FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 10 No. 18, September, 2026, pp 50 – 57
DOI: https://doi.org/10.33003/fjs-2026-1018-5733

DEVSECOPS: A Systematic Literature Review on Security Integration in Development and Operations

Kuye Oluwatobi, *Ridwan Kolapo, Temitope Olufunmi Atoyebi, Eru Akwuma Nathaniel and Solomon Ahiaba

Department of Information Technology, Faculty of Computing, Nile University of Nigeria, FCT Abuja, Nigeria.

*Corresponding authors' email: kolapo.olayinka@nileuniversity.edu.ng

Received: 12 Aug. 2026
Accepted: 11 Sep. 2026
Published: 18 Sep. 2026

ABSTRACT
In a world in which organizations are continually trying to deliver their software as quickly as possible. Keeping security strong throughout the development process is becoming a pressing challenge. Usually, security is put in at the end of the development process, typically resulting in high and extensive cost in order to rectify security loopholes in the software. DevSecOps changes that by weaving security into every level of the DevOps workflows and developing closer collaborations between the developers, operations, and security teams. This translates to software that is designed and developed security-first. This paper is a systematic literature review to examine the integration of security in the DevOps procedures. It explores the demands for successful DevSecOps implementations and the hurdles that institutions face when adopting it. Automated security testing, infrastructure as code, threat modeling and container security are used in DevSecOps practices to ensure the security of every level of the devops operations, as evidenced on an in-depth analysis of 23 scientific peer-reviewed articles published between 2020 and 2026. However, the review also uncovered some persistent challenges so as some of the solutions to these problems. Lack of existing literature is a gap with calls to develop more standardized frameworks and tools that can be used in the DevSecOps implementations to make them more readily adopted by organizations. Through investigating this area of growing interest in DevOps, it becomes possible to contribute to the knowledge of incorporating security more proactively into DevOps channels with the aim of creating more secure software systems.

Keywords: DevSecOps, Security Integration, Continuous Integration and Continuous Deployment (CI/CD), Shift-Left Security, Threat Modeling

INTRODUCTION

Today, in the world of technology, working teams constantly struggle. They must deliver software at hyper speeds, yet don't have the luxury of cutting corners on security. Traditional development models basically never included security in its development, and even when they did, it was typically done in the last stages, before the software went to market. This meant that a variety of vulnerabilities were likely to remain difficult and costly to patch, especially for the late fixes (Abiona et al., 2024). DevOps was launched in 2009 by Patrick Debois to change the industry by breaking down the barrier between Dev and Ops. Therefore, the software industry's software development capacities grew at an unprecedented rate (Prates & Pereira, 2025). But as teams have rushed to get their products to market with the help of modern continuous integration and continuous deployment (CI/CD) pipelines, security became a show stopper and there were obvious flaws in standard DevOps practices (Prates & Pereira, 2025). Because of the speed with which the work was being undertaken, it was easy to miss security shortcomings (Prates & Pereira, 2025). To address this friction, DevSecOps was created as a way to resolve this dilemma between the rapid development pace and security. More specifically, it aims to infuse security elements into development process from the very beginning of the development cycle. Security is not an inspection at the end of the day, but the fundamental principle of this is shift-left security. That is moving testing, threat modelling, automated compliance checks, and continuous monitoring to the earliest stages of the software development life cycle (Abiona et al., 2024; Prates & Pereira, 2025). It isn't about slowing developers down. Automating these safety nets in the CI/CD process means that teams can ensure a high level of security without compromising delivery speed, making the release of secure code a seamless, automatic routine. Apparently, making this shift is easier said than done. Organizations diving into DevSecOps often hit real-world friction from ingrained culture clashes and complex tool integrations to a shortage of hands-on expertise in security automation (Parashar et al., 2020).

Organizations also struggle to understand the cost-benefit relationship of DevSecOps implementations (Anjaria & Kulkarni, 2021). Nevertheless, it remains crucial for mitigating cyber threats, meeting regulatory requirements, and safeguarding application integrity (Abiona et al., 2024). This study conducts a Systematic Literature Review (SLR) to explore key practices, challenges, and best strategies for effective security integration within DevOps environments.

