International Journal of Information Security (2025) 24:11
https://doi.org/10.1007/s10207-024-00914-z

REGULAR CONTRIBUTION

DevSecOps practices and tools

Luís Prates[1,2] · **Rúben Pereira**[2]

Accepted: 16 October 2024

Abstract

Nowadays, software development happens at a fast pace. At the same time, Information Technology organizations face higher demands and competition while struggling with external threats such as cyberattacks. Therefore, many organizations adopt DevOps as a working culture to improve their Software Development Lifecycle (SDL). However, the success of DevOps adoption remains inconsistent, and recently, IEEE introduced a DevOps standard that might help improve DevOps adoption. The standard mentions DevSecOps as the security aspect of DevOps, adding security practices to the SDL from inception, but what are these practices or capabilities? Which tools can be used to implement these practices? Therefore, a Multivocal Literature Review was performed to identify DevSecOps practices and their definitions, and which tools can be used to implement them.

Keywords DevSecOps · SecDevOps · Practices · Capabilities · Secure software development · Security tools

1 Introduction

Nowadays, customers and users expect faster responses to their constantly changing requirements. As a result, IT organizations face higher demands for shorter software development release cycles [1]. At the same time, organizations must manage increasing demands with non-functional requirements such as security and compliance [2]. As a solution, organizations recourse to automation and increased deployment frequencies [3]. With the help of cloud computing, these strategies ultimately evolved into DevOps [4].

DevOps was defined as an organizational approach that stresses empathy and cross-functional collaboration within and between teams in software development organizations to operate resilient systems and accelerate the delivery of changes [5] [6]. The main benefits of adopting DevOps are faster time to market, increased quality, and increased organizational effectiveness [7]. Recent reports show that 70% of teams release code continuously, once a day, or every few days [8]. However, security reports state that the average data breach cost in 2022 hit a record value of 4.35 million dollars [9].

DevSecOps is the integration of security practices into DevOps [10]. There are two schools of thought towards DevSecOps, one that argues it should not exist as a separate label and that security is part of both the Dev and Ops domain. The other argues that it should be a separate label as the industry must have an explicit call to action toward security [6].

The authors of this paper consider that DevSecOps is a natural extension of DevOps, as argued by the IEEE Standard for DevOps [11].

The available literature already describes what are the perceived practices of DevSecOps. In [10], the perceived DevSecOps practices from the literature are listed.

This research aims to expand the current literature on DevSecOps practices providing researchers and practitioners with a solid foundation for further research. Furthermore, the study aims to identify tools used to support such practices, providing practitioners with a reference guide that can be leveraged to accelerate the adoption of security practices in organizations.

✉ Luís Prates
lfbps@iscte-iul.pt

Rúben Pereira
ruben.filipe.pereira@iscte-iul.pt

[1] ISCTE-IUL, 1649-026 Lisbon, Portugal

[2] INESC INOV-Lab, ISCTE-IUL, R. Alves Redol 9, 1000-029 Lisbon, Portugal

Published online: 05 November 2024

2 Background

2.1 DevOps

DevOps is merging Development (Dev) and Operations (Ops). Since the coining of the term, the industry had issues defining what DevOps was. The main challenge summarized in [12] was whether DevOps was a specific job that required development and IT operations skills. For others, DevOps is not only technical skills but also cultural aspects such as collaboration, automation, measurement, and sharing (CAMS) [4] [13].

According to Smeds et al. [14], the definition is expanded as a set of engineering practices influenced by cultural aspects and supported by technological enablers. [15] concludes that DevOps is a software development methodology that emphasizes collaboration and integration between development teams (responsible for creating and modifying software) and operations teams (responsible for deploying and maintaining software in production environments) where the goal is to streamline the software delivery process by breaking down silos and promoting a culture of shared responsibility, communication, and continuous improvement.

The perceived most common DevOps practices in literature are Continuous Integration, Continuous Delivery, Continuous Monitoring, and Continuous Testing [14] [16] [17] [18] [19] [20].

In [16], Amaro et al., discuss and verify that practices and capabilities are interchangeable terms to describe the application of a method within the DevOps context.

The perceived benefits of adopting DevOps found in the literature are Increased Release frequency, improved quality assurance (QA), and enhanced collaboration and communication. [19] [20] [21] [22] [23]

DevOps was initially coined in 2009 by Patrick Debois [24]. Since then, it is possible to find hundreds of papers studying the topic. The existing literature provides evidence-based research that DevOps is central to success in Software Development. Furthermore, major IT companies such as Google [25] or Dynatrace [26], publish yearly reports on the State of DevOps that provide valuable insights for all working within the Software Development context.

2.2 DevSecOps

DevSecOps term is a combination of the DevOps term merged with Security (Sec) [27] and represents the integration of security practices into DevOps [10]. In [28], it is stated that DevSecOps is important because it avoids having security as an afterthought, which can be costly. DevSecOps uses the same CAMS principles as DevOps [10] [29]. Table 1, lists and expands the DevOps CAMS principles with the DevSecOps.

Table 1 DevOps Principles Explained

Principle	DevOps	DevSecOps
Culture	Collaboration between the Development and Operations team [30] [31]	Collaboration with the Security team. Integration of Security in Development and Operations [32] [33]
Automation	Automation of build, deployment, and testing [34]	Automation of security [35]
Measurement	Measures from business indicators to system indicators [30]	Metrics that track security threats and vulnerabilities [29]
Sharing	Development and o0perations share knowledge, tools, and techniques [30]	Sharing challenges with security team improves the process [35]

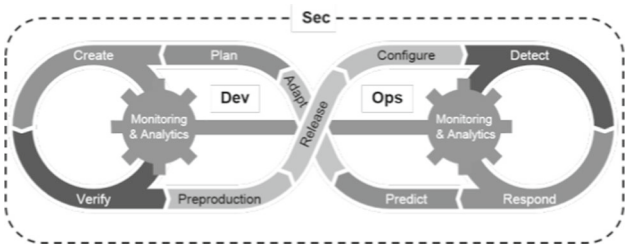

Fig. 1 DevSecOps flow (adapted from [38])

DevSecOps expands on these principles through Shift Security to Left [36]. This principle proposes including security practices from the earliest stages of the development cycle [37]. Figure 1 represents the expected DevOps flow with the integration of security practices.

In Fig. 1, it is possible to identify eleven phases in the DevSecOps Flow [38] [39] [40].

- Create- In this phase, the focus is on secure code development and collaboration among teams to integrate security from the start.
- Plan—Outlining security requirements, assessing risks, and ensuring compliance with regulations are done in this phase.
- Adapt – Implementation of practices that allow to quickly respond to change while maintaining security.
- Verify – In this phase, security tests and vulnerability identification are performed.
- Preproduction – Set of final checks and tests before releasing the application into production.
- Release – Continuous deployment of applications to production while ensuring compliant and secure configurations.

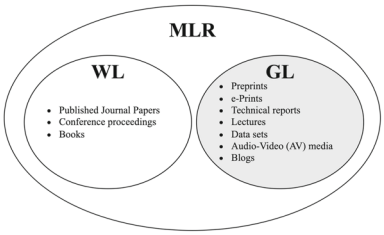

Fig. 2 Venn Diagram showing the spectrum of literature for MLR studies (Adapted from [39])

- Predict – Proactive anticipation of potential security issues.
- Respond – Preparation for managing security incidents to mitigate damages.
- Detect – Proactive and continuous monitoring that detects security threats in real-time.
- Configure – The focus is on maintaining and securing application and infrastructure configurations.

3 Multivocal literature review

This research adopts a Multivocal Literature Review (MLR) as the research methodology (RM). An MLR is a type of Systematic Literature Review (SLR) [41]. An SLR is a means of identifying, evaluating, and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest [42]. An MLR expands the SLR, but contrary to an SLR, which focuses on formal literature, also known as white literature (WL), it also incorporates grey literature (GL) like blogs, videos, web pages, and white papers [43], as shown in Fig. 2.

Since the early nineties, researchers have conducted MLRs in fields such as education [44] and Medicine [45]. Within Software Engineering (SE), MLRs are being undertaken as exploratory studies [46] [47] and are seen as beneficial due to the practical nature of the field [43]. A good argument is made in [48] that GL provides more current perspectives than the WL. It also has been reported that MLRs are useful in closing the gap between academic research and professional practice [41]. Although DevSecOps is still a recent topic, MLR has been previously applied to DevOps [16] and DevSecOps [10].

While GL can provide valuable insights and complement published literature, it also poses several challenges that researchers need to consider. Some of the challenges for researchers while selecting GL sources are accessibility, discoverability, bias, low quality, and lack of standardization [49] [41]. Additionally, source selection in GL is particularly time-consuming and difficult [41]. It is also recognized that GL sources have many different formats and levels of credibility, defined as shades of GL [50]. Table 2 categorizes the sources of GL in terms of credibility.

Table 2 Shades of Grey (adapted from [50])

1st Tier (High Credibility)	2nd Tier (Moderate Credibility)	3rd Tier (Low Credibility)
Books	Annual reports news articles	Blogs
Magazines	Videos presentations	Emails
Government reports	Wiki articles	Tweets
White papers		Product catalogues

DevSecOps is a topic within SE, therefore, for this research, it is preferable to include sources up to the 2nd tier, where it is expected to find valuable insights. The 3rd tier of GL sources is excluded, given the low credibility of those sources.

Therefore, to mitigate the challenges of working with GL sources, this research will adopt systematic guidelines for performing MLR in SE [43] that provide the research with a structured process to search, collect, and extract data from GL. Figure 3 represents the adopted process for the MLR in this research.

1. The first stage of the MLR is **planning the review** consisting of the following steps:
   - Establishing the need for an MLR in each topic.
   - Defining the MLR goal and raising its research questions.
2. The following stage is **conducting the review** which is composed of the following steps:
   - **Search process—**Searching either WL or GL is typically done via means of using defined search strings. Snowballing can also be used to complement the initial search.
   - **Source selection –** Definition of selection criteria and applying the criteria during the selection process.
   - **Study quality assessment** – Determining the extent to which a source is valid and free of bias.
   - **Data extraction –** Design forms of extracting data, and the procedures and logistics. It is required for that extraction. The possibility of automated data extraction and synthesis should be considered.
   - **Data synthesis** – Depending on the type of RQs and the type of data. An adequate data synthesis technique should be selected and used.
3. The final stage is **reporting the review:**
   - **Specifying the Dissemination Mechanism –** Researchers should select the dissemination mechanism. It can be an academic journal, a Practitioner-oriented journal, and magazine, or a web page.

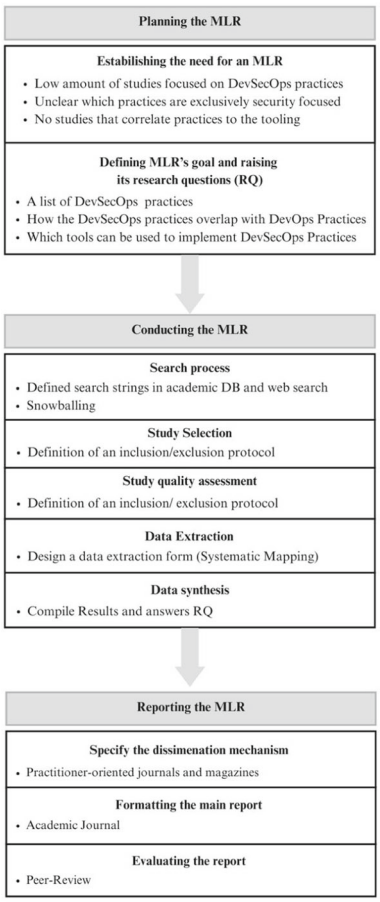

Fig. 3 The MLR process adopted in this research (adapted from [41])

- **Formatting the Report –** The chosen dissemination mechanism affects the report's size, style, structure, and contents.
- **Evaluating the Report –** Depending on the selected dissemination mechanism the evaluation of the report varies. Academic journals use peer reviews as the preferred evaluation technique.

4 Planning the MLR

This section corresponds to the first stage of the MLR process in Fig.3. This section starts by introducing the motivation for this research, and its usefulness is justified. Following the motivation sub-section, the research questions which are intended to be answered throughout the research are formulated. The adopted review protocol is presented in the final part of this section.

