Best Practices for Ensuring Security in DevOps: A Case Study Approach

Rajavi Desai and T N Nisha*
Symbiosis Centre for Information Technology,
Symbiosis International (Deemed University), Pune, Maharashtra, India.
Email: nisha@scit.edu*

Abstract. DevSecOps is an advancing area that is highly preferred in the field of Information technology. The main idea of DevSecOps revolves around how security could be integrated with DevOps. There exist certain tools and practices for implementing DevSecOps, but this area is not fully explored. The research serves to find out the best practices that could be implemented to enhance security in DevOps. The main objective is to provide best practices covering all the aspects of DevOps. The solutions provided in the research papers that were referred to, the results of these research papers, and reviews were considered for the study. From the data gathered a descriptive review of literature is formed and overviewed to develop all the possible effects and most widely used practices to implement DevSecOps the challenges that could be faced during DevSecOps implementation.
This paper aims to provide a range of options that could be implemented for enhancing the security of the system.

Keywords: DevOps, DevSecOps, security, Case study, SDLC.

1. Introduction
The recent application development process involves deploying advanced tools and complex computing services [1], which make the SDLC very agile and dynamic. The concept of DevOps was brought into practice to keep up the speed and agility. DevOps consists of various tools and services to enhance application development by reducing complexity and lowering the failure rates of new releases. It is a continuous process that works for all the phases of the application lifecycle, from planning to development to continuous testing and continuous deployment [2].

A survey result states that still, 43% of the IT professionals do not have the right tools to implement security throughout the SDLC lifecycle. Their focus is on meeting development goals over security. (Mike Vizard, 2019)[3]. Security becomes a challenge because of the high scalability and agility of these new
processes. Despite the challenges, it becomes necessary that security is not neglected. It can affect the organization's functioning and customer base and can cause financial or reputational damage or legal consequences. DevOps compliance and overall IT security are required to be implemented throughout the lifecycle of application development. When security is integrated with DevOps, it is called DevSecOps[4]. It includes focusing on development, IT operations, and security to have a seamless software delivery process.

The process of careful planning, designing, developing, deployment, and maintenance aligned with security at all the SDLC stages will avoid the rework caused by security if security is compromised anywhere [5]. A unanimous toolchain of technologies supporting coagulated automation could be implemented to achieve the systems' higher efficiency and maintain system standards. To maintain the reputation and high-quality product or service delivery for the clients, the company can clue out the weakness and the flaws in their software way in advance to avoid a loss of any kind or demean the company's value [6].

DevSecOps edges on the alignment of security throughout the business process to attain requisite outcomes. The alignment could be made possible by maintaining compliance, assessing vulnerabilities, and implementing controls [7]. It makes security practices repeatable, consistent, and embedded in development from start to finish. The implementation of DevSecOps should be transparent, aligned, high-performing and have quick releases for the development team to interact in real-time and automate the process of common responses to conditions, threats and security breaches, some quintessential system which are the backbone of the security system are necessary to be developed or incorporated without which the process of risk assessment or monitoring can get tedious. Furthermore, the security team should not stop there. Some new platform security and end-to-end protection should also be included to enhance the software/software development process [8].

The next section consists of a narrative literature review. A literature review is conducted to understand the implication and overall background of DevSecOps, including its importance, implementation techniques, and challenges [9].

2. Literature Review
Implementing DevOps practices alone is not sufficient; secure DevOps should be practiced so that security is maintained. DevOps process is practiced at a very large scale in the industry, which could be used by academics, healthcare, or government [10]. These entities would comply with general protection and security laws as mandated, which makes them highly regulated. Even though the environment is highly regulated (HRE), the SDLC of these HRE applications must be secure. There will always be innovations and alterations which will indirectly affect the DevOps process. Thus, a secure DevOps process must be implemented suitably by taking feedback to suit the HRE [11].