Development and Operations (DevOps)

It is hard to discuss DevSecOps without mentioning DevOps. DevOps is an acronym that combines the terms "development" and "operations." In the production cycle, DevOps describes how to expedite the processes from development to deployment (Parashar et al., 2020). DevOps was initially birthed in 2009 by Patrick Debois, and since the term's inception (Prates & Pereira, 2025), the industry has not been able to define what DevOps is, since one cannot distinguish it from continuous practices, which are many and so are overlapping (Rajapakse et al., 2022). Whether DevOps constitutes a specialised career requiring both development and IT operations expertise has been a central point of contention (Rajapakse et al., 2022). Others associate DevOps

with cultural elements such as collaboration, automation, measurement, and sharing (CAMS) in addition to technical expertise. DevOps went beyond the advancement of improving software delivery and evolved into a methodology that combines practical engineering techniques, smart technical tools and a deliberate cultural change. At its core, DevOps bridges the traditional divide between the developers building the software and the operations teams keeping it running in production (Prates & Pereira, 2025). By blending agile practices with closer day-to-day collaboration, it streamlines the whole process allowing teams to release updates far quicker, shrink the gap between dev and ops, and avoid the usual chaos that comes with deployment day (Azeem et al., 2022).

The definition describes a set of engineering practices informed by culture, powered by technology. Their purpose is to ease the process of delivering software by tearing down the silos and building shared ownership, trust, and a culture of continuous improvement (Prates & Pereira, 2025). Continuous integration, monitoring, testing and continuous delivery, are the most commonly used DevOps techniques (Prates & Pereira, 2025; Rajapakse et al., 2022).

Development, Security and Operations (DevSecOps)

In the communal DevOps paradigm, security is a shared responsibility rather than a distinct phase. In order to highlight the necessity of incorporating a security foundation into DevOps projects, some coined the phrase "DevSecOps." Security concerns increased as a result of large-scale data breaches and leaks, and other types of cyber incidents, forcing organizations to pay closer attention to security threats. In this case, there is the requirement for high levels of security integration into DevOps. Thus, the concept of DevSecOps was born in 2012 (Sánchez-Gordón & Colomo-Palacios, 2020) and security has to become an integral part of DevOps. Based on Anjaria and Kulkarni (2021), DevSecOps is an extension of DevOps that aims to combine security with DevOps practices. Another source defines DevSecOps as the security aspect of devops, suggesting the inclusion of security methods to the SDLC from inception (Prates & Pereira, 2025). According to Sánchez-Gordón and Colomo-Palacios (2020), DevSecOps is viewed as a necessary evolution of DevOps that seeks to foster cooperation between security, development, and operations teams in order to proactively incorporate security safeguards and procedures into the DevOps software development cycle. Similar to DevOps, previous studies on DevSecOps found that cultural elements C.A.M.S are crucial considerations (Sánchez-Gordón & Colomo-Palacios, 2020).

DevSecOps is an overview of security practices and principles into a devops model by establishing a higher amount of communication between the security, developer and operations people (Rajapakse et al., 2022). Variations of this phrase, such as SecDevOps and DevOpsSec, have been adopted by researchers and industry professionals (Casola et al., 2024; Rajapakse et al., 2022; Sánchez-Gordón & Colomo-Palacios, 2020). Nonetheless, the primary objective of each of these phrases remains the same: keeping security a top priority throughout the DevOps cycle.

It is an approach that weaves security into devops practices to ensure that security is not anymore, an afterthought, but a defining part of the SDLC (Akbar et al., 2022). It seeks to "shift security left" by embedding security practices at the beginning and throughout the DevOps pipeline (Sandu, 2024; Vangala, 2025). By shifting security left, DevSecOps enables early detection and mitigation of threats and vulnerabilities, shrinking the occurrence of security breaches and optimizing the overall security posture of software applications (Makani, 2024). Security integration in DevOps, commonly referred to as DevSecOps, has become an essential response to the growing need for resilient, secure software delivery. However, a number of issues remain in incorporating security procedures into contemporary software engineering. First, traditional security techniques are inapplicable due to their inability to match DevOps' agility and speed. Second, only limited research has been conducted on DevSecOps, so much about it remains poorly understood. One major issue that keeps software practitioners from incorporating security into their devops processes, is uncertainty about when and how to employ existing tools in automation (Rangnau et al., 2020).

