Many developers have taken a systematic approach to make productive products. The software development methodology is a framework to construct, plan, and control the process of developing applications. The application of effective methodology can undoubtedly determine the success of any project. On the other hand, when no methodology is applied then it may lead to a variety of difficulties that become prevalent during the process. Some of the most common methodologies are waterfall, agile, or DevOps which is again a very systematic adaptable approach.

In Project Cell at our institution, we adopted DevOps methodology for students' major projects that are challenging concerning timeline, skillset, efficiency and completeness. The implementation of this methodology was taken first with two projects build on two different methodologies. One with waterfall model and other with DevOps. The results of both the project were comparatively different from a project developed with DevOps show substantial improvement in the deliverables [1].

However, the teams fail to realize the importance of security until the end as no software security model was applied leading to application with insecure features. This enforced the teams to reengineer the process to apply security. The teams have realized the importance of security towards the testing time of projects resulting in various issues exposed to the deployment process of the projects. In this paper, we will discuss the background of the process adopted during our DevOps methodology and discuss some of the challenges we faced during the process that lead to the implementation of DevSecOps methodology.

II.DEVOPS

Every organization looking to enjoin the power of methodology when it comes to development, developers must refurbish the way their development works with underlying workflows. The approach of DevOps gained huge acceptance with faster, better collaboration and communication for team building, testing, and releasing software. It leads to outstanding agility, flexibility, and quality, as well as cost efficiencies. But DevOps also creates new and sometimes formidable challenges. Developers want to innovate and move software and products to market faster, while operations staff prefer to keep things stable and are rewarded for up-time reliability. These two contradictory approaches can make it difficult to align the two groups, and they also make it challenging to address security concerns[2].

Some of the common practices in DevOps models continuous integration and delivery, incorporating microservices, Infrastructure and policy as a Code, error monitoring and logging, and small and often software updates. All these practices when applied must be a good fit for the organization needs [2]. On the other hand, DevOps practice of development is characterized by speed: which deliver products in quick time. Reliability: Applications developed with this methodology will result in a lower failure rate. frequency: will support frequent deployment.

In our projects with project cell, we implemented DevOps and had our practices on these lines and were hoping that the product will be of more quality and efficient, with continuous testing procedure in place, however when end products eventually came up with security as big concern, however security practices were never a part of our DevOps projects. The expectations from our projects were reliability which is the main characteristics. Although we achieve some expectations still the application lead to vulnerabilities towards the testing phase. However if the vulnerabilities are tested in the last stage that is before deployment we may find the vulnerabilities and the cycle of fixing it and testing it is going to push our deployment timelines away and how much is the variance will depend on the number of vulnerabilities found and the time is taken to fix the same and it would also

978-1-7281-3841-1/19/$31.00 ©2019 IEEE

depend in which part of the entire development the changes have to be made.

In the process, the teams realized that when creating a new system, we need to think of security from the beginning. Security is a nonfunctional requirement and does not affect the functionality of the product because of this it is often pushed to the end of the process. Testing for security at the end can move deadlines without a realistic goal since the impact of the vulnerability are not fully known at that stage there can be a change of architecture costing a lot to time. This gave us a lot of uncertainty and inefficiency in the process. Lately moved security from being at the end of the process to be distributed throughout the pipeline.

III. DEVSECOPS

A. *Overview*

DevSecOps is a combination of development, security and operations. It makes the team responsible for application security by implementing security activities and decisions at the same measure and speed as development and operations tasks. An organization with a DevOps framework must look for a shift towards DevSecOps approach to achieve a higher level of proficiency in security. A DevSecOps framework applies tools that ensure security is built into applications rather than being attached on arbitrarily later. By ensuring that security is present during every stage of the software delivery lifecycle. DevSecOps is about introducing security earlier in the life cycle of application development, thus minimizing vulnerabilities and bringing security closer to IT and business objectives.[3][4].

