DevSecOps: Security Expertise a Key to Automated Testing in CI/CD Pipeline

Article · December 2020

CITATION
1

READS
923

1 author:

Bakary Jammeh
Bournemouth University
1 PUBLICATION **1** CITATION

SEE PROFILE

DevSecOps: Security Expertise a Key to Automated Testing in CI/CD Pipeline.

Bakary Jammeh
Department of Computing and Informatics
Bournemouth University
Poole, England.
s4926404@bournemouth.ac.uk

ABSTRACT

To keep up with the speed and agility of DevOps, Security should be involved throughout the development cycle from planning to monitoring, and organization teams needs to automate some of the security processes and practice to meet these demands. DevSecOps is the integrating and automating of security to help DevOps teams find issues earlier and mitigate them quickly while saving cost and time. We present some research questions to find out if teams with less or no security expertise can successfully integrate security testing in their CI/CD pipeline. We understand that access to security knowledge and expertise is key in DevSecOps, and every team member must understand their security responsibilities.

KEYWORDS

DevSecOps, Secure DevOps, Continuous Security, Automated software testing, CI/CD pipeline security, Secure development.

I. INTRODUCTION

The way Software is develop and offer has evolved from Software as a Product (SaaP) issued to individuals into Software as s Service (SaaS) to many users with different needs [1]. Mainly because devices that use software increases (IoT devices) and people are so much dependant on software application in our daily lives from homes, education, health, and financial sectors. The Security of these systems is ever increasingly becoming a major focus area for most organisations due to increase in cyber-attacks [2]. Consequently, new software development methodologies such as DevOps which promotes speed and agility needs to include a practice called DevSecOps to incorporate security testing during this rapid development cycle rather than waiting late in the end [1] [2] [3]. But before we go further it is important to introduce DevOps as a prerequisite to understanding DevSecOps.

DevOps

Development and Operations (DevOps) is based on collaboration and shared responsibility between the development team and operations team to improve product quality by solving problems together [2]. The concept was introduced in software development to extend the agile software development principles and it emerged to give speed and efficiency which software and IT teams are about [4] [5]. It promotes the merging and prioritizing of development and operations needs to tackle issues, agree on processes and products to automate during software development and deployment[1]. This automated process is known as Continuous integration and Delivery (CI/CD) pipeline.

CI/CD Pipeline

CI/CD is a well-known practice in DevOps to ensure fast delivery of new features and it enables development teams to deliver code changes constantly and consistently in production [6]. Continuous Integration (CI) enables developers to automatically integrate new code in a shared repository and check for errors at same time [2], while Continuous deployment enables frequent deployment of software in production environment when checks are passed. Continuous Delivery (CD) helps teams to keep the software in a releasable state for customer use and reduce release time so as to get quick feedback from users for improvements [1].

DevSecOps

Integrating security in DevOps practice brings about the term commonly known as DevSecOps. DevSecOps is a methodology in which security is present throughout the entire life cycle of application development rather than being an afterthought[3]. Traditionally, security has always been about exclusion and that means Development, Security, and Operations are in separated silos, and Security testing is done after the development is finished [7]. DevSecOps promotes cultural change in teams by putting security at the forefront of requirements, so everyone participates and ensure security is achieved in the project [2]. It is understood that DevSecOps isn't just about security automation, and there are some key principles such as Cultural change, Measurement and Sharing of knowledge [8]. We focus on automation of some security testing.

II. RESEARCH METHODS

This study is conducted to get an understanding of what DevSecOps means and to find out find if teams without security expertise can successfully adopt the practice and the benefits and challenges such teams might face. We use journals such as Google scholar, IEEE, ResearchGate etc. to find related literature using the above keywords listed. To clearly identify the goal of the paper, some research question was formulated:

RQ1: Why automate security testing?
RQ2: What/what not to automate and the security testing types in DevSecOps?
RQ3: What are the benefits of DevSecOps?

RQ4: Can teams without security expertise implement and adapt DevSecOps practice?
RQ5: What are the challenges in adopting DevSecOps?