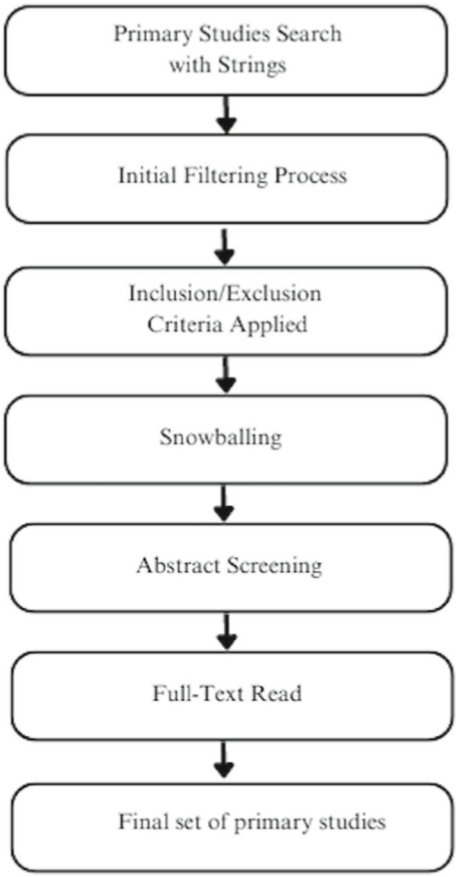

Fig. 4 - Review Protocol defined for this research

4.1 Motivation

Through approaches such as DevOps the speed at which Software Development is released has increased [7] increasing the pressure for the inclusion of security and compliance requirements [2]. There is available literature that identifies and details activities within DevSecOps [51] [10] [52]. However, the current literature does not offer a systematic mapping of DevSecOps practices both from WL and GL. In current literature, the same practice, such as deployment automation, can be mentioned for both DevOps [16] and DevSecOps [51] which indicates overlapping between practices. Previous literature states that one of the main challenges of adopting DevSecOps is the tooling selection [52].

Given the previously presented facts and considering that DevSecOps has been explored more from the industry side the usage of MLR includes industry sources to map DevSecOps practices and the tools that can be used to implement them. Additionally, by leveraging the systematic mapping

of DevOps practices found in [16] it is possible to compare which practices are mentioned in both approaches and how they are implemented in each approach.

5 Research questions

Based on the main purpose of this research, an investigation for scientific and industry-related work was done for DevSecOps practices and tools, which can be translated into the following research questions (RQ):

RQ1 Which DevSecOps practices are mentioned in the literature?

RQ2 Which tools can be used to implement the identified DevSecOps practices?

5.1 A review protocol

To find the primary studies, that might answer the proposed research questions, a search was started in December 2022 using various keywords. Two sets of search strings were used *("DevSecOps OR SecDevOps") AND ("Practices" OR "Capabilities")* and *("DevSecOps OR SecDevOps") AND ("Tools")* to find the primary studies. These search strings were applied to the following databases:

5.1.1 Databases:

- IEEE (https://ieeexplore.ieee.org/Xplore/home.jsp)
- Scopus (https://www.scopus.com/)
- ACM Digital Library (https://dl.acm.org/)
- Google Search (https://www.google.com)
- EBSCO (https://search.ebscohost.com)

The first set of studies is obtained through the search strings applied to the listed databases. To facilitate the search, and collection of GL from the web search engine a Python script was developed [53]. In the first phase, after the search is complete an initial filtering is applied followed by the application of the inclusion and exclusion criteria summarized in Table 3.

Following the application of the Inclusion and Exclusion Criteria, the Snowballing technique is used to obtain further relevant literature.

From that set of studies, the abstracts are screened to determine the relevance of the study to the research. Finally, a full-text read is performed on the remaining set of studies, intending to obtain the final set of primary studies to perform the review.

Figure 4 summarizes the review protocol defined and applied in this research.

Table 3 Inclusion and Exclusion Criteria

Inclusion criteria	Exclusion criteria
Mentions DevSecOps Practices or DevOps Security Practices or DevSecOps Tools	Literature that is inaccessible
Books, conferences, and journals	Vendors tool advertisements
Abstract search term match	Literature focused on topics not related to DevSecOps
Literature published after 2014	Duplicated Articles
Papers in english	

6 Conducting the MLR

This section describes how the review is conducted, which is the second phase of the MLR. In this phase, the search string is applied to the selected databases, followed by the filtering process to obtain the final set of studies. An analysis of the final set of studies concludes this section.

6.1 Study Selection

The initial search for the relevant primary studies starts with applying the defined search strings in the databases defined in the review protocol. This initial query to all metadata is defined as Filter 1 and resulted in 840 papers. The second filter (Filter 2) applies the search string to the abstracts resulting in 181 papers, which represents less than 30% of the number of papers returned in Filter 1, given that the abstract is a summary of the paper this difference is acceptable. This step was only applied to WL.

The study selection process is followed by applying the Inclusion and Exclusion criteria defined in Table 3 (Filter 3) resulting in 186 papers. From the 186 papers obtained in the previous Filter, the Snowballing technique is applied obtaining an additional fourteen papers resulting in a total of 200 papers. The primary studies selection is followed by Filter 4 which is focused on removing duplicated papers obtained from the different databases. Therefore obtaining 149 papers. The abstracts are read from that pool of papers (Filter 5) resulting in 74 papers, this step was only applied in WL. The final step of the primary study selection was the full text read (Filter 6) resulting in a total of 103 papers. Figure 5 resumes the primary study selection process. Table 4 summarizes the filtering process and studies obtained in each filtering stage

6.2 Data extraction analysis

The conclusion of the primary studies selection resulted in 103 publications eligible for data extraction. these studies are analyzed in terms of the publication type, and the publication year is presented here.

Table 4 MLR Filtering Summary

Database	Search string	Filter 1	Filter 2	Filter 3	Snowballing	Filter 4	Filter 5	Filter 6
Google	("DevSecOps OR SecDevOps ") AND ("Practices" OR " Capabilities" OR "Tools")	165	N/A	68	+ 14	65	N/A	51
Scopus	("DevSecOps OR SecDevOps ") AND ("Practices" OR " Capabilities" OR "Tools")	362	49	34		22	18	12
IEEE	("DevSecOps OR SecDevOps ") AND ("Practices" OR " Capabilities" OR "Tools")	48	43	28		15	14	11
EBSCO	("DevSecOps OR SecDevOps ") AND ("Practices" OR " Capabilities" OR "Tools")	165	42	25		10	9	8
ACM	("DevSecOps OR SecDevOps ") AND ("Practices" OR " Capabilities" OR "Tools")	99	46	30		19	17	16
	Total	895	209	201	215	154	75	103

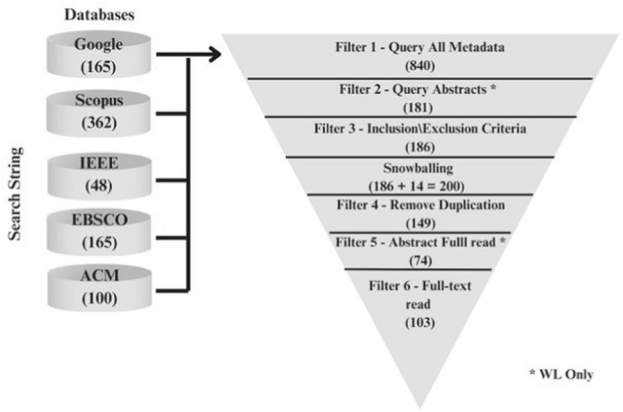

Fig. 5 Followed Multivocal Literature Review process

6.3 Contribution per literature type

From the Google search, 49,51% of the studies were obtained, this number also includes studies obtained through Snowballing given that those studies were also obtained through Google search. The white literature contribution of 50,48% of the studies was distributed in the following manner Scopus (12,62%), EBSCO (8,73%), IEEE (11,65%), and ACM (16,50%).
Figure 6 summarizes the distribution of primary studies per database (Fig. 7).

6.4 Studies distribution over the years

In Fig.8 it is observable that from 2018 to 2023 there was an increase in the number of studies published referring DevSecOps practices and tools. Not only the number of studies over the years has increased but is also diverse as can be seen in Fig.9. This demonstrates a growing interest in the last four years related to DevSecOps practices and tools, therefore, confirming the usefulness of this research in the area.

The first study from this set was the *SecDevOps: Is it a Marketing Buzzword* [33], which was followed webpage *"DevOpsSec: Securing software through Continuous Delivery"* from O'Reilly [36]. From GL, it was identified that several companies publish annual Tech Reports that cover DevSecOps. These Tech Reports in conjunction with the increase of several webpages demonstrate the increased interest in the topic. WL conference papers have also increased over the last few years showing that Academia is also researching the topic. Both WL and GL contributed almost the same number of publications as identified inFig. 7.

7 Reporting the MLR

In this section, the findings from MLR are reported. The two research questions are answered using the extracted data from the identified primary studies by the research protocol. This section ends with a brief discussion of the results followed by a reflection on what are the implications for researchers and practitioners.

7.1 RQ1 which devSecOps practices are mentioned in the literature?

In accordance with the English dictionary, a 'practice' signifies a habitual or customary action. Through a comprehensive literature review encompassing 103 publications, this study identified 39 specific practices commonly implemented by organizations adopting DevSecOps methodologies. Below, these practices are presented along with concise descriptions and the associated benefits derived from their implementation.

Table 5 DevSecOps Practices Mapping to Lifecycle Phase

Lifecycle phase	Practices
Create	Secure code practices
	Team collaboration
	Security champion
	Security training
	Shift left security
Plan	Security requirement gathering
	Threat modelling
	Risk assessment
	Compliance check
Adapt	Continuous integration
	Infrastructure as code
	Configuration management
Verify	Automated security testing
	SAST
	IAST
	DAST
	RASP
	Code review
Preproduction	Penetration testing
	Bug bounties
	Continuous deployment environment hardening
	Patch management
Release	Configuration management
	Software licensing
	Monitoring
Predict	Security auditing
	Logging
	Threat modelling
Respond	Security incident management alerting
	Red team
	Detect
	Monitoring
	Alerting
	Risk assessment
Configure	Secrets management
	Environment hardening
	Configuration management

Static Application Security Test (SAST) is mentioned in 65 publications. SAST is a practice that involves analyzing the source code, bytecode, or binary code of an application to identify and assess potential security vulnerabilities, such as code-level issues and design flaws [52] [54] [39]. SAST tools review the code without executing it [55]. Integrating SAST

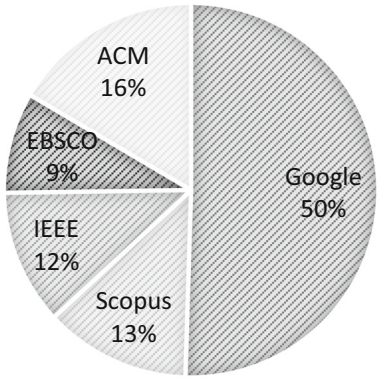

Fig. 6 Distribution of eligible primary studies per database

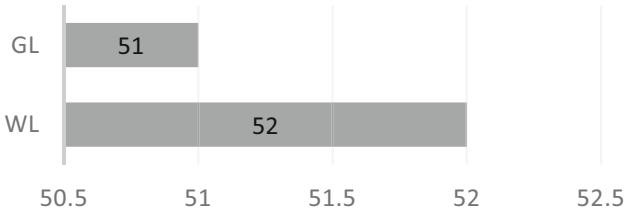

Fig. 7 Number of primary studies by literature nature

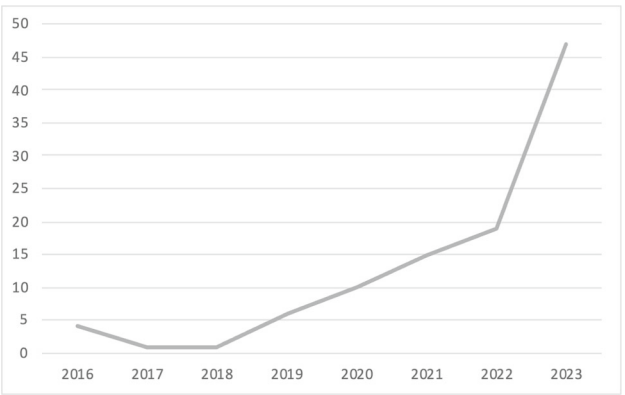

Fig. 8 Identified primary studies over the years

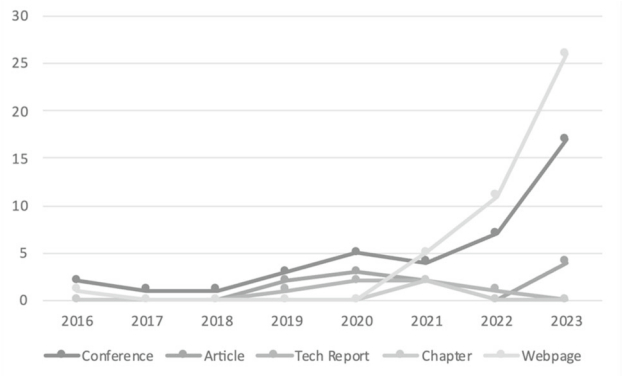

Fig. 9 Primary study types over the years

into the development pipeline, security issues can be identified early in the development process, allowing for timely resolution [56]. Implementing SAST results in the development of more secure software by addressing vulnerabilities before they become a part of the production environment, contributing to a proactive and secure software development process [57] [58].

Automated Security Testing is mentioned in 56 publications. This practice might refer to different types of tests, such as API testing, Web Application Testing, and Unit Testing but focusing on security. This practice consists of the use of automated tools and processes to systematically evaluate and verify the security of software and various stages of development. helping to find and address vulnerabilities early and consistently, ultimately leading to more secure and robust software and systems [33].

Dynamic Application Security Test (DAST) is mentioned in 58 publications. It is a security testing method that focuses on assessing web applications from the outside, simulating real-world attacks [59] [60]. DAST tools actively test applications by sending requests and analyzing responses to identify vulnerabilities, such as injection flaws and authentication issues. DAST is a dynamic and essential part of security testing as it helps identify and validate potential security weaknesses in running applications. Integrating DAST into the development pipeline allows teams to continuously assess the security of their applications, leading to more resilient and secure software products [61].