B. *Why DevSecOps*

In recent years organizations have experienced a huge change in development and process management. A major shift from dynamic provisioning shared resources to cloud computing has benefited development concerning speed, agility and cost, which in turn helped to advance application development. The ability to move to agile and DevOps methodologies made a big difference. In particular, DevOps which is the principle of integrating development and IT operations under the same unit helped with frequent feature releases to increased application stability [4]. On the other hand, security has not kept up with this leap of change, as it wasn't built to test code at the speed DevOps requires. This led to security the biggest difficulty for speedy application development.

C. *Benefits of DevSecOps*

Some of the benefits of DevSecOps are, it provides automation from the beginning of a process that reduces the chance of any mistakes, which can lead to vulnerabilities. Another major benefit is it reduces the need for security engineers to manually construct security supports. leaving teams free to work on policies [4].

IV.RELATED WORK

The authors in [5], discussed the lack of focus on DevSecOps workflow. The authors believe that this methodology helps in formalizing the gap between software and security. The authors further proposed the adaptation of Software security development life cycle SSDLC with the combination of DevOps and DevSecOps for agile development. The research was to focus on a designed framework to enforce Technology Development lifecycle.

The authors in [6] applied the concept of DevSecOps methodology to develop infrastructure as code (IaC) when writing insecure scripts to create development environments. This method will support in detecting security smells. that are recurring coding patterns indicative of security weakness and can potentially lead to security breaches. This methodology will help practitioners avoid insecure coding practices while developing infrastructure as code (IaC).

According to the authors in [7], DevSecOps guides implementation of IT Process. That may support the integration of cloud environment incorporating security practice and collaboration development with operation process. This methodology will also support automating the IT process in the development of applications that are fast and secure. The authors further proposed a model to reduce development time without the effect of quality and security.

The authors in [8] believe that many aspects involve the development of secure software. But the verification and validation of security must always be present, In the research, the authors proposed a security framework to guide the planning and definition phases of security requirements considering agile methods for application development and a DevSecOps approach.

In [9] the authors discuss the continuous deployment, the elapsed time for a change made by a developer to reach a customer can now be measured in days or even hours. To understand the emerging practices surrounding continuous deployment, the authors describe the DevOps and DevSecOps practices and environment used by these companies as they strive to develop secure and privacy-preserving products while making ultra-fast changes.

V. DEVSECOPS INTEGRATION WITH DEVOPS

Integrating security into DevOps to deliver DevSecOps requires new approaches, processes, and tools [4]. In our experiment introducing DevOps methodology for university student's capstone projects was successful as we have seen a considerable improvement in terms of deployment speed and end-user satisfaction, however, this implementation had not taken care of security perspective till the end. It was only in the end when the security tools were run on code and the vulnerabilities were identified.

The process of fixing this issue was going to delay our deployment as we don't know how much of application code must be changed, how much more testing is required and if there is any cost involved as part of the acquisition of any tool. The concept of DevOps was to have control over all aspects right from the start that is a 'shift Left' concept was adopted, but we observed that security was not implemented as part of DevOps process, but only towards the end of the deployment.

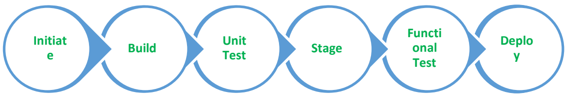


Fig. 1. The implementation process of DevOps

Case: The experiment results of DevOps in our project case were successful see figure 1, but the system came across with some of the problems that were recognized in the process of deployment. One instance was that we discovered that the configuration file retrieved from the repository for deployment was overwritten and the application was rendered nonfunctional. We had to take some approach to resolve this issue which was very vulnerable for the developed system.