Collaboration is a major challenge while dealing with the implementation of DevOps. To carry out DevOps practices, all the teams need to collaborate and on a single shared system [12]. Incorrect configurations or errors occurring knowingly or unknowingly can directly impact the system's functioning, affecting the operating of all the teams involved. It is also mentioned that the toolsets required for DevOps need proper training and attitude of the employees. The research was done on DevSecOps realities and opportunities for which 350 IT experiences at different enterprises aware of DevOps were interviewed. It showed that around half of them were failing to incorporate a secure CI/CD lifecycle. Few of the rest who were aware of DevSecOps faced difficulties in selecting the right approach, tools, and processes for implementing DevSecOps and have not implemented security at the initial stages [13].

Traditional security has always been about barring people from disclosing crucial information because of the security policies that were in place. However, DevSecOps is about promoting inclusion and collaborating people from the teams to work together. DevSecOps deals with implementing security from
the initial stages of the application development lifecycle to avoid costly mistakes that could be encountered by implementing security afterthought. The emphasis should be on integrating development teams and security teams so that the controls could be implemented throughout the lifecycle [14].

Even though DevOps encourages faster delivery, defect resolutions and creates an overall collaborative environment, it fails to include security. Thus, it is important to implement DevSecOps. DevSecOps outlines a set of guidelines to be implemented through tools, technology, and processes so the whole process could be made secure and more reliable. He suggests restructuring the process is using various automation software such as Jenkins. It would test all the releases during the time of integration to avoid piling defects, which may later act as vulnerability. It aims to modify the system, so that faster response time to vulnerabilities found could be achieved.

Maintaining a secure environment and work culture is important for ensuring security. Security culture can be created by providing proper training and adequate knowledge of the tools and technologies deployed. Awareness will reduce the possibility of errors occurring by deploying codes that are not suitable for certain systems and vulnerabilities could be tracked and avoided [15].

A system for implementing DevSecOps could be developed using open-source tools, which could be done by configuring various tools 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. Static and dynamic security checks 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 to monitor and analyze threats and thereby inform the programmer about any. In this way, by following certain critical and specific steps, the speed and security can be maintained hand in hand.

The popularity of DevOps is increasing because of its agile nature and speed. DevSecOps meet the need for security in DevOps. DevSecOps attempts to create and include modern security practices that can be included in DevOps' fast and agile world. It includes embedding security from the very beginning by following the most suitable way, including integrating tools with systems and development teams with security teams and maintaining compliance. Involving security experts from the beginning of the development cycle/process makes it easier to plan and lay out an agenda regarding integrating the security controls throughout the development process to avoid delays or create new issues by implementing security controls after systems are already up running. So, shifting the security to the left will increase the system's overall efficiency and reduce downtime.

3. Proposed Method
For the literature review, IEEE Xplore, Elsevier, and ACM digital library databases are referred. The keywords DevOps and DevSecOps were first searched individually, and then the following key phrases were searched: 1. DevOps and security 2. DevOps and limitations 3 and DevSecOps implementation.

All the research papers referred directly or indirectly answered at least one of the following questions:
1. What are the main aspects/components of DevOps where security needs to be integrated?
2. What are the limitations of DevOps about security?
3. What are the basic requirements for implementing DevSecOps?
4. How could security be implemented without affecting the agility of the system?

The solutions provided in the research papers that were referred to, the results of these research papers, and reviews were considered for the study. From the data gathered a descriptive review of literature is formed and overviewed to develop all the possible effects and most widely used practices to implement DevSecOps the challenges that could be faced during DevSecOps implementation.

This research aims to find out the best practices that could be followed to ensure security in DevOps, which is done by finding out possible ways in which DevSecOps implementation could be performed and
the challenges encountered during implementation. The research is carried out in two parts. First, data collection is performed, followed by data analysis.

3.1. Data collection

Data collection is performed by following the case study approach. A case study of three organizations that have implemented DevSecOps is done. The purpose of undertaking a case study is to get an in-depth understanding and insights into how the implementation of DevSecOps is practically performed.

For the case study, three companies are chosen. All the three companies selected are information technology-based, and they follow DevOps practices. They have also implemented DevSecOps at different levels and are following various practices for the same. The companies selected provide different services and have different areas of expertise so that different aspects could be taken into consideration. The descriptions of the companies selected are mentioned below. As the actual names of the companies selected are not to be disclosed, Company1, Comapany2, and Company3 are used instead.