Vulnerability scanning is mentioned in 44 publications. It is the practice of continuously identifying and assessing potential security vulnerabilities in the software and infrastructure through the usage of automated tools through the development and deployment pipeline [27] [40]. These tools can check software dependencies, and underlying infrastructure such as containers for vulnerabilities and then match them against known Common Vulnerabilities and Exposure (CVE) databases [27]. This proactive approach integrates security into the development process, allowing teams to find and address vulnerabilities early and consistently, promoting a more secure and resilient software development lifecycle [62].

Code Review is mentioned in 33 publications. This practice consists of the systematic examination of code by peers or experts to identify issues, vulnerabilities, and quality concerns [32]. It plays a crucial role in ensuring that code is not only functional but also secure and maintainable [60]. Code Review typically involves checking for security flaws, adherence to coding standards, and best practices [63]. By integrating Code Review into the DevSecOps pipeline, organizations can proactively identify security vulnerabilities and issues early in the development process promoting security-conscious coding and rectifying security-related flaws, contributing to more secure and reliable software applications [40].

Continuous Monitoring is mentioned in 41 publications. In the context of DevSecOps, continuous monitoring is the ongoing and automated process of tracking, collecting, and analyzing data related to the performance and security of software systems [64]. This practice integrates monitoring tools and practices into the DevSecOps pipeline, allowing for real-time visibility into the system's health and security posture [51] [65]. Therefore, this practice helps detect and respond to security incidents, performance issues, and anomalies promptly [66].

Continuous Integration (CI) is mentioned in 41 publications. CI is a fundamental DevSecOps practice that involves automatically and continuously integrating code changes from multiple developers into a shared repository [63] [67]. This integration process includes automated building, testing, validation of the code, and security checks ensuring that the code functions correctly and meets quality and security standards [57]. By making sure security checks are an integral part of the development pipeline CI helps identify security vulnerabilities and issues early in the development process, allowing for their prompt resolution, resulting in more secure and reliable software that can be deployed with confidence [55].

Threat Modelling (TM) is mentioned in 40 publications. TM is the systematic and proactive process for identifying and assessing potential security threats and vulnerabilities in software applications and systems. It involves analyzing the design, architecture, and functionality of the application to identify possible attack vectors and weaknesses [67]. TM helps development and security teams understand the security landscape and prioritize mitigation strategies early in the development process [39] [63] [68]. By incorporating TM organizations can effectively anticipate and address security concerns, resulting in more secure and resilient software products [58].

Version Control (VC) is mentioned in 26 publications. VC is the practice of managing and tracking changes to software code, configurations, and infrastructure throughout the development and deployment lifecycle [40]. It involves using version control systems (VCS), such as Git, to maintain a history of changes, enabling teams to collaborate, track modifications, and ensure that the latest code and configurations are used [69]. By using VC, organizations can enhance collaboration, ensure code integrity, and maintain security through audit trails and controlled access. The result is a more secure and well-organized software development process [70].

Shift Left Security (SLS) is mentioned in 36 publications. SLS is a practice that involves integrating security considerations and practices earlier in the software development process [56]. By shifting security "left" in the development

lifecycle, organizations aim to identify and address security issues as early as possible [40]. Promoting a proactive and preventative security mindset, with security becoming an integral part of the development process from earlier stages minimizes vulnerabilities and security risks, resulting in more secure and resilient software applications [27] [39].

Cross-team collaboration is mentioned in 36 publications. In the context of DevSecOps, this refers to incorporating security practices in the DevOps processes by promoting collaboration between the development teams, the operations teams, and the security teams [33]. This approach breaks down silos and encourages interaction between development, operations, and security teams to work together seamlessly resulting in a security-aware culture [40] ensuring that security is an integral part of the entire SDLC [69].

Software Composition Analysis (SCA) is mentioned in 27 publications. This practice refers to the activity of scanning and analyzing software components and libraries used in an application to identify vulnerabilities, licensing issues, and potential security risks [39] [67]. SCA tools examine the composition of an application, checking for known security vulnerabilities in third-party and open-source code [57]. By integrating SCA into the development process, organizations can proactively address security and compliance concerns related to software components [36], resulting in higher overall security and reliability of the software while also helping to meet compliance requirements.

Logging is mentioned in 21 publications. Logging is the practice of recording events, actions, activities, and other relevant in a system with the intent of generating a detailed log of what happens within an application or infrastructure [60]. Logs are a crucial source of information for troubleshooting, detecting malicious activity, and establishing accountability [58] [67]. In the context of DevSecOps by having an effective logging practice organizations can monitor and analyze security events and incidents in real-time, resulting in the ability to respond to security threats promptly. Additionally, by having security logs organizations have the necessary data to investigate security breaches and incidents, and the resulting insights can be used to better prepare for future attacks [64].

Penetration Testing (PT) is mentioned in 24 publications. PT is a proactive practice that is the systematic and controlled practice of simulating cyberattacks to identify vulnerabilities and weaknesses in the software applications and infrastructure of an organization that can be exploited [39] by a malicious actor. PT usually involves security experts attempting to breach security defenses and discover potential security flaws, even though there are tools for PT [52] [32]. The insights obtained from PT are useful to address security flaws and implement more security measures that contribute to a more secure environment [60].

Continuous Deployment (CD) is mentioned in 34 publications. This practice refers to when code base changes are automatically and continuously built, tested, and deployed to production environments [56]. In the context of DevSecOps, this practice emphasizes the integration of security throughout the entire process, ensuring that security testing and checks are an integral part of the automated pipeline [67]. Integrating security checks into the pipeline results in rapid and reliable software releases while also ensuring that security measures are consistently applied, helping to identify and mitigate security vulnerabilities at every stage of development and deployment [33] [71].

Security Training is mentioned in 27 publications. This refers to the creation of a structured educational program designed to equip development and operations teams with the necessary skills to address security concerns effectively and increase their security awareness [29] [62]. By providing ongoing security education, organizations can enhance the overall security posture of their software and infrastructure, ensuring that security is an integral part of the development process [40].

Environment Hardening (EV) is mentioned in 16 publications. EV is the act of proactively securing and fortifying infrastructure environments to reduce vulnerabilities and security risks [55]. EV typically involves implementing security controls, access restrictions, and configuration changes to make the environment less susceptible to cyberattacks and unauthorized access, which results in secure and resilient environments, reducing the attack surface [33] [52] and enhancing the overall security posture of applications and infrastructure.

Interactive Application Security Testing (IAST) is mentioned in 18 publications. IAST is a dynamic security testing approach that continuously assesses the security of applications during runtime [59] or the functional test stage [52]. IAST tools are integrated into the application through an agent and monitor its behavior in real-time, identifying vulnerabilities, and potential security issues as they occur [59]. IAST helps developers and security teams to rapidly detect the application's potential vulnerabilities [72] and address security flaws, increasing the detection speed of vulnerabilities results in a more secure and responsive software development process [36].

Secure Coding is mentioned in 18 publications. It is the practice of writing software code with a focus on security from the very beginning of the SDLC. It involves following security best practices, avoiding common vulnerabilities of the programming language, and proactively addressing security concerns during programming activities [62]. Some common principles are input validation, avoiding hard-coded credentials, and securely handling sensitive data. Writing

code with security awareness reduces the likelihood of introducing vulnerabilities and security flaws into the codebase, ultimately leading to more secure software applications [64].

Continuous Compliance is mentioned in 22 publications. This practice refers to maintaining and verifying compliance with regulatory and security requirements throughout the entire SDLC by integrating automated compliance checks into the CI/CD pipeline [67], ensuring that security and compliance controls are consistently evaluated and validated [33]. Automating compliance allows organizations to continuously track and demonstrate compliance with regulations and standards, reducing the risk of compliance violations and ensuring that security measures align with legal and industry requirements [60] [73].

Security Policies are mentioned in 12 publications. This practice refers to defining security policies and compliance requirements. The practice might refer to defining these policies as guidelines or as code or machine-readable scripts [74]. The code-based policies (Policies as Code) can be integrated into the CI/CD pipeline and used to automate and enforce security and compliance checks throughout the development and deployment process [75]. By defining and automating Security Policies it ensures that security and compliance are consistently applied, making it easier to detect and address violations early in the SDLC.

Gathering Security Requirements (GSR) is mentioned in 13 publications. In the context of DevSecOps, this refers to the systematic process of identifying and documenting the security needs and expectations for a new system [57] [32]. The gathering of security requirements requires collaboration between stakeholders, security experts, and development teams to define the specific security requirements, compliance requirements, and risk tolerance levels for the system being developed [63]. GSR is crucial in shifting security to the left and ensuring security is a concern from inception.

Configuration Management (CM) is mentioned in 17 publications. CM is the practice of systematically managing and controlling the configuration of software, infrastructure, and associated resources throughout their lifecycle. CM helps organizations maintain a standardized and secure environment by tracking and managing changes, identifying deviations, and ensuring that configurations align with security policies [40]. This is done by defining, documenting, and automating configurations to ensure consistency, reliability, maintainability, and security [66]. CM tools are crucial for security in the release phase since they provide visibility into the static configuration of a dynamic infrastructure [76].

Secrets Management is mentioned in 11 publications. It is the practice that involves securely storing, managing, and accessing sensitive information such as API keys, credentials, and encryption keys. The goal of the practice is to ensure that these secrets are protected throughout the software development and deployment lifecycle [40]. The implementation of this practice typically includes techniques such as secure storage, password rotation, and access controls [55] [63]. By implementing Secrets Management organizations can maintain the confidentiality and integrity of sensitive data, reducing the risk of unauthorized access and data breaches [61].

Risk Assessment is mentioned in 15 publications. This practice refers to the systematic process of identifying, evaluating, and prioritizing potential security risks and threats associated with software development and deployment [32] [39]. By assessing factors such as business impact, data sensitivity, and compliance requirements organizations can make informed decisions about which security measures they should invest in first based on the probability of security incidents and vulnerabilities [52].

Infrastructure as Code (IaC) is mentioned in 28 publications. IaC allows for the management and automation of infrastructure provisioning through code, rather than manual configuration [66]. Using declarative or imperative scripting languages, it is possible to define the desired state of infrastructure components, such as servers, networks, and databases [75]. IaC enables infrastructure to be version-controlled, tested, and deployed alongside software code, ensuring that the entire environment is consistent and reproducible [77]. Security measures can also be integrated directly into the infrastructure code, resulting in enhanced infrastructure security [36].

Run-time Application Self-Protection (RASP) is mentioned in 13 publications. RASP is a security approach that adds security protection to applications during their runtime. RASP tools rely on the integration of the protection engine into the application runtime environment to observe the associated execution flow and detect anomalous calls and behaviors likely to represent an active attack [59]. This real-time protection enhances the application's ability to defend against attacks, resulting in reducing the risk of security incidents and vulnerabilities in production environments [76].

Alerting is mentioned in 7 publications. In the context of DevSecOps, this practice refers to continually generating alerts in response to security incidents, anomalies, or events that require immediate attention [68]. This involves organizations integrating alerting systems into their infrastructure, allowing teams to receive real-time notifications about potential security threats or operational issues [78] [79]. This practice contributes to faster response times to security incidents.

Security Auditing is mentioned in 11 publications. This practice refers to regularly reviewing and assessing an organization's software development, deployment, and security practices to ensure compliance with standards, policies, and best practices [71]. Auditing should include various aspects of the SDLC such as code quality, security measures, access controls, and compliance with regulatory requirements [60]

[68]. This practice aims to identify areas of improvement, potential vulnerabilities, and non-compliance issues contributing to maintaining the integrity through the whole SDLC and creating more secure environments [36].

Red Team is mentioned in 8 publications. The practice refers to the usage of a specialized group or team of security experts to simulate cyberattacks and security breaches on an organization's systems, applications, or infrastructure [67]. The primary goal of a Red Team is to identify vulnerabilities and weaknesses in an organization's security posture [10] [40]. The lessons learned from Red Team activities are valuable in improving security and enhancing incident response procedures, which improves the overall security of the organization [36].

Continuous Reporting is mentioned in 6 publications. This practice in the context of DevSecOps involves the ongoing generation and delivery of reports that provide real-time insights into the security posture of a system or application throughout its lifecycle [33]. The generated reports allow teams to continuously monitor and assess the security status and enable the capacity to timely identification of security issues, making it easier to respond quickly and effectively to potential threats and vulnerabilities [56] [72].

Web Application Firewall (WAF) is mentioned in 8 publications. WAF is a layer of defense for web applications. This security solution has the capability of monitoring traffic between a Web Application (WA) and the Internet and protecting the WA by filtering and blocking malicious HTTP traffic [32]. By using WAF organizations aim to protect their WA against various cyber threats and ensure the integrity and confidentiality of data transmitted over the Internet resulting in an increased security posture [80].

Security Champions is mentioned in 8 publications. Security Champions are individuals or teams with specialized security knowledge and expertise who play a critical role in promoting and enhancing security practices within an organization [29]. Security Champions help raise security awareness, provide guidance on security best practices, and assist in integrating security into the development process therefore they are the first step to creating a cross-functional team focused on Application Security and Security Operations [40]. By acting as the bridge between security and development, Security Champions are essential to fostering a culture of security consciousness and ensuring that security is a top priority throughout the software development lifecycle.

