electronics

MDPI

Systematic Review

Holding on to Compliance While Adopting DevSecOps: An SLR

Xhesika Ramaj [1], Mary Sánchez-Gordón [1,*], Vasileios Gkioulos [2], Sabarathinam Chockalingam [3] and Ricardo Colomo-Palacios [1]

[1] Department of Computer Science and Communication, Faculty of Computer Sciences, Engineering and Economics, Østfold University College, 1757 Halden, Norway
[2] Department of Information Security and Communication Technology, Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology, 2802 Gjøvik, Norway
[3] Department of Risk, Safety and Security, Institute for Energy Technology, 1777 Halden, Norway
* Correspondence: mary.sanchez-gordon@hiof.no

Abstract: The software industry has witnessed a growing interest in DevSecOps due to the premises of integrating security in the software development lifecycle. However, security compliance cannot be disregarded, given the importance of adherence to regulations, laws, industry standards, and frameworks. This study aims to provide an overview of compliance aspects in the context of DevSecOps and explore how compliance is ensured. Furthermore, this study reveals the trends of compliance according to the extant literature and identifies potential directions for further research in this context. Therefore, we carried out a systematic literature review on the integration of compliance aspects in DevSecOps, which rigorously followed the guidelines proposed by Kitchenham and Charters. We found 934 articles related to the topic by searching five bibliographic databases (163) and Google Scholar (771). Through a rigorous selection process, we selected 15 papers as primary studies. Then, we identified the compliance aspects of DevSecOps and grouped them into three main categories: *compliance initiation*, *compliance management*, and *compliance technicalities*. We observed a low number of studies; therefore, we encourage further efforts into the exploration of compliance aspects, their automated integration, and the development of metrics to evaluate such a process in the context of DevSecOps.

Keywords: DevSecOps; compliance aspects; security compliance; systematic literature review

Citation: Ramaj, X.; Sánchez-Gordón, M.; Gkioulos, V.; Chockalingam, S.; Colomo-Palacios, R. Holding on to Compliance While Adopting DevSecOps: An SLR. *Electronics* **2022**, *11*, 3707. https://doi.org/10.3390/electronics11223707

Academic Editor: Juan M. Corchado

Received: 19 September 2022
Accepted: 6 November 2022
Published: 12 November 2022

1. Introduction

Compliance with industry standards and best practices is a focal point of information security within a DevSecOps environment [1], whether the organization is operating with medical systems, charging and billing systems, industrial control systems, cloud-based, or other critical systems. Regional and international markets require that products, production processes, services, or test methods adhere to standards and other legal requirements, such as laws and regulations. The European Union, for instance, adopted the General Data Protection Regulation (GDPR) in 2016 [2]. The United States does not have any laws like GDPR [3], but some states have introduced their own regulations, such as the California Consumer Privacy Act (CCPA) [4], the Massachusetts Data Protection Act [5], the Federal Information Security Management (FISMA) [6], the Family Educational Rights and Privacy Act (FERPA) [7], the Sarbanes–Oxley Act (SOX) [8], the Gramm–Leach–Bliley Act (GDLA) [9], and the Health Insurance Portability and Accountability Act (HIPPA) [10].

There are also national associations for standardization, such as the British Standards Institution (BSI) or the National Institute of Standards and Technology (NIST) in the United States [11]. Likewise, other non-governmental organizations made up of national standards bodies exist. Three well-known international standards organizations are the International Organization for Standardization (ISO), the International Electrotechnical Commission (IEC), and the International Telecommunication Union (ITU).

Besides the national efforts, an increasing number of industries have created their own legal or contractual obligations for compliance with industry standards; for instance, the Payment Card Industry Data Security Standard (PCI DSS) [12] in the finance industry. In the technology sector, there are other industry standards and frameworks—such as the Capability Maturity Model Integration (CMMI) [13], COBIT technical benchmarks [14], and SANS Top 20 Critical Security Controls [15]—that map their own objectives with the results of the service providers while providing guidance over their activities.

When an organization fails to comply with legal requirements like GDPR, it faces penalties and fines. For instance, Art. 83(5) of GDPR [2] allows imposing fines of up to 20 million euros for severe violations, or up to 4% of their total global turnover of the preceding fiscal year in the case of an undertaking. Meta is a recent example of being subject to a 17 million euro fine for breaching EU data privacy laws [16]. There is an even broader database of recipients fined for GDPR violations [17]. Nevertheless, such fines are not the only reason why compliance is becoming a trending topic.

As our society is moving forward to cyberspace, compliance is becoming the responsibility of the organizations that offer their services and products in such environments, e.g., online shopping, e-banking, or online healthcare systems. The responsibility goes beyond protecting what the organizations are delivering as their final product. It also includes the data protection of the customers while interacting with their systems and accessing their websites [3]. Thus, web accessibility becomes a subject of compliance as well, given the aim of turning the Internet into an all-encompassing space. In this regard, the World Wide Web Consortium (W3C) has created the Web Content Accessibility Guidelines (WCAG) [18], which have been adopted by some countries.

In this context, compliance methods and technologies aim to help organizations adhere to government regulations, industry standards, and also corporate policies. Moreover, automated compliance simplifies audits and avoids costly regulatory fines and lawsuits that certain regulations obtrude. The automatization has been validated and reported through approaches like DevOps. With an industry that is widely embracing and adopting DevOps [19], the integration of security processes in pipelines is turned into a must, which has resulted in DevSecOps. These pipelines not only support the detection of any security vulnerabilities, but also contribute to the adherence to standards and regulations [20]. In addition to the protection of customers that the adherence to regulations or the state of being compliant offers, it protects the product provider as well. Protection is provided in terms of concerns such as intellectual property, licensing issues, and customers' data privacy [3].

DevSecOps also opens the possibility of working inside the same space of both developers and operation teams, with other teams such as customer support, legal, and security [3]. In this situation, even though compliance issues are managed by other teams, a DevOps engineer is responsible for both applying recommended changes by the security experts and running compliance tests [3]. In addition, DevSecOps provides a set of methods and tools in support of compliance issues [3] which could reduce both the cost and time the teams spend [21].

