IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2024

DOI: 10.17148/IJARCCE.2024.13654

A Systematic Review of Security in DevOps: Best Practice and Tools

Komal Chavan[1], Prathamesh Benake[2], Ms Sheetal Bandekar[3]

Department of MCA, K.L.S. Gogte Institute of Technology, Belagavi, Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India[1]

Department of MCA, K.L.S. Gogte Institute of Technology, Belagavi, Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India[2]

Assistant Professor, Department of MCA, K.L.S. Gogte Institute of Technology, Belagavi, Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India[3]

Abstract: DevOps, you know, this area that's getting pretty popular in the tech world. It's all about security with DevOps to make things run smoother. There are tools and methods out there for doing DevOps. We're on a mission here to dig up those top-notch techniques that cover every nook and cranny of DevOps. We've sifted through research papers, analysed their findings, and read tons of reviews to get the lowdown on this topic. Taking all that info into account, we've crafted a detailed review of literature that highlights the effects and most common practices used in DevOps. Our goal is to provide a variety of options for beefing up system security while implementing DevOps. But it's not all rainbows and sunshine – there will be challenges along the way. This paper is your guide to navigating those hurdles and making sure your system is locked down tight against any threats looming out there in cyberspace.

Keywords: DevOps, DevOps Security, Systematic review

I. INTRODUCTION

What is DevOps?
The strategies, technologies, and cultural mindset of DevOps automate and integrate the processes that take place between IT and software development teams. Technology automation, cross-team communication and collaboration, and team empowerment are all prioritized [2].

What is DevOps security?
Combination of three words i.e. development, operations, and security is philosophy of DevOps security. Eliminating any obstacles that can stand in the way of software development and IT operations is the aim. [2]

The approach originated from the idea that, in order to guarantee the success of any business, information systems development and operations should be tightly interwoven activities. [1]

In today's software development processes, DevOps has become increasingly popular. The way that businesses develop and deliver products has been completely transformed by its capacity to optimize workflows and improve coordination between teams working on development and operational projects.

Consequently, organizations take advantage of the smooth merging of operations and development to produce software more quickly and effectively. Nevertheless, security and compliance issues are also brought about by this expedited development process. [6] By automating the software delivery infrastructure, DevOps focuses on teamwork amongst various departments within an organization to accomplish fast software and service distribution to end customers [7].

II. BEST PRACTICES IN DEVOPS

i. Power Combination:
When experts from several domains, such as operators and developers, collaborated, they devised a clever strategy to deal with the complex operations that arise in real life. To make these complex processes easier to understand, they divided them up into manageable portions. For example, they included performance management, infrastructure management, automating deployment, monitoring, handling logs, and adjusting configurations. [9]

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2024
DOI: 10.17148/IJARCCE.2024.13654

ii. Infrastructure automation:
In the past, building infrastructure components from the ground up was a major pain. Everything had to be done from scratch, including manually connecting to data centers, setting up servers, and paying cash for energy bills. Absolutely not enjoyable! I won't even begin to discuss manually connecting devices to networks. However, those drawn-out procedures have essentially become obsolete as a result of the cooperation between developers and operations. [9]

iii. Don't forget to test the hardware:
Security and efficiency of hardware are frequently overlooked. This could lead to erroneous test results or, worse yet, security flaws. You should always validate in accordance with security standards to ensure that your premium software is running properly. [9]

iv. Configuration management:
Back in the day, after the hardware was installed, people had to dig in and manually set up, execute, and adjust a number of programs. Fortunately, DevOps came to the rescue! DevOps uses intelligent automation for configuration management to directly address this issue. Software on hardware may now be easily configured with only a few commands thanks to the magic of DevOps. [9]