Patch Management is mentioned in 12 publications. This practice refers to continuously and proactively identifying and applying software patches and updates to address security vulnerabilities and maintain the security of software and systems [65]. By having this practice organizations ensure that security patches are integrated into the development and deployment pipeline, reduce the window of exposure to known vulnerabilities, enhance security, and minimize the risk of security breaches [64] [68].

Software Licensing is mentioned in 3 publications.

This practice in the context of software development refers to the terms that govern the use, distribution, and protection of software by establishing how users can legally use, copy, modify, and share software [32]. In the context of DevSecOps, understanding and managing software licenses is crucial to ensure compliance with open-source and third-party libraries used in the development process.

Chaos Engineering is mentioned in 3 publications. This practice refers to a proactive and focused activity of testing and assessing the resilience and security of software and systems by intentionally introducing controlled chaos, such as infrastructure failures or simulated cyberattacks [74] [76]. This approach helps organizations identify vulnerabilities and weaknesses under adverse conditions and prepare for real-world security incidents.

Bug Bounties is mentioned in 2 publications. This practice refers to creating a program where individuals are incentivized to discover and report security vulnerabilities in an organization's software applications, systems, or infrastructure. By incentivizing external experts to actively search for and disclose vulnerabilities, organizations expand their pool of security experts which in return results in the uncovering of vulnerabilities that may have gone unnoticed [36].

Security Incident Management is mentioned in 11 publications. This practice refers to the activities necessary to handle security incidents in an organization. These activities can include detecting the security breach with specific technologies, defining strategies to contain the incident, implementing procedures to resume normal activities, and post-incident inquiries that result in additional prevention measures [81] [82]. By implementing this practice, organizations prepare themselves for possible cyber-attacks resulting in less time to recover [83] and continuous learning from each incident increasing overall security over time.

Identity and Access Management (IAM) is mentioned in 11 publications. policies and technologies that ensure the right individuals have appropriate access to resources within a computing environment. Is the practice of managing digital identities by defining roles and permissions and enforcing access controls. Combining technologies such as Multifactor Authentication and policies such as the Principle of Least of Privilege helps achieve this resulting in an enhanced security posture [82] [84].

Table 6 summarizes the number of publications mentioning each practice.

An analysis of Fig. 10 reveals a notable convergence between White Literature (WL) and Grey Literature (GL) across the majority of the 39 identified DevSecOps practices. GL identified 39 practices, while WL contributed to the identification of 38, with only one practice, P38 – Chaos

Table 6 Number of Publications mentioning the practice

PID	Practice	Mentioned in	#
P1	SAST	[8] [33] [36] [38] [39] [40] [52] [54] [55] [56] [57] [58] [59] [60] [61] [63] [65] [72] [73] [74] [75] [76] [77] [78] [79] [80] [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95] [96] [97] [98] [99] [100] [101] [102] [103] [104] [105] [106] [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [118] [119] [120]	65
P2	DAST	[8] [36] [38] [39] [40] [52] [55] [56] [57] [58] [59] [60] [61] [65] [72] [73] [74] [75] [76] [77] [78] [79] [80] [82] [84] [83] [85] [87] [88] [89] [91] [93] [94] [95] [96] [97] [98] [99] [100] [102] [103] [104] [106] [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [118] [119] [120] [121]	58
P3	Automated security testing	[29] [32] [33] [36] [38] [39] [40] [51] [52] [56] [57] [58] [60] [61] [63] [64] [65] [66] [67] [68] [70] [71] [72] [73] [75] [76] [77] [79] [81] [82] [85] [87] [88] [90] [91] [95] [96] [97] [100] [106] [109] [110] [111] [112] [115] [116] [117] [121] [122] [123] [124] [125] [126] [127] [128] [129]	56
P4	Vulnerability scanning	[27] [29] [33] [36] [38] [39] [40] [51] [52] [55] [57] [61] [62] [63] [65] [67] [68] [73] [75] [76] [78] [79] [83] [86] [87] [89] [90] [91] [92] [97] [98] [106] [108] [110] [111] [113] [114] [117] [122] [121] [127] [129] [130] [131]	44
P5	Continuous monitoring	[8] [29] [32] [33] [36] [38] [39] [40] [51] [52] [60] [62] [64] [65] [66] [67] [68] [70] [71] [73] [79] [81] [82] [84] [91] [97] [100] [103] [106] [109] [110] [111] [113] [114] [115] [117] [120] [121] [126] [131] [132]	41
P6	Continuous integration	[33] [36] [38] [51] [52] [55] [57] [58] [59] [63] [65] [67] [70] [73] [78] [82] [87] [89] [90] [92] [95] [96] [98] [99] [100] [102] [106] [107] [110] [113] [114] [115] [118] [120] [121] [125] [130] [133] [134] [135] [136]	41
P7	Threat modelling	[32] [36] [38] [39] [51] [52] [58] [59] [60] [63] [64] [65] [67] [68] [72] [74] [76] [79] [82] [84] [89] [90] [100] [101] [106] [108] [110] [111] [112] [114] [115] [117] [119] [121] [122] [127] [128] [129] [133] [134]	40
P8	Shift left security	[8] [27] [36] [39] [40] [52] [56] [61] [62] [68] [72] [75] [81] [84] [86] [93] [95] [96] [99] [100] [106] [107] [109] [113] [114] [115] [117] [119] [121] [123] [125] [126] [128] [129] [137] [138]	36
P9	Collaboration	[10] [32] [33] [36] [39] [40] [52] [63] [67] [68] [69] [71] [75] [79] [80] [81] [82] [84] [95] [97] [99] [100] [109] [110] [111] [112] [113] [114] [115] [116] [119] [121] [126] [128] [131] [137]	36
P10	Continuous deployment	[27] [33] [36] [51] [52] [56] [64] [66] [67] [68] [71] [74] [82] [89] [90] [91] [98] [99] [100] [102] [106] [107] [110] [113] [114] [115] [118] [120] [125] [126] [133] [134] [135] [136]	34
P11	Code review	[32] [36] [38] [39] [40] [52] [57] [59] [60] [63] [64] [65] [67] [68] [69] [70] [76] [78] [84] [89] [90] [92] [95] [96] [106] [109] [119] [122] [123] [129] [131] [132] [138]	33
P12	Infrastructure as code	[36] [38] [59] [66] [74] [75] [77] [84] [89] [99] [100] [106] [107] [108] [109] [110] [112] [115] [116] [118] [119] [120] [125] [127] [131] [136] [139] [140]	28
P13	SCA	[36] [38] [39] [57] [58] [59] [67] [72] [73] [75] [76] [80] [83] [84] [87] [88] [89] [92] [93] [94] [99] [104] [108] [109] [116] [117] [118]	27
P14	Security training	[29] [32] [33] [38] [40] [52] [62] [64] [68] [72] [73] [81] [84] [87] [109] [110] [111] [112] [116] [117] [119] [126] [127] [129] [132] [131] [138]	27
P15	VCS	[27] [36] [38] [39] [40] [52] [56] [57] [63] [64] [65] [66] [67] [68] [69] [70] [81] [93] [95] [96] [98] [105] [106] [121] [125] [129]	26
P16	Penetration testing	[32] [36] [52] [39] [59] [60] [63] [64] [65] [67] [68] [78] [83] [86] [87] [98] [106] [107] [111] [116] [117] [123] [131] [132]	24
P17	Continuous compliance	[32] [33] [36] [52] [55] [60] [67] [68] [73] [81] [106] [107] [109] [110] [113] [114] [115] [118] [119] [120] [127] [128]	22
P18	Logging	[38] [52] [58] [60] [61] [64] [66] [67] [71] [76] [79] [83] [87] [95] [96] [100] [106] [116] [120] [121] [132]	21

Table 6 (continued)

PID	Practice	Mentioned in	#
P19	IAST	[36] [38] [52] [58] [59] [72] [75] [80] [87] [94] [103] [104] [106] [108] [109] [115] [121] [137]	18
P20	Secure coding	[36] [62] [63] [64] [65] [67] [73] [85] [98] [100] [102] [109] [112] [116] [121] [123] [125] [126]	18
P21	Configuration management	[32] [36] [40] [52] [60] [66] [67] [76] [91] [107] [109] [110] [116] [117] [119] [129] [139]	17
P22	Environment hardening	[33] [36] [52] [55] [61] [63] [64] [67] [68] [80] [84] [87] [106] [120] [121] [132]	16
P23	Risk assessments	[32] [36] [39] [52] [60] [65] [68] [81] [84] [109] [111] [114] [123] [126] [134]	15
P24	Security requirement gathering	[32] [39] [52] [57] [60] [63] [64] [67] [70] [106] [123] [129] [135]	13
P25	RASP	[36] [59] [76] [80] [83] [85] [87] [100] [104] [105] [109] [115] [140]	13
P26	Security policies	[32] [33] [38] [52] [64] [67] [71] [74] [75] [92] [125] [127]	12
P27	Patch management	[64] [65] [68] [76] [84] [98] [100] [106] [111] [116] [128] [129]	12
P28	Security incident management	[81] [82] [83] [84] [106] [109] [110] [111] [113] [115] [138]	11
P29	IAM	[82] [84] [106] [114] [116] [117] [125] [126] [127] [129] [138]	11
P30	Secrets management	[39] [40] [55] [61] [63] [64] [67] [78] [106] [119] [132]	11
P31	Security auditing	[36] [60] [65] [67] [68] [71] [82] [83] [106] [129] [138]	11
P32	Security champions	[29] [40] [70] [73] [109] [112] [117] [119]	8
P33	Red team	[10] [36] [39] [40] [67] [82] [109] [119]	8
P34	WAF	[32] [36] [80] [87] [100] [104] [105] [132]	8
P35	Alerting	[52] [67] [68] [71] [78] [79] [111]	7
P36	Continuous reporting	[33] [56] [66] [67] [72] [114]	6
P37	Software licensing	[8] [32] [121]	3
P38	Chaos engineering	[74] [76] [82]	3
P39	Bug bounties	[36] [119]	2

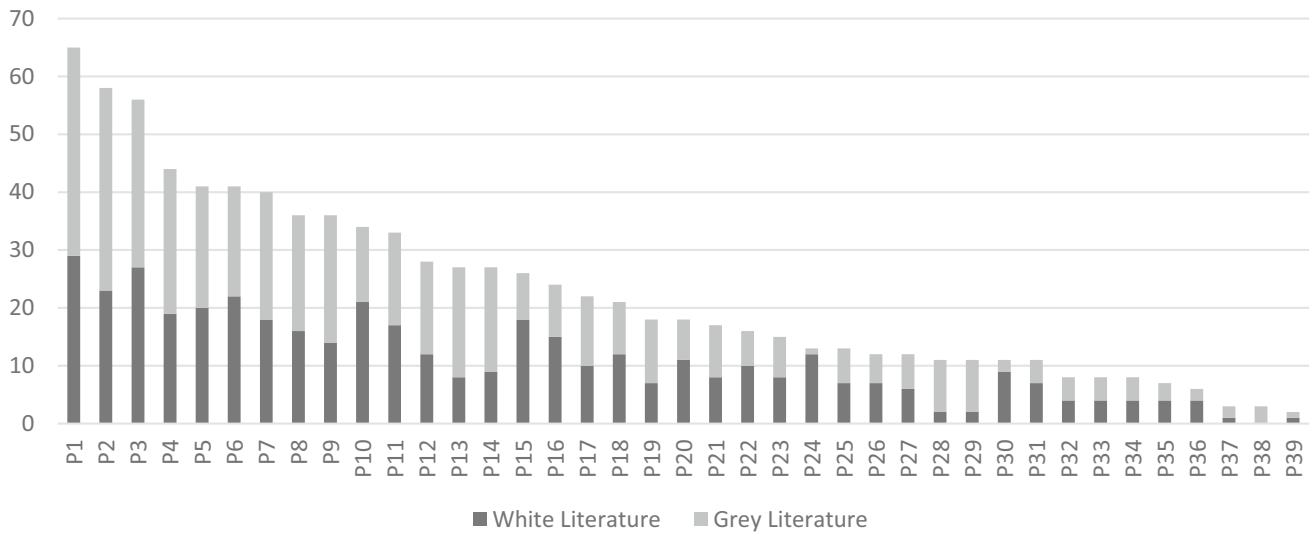

Fig. 10 Distribution of mentions by literature nature

Engineering, not extracted from WL primary studies. On average, WL contributed 49.13% and GL contributed 50.87% to each practice. This convergence underscores a clear alignment between academia and industry regarding the perceived landscape of DevSecOps practices.

In [38] the DevSecOps lifecycle phases are identified and represented in the diagram in Fig. 1. From the primary studies, it was possible to extract the DevSecOps practices which now can be mapped to the lifecycle phases as represented in Table 5.