Automated security testing, policy enforcement and toolchain integration are core practices for integrating security. For example, Putra and Siber (2022) have shown how to leverage Static Application Security Testing (SAST) with Dynamic Application Security Testing (DAST) in the CI/CD pipelines to support vulnerability detection. Likewise, Rangnau et al. (2020) did a case study which demonstrated that DAST tools like OWASP ZAP can be directly infused into pipelines, providing a continuous security test with real-time feedback to the developer. Many papers higlights just how critical the right tools and automation platforms are integrating devsecops. Popular tools such as SonarQube and Snyk sparked a detailed study by Makani (2024) which indicated that the automated scanner tools can save developers a lot of time, but they also have drawbacks: high false-positive rates when used by developers can quickly exhaust and frustrate them. Furthermore, it is not just about acquiring tools, as noted by Parashar (2020), real tangible security gains can only be realized when the organization is committed and continually training its staffs.

It is not only a technical, but also a cultural integration of security (Sánchez-Gordón & Colomo-Palacios, 2020). It is not enough to just align development, ops and security, it's the cornerstone of the entire approach, as Ashenden and Ollis (2020) and Lumpatki et al. (2022) note. Even the best tools typically are not used to their full potential or even adopted by most organizations if they don't have a shared attitude toward security. Athmakuri (2024) and Chittibala (2023) have found that a phased approach of implementing security controls in an organization over the entire SDLC provides a strong security posture without significantly slowing down the delivery speed. Technical, procedural and organizational are the three key dimensions of integration identified by research in Anjaria and Kulkarni (2021) and Sánchez-Gordón and Colomo-Palacios (2020) from a more general perspective. The real bridges between security and agile development involve practices such as continuous feedback loops, frequent developer training sessions, a definition of secure coding practices, and more. All of which are difficult to introduce to DevOps, as their reviews point out. It's a multination effort that is just as much a team alignment and continuous culture improvement effort as it is an automation and tooling effort.

Evolution of DevSecOps

Modern speed, and agile development through DevOps just weren't fast enough with older, more manual, security procedures. Therefore, devsecops was inevitable, it's really just built from the ground up, with security becoming a pro-active, rather than a re-active part of the development lifecycle, thanks to DevSecOps (Abiona et al., 2024). The development of devsecops dates back to initial efforts to merge security with CI/CD pipelines via security-focused DevOps tools and approaches. Along the journey of the

devsecops life cycle came key moments in published papers, guidelines, and security-specific software that transformed the idea of DevSecOps from concept to reality. Today the adoption rate is surging over industries, as organizations due to rapid change of cloud environments and advanced cybersecurity threats as well as evolving standards realize to do better than have the 'security first' approach at the tail end of the cycle, it has become essential and many organizations are in fact going forth to do exactly that. DevSecOps is also steadily accelerating its adoption across a large array of industries (Abiona et al., 2024).

There is a need for a balance between technical aspects of DevSecOps and the culture shift, which is one of the most dynamic areas of research and practice in software engineering.