Company 1 is an IT company that focuses on providing mobile application development and other specification such as chatbots and UI/UX for android as well as iOS. The company is medium-sized and has 150+ employees. The company is based in India.

Company2 provides solutions related to supply chain management, including warehouse and inventory management, supply chain tracking, vendor, and supplier management. The company has around 300 employees and has a large client base that includes global clients as well.

Company3 customizes software and web applications based on client requirements. They specialize in providing applications for financial management, banking, and insurance. Over 400 employees working on more than 25 big and small projects provide either new software/services or re-engineering the old technology.

Interviews were conducted for data collection. Nine individuals from the chosen companies who are in various positions were interviewed. The names are not mentioned because of confidentiality reasons. All the interviewed personnel knew DevSecOps or were directly involved in the implementation process. The interviews conducted were either on call or at the location. All the interviews were conducted in the English language. The initial part of the interviews aimed at gathering basic and background information about the company. The points such as the number of employees, DevOps practices in place, and the number of security teams working were covered.

The second part of the interviews was more specific towards the implementation of DevSecOps and covered three main questions. Table 1 shows the interview details.

1. Why did the company implement DevSecOps?
2. How did the company implement DevSecOps?
3. What were the challenges faced by the company during DevSecOps implementation?

| Companies | Mode of the interview | Duration | No. Of personnel interviewed | Position |
|-----------|-----------------------|----------|------------------------------|----------|
| Company1  | On location           | 00:23:00 | 2                            | Manager  |
|           |                       | 00:55:00 |                              | Security team lead |
| Company2  | On-call               | 00:47:00 | 4                            | Technical team lead |
|           |                       | 00:20:00 |                              | Security team member |
|           |                       | 00:32:00 |                              | Security team member |
|           |                       | 00:14:00 |                              | Server manager |
| Company3  | On-call               | 00:33:00 | 3                            | Technical head |
3.2. Data analysis

Descriptive data analysis is performed by combining the results of the literature review and a case study. The common challenges and the best practices are taken into consideration. By penning down the Literature review and the interviews and analyzing both, we will draw up the best practices and their actual implementation that various organizations are following. Interview results are mentioned in summarized form, and sensitive details are avoided.

4. Interview Results

The initial questions of the interviews covering the background details of the companies are summarized in Table 2.

| Companies   | No. Of Employees | No. of security teams | Main Products/ Services            |
|-------------|------------------|------------------------|-----------------------------------|
| Company1    | 150+             | 1 (8 members)          | Mobile application development    |
| Company2    | 300+             | 4                      | Supply chain management solutions  |
| Company3    | 400+             | 0 (Outsourced security processes) | Customized web applications |

All three companies have training programs planned. The developers are aware of secure coding, and it is ensured that secure coding is performed. All three companies have taken care of compliance and implemented controls accordingly.

The main focus is on the questions involving the implementation of DevSecOps. The interview discussion is summarized case-wise.

Case 1:

Why did Company1 implement DevSecOps? As the company started expanding its business, its customer base also increased. To keep with expansion and development, Company1 shifted to DevOps. A new architecture to support DevOps was developed. All the data was updated on the cloud, so it becomes important that the data is secured. The company implemented security and DevOps from the beginning to avoid any kind of interference that may be caused by neglecting security and protecting build upon

how did the Company1 implement DevSecOps? Company2 is at the very initial stage of DevSecOps implementation. New security experts were hired, and the recommendations were adopted to make the overall process secure. A new governance framework was followed so that compliance with PCI DSS is maintained to a greater extent. The employees were made accustomed to the new standards and procedures that were followed. They were given training on how to enhance the development methods and techniques. It includes deploying inbuilt security provisions that different programming languages support and developing secure codes from the beginning. The company also shifted from hybrid cloud to private cloud.

What were the challenges faced by Company1 during DevSecOps implementation? The company initially used traditional methodologies for software development. The tools and techniques used were not
very advanced. Completely discarding these tools and switching to more advanced ones was not a cost-effective solution.