From the identified practices and considering the number of mentions for each it is possible to rank them and divide them into three tiers. Tier 1 are the practices with more than 30 mentions and can be considered the core practices of DevSecOps. Tier 2 considers the practices with mentions between 10 and 29, based on the number of mentions these practices can be also considered important within DevSecOps. Tier 3 considers the practices with fewer than 10 mentions, the low number of mentions for these practices can be interpreted as nice to haves but not crucial for DevSecOps implementation.

Table 7 summarizes the practices by tiers.

Table 7 Practices ranked by tiers

Tier	Practices
Tier 1 (> = 30) Core	SAST
	DAST
	Automated security testing
	Vulnerability scanning
	Continuous monitoring
	Continuous integration
	Threat modelling
	Shift left security
	Collaboration
	Continuous deployment
	Code review
Tier 2 (< 30 > = 10) Important	Infrastructure as code
	SCA
	Security training
	VCS
	Penetration testing
	Continuous compliance
	Logging
	IAST
	Secure coding
	Configuration management
	Environment hardening
	Risk assessment
	Security requirement gathering
	RASP
	Security policies
	Patch management
	Security incident management
	IAM
	Secrets management
	Security auditing
Tier 3 (< 10) Nice to have	Security champions
	Red team
	WAF
	Alerting
	Continuous reporting
	Software licensing
	Chaos engineering
	Bug bounties

7.2 RQ2 – Which tools can be used to implement the identified DevSecOps practices?

To address this inquiry, a comprehensive inventory of all referenced tools from the identified literature was compiled. Initially, this survey yielded an extensive list exceeding 250 tools, necessitating the establishment of discerning selection criteria for practical efficacy. The objective was to streamline the initial pool of tools, prioritizing those most frequently mentioned and widely adopted.

In crafting the final list, preference was accorded to tools with substantial mentions, ensuring each practice was represented by at least three tools wherever possible. Furthermore, in cases where multiple tools were identified for a practice, priority was granted to those with more than two mentions. Additionally, for open-source tools, the popularity metrics such as the number of stars on their respective GitHub repositories were taken into consideration to further refine the selection process. GitHub stars is a feature that allows users to bookmark and show appreciation for repositories they find useful or interesting.

The catalog of tools corresponding to each identified practice is meticulously detailed and enumerated in Table 8.

Out of the 39 DevSecOps practices examined, tools were successfully identified for 27 practices. Within the subset of 25 practices with identified tools, at least three tools were identified for each practice, except for seven practices namely Dynamic Application Security Testing (DAST), Code Review, Interactive Application Security Testing (IAST), Policy as Code, Infrastructure as Code (IaC),

Table 8 List of tools identified for each practice

Practice	Tools	Mentioned in	#	WL	GL	GitHub Stars
SAST	Sonarqube	[29] [36] [52] [53] [54] [56] [59] [67] [74] [75] [78] [79] [81] [82] [85] [86] [87] [89] [90] [91] [92] [98] [106] [110] [114] [117] [118] [122] [139] [141] [142]	31	13	18	> 8.4 k
	Snyk	[27] [29] [55] [57] [61] [67] [74] [75] [86] [87] [100] [102] [110] [114] [119] [120] [122] [139] [142]	19	12	7	> 4 k
	Checkmarx	[29] [36] [55] [74] [75] [79] [86] [87] [89] [90] [91] [92] [110] [112] [114] [119] [142]	17	5	12	N/A
	Veracode	[29] [55] [79] [87] [89] [91] [110] [112] [114] [141] [142]	11	2	9	N/A
	Fortify	[36] [57] [61] [75] [79] [87] [90] [91] [110] [112] [142]	11	3	8	N/A
	Clair	[36] [55] [61] [67] [74] [91] [114] [118] [119]	9	5	4	> 9.9 k
	Coverity	[29] [36] [55] [94] [119] [142]	6	4	2	N/A
	Docker bench	[61] [67] [74] [75] [81]	5	2	3	> 8.8 k
	Contrast security	[36] [79] [87]	3	1	2	N/A
	Git leaks	[55] [74]	2	1	1	> 14.7 k
	Shiftleft	[75] [122]	2	0	2	> 700
	Codacy	[90] [92]	2	0	2	N/A
	Frama-C	[39] [85]	2	2	0	> 150
Automated security testing	Gautntlt	[33] [36] [122] [117]	4	2	2	N/A
	Selenium	[56] [67] [70] [117]	4	3	1	> 28 k
	BDD security	[36] [79] [122]	3	1	2	> 553
	Acutenix	[90] [92] [110]	3	0	3	N/A
DAST	OWASP zap	[33] [36] [39] [56] [57] [60] [61] [67] [74] [75] [78] [82] [83] [86] [87] [98] [100] [106] [110] [112] [117] [118] [119] [122] [142]	25	14	11	> 11.5 k
	HCL appscan	[75] [119] [142]	3	1	2	N/A
Vulnerability scanning	Trivy	[55] [74] [78] [83] [117] [118] [122]	7	2	5	> 3.5 k
	Retire.js	[29] [36] [55] [67] [74] [118]	6	4	2	> 3.4 k
	Anchore	[55] [61] [67] [74] [118]	5	3	2	N/A
	Brakeman	[36] [55] [74] [75]	4	2	2	> 6.8 k
	Checkov	[74] [75] [99] [139]	4	2	2	> 6.3 k
	Arachni scanner	[33] [118] [122]	3	1	2	> 3.6 k
	Bandit	[55] [74] [118]	3	1	2	> 5.7 k
	Tfsec	[74] [139]	2	1	1	> 6.5 k
Code review	Gerrit	[36] [67] [89] [122]	4	2	2	> 800
	Crucible	[36]	1	1	0	N/A
Continuous monitoring	Kibana	[36] [61] [66] [67] [79] [87] [91] [103] [106] [110] [118, 121] [122]	13	6	7	> 19 k
	Grafana	[61] [67] [79] [87] [103] [118] [121] [122]	8	3	5	> 59 k

Table 8 (continued)

Practice	Tools	Mentioned in	#	WL	GL	GitHub Stars
	Falco	[67] [74] [76] [118] [122]	5	1	4	> 6.7 k
	Nagios	[67] [79] [121] [132]	4	1	3	> 1.4 k
	Prometheus	[61] [67] [103] [121]	4	3	1	> 51 k
	Datadog	[36] [61] [79]	3	2	1	N/A
	Tripwire	[33] [76] [79]	3	1	2	> 800
	New relic	[33] [91]	2	1	1	N/A
	StatsD	[61] [67]	2	2	0	> 17 k
	Ganglia	[33] [67]	2	2	0	> 300
	Suricata	[67] [132]	2	1	1	> 3.8 k
Continuous integration	Jenkins	[36] [39] [60] [63] [66] [67] [70] [82] [83] [92] [100] [105] [110] [121] [122]	15	10	5	> 22 k
	GitLab CI	[55] [63] [82] [106] [112] [121] [122] [140]	8	4	4	N/A
	Github actions	[55] [57] [78] [102] [112] [122]	6	3	3	N/A
	Travis CI	[55] [121] [122]	3	1	2	N/A
	Circle CI	[55] [122]	2	1	1	N/A
Threat modelling	OWASP Threat Dragon	[39] [67] [74] [79] [87] [122]	6	2	4	> 700
	ThreatModeler	[79] [87] [89] [91] [114]	5	0	5	N/A
	IriusRisk	[79] [82] [87] [90] [100]	5	1	4	N/A
	ThreadFix	[33] [36]	2	2	0	N/A
	Threagile	[39] [74]	2	1	1	> 500
	Threatspec	[74]	1	1	0	> 200
	ThreatPlaybook	[74]	1	1	0	> 200
	Pytm	[74]	1	1	0	> 800
VCS	GitHub	[27] [36] [38] [39] [55] [57] [60] [64] [67] [69] [78] [85] [98] [99] [100] [105] [140]	17	15	2	N/A
	GitLab	[39] [52] [55] [56] [64] [66] [70] [85] [106] [122] [141]	11	9	2	N/A
	Git	[38] [60] [67] [69] [70] [86] [92] [121]	8	5	3	> 48 k
	Subversion	[67] [70] [121]	3	2	1	N/A
	Mercurial	[69] [121]	2	1	1	N/A
Team collaboration	Slack	[36] [57] [86]	3	3	0	N/A
	HipChat	[36] [66]	2	2	0	N/A
	Trello	[57]	1	1	0	N/A
	Mattermost	[67]	1	1	0	N/A
SCA	OWASP Dependency Check	[36] [55] [67] [74] [83] [106] [118] [122]	8	5	3	> 5.7 k
	BlackDuck	[36] [55] [61] [75] [94] [110] [119]	7	4	3	N/A
	Dependency Track	[74]	1	0	1	> 2.2 K
Logging	Logstash	[33] [36] [61] [67] [91] [103] [106] [110] [132]	9	6	3	> 13.5 k
	Elasticsearch	[36] [61] [67] [91] [103] [106] [110] [132]	8	5	3	> 66.5 k

Table 8 (continued)

Practice	Tools	Mentioned in	#	WL	GL	GitHub Stars
	Splunk	[33] [36] [66] [79] [81] [110] [132]	7	3	4	N/A
	Graylog	[67]	1	1	0	> 7 k
Penetration testing	BurpSuite	[61] [83] [110] [112] [114] [142]	6	2	4	N/A
	Metasploit	[29] [117] [119] [132]	4	2	2	> 32 k
	Astra	[75]	1	0	1	> 2.4 k
Continuous deployment	Jenkins	[36] [39] [60] [63] [66] [67] [70] [82] [83] [92] [100] [105] [110] [121] [122]	15	10	5	> 22 k
	Spinnaker	[67] [110]	2	1	1	> 9 k
	GoCD	[67]	1	1	0	> 7 k
	ArgoCD	[110]	1	0	1	> 16 k
IAST	Synopsyis Seeker	[75] [94]	2	0	2	N/A
	OWASP Benchmark	[54]	1	1	0	> 600
Security policies	Open Policy Agent	[74]	1	0	1	> 8.9 k
	Kyverno	[74]	1	0	1	> 4.9 k
Configuration management	Ansible	[36] [55] [60] [61] [67] [76] [81] [82] [89] [91] [103] [117] [120] [136] [139]	15	12	3	> 60 k
	Puppet	[36] [55] [61] [67] [70] [76] [82] [139] [140]	9	7	2	> 7.2 k
	Chef	[36] [55] [61] [67] [70] [76] [139]	7	6	1	> 7.4 k
	Saltstack	[36] [67]	2	2	0	> 16 k
Secrets management	Hashicorp vault	[36] [55] [61] [67] [74] [78] [132]	7	4	3	> 29 k
	Blackbox	[36] [55] [67]	3	3	0	> 6 k
	Git Secrets	[67] [74]	2	1	1	> 11 k
	Docker secrets	[55] [67]	2	2	0	N/A
Continuous compliance	Whitesource	[55] [75] [79] [86] [89] [90] [92] [110] [118] [119]	10	3	7	N/A
	OpenSCAP	[36] [39] [74] [114] [118]	5	2	3	> 1.3 k
	Cloud custodian	[55] [67] [74]	3	2	1	> 5.2 k
	ESLint security	[55] [57] [74]	3	2	1	> 2.1 k
	Terrascan	[55] [74]	2	1	1	> 4.3 k
IaC	Terraform	[36] [55] [60] [67] [74] [76] [110] [120] [125] [136] [139]	11	8	3	> 40 k
RASP	Imperva RASP	[36] [76]	2	1	1	N/A
	OWASP open RASP	[61]	1	1	0	> 2.6 k
	Hdiv	[67]	1	1	0	> 200
Auditing	Inspec	[55] [60] [67] [74] [79] [122]	6	3	3	> 2.8 k
	Findsecbugs	[36] [54] [67] [74]	4	3	1	> 2.2 k
	Nmap	[57] [67] [119]	3	3	0	> 8.9 k
	Kubeaudit	[55] [74]	2	1	1	> 1.8 k
Continuous alerting	Pagerduty	[33] [36] [79]	3	2	1	N/A
	Elastalert	[67] [79]	2	1	1	> 8 k
	Alertlogic	[36] [76]	2	1	1	N/A

Table 8 (continued)

Practice	Tools	Mentioned in	#	WL	GL	GitHub Stars
	Opsgenie	[79]	1	0	1	N/A
WAF	Signal sciences	[36] [79]	2	1	1	N/A
Chaos engineering	Chaos monkey	[36] [67] [74] [76] [82]	5	3	2	> 14 k
	Chaos mesh	[74]	1	0	1	> 6 k
	Chaos Kkube	[74]	1	0	1	> 1.5 k
IAM	Keycloak	[61]	1	1	0	> 19 k
	Boundary	[33]	1	1	0	> 3.8 k
Incident management	Pagerduty	[33] [36] [79]	3	2	1	N/A
	New relic	[33] [91]	2	1	1	N/A
	Opsgenie	[79]	1	0	1	N/A

Identity Access Management (IAM), and Web Application Firewall (WAF).

Notably, Static Application Security Testing (SAST), Vulnerability Scanning, and Monitoring practices exhibited a rich array of tools documented in existing literature. The cumulative tally of tools meeting the inclusion criteria for listing totaled 114. Table 9 summarizes the number of tools for each practice and the average contributions from both WL and GL to identify them.