Besides the team that manages the compliance issues, all the other teams should enhance their knowledge on the topic of compliance [3]. For instance, DevOps Audit Defense Toolkit [22] is a guide that leads the way for the involved teams towards auditing and compliance issues. There are also other initiatives and projects which work on integrating security and compliance into DevOps, such as AWS Labs [23], DevSecOps [24], Open DevSecOps [25], and Rugged DevOps [26]. The security checks from third-party auditors and penetration testers could be a source of information as well [3], and its possession is never of redundancy. It helps to enhance maintaining the control over the processes and prevent certain circumstances right on time.

Despite the competitive advantage and other benefits that a compliant DevSecOps environment offers, the research efforts on this topic require further consideration. Although some secondary studies about DevSecOps exist (see related work section), to the best of our knowledge, an overview of the state-of-the-art aspects of compliance in DevSecOps is

not available in the extant literature. Therefore, we aim to bridge this gap by reviewing research in this field. For this study, we follow the guidelines for systematic literature reviews (SLR) in software engineering proposed by Kitchenham and Charters [27].

The structure of this remaining article is as follows: Section 2 is focused on compliance and DevSecOps. Section 3 presents related works, while Section 4 describes the research approach, which is an SLR. Section 5 reports the findings of this SLR, while Section 6 discusses them in light of potential future research. Finally, Section 7 presents conclusions and future research directions.

2. Background

In this section, the authors provide the background information on two key concepts related to this study: *Compliance* and *DevSecOps*.

2.1. Compliance

Compliance is defined by the International Software Testing Qualifications Board (ISTQB) as adherence of a work product to standards, conventions, or regulations in laws and similar prescriptions [28]. By reviewing what is new in ISO 37301:2021 Compliance management systems, Marlow [29] recognized compliance as a continuous process aiming to monitor the organization to meet its obligations. The obligations of an organization are linked to its operational activities and the products that it offers.

Dealing with big data makes organizations subject to different issues related to privacy and security of information [30]. For instance, their web-based operations and products set the need for a consent mechanism, which affects web accessibility and becomes the subject of compliance issues [31]. In order to support privacy, security, and accessibility issues, a framework of regulations is needed as part of product requirements. On the other hand, developers and other involved teams have to bear in mind such requirements while designing, developing, testing, and deploying their products. This whole process can provide a chain of evidence to support the compliance of their products.

As mentioned above, there are many compliance and legal requirements all around the world, which regulate business functioning in different industries and countries. There are regulations, standards, and frameworks that provide guidelines and rules on security requirements. However, auditing and compliance activities could be automated to facilitate the compliance process. Their automation supports the integration into the DevSecOps pipelines and continuous performance throughout the whole software development life cycle (SDLC) [32]. Not only will the automated procedures decrease human interaction and the possibility of errors from manual work, but they will also impact the time and costs as well, driving processes towards efficiency [21].

2.1.1. Security Compliance

In order to understand security compliance, the concept of security needs to be defined as well, besides that of compliance explained above. Julisch [33] defines security as the state of being safe from potential threats. Ullah et al. [21] refer to security not only as a state of being safe but as a whole mechanism that contributes towards achieving such a state. Different tools contribute to the mechanism that incorporates security into a pipeline and performs automated checks. Casagni et al. [34] question if these checks are truly security-focused rather than compliance-focused. We posit that every organization should keep both in mind.

While defining both security and compliance can create a panorama of what security compliance is, the definition of security compliance itself differs from several studies. Julisch [33] defines security compliance as a conformance process with a given set of security requirements. Thus, security compliance makes certain that the system has the required level of security in place [21]. These requirements could be set on the use and configuration of different security mechanisms such as virus scanners, firewalls, patching, penetration testing, and so on. Daud et al. [35] define security compliance as the core activities to be

adhered to by employees in meeting security objectives, while for Ullah et al. [21] security compliance is a state of compliance within a given set of security requirements.

Consistent enforcement is vital to security compliance [36]. This fact indicates that all groups within an organization must be subject to compliance requirements checks where these requirements are embedded through policies and other related documents, such as standards and legislation. ISO/IEC 27001 [37] contains a dedicated annex about compliance with legal and contractual requirements. The objective is to avoid breaches of legal, statutory, regulatory, or contractual obligations related to information security and for any security requirements [38].

Security compliance not only shows how the security checks within an organization adhere to security standards but also demonstrates security reliability and trust for customers and partners [39]. Trust maintenance further assists in consolidating the brand name and assuring business continuity [3]. On the other hand, security compliance provides an overview of a security assurance program, but it does not specify which security approach would fit a company best [39]. That relies on the company itself, the activities the company is performing, and the products it is delivering.

2.1.2. Compliance as Code

While thinking about compliance in a DevSecOps environment, falling into the automation philosophy of DevSecOps does not seem to be avoidable. Automation is facilitated through codification by configuring resources into code, which is the first step for enabling compliance with security policies [40]. This process is supported by a set of automated tools and practices. Such tools allow embedding compliance requirements in the software product delivery pipeline. In turn, the embedded requirements will ensure that any code deployed is compliant with the industry regulations [41].

Many of the attacks mentioned in the OWASP Top 10 list [42] happen to be code injection attacks. The implementation of DevSecOps and expressing the configuration of the running system in the code itself could fix some of the targeted vulnerabilities of code injection attacks [43]. Furthermore, the automation of the security checks and processes allows exploratory testing and thinking beyond the manual tasks' performance [34]. Code compliance testing can be accomplished by integrating multiple test engines, such as fortify and appscan [44].

2.2. DevSecOps

DevSecOps was first introduced as a concept by Gartner [45] in 2012, with the aim to incorporate security into DevOps. The acronym stands for DEVelopment, SECurity, and OPerationS. Since then, several definitions have denoted DevSecOps as a concept.

DevSecOps is defined as an IT-processes-related set of practices that have incorporated a security approach to shorten development time and ensure software quality [46]. Software quality is measured in terms of code, security, and delivery mechanisms [47]. Other sources define DevSecOps as a paradigm for integrating the software development and operation processes, considering security requirements [48], a DevOps methodology for security [49], and DevOps embedded with security controls [50]. Yasar [51] defines DevSecOps as "DevOps done right" because mature DevOps practices speeding up the recovery process in case of any identified problem. The maturity of these practices relies on the fact that they are constantly tested, deployed, and validated, making sure that the software meets every requirement.