**Case 2:**
Why did Company2 implement DevSecOps? Company2 has been using fully functional DevOps methodologies with automation up to an extreme extent. The system faced an unexpected failure that affected the CI/CD pipeline building phase, which caused a downtime of almost 3.5 days. As supply chain tracking systems need continuous updating, a temporary shift was required from automated to manual updating of the system's data.

How did Company2 implement DevSecOps? Company2 has been using various customized systems for data analysis to predict stock refilling, raw material usage, etc., as per client requirements. The company decided to use the same prediction concept by monitoring and analyzing system behavior to find anomalies or suspected behavior. Company2 has used a similar technique for securing virtual data servers by monitoring and analyzing data traffic. An extra secure layer was added, which would act as an extended server.

What were the challenges faced by Company2 during DevSecOps implementation? Company2's main focus has only been agility. The shift of focus was to be made by collaborating security teams with development teams from the start, which had a major effect on the workflow, and getting accustomed to which was difficult for the employees.

**Case 3:**
Why did Company3 implement DevSecOps? Company3 had implemented DevOps a long time back, and the databases and the resources used were shared amongst all the teams. All the employees authorized had access to the data. An incident happened when a part of the code of a financial project was altered. There was no way to know whether it was done knowingly or unknowingly as there was no questioning. Due to this, rework had to be done, and the client faced difficulty due to the unavailability of the project's essential functionality.

How did Company3 implement DevSecOps? The company had already deployed security tools for a web application to assess common vulnerabilities such as XSS scripting and SQL injection. Privilege Management System and Change Management System were implemented, which would ensure continuous monitoring and control of user privileges; no change would go unnoticed and incorrect changes could be reverted without interfering with other functionalities of the system. These systems were integrated with the originals by using customized connectors which would prevent interference.

What were the challenges faced by Company3 during DevSecOps implementation? On implementing new systems, the working of the whole CI/CD pipeline was refined. The newly implemented systems: Privilege Management System and Change Management System, had to be incorporated with the original automation systems without affecting their efficiency.

5. **Results and Discussions**
Implementing DevSecOps requires an understanding of DevOps and its practices. It focuses on integrating security at all stages, including requirement gathering, development, testing, deployment, monitoring, and maintenance. Based on the case study and literature review, three important attributes are identified: Security could be ensured in DevOps. The attributes are training programs, integration of security tools/systems, and preventive measures.

Table 3 summarizes the ways and the best practices that the chosen companies are practicing.
Table 3 Company-wise DevSecOps implementation methods

| Companies | Training programs | Security tools | Security systems | Secure coding practices | Preventive controls and compliance |
|-----------|-------------------|----------------|------------------|-------------------------|-----------------------------------|
| Company1  | Yes               | No             | No               | Yes                     | Yes (PCI DSS)                     |
| Company2  | Yes               | No             | Yes (Configuration management system) | Yes | Yes |
| Company3  | Yes               | Yes            | Yes (Privileged access management and change management system) | Yes | Yes (ISO 9001: 2015 and ISO: 27001: 2013) |

One of the most common challenges that any organization may face while implementing DevSecOps is speed. DevOps' differentiating factor is speed. The speed of the process should not be compromised while integrating it with security. Security processes must match the pace of development processes, which could be managed by practicing security from the initial stages of SDLC. As this requires continuous monitoring and scanning, security processes such as scanning for vulnerabilities or traffic monitoring could be automated, which will help security keep up with the speed and agility of the DevOps process, which will ensure that the changes are made per the standards and guidelines that are in place. By including continuous monitoring or timely security checks throughout the CI/CD pipeline, engineers could be given quick alerts in a failed test case. Thus, automated security testing could save time and improve efficiency.

It becomes very important that the gap existing between development and security is addressed. While dealing with DevOps, maintaining agility is the top priority, so it would be beneficial if the security issues are taken care of throughout the SDLC lifecycle starting from the requirement gathering stage, which will not only improve the efficiency of the process but also reduce the cost that would be incurred with the fixing of security issues on later stages when the applications have become more mature.