III. BACKGROUND

The term DevOpsSec was first used in 2012 by Neil MacDonald to integrate security within DevOps practices without affecting speed and agility [9]. DevSecOps is a methodology in which security is integrated into the entire life cycle of application development and rather than being an afterthought [10]. Many organizations are introducing DevOps and DevSecOps in their SDLC and there are quite some astonishing figures. It is suggested that DevOps market might grow from $3.4b in 2018 to $10.3b in 2023 and DevSecOps to grow from $1.5b to $5.9b in 2023 [19]. The DevSecOps practice is based on five principles (CAMS) for successful implementation of security into the development life cycle [2].

Culture: DevSecOps culture promotes shared responsibility of security and promoting collaboration between the development, security, and operations team. Every department should integrate security in their work and that means security people should be involve from the project inception phase [2]. DevSecOps is about inclusion and working together as a team [7] and tends to eliminate the traditional practice of having separate silos.

Automation: DevSecOps is focus on 100% automation of security controls and processes in a way that will not reduce speed and agility [3]. The software testing activities are done automatically by using test tools (software) to do whatever human tester do manually [11]. And it is not just about testing and deploying, it also includes release management, configuration management, monitoring [12]. However, test automation cannot eliminate or entirely replace manual testing as it is not possible to automate all test cases [9]. Manual checks are vital in some cases as certain errors or issues such as authentication and authorization are impossible to be detected by automated testing tools [3].

Measurement: DevSecOps encourage the use of monitoring and metrics by teams to measure vulnerabilities and threats, which is important to keep records of performance and improve the quality of the software [3]. Everything that is relevant must be measured and teams cannot improve their product if measurements are left out [10].

Sharing: DevSecOps supports sharing of knowledge between all teams with the aim to integrate security into every process. It is the education and cross-training for each member of development, operation, and security team [8] about their security duties. The security processes can only be improved when teams constantly share challenges they face and how they can help each other out [3] [7].

IV. RELATED WORK

Implanting security within the CI/CD pipeline, monitoring and automated set up in production environment is very much useful in finding bugs and vulnerabilities especially in an inexperienced team with less or no security knowledge or expertise to manual find and assess vulnerabilities. Myrbakken and Colomo-Palacios described the meaning of DevSecOps, its benefits and challenges for adopting the practice and how it has evolved since it was first introduced [1]. The authors understood DevSecOps is meant to shift the mindset of everyone to ensure the security of the product.

Ahmed and Francis discussed the importance of security in DevOps processes and shared the challenges of DevOps processes without security considerations. They explained how to integrate security in an ongoing DevOps project and believed that missing security practice can cause the problem of insecure software at the end [13].

T. Rangnau, R. Buijtenen et al. presents how to implement continuous dynamic security testing in CI/CD pipeline and investigates the pitfalls of such testing. The authors were convinced that not many literature focus on dynamic Security testing (DAST) and explains how to integrate suitable tools to scan for vulnerabilities in a workflow[2]

Khan (2020) outlines the secure DevOps workflow and how organization can infuse continuous security testing in their Continuous delivery pipeline. Perhaps one of the most relevant work in terms of integrating security in DevOps environment [12]. Khan discussed security controls, tools, automated checks/testing and best practice to make sure the software is tested at each stage of development.

Whilst there are several studies and publications on DevOps and DevSecOps, to the best of our knowledge none of these discusses or focuses on teams without security knowledge. Our work investigates this research gap and to find out if such teams can effectively adapt DevSecOps, the benefits and challenges they might face.

V. RESULTS

RQ1: Why automate security testing?

Application security testing should be integrated with the CI/CD pipeline as traditional security testing methods cannot keep up with the speed and agility of DevOps [1] [13]. Development and automated testing at the infrastructure and service layer should be in the same continuous delivery flow (CI/CD) to tackle the problems with traditional development [12]. However, according to [3], 59% of organization failed to include dynamic and static security testing in their pipeline. Yasar and Kontostathis are convinced that the results these tools provide are helpful and should be carefully tracked and measured to better understand issues [3], and to make sure a quality software reaches production [7] [14]. Security expertise is needed to correctly configure automation and according to L. Bass, R. Holz, P. Rimba et al. securing a CI/CD pipeline is

complicated due to skills and knowledge required [2], and the different tools involve in running the workflow [15]

RQ2: What/what not to automate and the security testing types in DevSecOps?