The contribution to the identification of tools for each practice is also represented in Fig. 11. Visually analyzing the graphic, it is possible to identify that there is a balance of contributions from both WL and GL and only some outliers. From the 27 practices with tools identified 10 of the practices had an average contribution from both WL and GL in the range of 40% to 60%, while 14 of the practices have a trending contribution from one of the literature types in the range of 61% to 99%. Only 3 practices are 100% explored by one of the literature types.

Of the identified tools, 63%, as illustrated in Fig. 12, enjoy community support on GitHub, with some projects being tracked by thousands of users.

Table 9 Tools per practice summary

PID	Nº of Tools	% WL AVG	% GL AVG
P1	13	40,42	59,58
P2	2	44,47	55,33
P3	4	39,58	60,42
P4	8	46,49	53,51
P5	11	54,88	45,12
P6	5	50	50
P7	8	62,92	37,08
P9	4	100	0
P10	4	54,17	45,83
P11	2	75	25
P12	1	72,73	27,27
P13	3	39,88	60,12
P15	5	69,84	30,16
P16	3	27,78	72,22
P17	5	50,67	49,33
P18	4	68,01	31,99
P19	2	50	50
P21	4	85,87	14,13
P25	3	83,33	16,67
P26	2	0	100
P28	3	38,39	61,61
P29	2	100	0
P30	4	76,79	23,21
P31	4	68,75	31,25
P34	1	50	50
P35	4	41,67	58,33
P38	3	20	80

7.3 Results discussion

In the results of my study, I discovered a balanced contribution from both WL and GL in the exploration of DevSecOps practices and tools. This balance highlights that both academia and industry are actively engaged in developing and refining DevSecOps methodologies, suggesting a growing convergence between theoretical frameworks and practical, real-world applications. This balanced representation from both sources indicates that while academia provides rigorous, evidence-based insights, the industry offers valuable, hands-on experience, collectively advancing the DevSecOps topic. This pairing reflects the importance of collaboration between these two domains in shaping the future of secure and efficient software development.

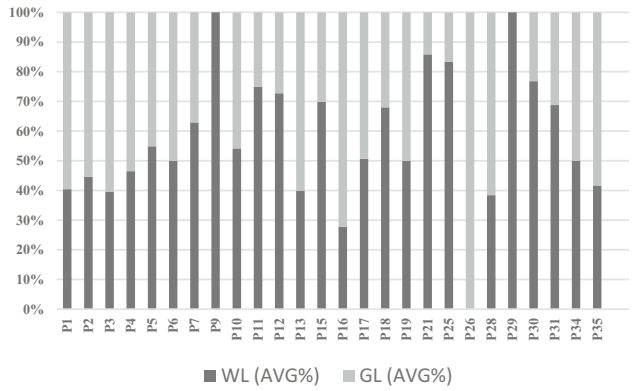

Fig. 11 Average contribution by literature type

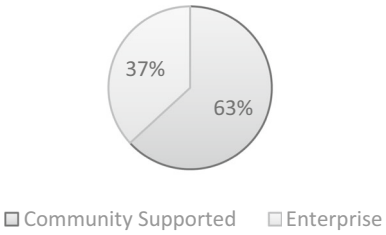

Fig. 12 Tools community supported vs enterprise

Additionally, in the last five years, the number of studies on DevSecOps more than doubled. The growing body of literature has become more diverse with researchers exploring several different perspectives on DevSecOps such as custom frameworks [85] [99] that address specific security challenges or in-depth research in practices recently related to DevSecOps such as RASP [104] [105]. DevSecOps managed to also establish itself as a centerpiece in the industry with many global organizations such as releasing annual reports on DevSecOps [96] [97] while other organizations regularly publish online articles on DevSecOps on their websites [72] [73]. The results of this study also indicate that DevSecOps continues to expand, in 2017, only seven practices were perceived as DevSecOps-related, with this study 32 additional practices were documented. The pool of identified tools, and the fact, that 63% of them are community-supported, suggest these practices are being adopted by several professionals. These results validate the growing importance of DevSecOps in the software development and cybersecurity landscape and that many of the automation within DevSecOps practices are key to addressing security challenges in fast-paced software development contexts.

7.4 Study implications

This study provides a solid foundation for further exploration in the DevSecOps domain. The combination of WL and GL strengthens the understanding of how DevSecOps is perceived from both theoretical and real-world perspectives.

Researchers can use this study as the groundwork for validating and expanding the identified practices, by empirically validating them across different contexts the results might confirm their relevance and effectiveness in those contexts. The resulting findings created a consolidated list of DevSecOps practices that can help organizations integrate security throughout their SDLC. The study proposes a division of practices into tiers based on the number of references found in the primary studies, this implies a gap analysis in literature, future studies can focus on these gaps and investigate why certain DevSecOps practices are underexplored or inconsistently mentioned in literature. The identification of specific tools opens the door for researchers to perform a more in-depth evaluation of how these tools are integrated into DevSecOps, professionals can use this study as a reference point for selecting tools that match their security needs. The practices and tools identified could serve as the building blocks for developing a standardized DevSecOps adoption framework. Future research could propose frameworks that combine the best practices and tools from WL and GL. In [143], a contemporary review study of this already expands on this by proposing a model named Challenge-Practice-Tool-Metric (CPTM) that can be used as the basis for DevSecOps framework design. The identified practices and tools can also be used by organizations to benchmark their current DevSecOps maturity level.

8 Conclusion & future work

The utilization of Multivocal Literature Review (MLR) in this study facilitated the identification of 103 pertinent primary studies. The discourse surrounding DevSecOps has exhibited a continuous expansion, with 47 of the primary studies pinpointed being published in 2023. Within the pool of studies, White Literature (WL) contributed 49% while Grey Literature (GL) contributed 51%. Analysis of the literature unearthed 39 distinct DevSecOps practices. Notably, there exists a convergence between WL and GL regarding the number of identified practices, with WL identifying 38 and GL identifying 39. Furthermore, there is alignment between academia and industry evidenced by the consistent number of mentions for each identified practice between WL and GL. The delineation of DevSecOps practices into three tiers (Tier 1 – Core, Tier 2 – Important, Tier 3 – Nice to have) reveals that core practices encompass Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), Automated Security Testing, Vulnerability Scanning, Continuous Monitoring, Continuous Integration, Threat Modeling, Shift Left Security, Collaboration, Continuous Deployment, and Code Review.

Initially, the identified primary studies referenced over 250 tools, prompting the establishment of selection criteria to streamline this abundance. Subsequently, a meticulous evaluation process culminated in identifying 114 pertinent tools tailored to 27 of the 39 DevSecOps practices. Notably, most of these tools boast robust support within the GitHub community, with some enjoying widespread adoption by thousands of users. The large pool of tools and community support suggests that DevSecOps practices are being widely adopted and have become central to addressing security challenges related to software development.

The initial aim of this study was to create a foundational body of knowledge into current DevSecOps practices and their associated tooling landscape, paving the way for deeper investigations into the practices themselves or the purpose of DevSecOps adoption frameworks through the combination of the practices and respective tools. It was possible, with this study, to identify several DevSecOps practices, define them, the tools that can be used to implement them and map them to DevSecOps flow.

The results validate that DevSecOps has been expanding in the last five years the body of literature has more than doubled. In 2017, only seven practices were perceived as DevSecOps-related [10], in this study, 32 additional practices were documented as related to DevSecOps.

Future research avenues may delve into the integration of these security practices within organizational frameworks and assess the current maturity levels thereof. Recently published studies already started to pursue these perspectives by measuring the importance of DevSecOps and how to integrate security practices into the DevOps pipeline [144], or by proposing custom tools [145] that support DevSecOps practices.

Moreover, there is potential for further inquiry into the identified tools, including surveys to gauge their popularity. Additionally, conducting a qualitative comparative analysis between tools utilized for implementing the same practice holds promise in elucidating their efficacy and suitability.

9 Limitations & threats to validity

This research is subject to several limitations related to the employed research methodology. MLRs have two major limitations: the first is associated with the quality and credibility of the sources and the second is selection bias.

MLRs offer a diverse pool of sources, both academic and non-academic literature, however, the latest are not peer-reviewed and may contain biases, lack of rigor, and transparency that could affect the validity of the synthesized conclusions.

Selection Bias is a common and major limitation of MLRs which sources are included or excluded. Although criteria for inclusion were defined it does not guarantee that during the study selection, certain types of studies or reports may have been unintentionally favored. This could result in an overrepresentation of a group of perspectives which limits the comprehensiveness of the review.

Moreover, MLRs also are affected by minor limitations, such as temporal bias, during the study selection it is possible to collect sources that may present outdated information that no longer applies to current practices or technologies. The inclusion of this literature might result in the synthetization of outdated perspectives. This represents a threat to the relevance of the review, especially in a fast-evolving field such as IT.

MLRs include both academic and non-academic literature which can result in a reduced heterogeneity in terms or concepts to describe the same phenomena which poses a challenge of synthesizing the findings across sources. This could lead to inconsistencies or misinterpretations in the review that might threaten the validity of the conclusions.

The final limitation of this study is the possibility of contextual differences due to the inclusion of industry reports as sources that might include perspectives specific to a region or industry and do generalize well across countries or sectors. These contextual differences might lead to conflicting findings or interpretations resulting in an incohesive understanding of the topic.

To mitigate these limitations a well-defined protocol was established to enhance the consistency of primary study selection and data extraction, thereby mitigating methodological threats, although it is admitted that there remains the possibility that relevant primary studies may have been inadvertently overlooked.

Acknowledgements This study was not funded by any institution.

Authors' Contributions L.P. wrote the main manuscript R.P. coordinated the research All authors reviewed the manuscript.

Funding Open access funding provided by FCT|FCCN (b-on).

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of Interests The authors declare no competing interests.

