ISSN: 2583-9888
DOI: doi.org/10.51219/JAIMLD/nagaraju-islavath/311

URF PUBLISHERS
connect with research world

Journal of Artificial Intelligence, Machine Learning and Data Science

https://urfpublishers.com/journal/artificial-intelligence

Vol: 1 & Iss: 1

Research Article

Shifting Security Left: Integrating DevOps with Secure Development Practices

Nagaraju Islavath*

Nagaraju Islavath, Independent Researcher, USA

Citation: Islavath N. Shifting Security Left: Integrating DevOps with Secure Development Practices. *J Artif Intell Mach Learn & Data Sci 2020*, 1(1), 1368-1371. DOI: doi.org/10.51219/JAIMLD/nagaraju-islavath/311

Received: 02 August, 2020; **Accepted:** 18 August, 2020; **Published:** 20 August, 2020

***Corresponding author:** Nagaraju Islavath, Independent Researcher, USA, E-mail: islavath.nagaraju@gmail.com

ABSTRACT

Traditionally, security in software development applications has become an added extra or incorporated at the concluding stages of the software development life cycle. However, the enhanced complication of contemporary software and the high risk of unauthorized access have determined the necessity of security integration at the earlier stage of software development. This is commonly referred to as 'shifting security left', where security is proactively built into the DevOps pipeline as first thought in the development process rather than being an additional consideration. In this work, the author looks at how the DevOps approach can be aligned with secure development processes to develop software with security in mind. Periodic security checks, security testing in all stages of the development process, utilization of automated tools and strong cooperation between security and development teams will minimize potential risks and improve application security conditions. This paper uses real-life scenarios and brief case studies to show that moving security left results in more secure, faster and more reliable applications.

Keywords: DevOps, secure development, Shift left, Security automation, CI/CD, DevSecOps, Security integration

1. Introduction

The technological era people find themselves in today demands security more than anything else. As the rate and level of sophistication of cyber-attacks rise, ensuring that applications are secure is an important issue for all companies. Nevertheless, classical trends in SDP management do not guarantee a proper focus on the security problem, which more often becomes an appendix, where threats are considered and eliminated only at product completion. This reactive approach exposes applications to any attack and may lead to expensive delays and security incidents[1]. The idea of security shifted left is to incorporate security into the development process starting from the first stage. API security tests should be carried out during the system planning process so that, when a problem is recognized, it can be eradicated as early as possible in the SDLC process. With this style of security, the likelihood of cyber theft or other attacks is minimized; the general security of applications is enhanced and overall deployment time for applications is less due to the detection of security problems at an earlier stage.

Shifting security left is integrating security into the development and deployment process upstream rather than downstream as a bolt-on. The discipline that makes this possible is DevOps, which blends development and operations in software delivery. It also means that DevOps needs to be about automation, collaboration and the continuous integration of features - and all of these concepts are critical to integrating security into the development process. When implementing integrated DevOps, special measures must be undertaken to protect an organization's applications and data from various threats due to security integration into each development phase[1]. This research work examines how the concept of moving security left can be implemented by adopting DevOps with security methods. The authors describe some of the typical issues of traditional security approaches, describe what it means to take security left and offer a plan for achieving secure DevOps. The paper also reviews case studies of organizations that have implemented DevSecOps to raise awareness and prove the viability of the DevSecOps model.

2. Main Body

Problem Statement

Previous security models in the development process are typically prevention, where security issues are detected and addressed once the bulk of the development process is done. The use of this approach has various demerits, as explained below. First, delay of security issues towards the later stages of the development cycle results in time consumption and cost escalation. When the vulnerabilities are discovered in the later stage, the developers may have to rewrite significant portions of code, which prolongs the time it takes to release the application and is costly[2]. Moreover, because security checks are performed during the end of the development cycle, there is inadequate time for application testing and securing it, increasing the possibility of creating applications with open security holes.