Despite the variety of definitions, security principles and practices lie at the heart of DevSecOps. The security principles and practices are integrated into DevOps through continuous communication, collaboration, and integration between development, operations, and security teams [52]. Furthermore, DevSecOps brings together other teams, such as customer support, legal, and marketing, to discuss the project's plans [3].

In light of the goal of DevSecOps to safeguard applications from potential threats [53], security is integrated throughout the whole SDLC and at each layer of the DevOps

pipeline [39]. Security is also integrated into DevSecOps as part of the culture. According to Casagni et al. [34], practitioners claim that an average developer might have security as an incorporated concept in his/her work culture, but there is no certainty that this is applied to compliance as well.

3. Related Work

Our work on this study was initiated with a preliminary search on the topic in order to identify the existing related studies. Despite a few secondary studies on DevSecOps that exist, none of them discuss in-depth compliance aspects and the way compliance is ensured in a DevSecOps environment. As the focus of our study is compliance instead of discussing it as one of the aspects of DevSecOps, we aim to identify compliance aspects. In this way, we extend the contributions list of existing studies with new insights.

Mohan and Ben Othmane [54] carried out a mapping study about security in DevOps that focuses on the main aspects of the DevSecOps trend. Compliance is one of the identified aspects besides the DevSecOps definition, security best practices, process automation, tools for DevSecOps, software configuration, team collaboration, availability of activity data, and information secrecy. Lie et al. [55] carried out a Multivocal Literature Review (MLR) to identify the regulatory compliance of DevOps in a regulated medical device context. They concluded that DevOps for such a context is a highly appealing approach and noted specific contradictions between DevOps and IEC 62304 medical device software. Lee et al. [56] conducted an SLR on the challenges of DevSecOps in small and medium-sized enterprises (SMEs). They concluded that SMEs could enhance security and compliance management without impacting efficiency by using DevSecOps. Moreover, they list some free license tools to be used in a DevSecOps environment.

Rajapakse et al. [57] conducted an SLR to identify 21 challenges that organizations face when adopting DevSecOps and provide possible solutions. They mention the inability to fully automate traditionally manual security practices to integrate into DevSecOps as one of the challenges related to practice; particularly, compliance (testing) practices were highlighted as one of its key points. As one potential solution, they proposed to adopt standards, policies, models, and service level agreements (SLAs) into testable criteria. Furthermore, they identified continuous automated compliance testing mechanisms as one of the essential features of the proposed solution. We aim to identify other compliance aspects—beside compliance testing—and the proposed approaches to ensure them.

4. Method

This study was carried out based on the guidelines for systematic literature reviews in software engineering proposed by Kitchenham and Charters [27]. This section presents the research aim, questions, and the search strategy followed. Furthermore, we describe the selection process of the relevant literature and the data extraction process.

4.1. Research Goal and Questions

This SLR is conducted keeping the following three specific objectives in mind: (i) identify the compliance aspects reported in the scientific literature, (ii) reveal how compliance is ensured in DevSecOps, and (iii) identify the literature trends of compliance in DevSecOps and identify research gaps in this field. Based on the above-mentioned goals, we list the research questions (RQs) as follows:

RQ1: Which compliance aspects are related to DevSecOps?

RQ2: How is compliance ensured in a DevSecOps environment?

RQ3: What are the emerging compliance trends in the DevSecOps literature?

Each of the above questions holds a clear motivation behind it. First, compliance is a very important issue that all organizations must consider regardless of the domain in which they operate; for instance, healthcare (medical information systems), financial (charging and billing systems), industrial (industrial control systems), or any other industry operating with critical systems. To follow up on the predefined compliance requirements,

the understanding of the compliance concept itself brings forward further considerations of its diverse aspects. Answering RQ1 will shed light on the compliance aspects that have been proposed in the literature.

Second, as the software industry is moving towards DevOps and DevSecOps recently, organizations must make sure that their activities address compliance requirements. Answering RQ2 will shed light on how organizations can adhere to standards and regulations in order to ensure compliance while adopting DevSecOps in their working environments.

Last, in order to put forward new insights and suggest future improvements, it is necessary to consider the current developments. The emerging compliance trends in the DevSecOps literature provide an overview of the body of knowledge in this field over time. Together with the findings from RQ1 and RQ2, this can lead interested researchers and practitioners towards further developments in this topic.

4.2. Search Strategy

The search strategy of this study was developed following the guidelines provided by Kitchenham and Charters [27]. Our search string contains both *Compliance* and *DevSecOps* as keywords. Thus, the search aims to include all the studies which discuss both concepts. Since there are other terms that can be used interchangeably for DevSecOps, e.g., *SecDevOps, DevOpsSec, Secure DevOps, or Rugged DevOps* [52], our final search string was formulated as shown below:

> *"compliance" AND ("devsecops" OR "secdevops" OR "devopssec" OR "secure devops" OR "rugged devops")*

In order to get the most out of the available literature about this topic, the search was conducted on five major databases: ACM Digital Library, IEEE Xplore, Wiley Online Library, SpringerLink, and ScienceDirect. We ran the search in December 2021 on those databases estimated as enough for this study. The number of sources available was 166, and these were the group that went under further inspection. Then, a duplication-identifying process followed up to retrieve the basic information from each source. We identified a total of 10 duplicates from the results which were excluded. There were also 11 studies in the German language that were excluded. Next, the first selection of the studies was conducted by reviewing the title, keywords, and abstract of each paper. This selection process resulted in 23 papers being selected. Table 1 shows the papers selected as relevant after performing this step.

Table 1. The results from each database.