Garousi and Mantyla [11], the decision of what and when to automate software testing is vital as wrong decision can be catastrophe if automation is not applied with the right tools and approach. However, what and when to automate is dependent on the project and technologies[7]. Garousi and Mantyla came up with five factors that influences what and what not to automate in software testing:

i. Software Under Test (SUT)
ii. Test case related factors
iii. Test tool related factors
iv. Human and Organisational factors
v. Cross-cutting factors

And according to Sabetto, below are some of the security testing to involve in DevSecOps to help identify vulnerabilities in a software and all of these testing requires good security skills and knowledge [16].

i. Static Application Security Testing (SAST): Scans for vulnerabilities in the source code before integrating codes together.
ii. Dynamic Application Security Testing (DAST): Scan for vulnerabilities after the application is deployed and running, it is a black box testing technique.
iii. Vulnerability Scanning and Penetration Testing: Automatically identify common vulnerabilities in the application against some predefined rules and exploiting them from attacker's perspective.
iv. Configuration Management and Compliance: Knowing how the application is configured and following regulation policies such as PCI-DSS.
v. Infrastructure as Code: Written scripts to be triggered automatically to run test and deploy securely without errors in a consistent manner.
vi. Continuous Monitoring: Consistent monitoring of production environment can help in finding issues for future improvements.

RQ3: What are the benefits of DevSecOps?

Organizations often state "We take your security and Privacy Seriously" but not many live by it [16]. Security must be taken seriously to avoid potential lawsuits and fines and according to Cope, strong security starts with software development [17]. Below are some of the benefits of DevSecOps Practice.

Security Automation: Manual Security testing has not kept with the speed an DevOps. Security must be continuously involved in each phase and run tests automatically to keep up with this leap of change [5], [6] [11]. Automating security enables expanding of workload and scope, makes controls faster and possible for error detection and reporting [6].

Shift Security to the left: The term "Shift to the left" means security has to be considered and involved from the start of the product development and it also means more security testing by developers rather than being handled at a later stage [17]. Including security earlier will lead to less errors and it is believed that Static analysis tools are one of the things to help shift security to the left as they can be run on codes earlier before deploying in an environment [4]. However, not many organizations have the expertise to make the best use of these tools [7].

Cost and Value: Security from the start of the project will enable teams find errors quicker and fixing them. Many believed that it can cost a fortune trying to fix bugs in production [1] [3] [7]. In the case of a cyber-attack, businesses can face a financial penalty due to a security hole being compromised or for non-compliance to some regulations such as GDPR and PCI-DSS [18].

RQ4: Can teams without security expertise implement and adapt DevSecOps practice effectively?

Implementing secure DevOps means teams must develop expertise and processes to discover and protect against threats and risk [2]. The authors are convinced that threat modelling and risk assessments are important practice of DevSecOps which requires knowledge and skills [2][7]. Threat modelling is a technique that must be employ to secure applications and APIs as it enables teams to identify and priorities possible threats malicious individuals might exploit [15]. And according to Rio, applying the wrong standards and controls is a popular mistake in organizations and that teams must know the language, framework and other technologies to configure security testing rules in the right way[4]. A good collaboration and understanding of security practices should be shared amongst teams with a complete change in peoples attitude towards security for a successfully DevSecOps adaption[1].

RQ5: What are the major challenges in adapting DevSecOps?

Organizational Change: Securing development is not easy due to the complexity of software[17]. DevSecOps implementation means organizations must adopt change and the barriers between the security teams, development and operation team must be eliminated [2] [11].

Security Knowledge: Security methods in DevOps must be agile and these methods needs to be understood and accepted by all teams [2]. Skills and staff training are required to build security in DevOps process[8].

Tools and Configurations: Getting the right tools and the right configurations is vital to finding bugs and other issues.

And understanding the abilities and limitations of these tools is a major challenge for teams[4].

Teams and Tools: Integration of all the different teams and the vast variety of tools needed in CI/CD pipeline can be difficult. Choosing the best tool that best suites all the teams and mutual agreement on requirements and controls can be ambiguous [8].

VI. CONCLUSION

In this paper, we researched what DevSecOps means, and if organization teams without security knowledge or expertise can successfully adopt DevSecOps. We want to find out if the practice can be adopted without a dedicated security personnel or team. We also identified the key principles teams should adhere to, the benefits of successfully implementation and some of the challenges of DevSecOps.