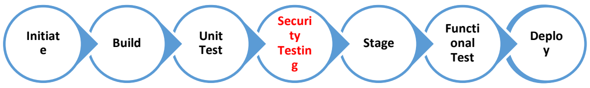


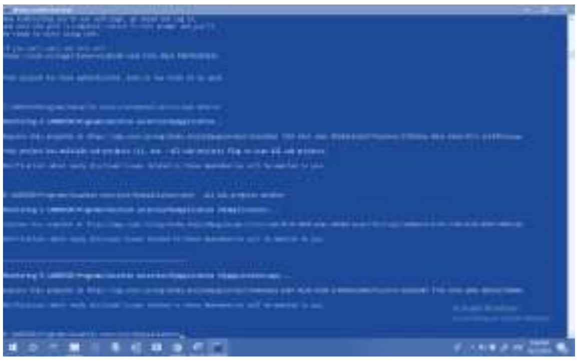
Fig2. Final pipeline implementing DevOps with Security

Procedure: We implement security right at the start of the project from the initialization stage, now we deposited our code in GitHub repository for continues checking available for all the members. This repository code was pulled into our build system and checked with npm package for dependencies. Then the testers applied unit testing, staging, functional testing to check for discrepancies to deploy on the production server. With this procedure shown from Fig 2, we secured the entry points to the system, we secured our repository of base images, ensure that the users were not pulling random code from GitHub. This is the stage where we believe that the code is still not complete and keeping in mind the vulnerabilities and analysis of tools that are appropriate for the security of an application.

The next stage for our process was more crucial as shown in the below figure 3, where we introduced the security gate after unit testing. This step was taken before staging the program, the gate introduced was with the help of tool Snyk for dependency checks and to enable vulnerability advisor to identify vulnerabilities as these tools have its database called Commonly known Vulnerabilities (CVE). It has been noticed that it's recommended to have advisory from more than one such tool. However, in our project, we tried with using only Snyk.

Fig 3. Snyk Install

Experiment Procedure: The files from GitHub repository can now be pulled into a directory or in another case GitHub access can be directly connected to Snyk. Then we installed Npm for checking dependencies and Snyk commands are executed on the Command Line Interface (CLI).



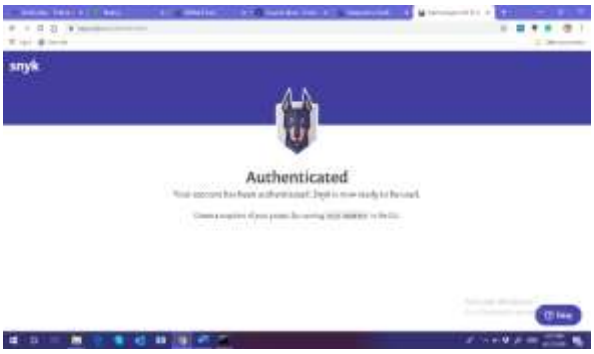


Fig 4. Snyk Authenticate

It was observed that Snyk pulls up all the vulnerabilities and shown in the figure 4-7 and then we worked towards fixing these vulnerabilities before moving code for staging and functional testing. This change security model helps us to overcome a lot of vulnerabilities like "Insecure Encryption", 'Unexpected code execution', 'Denial of Service', 'Directory Traversal'. These results have made the application more secure and robust.