Database	Initial Search	Remove Duplicates and Other Languages	Read Title, Keywords, and Abstract	Full Text Reading
ACM Digital Library	19	19	11	1
IEEE Xplore	3	3	3	1
ScienceDirect	15	15	6	2
Springer Link	103	82	3	2
Wiley	26	26	0	0
Total studies	166	145	23	**6**
Snowballing				2
Google Scholar	771		27	7
Primary studies				**15**

These 23 papers required further full-text consideration in order to make an informed decision. During full-text reading, we identified studies that mention compliance or DevSecOps, but not as their primary focus. Their relevance was assessed by their ability to provide empirical evidence to address at least one of our RQs. Notes were kept for every included or excluded paper. The first author conducted a thorough review using an excel

sheet with detailed information about the selected studies. The final list of selected studies from the first author ([1]) was validated by the other authors as well. Two studies ([43,58]) initially excluded by the first author were added in the final list after a discussion. At the end of the process, a total of six studies were included.

Due to the scarce number of studies, a snowballing process was performed on the identified secondary studies [54]. After a thorough review process on the list of references of each secondary study, two studies ([59,60]) were identified as relevant. This raised the total number of papers selected to eight. Moreover, the search string was executed in Google Scholar in February 2022. As a result, seven studies were selected, and the total number of primary studies was 15 (see details in Section 5.3).

All the available papers were stored in a reference manager (Zotero). More information about the inclusion and exclusion process is given in the following section.

4.3. Selection Criteria

The goal of the selection criteria of this study is to identify those studies that provided relevant empirical evidence related to our RQs [27]. Therefore, the studies were selected based on the following inclusion and exclusion criteria.

4.3.1. Inclusion Criteria

- No restrictions were applied to the publication year. Even though we do not limit the time when a study was published, the selected studies were published within the period of 2012–2022. This makes sense since DevSecOps was first mentioned in 2012.
- The study must be available as a full-text article. Full-text access to licensed content was limited to Østfold University College's contractual arrangements with publishers.
- The study is included if it addresses at least one compliance aspect integrated into DevSecOps.

4.3.2. Exclusion Criteria

- The study is excluded if it is not written in the English language (there were 11 studies in the German language);
- If the study is available in more than one database, the versions available in databases other than the one that provides the study for download are excluded (no study was available in different databases);
- If the study is available twice, once as a book chapter and then as a whole book, then the book must be excluded and only the relevant chapter should be considered (there were 10 studies in this scenario);
- If the study is inaccessible, it is excluded (20 studies were identified in this case however only one of them was selected for full-text reading after reading the title, abstract, and keywords. Despite that licensed content, we received access later, and the study was included as primary studies.)

4.4. Data Extraction and Selection Process

The extraction process was carried out in such a way that a researcher conducted the data extraction, and then another researcher reviewed it. The extraction process was performed using an excel sheet that included (i) the basic information about the selected studies, (ii) an overview of those studies, and (iii) specific areas of the research topic. Figure 1 provides an overview of the whole research process.

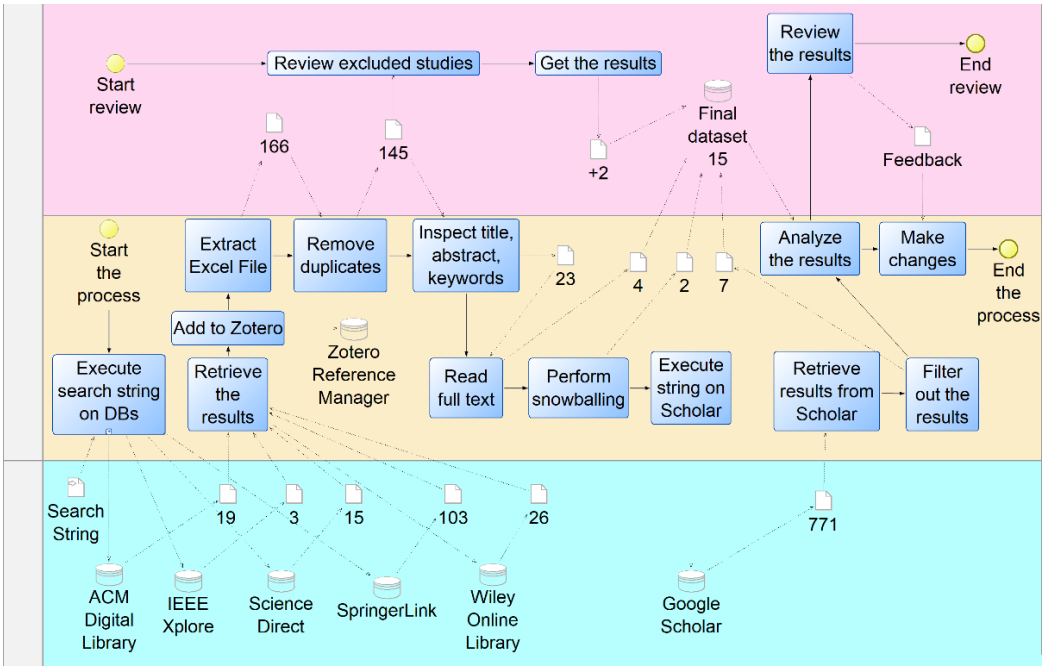

Figure 1. Overview of our Research Process.

5. Results

The findings presented in this section answer the three RQs proposed in this study. Those facilitated our work in categorizing compliance aspects, identifying the approaches to ensure compliance aspects, and revealing the compliance trends in the literature on this topic.

5.1. Categorizing Compliance Aspects (RQ1)

Table 2 provides an overview of the compliance aspects identified by this SLR. Those compliance aspects can be broadly classified into three categories: initiation, management, and technicalities. This section provides an overview of each category and discusses in detail the aspects belonging to each of the categories.

Table 2. Overview of Compliance Aspects in DevSecOps.