All the aspects of DevOps need to be made secure to establish secure DevOps. The first and the most important one is the Agile framework. It provides a software development process that is iterative, focuses on collaboration, and supports self-organizing, cross-functional teams to deliver quality software in less time. Another significant aspect is automation. DevOps is marked by implementing automation throughout the SDLC lifecycle, including automated monitoring, testing, and delivery/deployment pipeline.

Both of these aspects could be made more secure by using various security tools available. Depending upon the importance, the system's functionalities can be prioritized, and accordingly, the tools could be selected. These are the important functionalities that should be reviewed while selecting the tools.

1. Whether the tool is supporting multiple programming languages?
2. Whether the system’s agility is compromised on integration with the tool?
3. Whether the tool can identify all the bugs/vulnerabilities in the system?
4. Whether the tool can identify these bugs/vulnerabilities at an early stage?
5. Whether the tool can fix these bugs/vulnerabilities?
6. Whether the tool can provide inception and monitoring features throughout the lifecycle?
7. Whether the tool can deal with false positives or noise?
Here are few tools that could be used: Microsoft Secure DevOps Kit For Azure, Aqua security, logz.io security analytics, SonarQube, etc.

Another area that requires security implementation in DevOps is IT infrastructure. In order to implement DevOps, a well-capable infrastructure with planned workflows is required. As it forms the DevOps base, various security controls and guidelines are needed to be implemented. Infrastructure as code is another aspect popular in DevOps. It includes managing all the infrastructure components like networks, topologies, and virtual machines using configuration files rather than using hardware configurations, which brings another aspect: configuration management. Maintaining secure configuration and configuration changes holds great significance. The changes made/ codes added should be secure and done only by privileged users.

Innovation and alterations would always be there when technology is used. These alterations could have a negative impact, and the need to revert the change might arise. This process becomes seamless if there is a record maintained for the changes made, by whom, and all the components affected, which ensures that any unsecured changes are avoided or recorded if already made. For this, a change in the management system could be integrated with the original system.

Another threat to security is malicious insiders. The DevOps process would yield a large amount of data shared amongst all the departments. There are multiple accounts with administrator privileges involved in taking care of the privileges given to the employees/ other user accounts. For this, automated privilege access management systems could be incorporated so that all the user activities, including that of the admin, could be continuously monitored and recorded. These records/logs could also be used as evidence for audit trials.

Right tools and systems could be chosen by prioritizing core tasks, but their integration is not always straightforward. Workflows of the traditional business processes could be refined to increase the effectiveness of the add-ons. However, the main challenge comes during the integration of the tools/ systems in the original environment. Integration of tools and security systems is a crucial but not an effortless task. Such change management, configuration management, and privilege access management systems could be integrated using a connectivity stack. Connectors suitable for different systems are either already available or customized, ensuring seamless integration without compromising the original system’s speed and agility.

Another definite challenge that comes with implementation is collaboration. It becomes important to acclimatize a working environment in which the security team compliments the development team throughout the lifecycle. Dependencies between the teams should be shaped in such a way that their efficiency is not compromised. Proper training programs should be conducted to overcome this challenge. While dealing with complex DevOps processes, all the users directly or indirectly associated with the system should have proper training and knowledge about its working. They should be trained on using the technology/ toolsets used so that chances of human errors reduce and any anonymity could be reported. Proper knowledge about the guidelines to be followed should be conveyed to maintain security at the individual level. It may include company-made rules such as password policies or agreements such as non-disclosure agreements. Proper training and evaluation can help developers write clean code.

Apart from these, there are general preventive measures that could be taken while implementing DevSecOps.

- Maintaining general compliance and protection.

In the system and the organization, using the system should be following the standards and compliance. Compliance monitoring should be performed at all the stages of the software lifecycle, which will make the environment highly regulated as the suitable controls would already be in place.
• Secure coding practices.
Code delivery should be done in small fragments, which will be beneficial because quality checks would be done continuously. Any alterations could be made easily. Only the particular fragments related are required to be checked in case of errors, which will ensure error-free CI/CD. Codes should be made more secure by continuous analyses and deployment of inbuilt security functionalities provided by different programming languages so that vulnerabilities and loopholes would not go unnoticed. It also involves establishing and adhering to coding standards. Not practicing secure coding may cause many software security risks, such as a breach of an organization's confidential information.