v. Monitoring:
Previously, in order to detect any problems, developers and operations personnel had to manually monitor every aspect of the network. I thought it was really annoying and ineffective. But then DevOps entered the picture and altered everything. Everything is now automated to ensure that no one overlooks those important notifications regarding system malfunctions. Ticketing systems, chat features, and monitoring tools are all effortlessly linked with them. And believe me when I say that these instruments are not to be trifled with; when an issue arises, they break through the clutter and sound the alarm. The days of warnings getting mixed up on one system and forgotten are long gone. DevOps systems send out messages faster than you can through a variety of methods.[9]

vi. Log management:
Monitoring every phase of operation is essential as businesses and organizations expand and change. Before an application goes live, development and operations teams can better understand its behaviour thanks to the invaluable assistance of log and event management. Although some may contend that log management necessitates additional stages in the development process, prevention is always preferable to treatment. Organizations can avoid headaches when the product is used in the real world by resolving concerns during the development process. Businesses can facilitate continuous development, minimize needless effort, and guarantee a flawless user experience by adopting observability methods like log management. Now go ahead and confidently explore the world of observability. [9]

III. METHODS TO IMPROVE SECURITY IN DEVOPS

i. Continuous vulnerability scanning and penetration testing:
All DevOps teams should place a high premium on doing continuous vulnerability scanning and penetration testing. Prior to delivering a product, vulnerabilities, misconfigurations, and out-of-date software versions are found by routinely checking codes and dependencies for vulnerability scans. In order to stop attackers from taking advantage of the vulnerabilities found, security teams can mitigate them. Using automated scanning technologies and incorporating them into the DevOps pipeline to conduct regular security checks and offer immediate response is one of the most efficient approaches. [6]

ii. Automating security compliance checks:
Automation of security compliance guarantees compliance with industry standards and legal requirements for DevOps environments. DevOps deployments are more favorable than manual compliance checks because of their dynamic nature. Additionally, manual compliance checks take a lot of time and are prone to mistakes.

Fortunately, DevOps teams may evaluate and certify compliance against predetermined criteria using automated tools and frameworks, which streamline the process. Furthermore, automating compliance tests into the DevOps pipeline helps businesses monitor and assess the compliance state of their systems on a continual basis. Automated compliance checks lower the chance of compliance violations and associated fines by quickly identifying deviations or non-compliance issues. Additionally, automation produces auditable reports and records, making it easier to prove compliance in audits. [6]

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2024
DOI: 10.17148/IJARCCE.2024.13654

iii. Implementing secure Coding:

Using secure coding lessens the chance of creating software system vulnerabilities. Thus, when writing code, developers should abide by industry-accepted best practices and secure coding rules. Input validation, output encoding, and appropriate error handling are examples of secure coding techniques that help stop typical security problems including buffer overflows, injection attacks, and cross-site scripting (XSS). In this sense, companies ought to give developers access to secure coding tools and secure coding training. They should also set up code review procedures that concentrate on security issues in order to identify and address possible security risks early in the development cycle. Additionally, strengthening the DevOps environment security posture is the uniform enforcement of secure coding principles throughout the firm. [6]

iv. Combining DevOps procedures with security:

Using a DevSecOps methodology is necessary to mitigate security concerns in DevOps environments. Security procedures are integrated into the software development lifecycle by DevSecOps. The strategy also places a strong emphasis on coordination and cooperation between the teams responsible for development, operations, and security from the very beginning of the software development life cycle. One essential DevOps practice is DevSecOps. Organizations can conduct security evaluations, threat modeling, and risk assessments alongside development and deployment activities by integrating security into every stage of the DevOps process. Because of this, the automated approach makes it possible to include important security needs like vulnerability scanning, security testing, and secure coding standards. Using such a comprehensive strategy incorporates best practices and security standards at every stage of the development and operations cycle. [6]

v. Developing a security culture within DevOps teams:

Fostering a proactive and security-aware mentality within DevOps teams is facilitated by a strong security culture. More significantly, security barriers that impede time-to-market or produce subpar software are removed when a security culture is deeply embedded throughout the whole product development process. In order to inform developers, operations staff, and other team members about secure coding techniques, security principles, and new threats, businesses should give priority to security training and awareness initiatives. Encourage team members to think about security implications during system deployments, code reviews, and product design by cultivating a security culture. As a result, team members become increasingly watchful in spotting and disclosing possible security flaws, improving the DevOps environment's overall resilience. [6]

IV. CHALLENGES

i. Cloud Security:

Another challenge has emerged as cloud-first architecture has grown in popularity. Because network boundaries in the cloud are less distinct than in an on-premises implementation, there is a significantly greater attack surface. Almost any provided resource can be made available to all kinds of internet traffic with just a few clicks or lines of code. Traditional network safety relied heavily on the premise that networks will be well-defined, with few, well-established ports of entry and departure [11].

ii. Workload containerization:

This also introduces new security variables. There are more attack pathways to be watched over and protected against, though, due to the added complexity of the orchestration, networking, and underlying engine [11].

iii. Collaboration:

These new use cases were simply overlooked in the development of the first security solutions and procedures. Divided security and engineering teams cannot coexist in a DevOps-first culture. Security and engineering are duplicating operational effort and information flow when they operate in separate silos when they could just combine. Unified pipelines are essential for team collaboration and information sharing from a single source. The fraud department frequently receives feeds from the security and infrastructure teams, each of which is operating an entirely different and parallel event pipeline. Similarly, security and application teams frequently operate their own Splunk agents on different boxes within organizations [11].

V. LIMITATIONS

Since we collected the required Internet artifacts using seven search keywords, we are unable to declare that the set of Internet artifacts in our study is complete. We make no claims as to the completeness of the security practices that have been discovered for DevOps security integration. [6] The limited number of companies examined prevents us from making a solid case for the generalizability of our findings.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2024
DOI: 10.17148/IJARCCE.2024.13654

We did not investigate the potential correlation between the nine DevOps organizations of interest's software quality and their use of automation activities. Furthermore, we did not address the possibility that the application of the ten security activities, the five automation activities, or the four security practices was influenced by the degree of cooperation amongst various teams. We leave it as research guidelines for other studies to explore these constraints. [4]

VI. FUTURE DIRECTIONS

i. Trends and New Technologies in DevOps Security and Compliance:

The security and compliance landscape for DevOps is always changing due to new developments in technology and market trends[6]. Adopting artificial intelligence (AI) and machine learning (ML) for threat detection and vulnerability management, utilizing containerization and orchestration platforms for safe and scalable deployments, and incorporating security testing into DevOps pipelines through technologies like DevSecOps are a few major areas of focus[8]. The security and compliance procedures in DevOps environments might be improved by investigating and utilizing this cutting-edge technology[6].

ii. The effect of changing regulations on DevOps techniques:

Since regulatory environments are dynamic, DevOps practices are greatly impacted by changes in the regulatory landscape[6]. For this reason, companies need to be aware of modifications to data protection laws, privacy legislation, and industry-specific compliance standards. Additionally, in order to prevent compliance violations and the related fines, DevOps companies need modify their security controls and processes to comply with new regulations. To maintain compliance, firms should keep an eye on changes in regulations and proactively modify their DevOps procedures.[10]

iii. Opportunities and challenges for cloud-native DevOps environments security:

Securing DevOps environments gets harder as cloud-native designs become more common. The management of security and compliance among many cloud providers, the mitigation of the particular security threats associated with cloud-native apps, and the maintenance of uniform security measures throughout the dynamically scalable infrastructure present challenges. Cloud workload protection platforms and cloud security posture management are two examples of cloud-native security technologies and services that can be leveraged in cloud-native DevOps environments. By seizing these chances and tackling the related obstacles, companies can attain strong security and adherence in their cloud-based DevOps processes. [6]