Category	#	Compliance Aspects	Primary Studies
Initiation	1	Compliance Issues	[1,3,61–63]
	2	Compliance Requirements	[1,58,60–63]
Management	3	Compliance Awareness and Training	[1,43,61,62]
	4	Compliance Automation	[1,20,50,61,63]
	5	Compliance Testing and Verification	[1,3,20,43,60,61,63–66]
	6	Compliance Validation	[1,50,60,61,63]
	7	Compliance Control and Monitoring	[1,3,43,59,67]
	8	Compliance Assessment/Evaluation	[1,61,62]
Technicalities	9	Compliance as Code	[1,20,43,50,66]
	10	Compliance Tools	[1,20,62,66]

5.1.1. Initiation

Compliance Issues: The lifecycle of compliance starts with a *compliance initiation* process. Such a process must define the DevOps processes themselves in order to make them available and ensure they are aligned with security standards, frameworks, or best practices [1]. Organizations going through mergers [62] are a particular case that stresses the need to conduct a detailed assessment identifying the applicable regulatory compliance. The definition of processes and an understanding of the DevOps approach help to identify

compliance issues [1]. Gaps in timely communication and collaboration are considered critical in introducing *compliance issues* [63]. Practitioners that are knowledgeable of security *compliance issues* in DevOps projects [61] are considered a valuable source for receiving more insights on such issues. Moreover, a comprehensive guide as suggested by [3] can strengthen practitioners' preparation on *compliance issues*.

Compliance Requirements: After defining compliance issues and the targeted state of security compliance, organizations have to express the target state of compliance in terms of requirements [1]. These requirements determine what needs to be accomplished to fulfill the security mandate [33]. *Compliance requirements* are considered as a nonfunctional concern for DevOps security [58]. This indicates that without a proper definition of compliance requirements, DevOps processes cannot securely function. Nevertheless, in order to be easily defined and understood by both humans and machines, compliance *requirements* need to be set into a language [68]. Inspec is an example of a Domain-Specific Formal Language (DSL) used for defining technical compliance requirements, therefore contributing to continuous and automated compliance checks [1]. However, defining the compliance requirements is not always an undemanding process. The complexity of such a process increases while moving toward more complex systems on a larger scale.

An approach to modeling *compliance requirements* [60] revealed the need for a definition of the *compliance requirements* of the software by vendors themselves before the delivery of the product. Once an organization identifies its internal requirements, it can use such a framework of requirements for validating its compliance with guidelines available at the organization [62]. In addition to internal frameworks, an organization can partner with third parties to address *compliance requirements*. The Alzheimer's Therapeutic Research Institute (ATRI) is an example of a collaboration with a third party to establish a compliant IT infrastructure for their system [63]. Since ATRI must deal with health data, ATRI partnered with Amazon Web Services Professional Services for a HIPPA-compliant infrastructure.

5.1.2. Management

Compliance Awareness and Training: Moreover, due to the fact that compliance is managed internally or through third parties offering compliance services, internal resources need to be assigned. This will ensure that compliance is being implemented and adhered to [1]. In order to properly assign the resources, each organization should organize workshops and host training sessions about compliance with security guidelines in order to appraise the IT staff [43]. These training sessions can provide information that may vary from the current situation and existing tools for addressing compliance issues to research efforts for future improvements and new tools.

The levels of awareness about tools could influence the activities of the standards implemented into pipelines as well [61]. Lack of awareness of the available compliance tools is one of the reasons that hamper the successful adoption of an organization's compliance processes [62]. This highlights not only the state of being aware but also the inclusion of all stakeholders in the security compliance process [69]. By discussing security concerns in the DevOps environment with them, both the development and operational teams can play a more proactive role in ensuring the compliance with standards [54]. In particular, greater transparency between developers, security experts, and operation staff allows the respective staff to not only provide input into what is being developed but also define security checks inside a DevSecOps environment [1]. In this sense, even top management must be subject to compliance checks [36].

Compliance Automation: All parties involved in the application development and operations processes have seen these processes evolve in tandem with the advancements of DevSecOps technologies. Such technologies advocate automation which is considered a backbone of DevSecOps workflow implementation [70]. Automated security has begun to consider the panorama of challenges by bridging the gap between compliance with security standards and the software engineering environment itself [20]. The opportunity to integrate compliance aspects into the DevOps methodology makes the process a matter

of automation as well. Abrahams and Langerman [1] explored the feasibility of the implementation of compliance automation into a DevOps pipeline. In particular, they tested "automated configuration management", "automated compliance testing", and "system and network architectures" to prove the ability of the DevOps principles to optimize the compliance process without hindering the velocity. Security Standard Compliant (S^2C) DevOps pipeline specification was used as the main artifact to estimate the possibilities to achieve compliance automation in [61]. In support of this approach, the authors have put further efforts into automating pipelines for security compliance using the pipeline specifications of the IEC 62443-4-1 standard. Abrahams and Langerman [1] proved that the automation of compliance into a DevSecOps pipeline does not hinder the velocity of the compliance process. Furthermore, Kumar and Goyal [50] show that automating security compliance contributes to the readiness of the system for audit and assessment processes. The compliant environment established by the Alzheimer's Therapeutic Research Institute (ATRI) in collaboration with Amazon Web Services (AWS) [63] applies the automation of compliance into a DevSecOps environment using automated methods for classifying data, detecting threats, configurations, and firewalls for web applications.

Compliance Testing and Verification: *Compliance Testing* is also known as conformance testing. This is the process of testing compliance requirements for their adherence to industry standards and benchmarks [1]. On the other hand, *Compliance Verification* checks, verifies, and attains the configuration of the infrastructure, platform, and application conform to the security control requirements and implementation approach by design [66].

Tools such as Chef InSpec, RedHat OpenScap, UpGuard1, and CIS-CAT allow practitioners to execute compliance tests [60]. Compliance testing can be ensured not only by using compliance testing tools, but also while using automated security tools. The use of security tools as an embedded component allows the automation of security testing [43]. Security activities such as Third-Party Vulnerability Scanning, Static/Dynamic Application Security Testing [20], security requirements testing, threat mitigation testing, vulnerability testing, and penetration testing [61] were performed to test for security risks. Security testing itself contributes to the application of compliance testing. For instance, Kryptowire—rebranded as Quokka—[71] is an applications testing provider that tests whether Android applications comply with international regulations, such as GDPR, the Open Web Application Security Project (OWASP), and the National Information Assurance Partnership (NIAP).

Alongside other security tests, the need to scan container images for security threats or vulnerabilities is mentioned as well [43]. Because container images are derived from the container, the compromise of the container will be inherited from the container images too. For example, a security audit of Docker application container images can be conducted throughout the SDLC to verify compliance with the OWASP Container Security Verification Standards [65].