Table 1: Summary of Reviewed Literature

Article	Author(s)	Year	Method Used	Problem Addressed	Summary of Results	Limitations	Ref.
Putting the Sec in DevSecOps: Using Social Practice Theory to Improve Secure Software Development	Ashenden, Debi Ollis, Gail	2020	Social Practice Theory	Behavioural factors in DevSecOps adoption	Cultural practice greatly influences security integration	Limited empirical validation	(Ashenden & Ollis, 2020)
Security as Culture: A Systematic Literature Review of DevSecOps	Sánchez-Gordón, Mary Colomo-Palacios, Ricardo	2020	Systematic Literature Review (SLR)	Undefined DevSecOps cultural and conceptual foundations	Identified culture as a key pillar of DevSecOps adoption; explored cultural aspects influencing adoption; highlighted the importance of security as a shared responsibility	Theoretical findings without field validation	(Sánchez-Gordón & Colomo-Palacios, 2020)
Challenges and Solutions when Adopting DevSecOps: A Systematic Review	Rajapakse, Roshan N. Zahedi, Mansooreh Babar, M. Ali Shen, Haifeng	2022	Systematic Literature Review	Common DevSecOps adoption challenges	Grouped key obstacles and possible solutions; identified 21 challenges and 31 solutions, categorised into themes such as People, Practices, Tools, and Infrastructure	Limited coverage of emerging DevSecOps tools	(Rajapakse et al., 2022)
Integration of Security in the DevOps Methodology	Sermpezis, Emmanouil Karapiperis, Dimitrios Tjortjis, Christos	2024	Conceptual Analysis and Literature Synthesis	Security not incorporated at early stages of the software development lifecycle within DevOps practices	Proposes the adoption of automation and several best practices, including security-as-code and shift-left security	Largely theoretical; lacks empirical validation or case studies demonstrating practical application	(Sermpezis et al., 2024)
Implementation of DevSecOps by Integrating Static and Dynamic Security Testing in CI/CD Pipelines	Putra, Agung Maulana Siber, Politeknik Kabetta, Herman	2022	Prototype Implementation	Integration of security testing tools in CI/CD	Combines SAST and DAST for security in pipelines	Limited scope of tools and testing environment	(Putra et al., 2022)
Effective DevSecOps Implementation: A Systematic Literature Review	Anjaria, Dhaval Kulkarni, Mugdha	2021	Systematic Literature Review	Lack of a consolidated review on DevSecOps implementation	Outlined factors contributing to effective DevSecOps implementation and reconciled challenges with mitigation strategies; identified key enablers, barriers,	Did not include industrial case validation; limited by the scope of reviewed literature	(Anjaria & Kulkarni, 2021)

Article	Author(s)	Year	Method Used	Problem Addressed	Summary of Results	Limitations	Ref.
					and metrics for success		
Implementing "DevSecOps as a Culture"	Lumpatki, Sairaj S. Patwardhan, Swapnaja Kulkarni, Mukul	2024	Qualitative Study	Cultural and strategic challenges of DevSecOps	Emphasized improved collaboration and faster detection of vulnerabilities; discussed resistance to change, lack of training, and poor communication; proposed a culture-first approach to adoption	No quantifiable data or adoption metrics	(Lumpatki et al., 2022)
DevSecOps: Integrating Security into the DevOps Pipeline	Chittibala, Dinesh Reddy	2023	Integration Framework Proposal	Security often excluded in fast DevOps pipelines	Outlined a secure CI/CD framework with automation; highlighted the significance of embedding security throughout the SDLC, emphasizing strategies like "Security as Code" and automated security tool integration within CI/CD pipelines	Tool-specific; not generalised across stacks	(Chittibala, 2023)
The Emergence and Importance of DevSecOps: Integrating and Reviewing Security Practices within the DevOps Pipeline	Abiona, Oluwatosin Oluwatimileyin Oladapo, Oluwatayo Jacob Modupe, Oluwole Temidayo Oyeniran, Claudius Adewusi, Adebunmi Okechukwu Komolafe, Abiola Moshood	2024	Literature Review and Integration Model	Need to review how security is embedded in DevOps	Emphasized the value of security automation and early integration; explored integration patterns and common security gaps; proposed strategic adoption of DevOps frameworks and the need for proactive security integration to mitigate risks without hindering agility	Limited regional diversity in selected literature; lacked empirical validation and was primarily theoretical	(Abiona et al., 2024)
Integrating Security into the DevOps Pipeline	Athmakuri, Naveen	2024	Conceptual Framework	Missing security automation in DevOps pipelines	Proposed a model for continuous security integration	Mostly theoretical, without security details	(Athmakuri, 2024)
DevSecOps: Integrating Security into the DevOps Lifecycle	Vangala, Vidyasagar	2025	Conceptual Analysis	Lack of integrated security in the DevOps lifecycle	Emphasizes automation and early testing as critical components; identifies culture, tool selection, and team collaboration as key success factors; proposes	No empirical or case-study support	(Vangala, 2025)

Article	Author(s)	Year	Method Used	Problem Addressed	Summary of Results	Limitations	Ref.
					an integrated DevSecOps model for improved security posture		
DevSecOps Practices and Tools	Prates, Luís Pereira, Rúben	2025	Multivocal Literature Review	Identifying and consolidating effective DevSecOps practices and the tools that support them	Conducted an MLR to collate DevSecOps practices and associated tools; identified 39 distinct practices, with tools mapped to 27 of them; provided a taxonomy highlighting areas with robust tool support and those needing further development	Limited by the scope of available literature; may not encompass all emerging practices or tools in the rapidly evolving DevSecOps landscape	(Prates & Pereira, 2025)