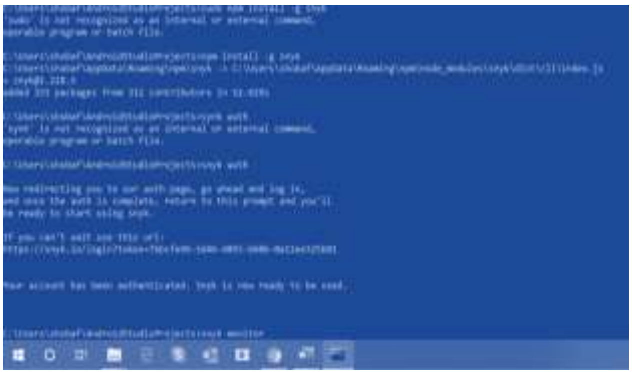


Fig 5. Snyk Authentication complete

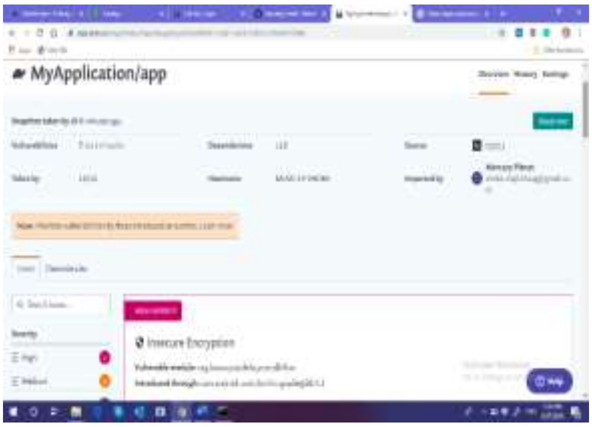


Fig 6. Snyk check on build files from GitHub repository

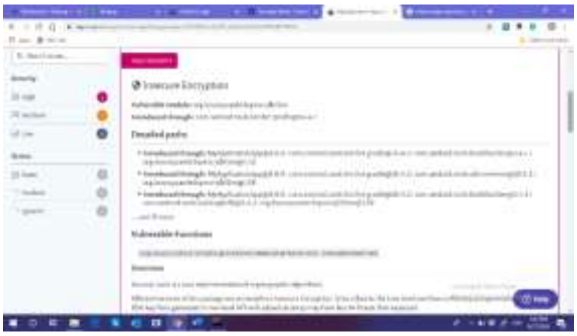


Fig7. Details of Insecure Encryption

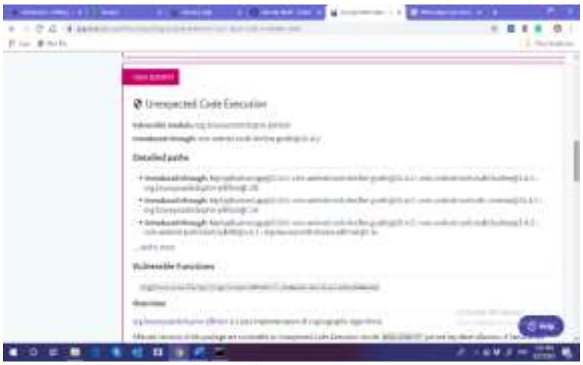


Fig8. Show Unexpected Code Execution

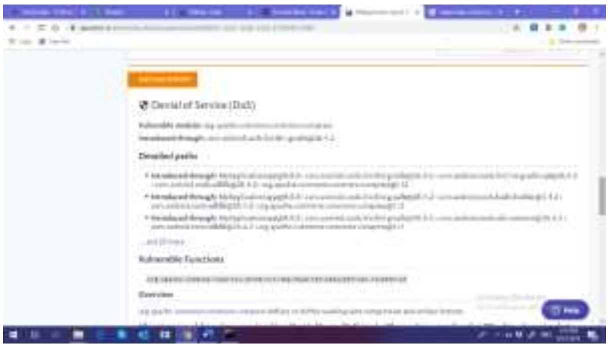


Fig9.Denial of Services Error

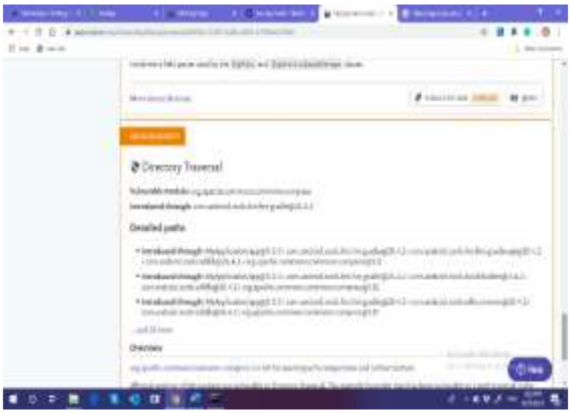


Fig.10 Directory Traversal

VI. CHALLENGES

Every Process faces an uphill task so is with implementing DevSecOps. In our process of implementing the procedure in student’s capstone project, the teams faced many challenges. First, the integration of individuals to work in a group connected with each other rather than working independently was a big challenge. Since the project was already towards the end not everybody was ready to return to development and amend changes instantly. This process again worked gradually facing enough human resources. Further adopting a new system again after members have already become deeply accustomed to present development processes was a big task. Another challenge was lack of experience by the members in new trends and needs time to explore the unfamiliar procedures. This took more time for people to be familiar with the right tools required for change.