Other efforts have been made to support compliance testing; for example, presenting new approaches that enable the execution of compliance tests. COMET [60] represents such efforts in the shape of a web-based continuous compliance testing framework, which allows users to build continuous compliance testing scenarios. Another effort was conducted by ATRI to develop automated testing and validation methods that address the challenges with regulatory agencies in conducting multicenter Alzheimer's disease and Alzheimer's disease-related dementias (AD/ADRDs) clinical trials [63]. Furthermore, Peldszus et al. [64] proposed incremental rule-based security violation patterns for security compliance checks.

Compliance Validation: This is briefly mentioned in the scientific literature and the context in which this aspect is positioned differs among authors. Abrahams and Langerman [1] mentioned "code validation". Taking into consideration the *"Compliance as Code"* aspect (see Section 5.1.3), a compliance validation process could be implemented by validating the code in which the compliance solutions are embedded. Some of the selected studies [50,60,61,63] mentioned the validation process by passing along with testing and verification. Kumar and Goyal [50] presented the embedding of security

testing into tasks as a form of automated security validation, while Steffens et al. [60] posed the validation process as an activity that provides insights on compliance state without necessarily running the tests. On the other hand, experts like Nygard [72], with over 20 years of experience across large enterprises, define validation as the process of comparing the evidence to particular constraints in order to demonstrate the compliance of the system with specific requirements.

Compliance Control and Monitoring: The time differences between the manual and continuous compliance assessment noted by Abrahams et al. [1] indicate that continuous *compliance control* is a model aiming to optimize the process. Such an automated process will enlighten the loads that organizations may carry while trying to adhere to industry standards. Nevertheless, a product approved by Quality Assurance (QA) does not always comply with all the regulations imposed in the global market [3]. This fact emphasizes the importance of continuous control and monitoring. Furthermore, it suggests an adaption to the ongoing changes in standards and new regulations that may apply.

According to a recent review of the literature [73], implementation of controls, besides maintaining compliance and assessing vulnerabilities, are key factors that contribute to security alignment throughout the business process. This analogy can be extended to compliance alignment. Desai and Nisha [67] explored best practices for ensuring security in DevOps, including preventive controls and compliance checks. According to them, compliance monitoring is an all-stages process that will pave the way for further compliance controls throughout the software lifecycle.

Continuous *compliance monitoring* refers to all the operations by constantly checking the adherence to standard regulations, thereby preventing the occurrence of noncompliance flags [43]. The *compliance monitoring* process can be supported by metrics, such as those defined in a composed service-level agreement (SLA) [59] in the MUSA framework, which is an open source integrated tool suit developed by project MUSA [74]. The monitoring functionality in the MUSA framework is supported by the MUSA Security Assurance Platform [59].

Compliance Assessment/Evaluation: *Compliance assessment* is the continuous process that follows compliance testing [44]. In this regard, different application security testing approaches, such as Dynamic Application Security Testing (DAST) and Software Composition Analysis (SCA), are used to assess compliance. SCA centers on license compliance while DAST is essential for industry-standard compliance [75] that is also the focus of this review. Following the DAST approach, OpenSCAP [76] is an initiative focused on Security Content Automation Protocol (SCAP). It provides multiple tools in support of compliance and vulnerability assessment, e.g., OpenSCAP Base, OpenVAS, and OWASP ZAP [32]. In addition to the internal assessment using different available tools, auditing and governance services can be purchased by third parties, such as AWS. The solution of AWS for compliance auditing is AWS CloudTrail [77]. This solution not only protects the customers' accounts but also saves the organization from possible fines. On this behalf, compliance with multiple regulations such as SOC, PCI, and HIPAA relies on using CloudTrail logs [78].

Different methods for assessing compliance have arisen, regardless of whether the process is performed internally, externally, manually, or in an automatic way. One approach used by [1] is evaluating the compliance assessment timeframe intervals with industry standards. The selected standards and benchmarks were ISO 27001/2, COBIT Technical Benchmarks, and SANS Top 20 Critical Security Controls. Another approach, as suggested by [61] about compliance with IEC 62443-4-1 standard [79], could be that compliance auditors evolve their artifacts as templates. This will allow them to perform compliance assessments of DevOps pipelines, with regard to the specific standards.

In addition to the assessment approach used, several criteria might impact the *compliance evaluation* process. In this regard, Vadlamudi and Sam [62] list the location, size, and global presence of the organization among the criteria that could set a difference between one evaluation process and another.

5.1.3. Technicalities

Compliance as Code and Infrastructure Configurations: Shields [66] describes *Compliance as Code* (*CaC*) as a strategy of embedding compliance solutions on the code that meets the security requirements. The implementation will endorse the automation of the compliance process and pave the way for meeting security requirements, given the improvement of code security. CaC, the same as Infrastructure as Code (IaC), comes as an extension of the Security as Code concept and optimizes the development process of secure applications by reusing code controls [50].

Infrastructure configurations can also be standardized while being written as lines of code. This will enable both the automation of the configuration management and the audit of activities [1]. On the other hand, the automation of system configurations itself will ensure that manual errors are avoided and compliance is ensured [43]. Moreover, it will facilitate the compliance verification process, given the security configuration applied and compliance checks on the configuration [20].

Compliance Tools: While considering the management processes of compliance, an important aspect is the tools used in support of such processes. The need for an automated security compliance tool was described in 2013 by Ullah et al. [21]. Back then, the concept of DevSecOps was at its beginning and not yet a buzzword [80]. Authors identified the necessity of tools in a cloud environment, and yet it is relevant for other environments as well, including DevSecOps.