References

1. J. Humble and D. Farley, Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation, Addison-Wesley Professional, 2010.
2. B. Fitzgerald and K. Stol, "Continuous Software Engineering and beyond: Trends and Challenges," in *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*, Hyderabad, India, 2014.
3. Wettinger, J., Breitenbücher, U., Leymann, F.: DevOpSlang – Bridging the Gap between Development and Operations. Lect. Notes Comput. Sci. **8745**, 108–122 (2014)
4. S. K. Bang, S. Chung, Y. Choh and M. Dupuis, "A Grounded Theory Analysis of Modern Web Applications: Knowledge, Skills, and Abilities for DevOps," in *Proceedings of the 2nd Annual Conference on Research in Information Technology*, Orlando, Florida, USA, 2013.
5. A. Dyck, R. Penners and H. Lichter, "Towards Definitions for Release Engineering and DevOps," *2015 IEEE/ACM 3rd International Workshop on Release Engineering,* p. 3, 2015.
6. Labs, Puppet, "2021 State of DevOps Report," Puppet Labs, 2021. [Online]. Available: https://media.webteam.puppet.com/uploads/2021/07/Puppet-State-of-DevOps-Report-2021.pdf. [Accessed September 2022].
7. G. Kim, "Top 11 Things You Need to Know About DevOps," 2012. [Online]. Available: https://www.itrevolution.com/wp-content/uploads/2012/11/11things.pdf. [Accessed 5 October 2022].
8. Gitlab, "Global DevSecOps Survey Thriving in an insecure world," 2022. [Online]. Available: https://about.gitlab.com/developer-survey/. [Accessed 5 October 2022].
9. I. Security, Cost of a Data Breach Report 2022, 2022.
10. H. Myrbakken and R. Colomo-Palacios. 2017. DevSecOps: A Multivocal Literature Review. in *SPICE 2017. Communications in Computer and Information Science*. Springer International Publishing. cham
11. IEEE Computer Society Software & Systems Engineering Standards Committee, "IEEE Standard for DevOps: Building Reliable and Secure Systems Including Application Build, Package, and Deployment," *IEEE Std 2675–2021,* pp. 1–91, 2021.
12. Roche, J.: Adopting DevOps practices in quality assurance. Commun. ACM **56**, 38–43 (2013)
13. L. E. Lwakatare, P. Kuvaja and M. Oivo. Dimensions of devops in *Agile Processes in Software Engineering and Extreme Programming: 16th International Conference, XP 2015*. Helsinki. Finland.
14. Smeds, J., Nyborn, K., Pores, I.: DevOps: A Definition and Perceived Adoption Impediments. Agile Proce. Softw. Eng. Extreme Program. **212**, 166–177 (2015)
15. Kim, G., Humble, J., Debois, P., Willis, J.: The DevOps Handbook, Portland, OR. IT Revolution Press, USA (2016)
16. Amaro, R., Pereira, R., da Silva, M.M.: Capabilities and practices in DevOps: a multivocal literature review. EEE Trans. Softw. Eng. **49**, 883–901 (2023)
17. Google - Dora, "DevOps Research and Assessment," Google, [Online]. Available: https://cloud.google.com/architecture/devops. [Accessed 1 June 2023].
18. M. Virmani. 2015. Understanding DevOps & bridging the gap from continuous integration to continuous delivery, in *Fifth International Conference on the Innovative Computing Technology*, Galacia, Spain.
19. M. Senapathi, J. Buchan and H. Osman. 2018. DevOps Capabilities, Practices, and Challenges: Insights from a Case Study," in *In Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering*, New York, NY, USA
20. R. Jabbari, N. bin Ali, K. Petersen and B. Tanveer. 2016. What is DevOps? A Systematic Mapping Study on Definitions and Practices," in *Proceedings of the Scientific Workshop Proceedings of XP2016*, New York, NY, USA.
21. L. Riungu-Kalliosaari, S. Mäkinen, L. Lwakatare, J. Tiihonen and T. Männistö. 2016. DevOps Adoption Benefits and Challenges in Practice: A Case Study," *Product-Focused Software Process Improvement,* no Lecture Notes in Computer Science
22. Floris, E.: DevOps is Simply Interaction Between Development and Operations," in *First International Workshop DevOps 2018*. Chateau de Villebrumier, France (2019)
23. Sánchez-Gordón, M., Colomo-Palacios, R., "Characterizing DevOps Culture: A Systematic Literature Review.: Software Process Improvement and Capability Determination no Communications in Computer and Information Science. Springer International Publishing, Cham (2018)
24. New Relic, "New Relic: What is DevOps," New Relic, [Online]. Available: https://newrelic.com/devops/what-is-devops. [Accessed 25 June 2023].
25. Google, "State of DevOps," Google, 2022. [Online]. Available: https://cloud.google.com/devops/state-of-devops. [Accessed 2 July 2023].
26. Dynatrace, "State of DevOps," Dynatrace, 2023. [Online]. Available: https://www.dynatrace.com/monitoring/solutions/devops-report/?utm_source=google&utm_medium=cpc&utm_term=devops&utm_campaign=emea-south--devops-devops&utm_content=none&utm_campaign_id=15353438569&gclsrc=aw.ds&gclid=Cj0KCQjwkqSlBhDaARIsAFJANkhyCEp9G-PuHPWKX. [Accessed 2 July 2023].
27. A. Zaheeruddin and F. C. Shoba, "Integrating Security with DevSecOps: Techniques and Challenges," in *International Conference on Digitization*, Sharjah, UAE, 2019.
28. Carter, K.: Francois Raynaud on DevSecOps. IEEE Softw. **34**, 93–96 (2017)
29. N. Tomas, J. Li and H. Huang. 2019. An Empirical Study on Culture, Automation, Measurement, and Sharing of DevSecOps," in *International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, Oxford. UK.
30. Humble, J., Molesky, J.: Why enterprises must adopt devops to enable continuous delivery. Cutter IT Journal **24**, 6–12 (2011)
31. Davis, J., Daniels, K.: Effective DevOps. O'Reilly Media, USA (2016)
32. Rahman, A.A.U., Williams, L.: Software Security in DevOps: Synthesizing Practitioners' Perceptions and Practices," in *IEEE/ACM International Workshop on Continuous Software Evolution and Delivery*. Austin, Texas, USA (2016)
33. V. Mohan and L. B. Othmane, "SecDevOps: Is It a Marketing Buzzword? - Mapping Research on Security in DevOps," in *11th International Conference on Availability, Reliability and Security*, Salzburg, 2016.
34. Hüttermann, M.: DevOps for Developers, New York. Springer Science, USA (2012)
35. Wilson, G.: DevSecOps. Rethink Press, UK (2020)
36. Bird, J.: DevOpsSec, Sebastopol, CA. O'Reilly, USA (2016)
37. Candel, J.: Implementing DevSecOps with Docker and Kubernetes. BPB Publications, India (2022)
38. N. MacDonald and I. Head, "DevSecOps: How to seamlessly integrate security into devops," Gartner, 2016.
39. S. Dupont, G. Ginis, M. Malacario, C. Porretti, N. Maunero, C. Ponsard and P. Massonet, "Incremental Common Criteria Certification Processes using DevSecOps Practices," in *IEEE European Symposium on Security and Privacy Workshops*, 2021 IEEE European Symposium on Security and Privacy Workshops, 2021.

40. R. Mao, H. Zhang, Q. Dai, H. Huang, G. Rong, H. Shen, L. Chen and K. Lu. 2020.Preliminary Findings about DevSecOps from Grey Literature in *IEEE 20th International Conference on Software Quality, Reliability and Security*, Macau, China
41. Garousi, V., Felderer, M., Mäntylä, M.: Guidelines for including the grey literature and conducting multivocal literature reviews in software engineering. Inf. Softw. Technol. **106**, 101–121 (2017)
42. Kitchenham, B.: Procedures for performing systematic reviews." Keele. UK, Keele Univ. **33**, 1–26 (2004)
43. V. Garousi, M. Felderer and M. V. Mantyl. 2016. The Need for Multivocal Literature Reviews in Software Engineering: Complementing Systematic Literature Reviews with Grey Literature," in *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, Limerick, Ireland, 2016.
44. Patton, M.Q.: Towards Utility in Reviews of Multivocal Literatures. Rev. Educ. Res. **61**, 287–2982 (1991)
45. Hopewell, S., McDonald, S., Clarke, M., E. M.: Grey literature in meta-analyses of randomized trials of health care interventions.". Cochrane database syst rev (2017). https://doi.org/10.1002/14651858.MR000010.pub3
46. Tom, E., Aurum, A., Vidgen, R.: An exploration of technical debt. J. Syst. Softw. **86**, 1498–1516 (2013)
47. Garousi, V., Mäntylä, M.V.: When and what to automate in software testing? A multi-vocal literature review. Inf. Softw. Technol. **76**, 92–117 (2016)
48. Adams, J., Hillier-Brown, F.C., Moore, H.J., Lake, A.A., Araujo-Soares, V., White, M., Summerbell, C.: Searching and synthesising 'grey literature'and 'grey information'in public health: critical reflections on three case studies. Syst. rev. **5**, 1–11 (2016)
49. Ogawa, R.T., Malen, B.: Towards rigor in reviews of multivocal literatures: applying the exploratory case study method. Rev. Educ. Res. **61**, 265–286 (1991)
50. Adams, R.J., Smart, P., Huff, A.S.: Shades of grey: guidelines for working with the grey literature in systematic reviews for management and organizational studies. Int. j. manag. Rev. **19**(4), 432–454 (2017)
51. Zaydi, M., Nassereddine, B.: DevSecOps practices for an agile and secure IT service management. Int. J. Inf. Decis. Sci. **23**, 134–149 (2020)
52. R. N. Rajapakse, M. M. Zahedi, A. Babar and H. Shen. 2022. Challenges and solutions when adopting DevSecOps: A systematic review," *Information and Software Technology,,* vol. 141
53. L. Prates, "Prates23 - Github," 30 January 2023. [Online]. Available: https://github.com/Prates23/devsecops-practices-mlr. [Accessed 30 January 2023].
54. M. Christakis, T. Cottenier, A. Filieri, L. Luo, M. N. Mansur, L. Pike, N. Rosner, M. Schäf, A. Sengupta and W. Visser. 2022. Input splitting for cloud-based static application security testing platforms in Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, New York, NY, USA, 2022.
55. F. Angermeir, M. Voggenreiter, F. Moyón and D. Mendez. 2021. Enterprise-driven open source software: a case study on security automation," In Proceedings of the 43rd International Conference on Software Engineering: Software Engineering in Practice. 278-287
56. T. Rangnau, R. v. Buijtenen, F. Fransen and F. Turkmen. 2020. Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines," in *EEE 24th International Enterprise Distributed Object Computing Conference (EDOC)*, Eindhoven, Netherlands.
57. S. Chalishhafshejani, B. K. Pham and M. G. Jaatun. 2022. Automating Security in a Continuous Integration Pipeline," *Proceedings of the 7th International Conference on Internet of Things, Big Data and Security,* pp. 231 - 238.
58. J. A. Morales, T. P. Scanlon, A. Volkmann, Y. Yankel and H. Ysar. 2020. Security impacts of sub-optimal DevSecOps implementations in a highly regulated environment," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, New York, NY, USA.
59. Chernyshev, M., Baig, Z., Zeadally, S.: Cloud-native application security: risks, opportunities, and challenges in securing the evolving attack surface. Computing **54**, 47–57 (2021)
60. D. Granata, Rak, M. and G. Salzillo. 2022. MetaSEnD: A Security Enabled Development Life Cycle Meta-Model," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, Vienna, Austria.
61. P. Billawa, A. B. Tukaram, N. Ferreyra, J. Steghöfer, R. Scandariato and G. Simhandl. 2022. SoK: Security of Microservice Applications: A Practitioners' Perspective on Challenges and Best Practices," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, New York, NY, USA, 2022.
62. R. Desai and T. N. Nisha. 2021. Best Practices for Ensuring Security in DevOps: A Case Study Approach, *Journal of Physics: Conference Series,* vol. 1964, no. Advances in Computer Science Engineering, 2021.
63. F. Moyón, R. Soares, M. Pinto-Albuquerque, D. Mendez and K. Beckers. 2020. Integration of Security Standards in DevOps Pipelines: An Industry Case Study, *Lecture Notes in Computer Science.*12562, 2020.
64. R. Brasoveanu, Y. Karabulut and I. Pashchenko, "Security Maturity Self-Assessment Framework for Software Development Lifecycle," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, Vienna, Austria, 2022.
65. Woody, C., Chick, T., Reffett, A., Pavetti, S., Laughlin, R., Frye, B., Bandor, M.: DevSecOps Pipeline for Complex Software-Intensive Systems: Addressing Cybersecurity Challenges. J Syst, Cybern Inf: JSCI **18**(5), 31–36 (2020)
66. H. Yasar and S. E. Teplov, "DevSecOps In Embedded Systems: An Empirical Study Of Past Literature," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, Vienna, Austria, 2022.
67. R. Kumar and R. Goyal. 2021. Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC)," *Innovative Data Communication Technologies and Application.* Lecture Notes on Data Engineering and Communications Technologies. Springer Singapore. Singapore. 415-432
68. Hong, J.: Component analysis of DevOps and DevSecOps. J. Korea Conver. Soc. **10**, 47–53 (2019)
69. D. Ashenden and G. Ollis. 2020. Putting the Sec in DevSecOps: Using Social Practice Theory to Improve Secure Software Development. in *New Security Paradigms Workshop, NSPW 2020 - Post-Proceedings*, USA, 2020.
70. J. Nguyen and M. Dupuis. 2019. Closing the Feedback Loop Between UX Design, Software Development, Security Engineering, and Operations, in *The 20th Annual Conference on Information Technology Education*, Tacoma. WA. USA
71. Mohan, V., Othmane, L.B., Kres, A.: "BP: Security Concerns and Best Practices for Automation of Software Deployment Processes An Industrial Case Study," in *IEEE Cybersecurity Development (SecDev)*. MA, USA, Cambridge (2018)
72. AWS, "What is DevSecOps," AWS, [Online]. Available: https://aws.amazon.com/what-is/devsecops/?nc1=h_ls. [Accessed 13 June 2023].
73. Jfrog, "What is DevSecOps," Jfrog, [Online]. Available: https://jfrog.com/devops-tools/what-is-devsecops/. [Accessed 11 June 2023].