One drawback of conventional security strategies is that the development and security teams rarely work together. In most organizations, security is an alien concept entirely different from the development team, with security folks getting to work only when the application is nearly done (Perera et al., 2016)[3]. This distinction may result in misunderstanding and mismatch of goals, where developers are concerned about performance and usability, while security professionals are concerned with threats. It states that if these teams are not integrated, security problems are not considered or are considered when it is too late, consequently producing insecure applications.

The sophistication of modern applications also contributes to the problems due to the constant evolution of the software developed. Today's Applications are developed and deployed using technologies such as cloud, microservices, containers and APIs, bringing in new forms of security threats. As the dimensions of attack vectors grow, it is only challenging to remediate them conventionally. Further, with continuous integration/continuous delivery (CI/CD) pipelines, the development rate increased much more, meaning there is even less time to dedicate to secure testing. Last but not least, the modern security environment has ensured that mere reactive security management strategies are not useful anymore[3]. Threats in cyberspace are evolving, and the threat agents are exploiting weaknesses in both application systems and those of the underlying infrastructure. While organizations incorporate DevOps into their software delivery models, security must also follow to mitigate these new-age threats.

Solution: Shifting Security Left with DevOps

Moving security left means making security a part of the DevOps process, where security becomes built and is part of a continuous process. The purpose is to detect problems as early as possible, limit security failures in production and make security everyone responsible in development, operations and security teams. DevSecOps means integrating security measures into the development and production processes characteristic of DevOps methodologies.

When we talk about moving security to the left, one of the major approaches is integrating security testing into the CI/CD pipeline. Security testing tools must be incorporated into the process to scan the code for threats when written, tested, and deployed[4]. Some techniques include static application security testing (SAST), dynamic application security testing (DAST), and scanners that detect known vulnerabilities in third-party libraries and dependencies. Implementing these security scans automatically makes it possible to detect security holes before insecure code makes its way to the production environment.

Another important factor discussed as a part of shifting security left is collaboration. In a classical development model, security teams and developers are divided where cooperation is either rare or insufficient. DevSecOps, on the other hand, strives for proper cooperation between these teams, as security concerns must involve everyone from the project's onset[4]. Security specialists collaborate with programmers to integrate proper security concerns during application development. This form of a working relationship assists in closing the gap between the development process and the security approach, thus keeping the two groups' objectives aligned.

Shifting security left also consists of another practice called Security as Code (SaC). It also allows organizations to automate policy conformity in security provisions and configurations they apply throughout their computing infrastructure. Software like Ansible, Chef and Puppet enable an organization to write security into code and ensure every system built meets these security prerequisites[5]. This makes it easier to avoid configuration mistakes and allows security policies to be properly implemented across environments.

Attention to feedback and constant checking are important aspects of shifting security to the left. Essentially, DevSecOps focuses on the continuous security process because security is checked and resolved in parallel with the software development process. Currently, monitoring tools like Prometheus, Grafana and ELK Stack are used to keep track of their applications and infrastructure security to act when there is a threat[5]. The positive feedback loop ensures that security and development teams are always on the same page regarding the application's security status and making necessary changes. Last but not least, training and education also play an important part in the direction of DevSecOps. One way is to educate developers on proper security coding mechanisms. This is to make sure that developers create codes with security in mind. In this way, knowledge and tools can be given to the developers, eliminating potential security threats and making security an extensive part of the development process.

Use Cases

They should shift security left, as one of the apt examples of the financial services industry shows. A big global bank, which experienced growing cyber risks, decided to include security into its CI/CD process, implementing DevSecOps strategies. The bank's IT automated security checks by utilizing both SAST and DAST tools to detect the problems early in the development phase. Consequently, shifting security to the left helped the bank cut the number of vulnerabilities in the applications in half and increase the velocity of its cycles[6]. Integrating the D and security teams made security become everyone's duty and there were more secure applications to market in shorter periods.

DevSecOps has become relevant in guarding electronic health record (EHR) systems in the healthcare sector. A healthcare provider that the organization has been working with was in the process of its digital transformation and unfortunately encountered problems in protecting EHR applications and, at the same time, meeting the requirements of healthcare legislation[6].