Abrahams and Langerman [1] highlighted the efforts of developing compliance tools for organizations that intend to transfer their activities to a DevSecOps environment. The capabilities of these tools vary from compliance testing to the automation of code validation. However, as already mentioned in "*Compliance Testing and Verification*" (see Section 5.1.2), compliance testing can be ensured by using automated security tools. Angermeir et al. [20] identified a list of security tools utilized in the continuous integration of enterprise-driven open source software split between security activities. Among other security activities, [20] distinguished compliance-related activities, such as secure configuration/hardening and compliance/hardening checks. The first activity considers tools such as Chef, Ansible, Terraform, Puppet, Phan, Seccomp, CloudCustodian, and Turbot. The second activity considers tools such as Bane, Ansible-Lint, Blackbox, Puppet-Lint, Inspec, Kube Audit, Foodcritic, Cookstyle, and conftest. Other tools such as Chef InSpec, HashiCorp Sentinel, and OpenSCAP are mentioned by Shields in [66] on behalf of the "compliance as code" and "policy as code" approach. It is also worth mentioning that the tools identified in [3,20] do not explicitly test compliance. Those are automated security tools that, while scanning for security issues, may generate non-compliance alerts. The use of these tools contributes indirectly to achieving compliance testing. In addition, Shields [66] mentions other security testing tools for the different testing types. For example, Static Application Security Testing (SAST) uses tools such as Chexmarx, Fortify, and SonarQube; DAST uses tools such as Burp Suite, WebInspect, and Zap; and Interactive Application Security Testing (IAST) uses tools such as Veracode and Contrast Security.

Beyond technical compliance, Abrahams and Langerman [1] identified non-technical compliance criteria for processes that cannot be automated. Thus, the use of security tools and solutions is not applicable. In this scenario, auditors and security experts must have a solid understanding of DevOps practices as a prerequisite. As a result, the information required to ensure that an organization is compliant with security standards and best practices must be available to auditors. *Documentation* that defines the security phase in the DevSecOps lifecycle is a viable solution. In fact, the need for organizations to invest in creating comprehensive documentation was highlighted in [62].

5.2. Approaches to Ensure Compliance Aspects (RQ2)

Abrahams and Langerman [1] explored compliance automation focusing on the concept of compliance at a velocity, referring to the process of achieving compliance with industry standards without impacting the product delivery time, thus not hindering the

velocity of the pipeline. As a result, a set of steps must be implemented within the organization to ensure regulatory compliance. During these steps, the desired state of security compliance is defined, and compliance requirements are translated into a language that both humans and machines understand and then tested and checked for validity. According to [58], the lack of proper compliance requirements can put the security of a DevOps environment into question.

Likewise, Moyón et al. [61] addressed how to achieve security compliance with a minimal lead time impact. To do so, they highlighted the need to understand how to integrate security standard requirements into DevOps pipelines. Then, an approach finalized with the S^2C DevOps Pipeline was proposed. The S^2C DevOps pipeline aims to both raise awareness of compliance issues in the context of DevOps and achieve compliance automation [61]. Peldszus et al. [64] also made a point in the lead time by proposing the solution of incremental security compliance checks. This solution suggests checking only the changed parts, therefore avoiding the re-execution of the entire compliance checks.

In light of the discussion about pipeline velocity, different tools offer their contribution to its measurement. Zeeshan [3] presents the Gitlab option to preview the time spent on compliance and security testing. Discussing measurements, Kumar and Goyal [50] suggested a couple of metrics for measuring the velocity objective of businesses through their conceptual model for Automated DevSecOps using Open-source Software Over the Cloud (ADOC). The ADOC model addresses the continuous security requirements and maps them to address the challenges faced during DevSecOps adoption. The model defines a workflow instated with different security controls. Security Audit and Compliance activity are mapped with the workflow stage of operation. Beyond the implementation of the ADOC model and the use of automation tools, the authors believe that a security-first culture and compliance mindset can create a real difference in delivering security with velocity. Organizing security training sessions or workshops contribute to enhancing such a mindset. Indeed, Chatterjee [43] lists training among the most vital elements of a DevSecOps framework.

It is worth mentioning that compliance assurance is a continuous process, as are the efforts to automate and ensure both security testing and compliance. Steffens et al. [60] proposed a software-centric approach that sets the stage for continuous compliance testing. Through COMET, a web-based framework for continuous compliance testing, the authors demonstrate how their approach can be easily automated and embedded into a continuous delivery process.

Additional efforts for ensuring security compliance are required since security knowledge is evolving. In this regard, Peldszus et al. [64] provided prototypical implementation on ensuring security compliance on systems subject of expanded knowledge. On the other hand, the efforts for ensuring security compliance have given rise to several tools to facilitate the process (see "Compliance Tools").

5.3. Emerging Compliance Trends in DevOps literature (RQ3)

The search process resulted in the selection of 15 primary studies. Table 3 shows the basic information of the primary studies, i.e., database, reference, year of publication, title, and whether the primary study was published in a journal (J), conference (C), or book chapter (B). There is an equal number of conference and journal studies.

Figure 2 illustrates the number of relevant studies published each year. Those studies fall into the timeframe 2018–2022. Recent GDPR violations and fines for well-known companies, such as Google, Amazon, and Facebook Meta, have turned security compliance into a buzzword and have possibly increased the interest in the topic. Since 2020, there has been an increased interest in discussing compliance aspects in the context of DevSecOps. Selected studies discuss medical information systems, charging and billing systems, industrial control systems, cloud-based systems, software systems, and other critical systems.

Table 3. Overview of Primary Studies.

Database	Reference	Year	Type *	Title
ACM Digital Library	[58]	2020	C	Towards a Hypothetical Framework to Secure DevOps Adoption: Grounded Theory Approach
IEEE Xplore	[1]	2018	C	Compliance at Velocity within a DevOps Environment
	[62]	2021	C	A Novel Approach to Onboarding Secure CloudNative Acquisitions into Enterprise Solutions
	[65]	2021	C	Security Audit of Docker Container Images in Cloud Architecture
	[20]	2021	C	Enterprise-Driven Open Source Software: A Case Study on Security Automation
ScienceDirect	[64]	2021	J	Ontology-driven evolution of software security
	[50]	2020	J	Modeling Continuous Security: A Conceptual Model for Automated DevSecOps using Open-source Software Over the Cloud (ADOC)
Springer Link	[61]	2020	C	Integration of Security Standards in DevOps Pipelines: An Industry Case Study
	[43]	2021	B	Security in DevOps and Automation
	[3]	2020	B	Compliance and Security
Wiley	[59]	2019	J	Service Level Agreement-based GDPR Compliance and Security Assurance in (multi)Cloud-based Systems
Other	[60]	2018	J	Towards Data-driven Continuous Compliance Testing
	[67]	2021	J	Best Practices for Ensuring Security in DevOps: A Case Study Approach
	[66]	2020	B	The Secret to Achieving a Faster ATO
	[63]	2022	J	ATRI EDC: A Novel Cloud-native Remote Data Capture System for Large Multicenter Alzheimer's Disease and Alzheimer's Disease-related Dementias Clinical Trials

* C = Conference, J = Journal, B = Book Chapter.