• Continuous evaluation and feedback. impart
For every new method followed or technology used, continuous threat modeling and risk assessment should be done. Employees should be asked to give regular feedback about the system's working so that any unusual changes could be looked after at the initial stages.

6. Conclusion
The most common challenges that any organization could encounter while implementing DevSecOps can be categorized into these three areas: speed, collaboration, and integration. The most common and effective best practices followed can be covered by considering four main indicates. The first is training. Accurate training programs can push the extent to which security is protected. Proper training programs should cover the importance of security and individual contributions that are expected to maintain security. The knowledge of the information systems driving the organization should be imparted. As the indifferences caused between the teams can directly affect efficiency, it becomes important that everyone is familiar with the working environment. The second is the deployment of security tools or systems. Once integration with the original systems is done, continuous monitoring and fixing would be performed throughout the development process. The last is taking preventive measures, which involves securing all the components like codes, databases, networks, and physical environment, which could be done by placing various controls, following secure coding, and filtering the traffic, which will reduce the chances of security being compromised. These challenges could prevail over, and security could be preserved by practicing these best practices or their combinations.

References
[1]. Akond, A., & Laurie, W. (2016). Software Security in DevOps: Synthesizing Practitioners’ Perceptions and Practices. 2016 IEEE/ACM International Workshop on Continuous Software Evolution and Delivery (CSED) (pp. 70-76). Austin, TX, USA: IEEE.
[2]. BRADLEY, T. (2018). 451 Research Report Highlights Value of DevSecOps for the CI/CD Pipeline.
[3]. Bruneo, D., Fritz, T., Keidar-Barner, S., Leitner, P., Longo, F., Marquezan, C., et al. (2014). CloudWave: Where adaptive cloud management meets DevOps. 2014 IEEE Symposium on Computers and Communications (ISCC) (pp. 1-6). Funchal, Portugal: IEEE.
[4]. Carter, K. (2017). Francois Raynaud on DevSecOps. IEEE Software, 34 (5), 93 - 96.
[5]. Gomes, K. (2018). The Importance of DevSecOps. DeKalb, Illinois.
[6]. Hamunen, J. (2016). Challenges in adopting a Devops approach to software development and operations.
[7]. Håvard, M., & Ricardo, C.-P. (2017). DevSecOps: A Multivocal Literature Review. International Conference on Software Process Improvement and Capability Determination.770, pp. 17-29. Springer, Cham.
[8]. James, R. (2013). Adopting DevOps practices in quality assurance. *Communications of the ACM, 56* (11).

[9]. Lee, J. S. (2018). The DevSecOps and Agency Theory. *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 243-244.

[10]. Longstaff, T., & Yasar, H. (2019). *Build Secure Application with DevSecOps!*

[11]. Luís, P., João, F., Miguel, S., & Rúben, P. (2019). DevSecOps Metrics. *EuroSymposium on Systems Analysis and Design*. 359, pp. 77-90. Springer, Cham.

[12]. Nora, T., Jingyue, L., & Huang, H. (2019). An Empirical Study on Culture, Automation, Measurement, and Sharing of DevSecOps. *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. Oxford, United Kingdom, United Kingdom: IEEE.

[13]. Rahul, B. S., Prajwal, K., & Manu, M. N. (2019). Implementation of DevSecOps using Open-Source tools. *International Journal of Advance Research, Ideas, and Innovations in Technology, 5* (3).

[14]. Thomas, F. D., Christina, P., & André, v. H. (2018). Exploiting DevOps Practices for Dependable and Secure Continuous Delivery Pipelines. *2018 IEEE/ACM 4th International Workshop on Rapid Continuous Software Engineering (RCoSE)*, (pp. 27-30). Gothenburg, Sweden, Sweden.

[15]. Ramya, R. S., Nanda Kishore, D. Sejal, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik. "Predicting Social Emotions based on Textual Relevance for News Documents." In 2019 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), pp. 1-4. IEEE, 2019.