When the provider transitioned to DevSecOps, it incorporated the security assessment into the CI/CD process, where every piece of code was scanned for vulnerabilities before release. Moreover, the provider introduced constant monitoring mechanisms through which it could obtain actual information about applications' security to address dangers adequately. Therefore, this approach enhanced the security of the EHR system while considering the EHR industry regulations.

It has especially helped keep e-commerce platforms secure in the retail industry by incorporating DevSecOps. An e-commerce giant recently experiencing a surge of cyber threats aimed at its pay-per-click mechanism embraced DevSecOps implementation. The retailer made security testing tools part of continuous integration and delivery, which check the code and the infrastructure for any security issues[7]. Security left proved beneficial to the retailer as it is aimed at preventing future attacks on the e-commerce platform and stopping attackers from exploiting such weaknesses that they identified in the platform.

DevSecOps has been applied in the government sector to protect key facilities. A federal agency responsible for dealing with sensitive information had issues regarding the safety of its applications and structures. Thanks to the powerful concept of DevSecOps, the agency was able to create an understanding of security checks that would automatically scan all applications before their deployment. Also, Security as Code practices were adopted to automate security policies across the agency, ensuring proper security was applied across every system. Through this approach, the security of the agency's infrastructure was enhanced. The efforts and time needed in handling security were minimized.

Thirdly, the telecommunications industry has also witnessed the impact of shifting security left. A telecom major handling a huge cloud environment struggled with application security and work productivity[7]. Back then, the company embraced the DevSecOps model, which means that it is possible to automate security checks that should be done on the cloud and put on certain security policies. This approach enhanced the safety of applications belonging to the company and lessened the chances of disruption of business processes due to security breaches.

Impact

Consequently, security can be and must be shifted left, and the effects of this shift are great for both security and everyday operations. The first is the decrease in vulnerabilities, which is one of the most evident benefits to be expected out of the program. The proposed implementation methodology associates security checks with the software development life cycle, which means that security issues can be detected early and it is impossible to let them creep into the productive environment[8]. This proactive measure enhances the security of applications to help decrease the vulnerability of cyber-attacks and data breaches.

The other benefit of shifting security left includes the faster cycles of development. Security controls were traditionally implemented at the end of the development life cycle, as in other development models; this has drawbacks in congesting the development cycle[8]. Consuming security as a part of the CI/CD pipeline helps organizations incorporate security testing into their systems to help them detect flaws before they become major issues. This automation attenuates the need to perform a security test manually while enhancing organizations' ability to release code more frequently.

Another advantage of shifting security is that there will be better relations between the development and security teams. DevSecOps promotes DevOps adoption by these teams, where security is integrated from the project's onset. It contributes to the fact that independent specialists in development and security studies work together, facilitating the creation of more secure applications while shortening the time required to launch them[9]. Also, the integration promotes ownership because all the personnel working on the application are responsible for its security.

Likewise, shifting security left enhances security's scalability since testing is considered from the onset. Most business applications are nowadays implemented either on the public and private cloud, local infrastructure, or even a combination of these environments, making it impossible to employ security policies through manual means. With automated security checks and configuration, the organization has an assurance that the implemented security policies will be runnable and consistent in all environments to prevent misconfiguration and security gaps[9]. This scalability is particularly important in cloud-native environments and applications delivered across highly flexible, distributed architectures.

Last but not least, the role of constant supervision or open feedback mechanisms can hardly be overemphasized. With frequent monitoring, it's possible to address security flaws as soon as they are produced, thereby minimizing the chances of glaring vulnerabilities being hammered in production[9]. The feedback loops allow both the developers and the security teams to be aware of the security state of an application and take corrective measures where necessary. This constant feedback enhances the fundamental security of the application and decreases the time and effort needed to manage such issues.