MATERIALS AND METHODS

This research utilized the Systematic Literature Review (SLR) approach for searching, critically appraising, and summarizing available evidence-based research relating to DevSecOps practices, prevailing challenges, and methods for their adoption to give a complete summary of the real-time integration of security into the DevOps pipeline. This structured approach is consistent with recent trends in related technology-adoption research, most of which favors a structured over a framework-based, bibliometric, or meta-analytic design for its conceptual clarity and its ability to be used for identifying trends (Isa et al., 2026). The following review questions guided this Systematic Literature Review:

ReQ1: How is security integrated into DevOps pipelines and what are the key practices that define DevSecOps?

ReQ2: What are the main challenges organizations encounter when adopting DevSecOps?

ReQ3: What best practices and strategies can be derived from existing literature that support effective DevSecOps implementation?

To guide this review, we structured our research scope using the PICOC model, as shown in Table 2.

Table 2: PICOC Framework

Component	Description
Population	Software development teams/organizations using DevOps
Intervention	Security integration
Comparison	Traditional DevOps
Outcome	Enhanced security posture, reduced vulnerabilities, improved collaboration, automation
Context	Software development environments using DevOps (agile, cloud-native, CI/CD environments)

Database Sources

Relevant studies were sourced from the following digital libraries: ACM Digital Library, JSAER, SpringerLink, IEEEXplore, ResearchGate, Academia, ScienceDirect, Semantic Scholar and IJFMR.

Search String

("DevSecOps") OR ("Security Integration") OR ("Secure DevOps") AND ("DevOps") OR ("software development teams") OR ("CI/CD") OR ("agile development") AND ("security practices") OR ("security automation") OR ("Security tools") AND ("implementation") OR ("adoption") OR ("challenges") OR ("benefits") OR ("case study") AND ("softwar development lifecycles") OR ("continuous delivery") OR ("cloud-native")

Inclusion and Exclusion Criteria

Each publication was considered against predetermined inclusion and exclusion criteria so as to established which articles fell within the scope of the review, as detailed in table below.

Table 3: Inclusion and Exclusion Criteria

Criteria Type	Description
Inclusion	Peer-reviewed journal articles; studies focused on DevSecOps or security integration in DevOps; published between 2020 and 2026; English language
Exclusion	Non-English articles; articles irrelevant to security integration and DevSecOps; non-peer-reviewed papers; duplicate studies; studies published before 2020

RESULTS AND DISCUSSION

We found the articles on our study selection page on a base 34 articles, from which the duplicates were removed, and from there we started applying inclusion-exclusion to achieve a much more manageable 23 articles to read that were peer-reviewed. Each article used, where possible is based between 2020-2026. All the articles contain a selection of empiric, real-world examples, theoretical, and systematic literature studies that provided a basis for understanding this field of research.

Data Analysis

Figure1 shows the distribution of articles across the different sources searched. Figure 2 presents the number of accepted articles retained per sources after the screening process. Figure 3 illustrates the distribution of the final 23 articles by year of publication.