74. M. Sotti, "Ultimate DevSecOps Library," [Online]. Available: https://github.com/sottlmarek/DevSecOps. [Accessed 13 June 2023].
75. J. Hirschauer, "Top 10 Best Practices for DevSecOps," Harness, 11 May 2022. [Online]. Available: https://www.harness.io/blog/best-practices-devsecops. [Accessed 10 June 2023].
76. K. Zettler, "DevSecOps Tools," Atlassian, [Online]. Available: https://www.atlassian.com/devops/devops-tools/devsecops-tools. [Accessed 10 June 2023].
77. T. Scanlon and J. Morales, "Revelations from an Agile and DevSecOps Transformation in a Large Organization: An Experiential Case Study," in *ICSSP'22: 16th International Conference on Software and System*, Pittsburgh,PA, USA, 2022.
78. Aquasec, "DevSecOps Tools: 9 Ways to Integrate Security Into the SDLC," Aquasec, [Online]. Available: https://www.aquasec.com/cloud-native-academy/devsecops/devsecops-tools/. [Accessed 10 June 2023].
79. N. S. Gill, "A Guide to DevSecOps Tools and Continuous Security For an Enterprise," Xenonstack, 16 May 2023. [Online]. Available: https://www.xenonstack.com/blog/devsecops-tools. [Accessed 11 June 2023].
80. D. Weeks and D. Schleen, "DevSecOps Community Survey," Sonatype, 2020.
81. P. Kumari, "DevSecOps best practices and strategies you can't ignore," 26 June 2023. [Online]. Available: https://www.softwebsolutions.com/resources/devsecops-best-practices.html. [Accessed 21 December 2023].
82. V. Nastenko, "Integrating Security in DevOps: Best Practices, Tools, and Challenges," 3 April 2023. [Online]. Available: https://tech-stack.com/blog/integrating-security-in-devops-best-practices-tools-and-challenges/. [Accessed 20 December 2023].
83. Lombardi, F., Fanton, A.: From DevOps to DevSecOps is not enough. CyberDevOps: an extreme shifting-left architecture to bring cybersecurity within software security lifecycle pipeline. Software Qual. J. **31**(2), 619–654 (2023)
84. A. Irwin, "DevSecOps Best Practices Checklist," 9 October 2023. [Online]. Available: https://aptori.dev/blog/devsecops-best-practices-checklist. [Accessed 20 December 2023].
85. T. G. Espinha, U. Lechner and M. Pinto-Albuquerque, "Sifu - a cybersecurity awareness platform with challenge assessment and intelligent coach," *Cybersecurity,* vol. 3, 2020.
86. Z. Seremet and K. Rakić, "Best Approach to Security in Azure Devops," *DAAAM International Scientific Book,* pp. 223–230, 2021.
87. S. Ingalis, "10 Best DevSecOps Tools," ESecurityPlanet, 17 May 2022. [Online]. Available: https://www.esecurityplanet.com/products/devsecops-tools/. [Accessed 10 June 2023].
88. Fossa, "5 Must-Have DevSecOps Tools," Fossa, [Online]. Available: https://fossa.com/blog/must-have-devsecops-tools/. [Accessed 13 June 2023].
89. S. Manjaly, "The Top 10 Best DevSecOps Tools for 2023," Invgate, 29 December 2022. [Online]. Available: https://blog.invgate.com/devsecops-tools. [Accessed 10 Jube 2023].
90. S. Pickard, "10 Best DevSecOps Tools," PCWDLD, [Online]. Available: https://www.pcwdld.com/best-devsecops-tools/. [Accessed 11 June 2023].
91. Tigera, "16 Amazing DevSecOps Tools to Shift Your Security Left," Tigera, [Online]. Available: https://www.tigera.io/learn/guides/devsecops/devsecops-tools/. [Accessed 11 June 2023].
92. E. Corrales, "Best DevOps and DevSecOps Tools," Developer, 24 May 2022. [Online]. Available: https://www.developer.com/project-management/devsecops-tools/. [Accessed 11 June 2023].
93. Aquasec, "SecDevOps in Your Organization: A Practical Guide," [Online]. Available: https://www.aquasec.com/cloud-native-academy/devsecops/secdevops/. [Accessed 14 10 2023].
94. Synopsys, "What is DevSecOps," [Online]. Available: https://www.synopsys.com/glossary/what-is-devsecops.html. [Accessed 13 June 2023].
95. Gitlab, "Mapping the DevSecOps Landscape," Gitlab, 2020.
96. Gitlab, "A maturing DevSecOps landscape," Gitlab, 2021.
97. Contrast Security, "The State Of DevSecOps Report," Contrast Security, 2021.
98. S. Nocera, S. Romano, R. Francese and G. Scanniello. 2023. A Large-scale Fine-grained Empirical Study on Security Concerns in Open-source Software, in *9th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Durres,Albania,.
99. M. A. Aljohani and S. S. Alqahtani. 2023. A Unified Framework for Automating Software Security Analysis in DevSecOps," in *International Conference on Smart Computing and Application (ICSCA)*, Hail, Saudi Arabia
100. L. A. Nikolov and A. P. Aleksieva-Petrova. 2023. Action Research on the DevSecOps Pipeline in *International Scientific Conference on Computer Science (COMSCI)*, Sozopol, Bulgaria, 2023.
101. A. D. Tran, K. Yskout and W. Joosen. 2023. AndrAS: Automated Attack Surface Extraction for Android Applications, in *EEE 23rd International Conference on Software Quality, Reliability, and Security (QRS)*, Chiang Mai, Thailand, 2023.
102. M. Marandi, A. Bertia and S. Silas. 2023. Implementing and Automating Security Scanning to a DevSecOps CI/CD Pipeline in *World Conference on Communication & Computing (WCONF)*, Raipur, India, 2023.
103. Basinya, E.A., Malyshev, E.A.: "The Creation and Integration of the Technological Workflow for Software and Hardware-Software Development," in *EEE XVI International Scientific and Technical Conference Actual Problems of Electronic Instrument Engineering (APEIE)*. Russian Federation, Novosibirsk (2023)
104. M. Ji, M. Yin and Y. H. Zhou. 2023. Application of static taint analysis in RASP protection strategy," in *Proceedings of the 2022 International Conference on Cyber Security (CSW '22)*, New York, NY,USA, 2023.
105. P. Le-Thanh, T. Le-Anh and Q. Le-Trung. 2023. Research and Development of a Smart Solution for RuntimeWeb Application Self-Protection," in *Proceedings of the 12th International Symposium on Information and Communication Technology (SOICT '23)*, New York, NY, USA
106. W.-T. Lee and Z.-W. Liu. 2023. Microservices-based DevSecOps Platform using Pipeline and Open Source Software," *Journal of Information Science and Engineering,* vol. 39, pp. 1117–1128
107. R. Kimani. 2023. Implementing DevSecOps Best Practices," [Online]. Available: https://thenewstack.io/devsecops-implementation-best-practices/. [Accessed 20 December 2023].
108. Microsoft. 2023. What is DevSecOps? [Online]. Available: https://www.microsoft.com/en-us/security/business/security-101/what-is-devsecops. [Accessed 20 December 2023].
109. Paloalto Networks. 2023. How to Transition from DevOps to DevSecOps, [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/devops-to-devsecops. [Accessed 20 December 2023].
110. N. S. Vardhan. 2023. "What is DevSecOps," 19 June 2023. [Online]. Available: https://www.opsmx.com/blog/what-is-devsecops/. [Accessed 20 December 2023].
111. GuardRails. 2023. "Maximizing Security with DevSecOps: Top 5 Mistakes to Avoid," 26 July 2023. [Online]. Available: https://www.guardrails.io/blog/maximizing-security-with-devsecops-top-5-mistakes-to-avoid/. [Accessed 20 December 2023].
112. M. Kopacki. 2023. 5 DevSecOps best practices that can help you build secure applications, faster," 30 October 2023. [Online]. Available: https://www.kellton.com/kellton-tech-blog/

devsecops-best-practices-that-helps-build-secure-applications. [Accessed 20 December 2023].
113. A. Raizada. 2023. A Deep Dive into DevSecOps Best Practices for Secure and Efficient Software Delivery," 2023. [Online]. Available: https://copperdigital.com/blog/devsecops-best-practices-secure-software-delivery/. [Accessed 20 December 2023].
114. A. Spasojevic. 2023. What is DevSecOps," 2023. [Online]. Available: https://phoenixnap.com/blog/what-is-devsecops. [Accessed 21 December 2023].
115. R. Kolodiy. 2023. DevOps vs DevSecOps," 27 November 2023. [Online]. Available: https://www.techmagic.co/blog/devops-vs-devsecops/. [Accessed 21 December 2023].
116. J. Peterson. 2023. Mastering SDLC Security: Best Practices, DevSecOps, and ASPM," 12 December 2023. [Online]. Available: https://cycode.com/blog/mastering-sdlc-security-best-practices/. [Accessed 22 December 2023].
117. A. Prasad, "DevSecOps in Modern Software Development," 26 October 2023. [Online]. Available: https://semaphoreci.com/blog/devsecops. [Accessed 22 December 2023].
118. R. Singh, "Top DevSecOps Tools for 2023: Open Source Solutions for Enterprises," 16 May 2023. [Online]. Available: https://ranjaniitian.medium.com/top-devsecops-tools-for-2023-open-source-solutions-for-enterprises-7c146f80b325. [Accessed 22 December 2023].
119. Zhou, X., Mao, R., Zhang, H.: Revisit security in the era of DevOps: An evidence-based inquiry into DevSecOps industry. IET Software **17**(4), 435–454 (2023)
120. L. Verderame, L. Caviglione, R. Carbone and A. Merlo. 2023. SecCo: Automated Services to Secure Containers in the DevOps Paradigm," in *Proceedings of the 2023 International Conference on Research in Adaptive and Convergent Systems (RACS '23)*, New York, NY, USA
121. Gitlab, "Global Developer Report: DevSecOps," Gitlab, 2019.
122. E. DevOps, "Best DevSecOps Tools for 2022 | Open Source Enterprise," 29 June 2022. [Online]. Available: https://medium.com/@devops.ent/best-devsecops-tools-for-2022-open-source-enterprise-a5d13455b90. [Accessed 11 June 2023].
123. C. Weir, S. Migues, M. Ware and L. Williams. 2021. Infiltrating Security into Development: Exploring theWorld's Largest Software Security Study," in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Athens, Greece
124. P. Thantharate and A. T, "GeneticSecOps: Harnessing Heuristic Genetic Algorithms for Automated Security Testing and Vulnerability Detection in DevSecOps," in *6th International Conference on Contemporary Computing and Informatics (IC3I)*, Gautam Buddha Nagar, India, 2023.
125. Vakhula, O., Opirskyy, I., Mykhaylova, O.: Research on Security Challenges in Cloud Environments and Solutions based on the "security-as-Code" Approach in CEUR Workshop Proceedings, p. 2023. Kyiv, Ukraine (2023)
126. M. Thevarmannil, "Top 10 DevSecOps Best Practices for 2023," 1 July 2023. [Online]. Available: https://www.practical-devsecops.com/devsecops-best-practices/. [Accessed 20 December 2023].
127. N. Rini, "DevSecOps Best Practices to Implement," 12 October 2023. [Online]. Available: https://www.techrepublic.com/article/devsecops-best-practices/. [Accessed 20 December 2023].
128. D. Hopper, "7 Essential DevSecOps Best Practices Every Development Team Should Implement," 19 May 2023. [Online]. Available: https://www.mayhem.security/blog/7-essential-devsecops-best-practices-every-development-team-should-implement. [Accessed 20 December 2023].
129. G. Andrews, "Top 8 DevSecOps Best Practices," 2023. [Online]. Available: https://www.akto.io/blog/top-8-devsecops-best-practices. [Accessed 20 December 2023].
130. Everable, "Five Core Capabilities for Every DevSecOps Environment," Everable, 3 August 2022. [Online]. Available: https://www.everable.com/blog/five-core-capabilities-for-every-devsecops-environment. [Accessed 11 June 2023].
131. J. Marsal, "What is DevSecOps? And what you need to do it well," 19 January 2023. [Online]. Available: https://www.dynatrace.com/news/blog/what-is-devsecops/. [Accessed 20 December 2023].
132. A. Baig, "DevOps - 15 DevSecOps Best Practices," DevOps.com, 15 April 2022. [Online]. Available: https://devops.com/15-devsecops-best-practices/. [Accessed 13 June 2023].
133. Flora, J., Gonçalves, P., Antunes, N.: Intrusion Detection for Scalable and Elastic Microservice Applications," in *IEEE 28th Pacific Rim International Symposium on Dependable Computing (PRDC)*. Singapore, Singapore (2023)
134. Simopoulos, D., Wolf, A.: "Overcoming New Technologies Challenges in IoT Security Labs: Strategies for Effective Adaptation," in *IFIP Networking Conference (IFIP Networking)*. Barcelone, Spain (2023)
135. J. A. Morales and H. Yasar, "Experiences with Secure Pipelines in Highly Regulated Environments," In: *Proceedings of the 18th International Conference on Availability, Reliability and Security*, New York, NY, USA, 2023.
136. E. Villanueva, I. Torres, E. Osaba, S. Canzoneri, A. Franchini and L. Blasi, "PIACERE Integrated Development Environment. In: *Proceedings of the 3rd Eclipse Security, AI, Architecture and Modelling Conference on Cloud to Edge Continuum (ESAAM '23)*, New York, NY, USA, 2023.
137. M. Sánchez-Gordón and R. Colomo-Palacios, "Security as Culture: A Systematic Literature Review of DevSecOps," In: *ICSEW'20 Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, Seoul, Republic of Korea, 2020.
138. Codacy, "What Is DevSecOps? Shift Security Left in Your DevOps Lifecycle," 7 November 2023. [Online]. Available: https://blog.codacy.com/what-is-devsecops. [Accessed 21 December 2023].
139. P. R. Reddy Konala and V. B. D. Kumar. 2023. SoK: Static Configuration Analysis in Infrastructure as Code Scripts, in *IEEE International Conference on Cyber Security and Resilience (CSR)*. Venice. Italy.
140. Rahman, A., Parnin, C.: Detecting and characterizing propagation of security weaknesses in puppet-based infrastructure management. IEEE Trans. Softw. Eng. **49**, 3536–3553 (2023)
141. Trustradius, "DevSecOps Tools," Trustradius, [Online]. Available: https://www.trustradius.com/devsecops. [Accessed 20 11 2023].
142. M. Thevarmannil, "Best DevSecOps Tools List for 2024," 27 April 2023. [Online]. Available: https://www.practical-devsecops.com/devsecops-tools/. [Accessed 22 December 2023].
143. Zhao, X., Clear, T., Lal, R.: Identifying the primary dimensions of DevSecOps: A multi-vocal literature review. J. Syst. Softw. **22**, 112063 (2024)
144. Abiona, O., Oladapo, O., Modupe, O., Oyeniran, O., Adewusi, A., Komolafe, A.: The emergence and importance of DevSecOps: Integrating and reviewing security practices within the DevOps pipeline. World J. Adv. Eng. Techn. Sci. **11**, 127–133 (2024)
145. N. Bernardino, B. Sequeira, E. Piza, F. Henriques, F. Neves and C. I. Reis. 2024. Enhancing DevSecOps: Three Custom Tools for Continuous Security," in *IEEE 11th International Conference on Cyber Security and Cloud Computing (CSCloud)*. Shanghai. China