Scope

Of course, moving security to the left is not limited to any particular industry or application. In the financial services industry, with customer records and financial transactions that the firms collect, process and manage for ourselves or our customers, the influence of DevSecOps is crucial to give the tools and practices required to build applications secured from within. Two of the most valuable features are the automatic ability to perform security checks and policy enforcement. This industry is especially vulnerable to even the slightest security risks[10]. In the healthcare industry, where patient information protection is regulated, the shift of security left enables organizations to guarantee compliance with regulation in addition to protecting sensitive application data. Security checks using automation and the tools of continuous monitoring of the applications can keep the healthcare providers HIPAA compliant and enhance the security level of the applications concurrently.

In the retail business, where e-commerce portals are the most constantly attacked by hackers, shifting security enables a preventive approach to protect online payment gateways and customer information[11]. Fault injection security checks mean flaws are tested and fixed before exploitation, thus enhancing the security of e-commerce websites. More specifically, the authorities need security left shifting in the government sector since the security of critical applications is vital for the country's security. Control of security checks and enforcement of security policies in large-scale infrastructure systems ensures that large government agencies can continue to manage the security of their systems without dedicating substantial time to it.

Last but not least, in the telecommunications industry, where managing big cloud infrastructures to deliver services to millions of people is preeminent, shifting security guides the applications with the tools and practice to be secure, integrated, and sound[11]. The effectiveness of security check automation and continuous monitoring amenities allow telecommunications companies to sustain the dependableness of their schemes while at the least providing hale responses to potential threats.

3. Conclusion

Security becomes part of the process by intertwining DevOps with developing applications, which is the best way to achieve the best results in building secure applications from the ground up. Automated security checks, collaboration between the development and security teams and Security as Code changes help minimize vulnerability and maximize security advantages while minimizing the total time to release changes. Security left is a vast move as it improves both security and operation/development functionality, thereby being aprinciple any organization aiming to counter new emerging threats has to undertake. Thus, DevOps' integration with secure development practices is the most rational and efficient way to implement the 'Security Left Shift' concept[11]. The cases and practices described in this paper show the implementation of automating security checks, cooperation between teams and permanent monitoring of applications, as it means that security cannot be a supplement or an extra phase of application development. Still, it has to be a fundamental part of application development. It is a process that will become even more essential for organizations as more of them adopt DevOps practices and want a better stance toward the security of their applications, scalability and ability to resist new threats.

4. References

1. Ur Rahman, A. A., & Williams, L. (2016, April). Security practices in DevOps. In Proceedings of the Symposium and Bootcamp on the Science of Security 2016;109-111.
2. Rajkumar M, Pole AK, Adige VS, Mahanta P. DevOps culture and its impact on cloud delivery and software development. In 2016 International Conference on Advances in computing, communication & automation (ICACCA)(Spring). IEEE 2016;1-6.
3. Perera P, Bandara M, Perera I. Evaluating the impact of DevOps practice in Sri Lankan software development organizations. In 2016 sixteenth international conference on advances in ict for emerging regions (icter). IEEE 2016;281-287.
4. MohanV, Othmane LB. Secdevops: Is it a marketing buzz word?-mapping research on security in devops. In 2016 11th international conference on Availability, reliability and Security (ARES) . IEEE 2016;542-547.
5. Mohamed SI. DevOps shifting software engineering strategy Value based perspective. International Journal of Computer Engineering 2015;17(2):51-57.
6. Lwakatare LE. DevOps adoption and implementation in software development practice: concept, practices, benefits, and challenges 2017.
7. Hamunen J. Challenges in adopting a Devops approach to software development and operations (Master's thesis) 2016.
8. Erich FM, Amrit C, Daneva M. A qualitative study of DevOps usage in practice. Journal of software: Evolution and Process, e1885.2017;29(6).
9. Colavita, F. DevOps movement of enterprise agile breakdown silos, create collaboration, increase quality, and application speed. In Proceedings of 4th International Conference in Software Engineering for Defence Applications: SEDA 2015 Springer International Publishing 2016;203-213.
10. Bou Ghantous G, Gill A. DevOps: Concepts, practices, tools, benefits, and challenges. PACIS2017 2017
11. Battina DS. Best practices for ensuring security in Devops: A case study approach. International Journal of Innovations in Engineering Research and Technology 2017;4(11):38-45.