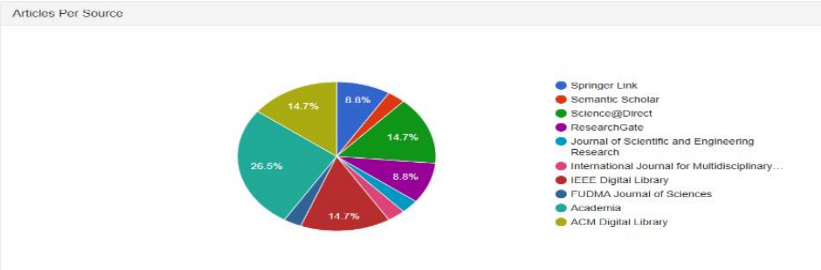

Figure 1: Articles Per Source

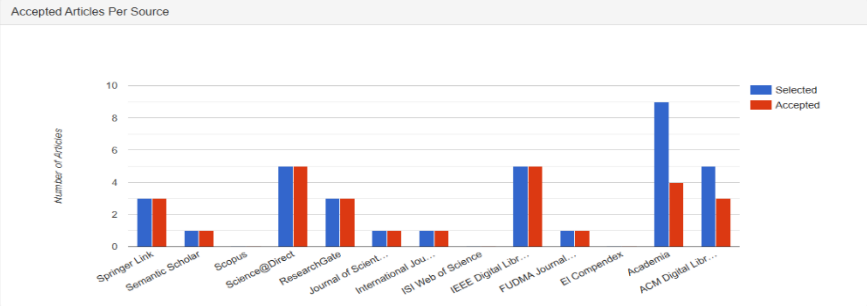

Figure 2: Accepted Articles by Source

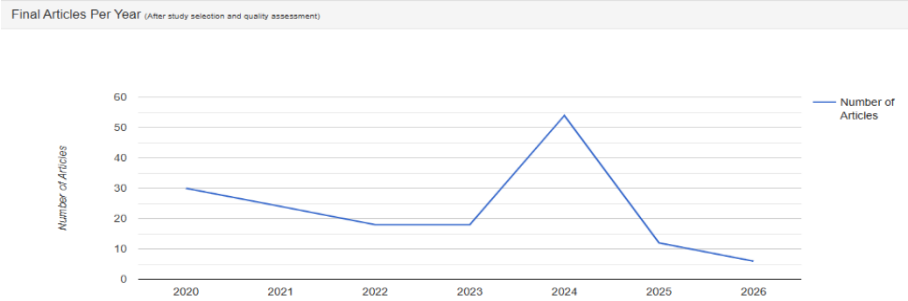

Figure 3: Final Articles Per Year

Results Based on Research Questions

ReQ1

How is security integrated into DevOps pipelines, and what are the key practices that define DevSecOps?

The literature emphasizes the importance of early security integration in software development, also known as "shifting security left," to identify and mitigate exploits early in the software lifecycle. Automated security testing plays a central role in this regard, involving the integration of SAST and DAST tools into CI/CD operations to identify vulnerabilities at various stages (Putra et al., 2022; Veeramachaneni, 2023). Threat modeling is similarly emphasised, typically conducted during the design phase to anticipate and mitigate risks before implementation begins (Ashenden & Ollis, 2020). Another key practice, security as code, involves embedding security policies directly into the codebase and pipeline scripts to ensure consistent enforcement (Athmakuri, 2024). Container security is also widely discussed, with tools such as Trivy, Docker Bench, and Clair employed to scan container images for known vulnerabilities (Lokiny & Nandanampati, 2020). Finally, continuous monitoring, through ongoing logging, scanning, and behaviour analysis, helps identify threats in real-time environments (Chittibala, 2023). Collectively, these practices form the core of the DevSecOps concept, which prioritises continuous feedback loops, cross-functional collaboration, and automation. Thus, the findings above satisfy the first objective: to investigate how security is integrated into DevOps workflows.

ReQ2

What are the main challenges organizations encounter when adopting DevSecOps?

The investigation identified a number of barriers that prevent DevSecOps from being effectively adopted across several of the selected studies. Cultural resistance emerged as a recurring concern, as development and security teams frequently work in silos, leading to a breakdown in communication and misaligned priorities (Lumpatki et al., 2022). Another common issue reported was the complexity of integration with tools, with organizations having trouble integrating security tools with fast DevOps development cycles and maintaining interoperability (Rajapakse et al., 2022). Lack of security skills among developers and operations staff is another frequent impediment to adoption, with the effectiveness of security integration limited by this challenge (Abiona et al., 2024). Another challenge is the latency introduced by performance overhead due to the

scanning process that occurs during deployments (Makani, 2024). Last but not least, there is a lack of a standard framework, in that there is no common approach to the implementation of DevSecOps and it results in variations of the implementation between organizations (Anjaria & Kulkarni, 2021).
The findings above satisfy the second objective identifying common hurdles encountered by organizations when adopting DevSecOps, including cultural, technical, and operational issues.