Two teams were working for our capstone projects for a duration of four months with different metrics and tools they used. Combining the teams at this stage of the project for a mutual agreement to integrate or not was a tradeoff keeping in mind the milestones of the project was again challenging. Again, choosing the best tools that fit well for all the teams was ambiguous. One big challenge was to appropriately integrate all teams, applications, and tasks in order to build, deploy, and test in a continuous manner. It’s not an easy task to bring together tools from various departments and synchronize them on one platform.

Implementing security in continues development was another challenge as this is seen as something that comes towards the end of development, getting security to familiarize to the DevOps process was again challenging. It is been observed that teams were about to give up the projects because a lot of time was wasted in trying to put things in place and make task work perfectly. Nevertheless, adopting DevSecOps procedure was a complicated process, but once completed, it improves the entire operation. One last observation was that when achieving to get flawless security at every phase of development was tedious work for developers.

VII. Conclusion

In this research article, we have been able to discuss the limitations in DevOps methodology and see how missing security model can cause the problem of insecure development. We have also discussed the various techniques of DevOps and DevSecOps can support the development process. We have seen the challenges that we faced during the process without DevSecOps. Our explanation also provides which of the stages where DevSecOps can be adopted. The result shows that with DevOps we can develop a secure application ensuring the constraints of DevOps methods like speed. Our future work will be to implement the same concept on a project with a larger scale and scope.

Acknowledgement

The authors would like to acknowledge the valuable inputs given by the professors & colleagues at MAHE, Dubai U.A.E. Further, we are very grateful to the students and Project cell members of School of Engineering and IT for their active participation. We would like to thank the facilities department, School of Engineering and IT for providing the logistics and infrastructure for completing this research project.

References

[1] Ahmed Z, Francis S, “Introducing DevOps Methodology for University Students Capstone Projects”, Recent Advances in Science Engineering & Management, Journal Research Direction No 45489, Volume 6 Issue 11 May 2019.

[2] Ethan Thorpe, *A Comprehensive Beginners Guide to Learn DevOps Step by Step,* Amazon Digital Services, 2019

[3] ForcePoint, (2019, September), What is DevSecOps?, [Online]. Available: https://www.forcepoint.com/cyber-edu/devsecops.

[4] DevOps, (2018, January), Doug Drinkwater, What is DevSecOps? Developing more secure applications, [Online], Available: https://www.csoonline.com/article/3245748/what-is-devsecops-developing-more-secure-applications.html.

[5] Jessica Nguyen, Marc Dupuis, “Closing the Feedback Loop Between UX Design, Software Development, Security Engineering, and Operations”, SIGITE '19: Proceedings of the 20th Annual SIG Conference on Information Technology Education.

[6] Akond Rahman, Chris Parnin, Laurie Williams, “The seven sins: security smells in infrastructure as code scripts”, ICSE '19: Proceedings of the 41st International Conference on Software Engineering, IEEE Press, 2019.

[7] Sara B. O. Gennari Caraturan, Denise Hideko Goya, “Major challenges of systems-of-systems with cloud and DevOps: a financial experience report”, SESoS-WDES '19: Proceedings of the 7th International Workshop on Software Engineering for Systems-of-Systems and 13th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems, IEEE Press, May 2019.

[8] Sara B. O. Gennari Carturan, Denise Hideko Goya, “A systems-of-systems security framework for requirements definition in cloud environment”, ECSA '19: Proceedings of the 13th European Conference on Software Architecture - Volume 2, ACM, Sep 2019.

[9] Laurie Williams, “Continuously integrating security”, SEAD '18: Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment, ACM, May 2019.