VII. CONCLUSION

An developing idea in agile SD is DevOps. Agile SD is thought to use DevOps as an automated toolchain. Organizations are displaying a great deal of interest in DevOps, but many are unclear about what DevOps is and how it works, as well as the benefits and problems that it presents. In order to help the informed and lower-risk adoption of DevOps, this article aims to provide such clarity and knowledge. As a result, this work used the SLR method to methodically identify a final group of thirty papers. Subsequently, a thorough analysis of these articles was conducted in order to extract pertinent data and create catalogues of DevOps techniques, concepts, benefits, and obstacles. These catalogues offer a body of collective DevOps knowledge that practitioners and scholars can utilize to deepen their understanding and facilitate the successful implementation of the DevOps methodology in their particular setting.

REFERENCES

[1]. DevOpscritical success factors —Asystematicliterature review- Nasreen Azad∗, Sami Hyrynsalmi https://doi.org/10.1016/j.infsof.2023.107150
[2]. Effective DevSecOps Implementation: A Systematic Literature Review Dhaval Anjaria1; Mugdha Kulkarnihttps://revistageintec.net/old/wp-content/uploads/2022/03/2514.pdf
[3]. A Complete Guide to DevOps Best Practices Justin Onyarin Ogala Department of Computer Science, Faculty of Computing, University of Delta, Agbor, Delta State, Nigeria https://www.researchgate.net/publication/360066207
[4]. Software Security in DevOps: Synthesizing Practitioners, Perceptions and Practices Akond Ashfaque Ur Rahman, Laurie Williams https://dl.acm.org/doi/10.1145/2896941.2896946
[5]. Best Practices for Ensuring Security in DevOps: A Case Study Approach Rajavi Desai and T N Nisha https://iopscience.iop.org/article/10.1088/1742-6596/1964/4/042045/meta
[6]. Compliance And Audit Challenges In Devops: A Security Perspective Sumanth Tatinenihttps://www.irjmets.com/uploadedfiles/paper//issue_10_october_2023/45309/final/fin_irjmets1697523531.pdf

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2024

DOI: 10.17148/IJARCCE.2024.13654

[7]. Security Practices in DevOps https://dl.acm.org/doi/10.1145/2896941.2896946

[8]. Prioritization Based Taxonomy of DevOps Security Challenges Using PROMETHEE Saima Rafi, Wu Yu, Muhammad Azeem Akbar, Ahmed Alsanad, And Abdu Gumaei https://ieeexplore.ieee.org/document/9104690

[9]. DevOps and Its Practices Ravi Teja Yarlagaddahttps://papers.ssrn.com/sol3/papers.cfm?abstract_id=3798877

[10]. The emergence and importance of DevSecOps: Integrating and reviewing security practices within the DevOps pipelineOluwatosin Oluwatimileyin Abiona 1, Oluwatayo Jacob Oladapo 2, Oluwole Temidayo Modupe 3, Oyekunle Claudius Oyeniran 4, Adebunmi Okechukwu Adewusi 5 and Abiola Moshood Komolafe 6. https://www.researchgate.net/publication/379428586_The_emergence_and_importance_of_DevSecOps_Integrating_and_reviewing_security_practices_within_the_DevOps_pipeline

[11]. Best Practices For Ensuring Security In Devops: A Case Study Approach Dhaya Sindhu Battina https://www.researchgate.net/publication/357033114_BEST_PRACTICES_FOR_ENSURING_SECURITY_IN_DEVOPS_A_CASE_STUDY_APPROACH

[12]. DevOps in practice: A multiple case study of five companies Lucy Ellen Lwakatare a,Terhi Kilamo b, Teemu Karvonena, Tanja Sauvolaa, Ville Heikkiläc, Juha Itkonenc, Pasi Kuvajaa, Tommi Mikkonend, Markku Oivoa, Casper Lassenius https://www.researchgate.net/publication/334007673_DevOps_in_Practice_A_Multiple_Case_study_of_Five_Companies