ReQ3
What best practices and strategies can be derived from existing literature that support effective DevSecOps implementation?
A number of best practices emerged from the literature that can be applied to the adoption DevSecOps. One recommended pratices is to provide security training for developers and DevOps engineers, recommending that the training be continuous training for developers, DevOps engineers, and architects (Ashenden & Ollis, 2020; Rajapakse et al., 2022; Sánchez-Gordón & Colomo-Palacios, 2020). Another recommended practice is incorporating security tools in the development process early in the process and automating those security tools within pipelines, as automation within the CI/CD development pipelines reduces manual effort that is required for development (Putra et al., 2022). Several studies advocate establishing a "security champion" role within DevOps teams, whereby development teams assign security advocates to ensure the implementation of secure coding practices (Sánchez-Gordón & Colomo-Palacios, 2020). Adopting policy-as-code and infrastructure-as-code to enforce consistency was another recurring recommendation, involving the use of code to enforce security configurations and compliance from the ground up (Kumar & Goyal, 2020). Organizations are further encouraged to start small, through pilot projects, before scaling DevSecOps practices organization-wide as maturity increases (Parashar et al., 2020). Lastly, routine security audits, integrating security reviews and code audits into the CI/CD lifecycle, were identified as essential for detecting issues early (Anjaria & Kulkarni, 2021; Chittibala, 2023; Putra et al., 2022).
The findings above satisfy the third objective: to recommend best practices to guide organizations in achieving effective and sustainable DevSecOps implementation.

Thematic Synthesis
Four major themes emerged from the review:

i. *Security Automation* – integration of tools and automated checks.
ii. *Culture and Collaboration* – importance of communication between Dev, Sec, and Ops.
iii. *Toolchain Integration* – managing the complexity of tools.
iv. *Frameworks and Standards* – lack of industry-wide consensus on implementation.

The study conducted a systematic literature review on security integration in DevOps, analyzing 23 peer-reviewed articles from 2020-2026 to identify challenges and best practices. The articles were reviewed using a well-defined protocol that include the formulation of research questions, study selection, quality assessment, data extraction, and thematic synthesis. Three key findings emerged from the review, in relation to security integration, the majority of the studies support shifting security to the left within the SDLC to ensure that security processes are performed in the development stage of the SDLC rather than the testing stage. Regarding challenges to implement such strategies, the major barriers identified include organizational resistance to cultural change, integration complexity, a lack of relevant skills, and lack of existing frameworks or standard methodologies for implementing DevSecOps. Finally, in terms of best practices, effective strategies include automating security checks, providing regular training for development teams in devsecops concepts, establishing security champions, and gradual phased implementation of the concepts.

CONCLUSION
DevSecOps, the integration of security into devops pipelines, is crucial for developing a secure and high-quality software in fast-paced development environments. However, there are challenges to implementing DevSecOps efficiently. Through conducting a literature review of the available information on the topic, it becomes evident that while there is much information on DevSecOps and its adoption, there is need for further empirical studies and industry frameworks for scalable security integration.

RECOMMENDATION
Based on these findings, several recommendations can be made. The first of which is to develop standardized and industry accepted frameworks for DevSecOps practices. Additionally, cross-functional collaboration should also be encouraged to develop a culture of security within their DevOps environment. Organizational efforts should also be made to automate security practices and tools within their CI/CD environments. Finally, DevOps practitioners should undergo continuous education and hands-on training in secure coding and DevSecOps principles to sustain long-term, organization-wide adoption.

REFERENCES
Abiona, O. O., Oladapo, O. J., Modupe, O. T., Oyeniran, C., Adewusi, A. O., & Komolafe, A. M. (2024). The emergence and importance of DevSecOps: Integrating and reviewing security practices within the DevOps pipeline.

Akbar, M. A., Smolander, K., Mahmood, S., & Alsanad, A. (2022). Toward successful DevSecOps in software development organizations: A decision-making framework. *Information and Software Technology*, *147*, Article 106894. https://doi.org/10.1016/j.infsof.2022.106894

Anjaria, D., & Kulkarni, M. (2021). Effective DevSecOps implementation: A systematic literature review. *Revista Gestão Inovação e Tecnologias*, *11*(4), 4931–4945. https://doi.org/10.47059/revistageintec.v11i4.2514

Ashenden, D., & Ollis, G. (2020). Putting the sec in DevSecOps: Using social practice theory to improve secure software development. In *Proceedings of the ACM International Conference Proceeding Series* (pp. 34–44). https://doi.org/10.1145/3442167.3442178

Athmakuri, N. (2024). Integrating security into the DevOps pipeline. *International Journal for Research in Applied Science and Engineering Technology*, *12*(6), 1861–1866. https://doi.org/10.22214/ijraset.2024.63411

Azeem, M., Smolander, K., Mahmood, S., & Alsanad, A. (2022). Toward successful DevSecOps in software development organizations: A decision-making framework.