We found out that DevSecOps is understood by many as the integration of security testing and practice into the DevOps development Methodology and that it should be everyone's responsibility to ensure security is achieved. After reviewing some of the literature, it become clear that expertise and a good understanding of security needed to implement security, and that knowledge needs to be share amongst teams' members to avoid security illiterate.

REFERENCES

[1] H. Myrbakken and R. Colomo-Palacios, "DevSecOps: A multivocal literature review," *Commun. Comput. Inf. Sci.*, vol. 770, no. September, pp. 17–29, 2017, doi: 10.1007/978-3-319-67383-7_2.

[2] T. Rangnau, R. v. Buijtenen, F. Fransen, and F. Turkmen, "Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines," pp. 145–154, 2020, doi: 10.1109/edoc49727.2020.00026.

[3] H. Yasar and K. Kontostathis, "Where to Integrate Security Practices on DevOps Platform," *Int. J. Secur. Softw. Eng.*, vol. 7, no. 4, pp. 39–50, 2017, doi: 10.4018/ijsse.2016100103.

[4] S. Mansfield-Devine, "DevOps: finding room for security," *Netw. Secur.*, vol. 2018, no. 7, pp. 15–20, 2018, doi: 10.1016/S1353-4858(18)30070-9.

[5] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "DevOps," 2016.

[6] L. Williams, "Continuously integrating security," *Proc. - Int. Conf. Softw. Eng.*, pp. 1–2, 2018, doi: 10.1145/3194707.3194717.

[7] R. B. Salesforce and K. Carter, "SOFTWARE ENGINEERING Francois Raynaud on DevSecOps," no. October, pp. 93–96, 2017.

[8] M. Sánchez-Gordón and R. Colomo-Palacios, "Security as Culture: A Systematic Literature Review of DevSecOps," *Proceedings - 2020 IEEE/ACM 42nd International Conference on Software Engineering Workshops, ICSEW 2020*, pp. 266–269, 2020.

[9] R. Kumar and R. Goyal, "Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC)," *Comput. Secur.*, vol. 97, p. 101967, 2020, doi: 10.1016/j.cose.2020.101967.

[10] J. Caraballo-vega, "Pipelines Use Case : Docker Container Scanning BUILD CLEANUP Use Case : Black Box Enumeration of System," no. August, 2019.

[11] V. Garousi and M. V. Mäntylä, "When and what to automate in software testing? A multi-vocal literature review," *Inf. Softw. Technol.*, vol. 76, no. April, pp. 92–117, 2016, doi: 10.1016/j.infsof.2016.04.015.

[12] M. O. Khan, "Fast Delivery, Continuously Build, Testing and Deployment with DevOps Pipeline Techniques on Cloud," *Indian J. Sci. Technol.*, vol. 13, no. 5, pp. 552–575, 2020, doi: 10.17485/ijst/2020/v13i05/148983.

[13] Z. Ahmed and S. C. Francis, "Integrating Security with DevSecOps: Techniques and Challenges," *Proceeding 2019 Int. Conf. Digit. Landscaping Artif. Intell. ICD 2019*, pp. 178–182, 2019, doi: 10.1109/ICD47981.2019.9105789.

[14] J. Wolf and S. Yoon, "Automated Testing for Continuous Delivery Pipelines," pp. 1–12.

[15] L. Bass, R. Holz, P. Rimba, A. B. Tran, and L. Zhu, "Securing a deployment pipeline," *Proc. - 3rd Int. Work. Release Eng. RELENG 2015*, no. May, pp. 4–7, 2015, doi: 10.1109/RELENG.2015.11.

[16] R. Sabetto, "DevSecOps – Security and Test Automation Vibha Dhawan ▪ Clearly describe how Security and Testing can be integrated into a," no. March, 2019.

[17] R. Cope, "Strong security starts with software development," *Netw. Secur.*, vol. 2020, no. 7, pp. 6–9, 2020, doi: 10.1016/S1353-4858(20)30078-7.

[18] Capita, "GDPR and data protection in the payments environment - an overview," *Pay360*, 2018.

[19] Markets, 2018: DevSecOps [Online] www.marketsandmarkets.com Available From: https://www.marketsandmarkets.com/PressReleases/devsecops.asp [Accessed 10 Nov 2020].

[20] devsecops, 2015: devsecops [Online] https://www.devsecops.org/blog/2015/2/15/what-is-devsecops [Accessed 10 Nov 2020].