Information and Software Technology, *147*, Article 106894. https://doi.org/10.1016/j.infsof.2022.106894

Casola, V., De Benedictis, A., Mazzocca, C., & Orbinato, V. (2024). Secure software development and testing: A model-based methodology. *Computers & Security*, *137*, Article 103639. https://doi.org/10.1016/j.cose.2023.103639

Chittibala, D. R. (2023). DevSecOps: Integrating security into the DevOps pipeline. *12*(12), 2074–2078.

Isa, S. A., Atoyebi, T. O., Kolapo, R., Kirubakaran, P., Nathaniel, E. A., & Ahiaba, S. (2026). SYSTEMATIC REVIEW ON THE IMPACTS OF DIGITAL TRANSFORMATION ON CORPORATE INNOVATION PERFORMANCE. *FUDMA Journal of Sciences*, *10*(1), 52-56. https://doi.org/10.33003/fjs-2026-1001-4176

Kumar, R., & Goyal, R. (2020). Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC). *Computers & Security*, *97*, Article 101967. https://doi.org/10.1016/j.cose.2020.101967

Lokiny, N., & Nandanampati, R. (2020). DevSecOps: Integrating security into DevOps with AI in cloud. *7*(10), 239–242.

Lumpatki, S. S., Patwardhan, S., & Kulkarni, M. (2022). Implementing 'DevSecOps as a culture.' *International Journal of Multidisciplinary Research (IJFMR)*, *4*(1), 1–5.

Makani, S. T. (2024). DevOps security tools: Evaluating effectiveness in detecting and fixing vulnerabilities (Vol. 1).

Parashar, A., Dwivedi, A., Kumar, A., & Ahmad Khan, A. (2020). DevSecOps: A case study on a sample implementation of DevSecOps. *International Journal of Engineering and Applied Sciences Technology*, *5*(2), 156–158. https://doi.org/10.33564/ijeast.2020.v05i02.022

Prates, L., & Pereira, R. (2025). DevSecOps practices and tools. *International Journal of Information Security*, *24*(1), 1–25. https://doi.org/10.1007/s10207-024-00914-z

Putra, A. M., Siber, P., Kabetta, H., & Siber, P. (2022). Implementation of DevSecOps by integrating static and dynamic security testing in CI/CD pipelines. https://doi.org/10.1109/ICOSNIKOM56551.2022.10034883

Rajapakse, R. N., Zahedi, M., Babar, M. A., & Shen, H. (2022). Challenges and solutions when adopting DevSecOps: A systematic review. *Information and Software Technology*, *141*, Article 106700. https://doi.org/10.1016/j.infsof.2021.106700

Rangnau, T., Buijtenen, R. V., Fransen, F., & Turkmen, F. (2020). Continuous security testing: A case study on integrating dynamic security testing tools in CI/CD pipelines. In *Proceedings of the 2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC)* (pp. 145–154). https://doi.org/10.1109/EDOC49727.2020.00026

Sánchez-Gordón, M., & Colomo-Palacios, R. (2020). Security as culture: A systematic literature review of DevSecOps. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops (ICSEW 2020)* (pp. 266–269). https://doi.org/10.1145/3387940.3392233

Sandu, A. K. (2024). DevSecOps: Integrating security into the DevOps lifecycle for enhanced resilience. https://www.researchgate.net/publication/380629263

Sermpezis, E., Karapiperis, D., & Tjortjis, C. (2024). Integration of security in the DevOps methodology. *Journal of Global Information Technology*, *15*(3), 45–52.

Vangala, V. (2025). DevSecOps: Integrating security into the DevOps lifecycle.

Veeramachaneni, V. (2023). A systematic review of DevSecOps: Bridging security and agile development for resilient software systems. *21*(07), 1251–1255. https://doi.org/10.48047/nq.2023.21.7.nq23114