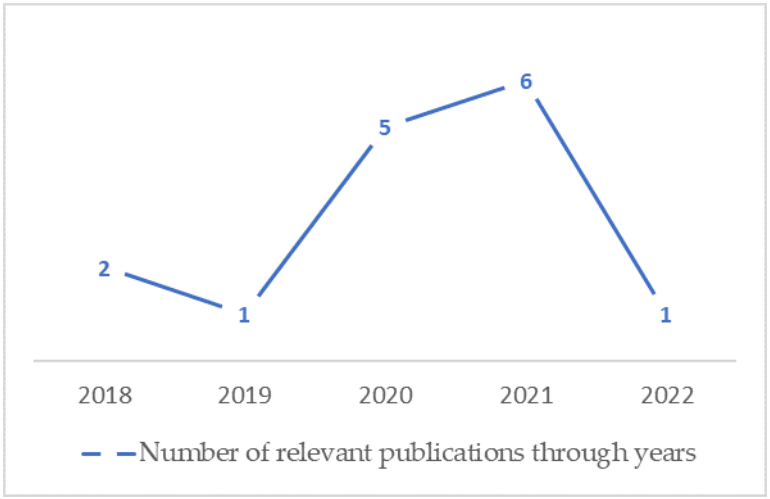

Figure 2. Number of Primary Studies by Year.

In order to track the emerging compliance trends in the context of DevSecOps, we have created a timeline of the compliance aspects, shown in Table 4. The years are positioned horizontally together with the references of each selected publication, while the identified aspects are positioned vertically. There is no clear pattern in the compliance aspects discussed in each of the publications. Abrahams et al. [1] discussed all the aspects identified in this review. *"Compliance Testing and Verification"* seems to be the most discussed aspect by various studies (10) throughout the years (see Table 4). In the current year, compliance aspects, such as compliance issues, requirements, automation, testing and verification, and compliance validation, demonstrate an ongoing interest in these aspects.

Table 4. Overview of Compliance Aspects by Year.

Category		Aspects	2018		2019		2020			2021						2022	Total
			[1]	[60]	[59]	[50]	[61]	[58]	[66]	[3]	[64]	[43]	[67]	[62]	[65]	[20]	[63]	
Initiation	1	Issues	X				X			X				X			X	5
	2	Requirements	X	X			X	X						X			X	6
Management	3	Awareness and Training	X				X					X		X				4
	4	Automation	X			X	X									X	X	5
	5	Testing and Verification	X	X			X		X	X	X	X			X	X	X	10
	6	Compliance Validation	X	X		X	X										X	5
	7	Control and Monitoring	X		X					X		X	X					5
	8	Assessment/ Evaluation	X				X							X				3
Technicalities	9	Compliance as Code	X			X			X			X				X		5
	10	Tools	X						X					X		X		4
		Total	10	3	1	3	7	1	3	3	1	4	1	5	1	4	5	

6. Discussion and Future Work

Limitations to our research approach arise from the low number of studies that discuss the integration of compliance aspects into DevSecOps. Another limitation is related to the domains where the integration of compliance aspects could be applicable. Given that selected studies discuss industries such as healthcare or finance, we suggest further work on other industries operating with critical systems. Despite the limitations, since some of the studies include more than one compliance aspect discussed, our approach is applicable for perceiving significant results.

Selected studies highlighted the importance of both fully understanding DevOps by security experts [1] and creating a security-first culture [50], which will facilitate their work. In light of that, they can build compliance methods that test compliance criteria in the context of DevOps which do not hamper the expected benefits, such as agility or velocity in time to market [1]. We believe that resistance to embracing new methodologies of working in general, specifically compliance activities, is another cultural factor to overcome. Detecting a security flaw and/or compliance issue in a later phase will require extra time and effort. In this scenario, we consider *compliance by design* as a holistic approach to introduce preventative controls—and where possible, automated tests—to reduce the potential issues throughout different stages of the SDLC. It also implies a further perspective of thinking on compliance *"compliance analytics"* which could be a new aspect for further research as well.

Taking into consideration the recommendation by Kumar et al. [50], for further research on cognitive security controls using data analytics, we believe that data analytics observations could apply to compliance controls as well. This could be a topic of interest for those working on data analytics and willing to explore other contexts in which those could be used. Although gaining more insights into compliance testing challenges is relevant and necessary, as mentioned by Steffens et al. [60], we believe that the whole compliance process is challenging in a DevSecOps context. Thus, evaluating the integration of compliance aspects identified in this review and identifying their challenges could be another area of research interest. In particular, it could be interesting to further work on compliance validation, since two studies [61,72] reported challenges for the automation and the high manual performing rate of this process.

While delivering cloud services, different open-source software (OSS) tools, including compliance tools, can be used. Kumar et al. [50] recommended further research on the interworking of multiple OSS, while Angermeir et al. [20] aimed to analyze through empirical methods the top used security tools in OSS projects, in order to make recommendations to practitioners. In the context of our study, we suggest considering the dynamics of

the compliance tools and analyzing which combinations of compliance tools are most commonly used. Another area for further research could be considering the metrics on the performance of each compliance tool, thus conducting a feasibility analysis on the use of tools in different scenarios. In addition to the use of different tools, the phase in which an enterprise is currently operating could be challenging as well. Vadlamudi and Sam [62] showed a case of mergers and acquisitions and the need to strengthen the security posture of the mergers even in the post-acquisition phase. Thus, we believe that while thinking about compliance, the compliance tools being used, or other compliance aspects, we should also consider the external and internal conditions that an enterprise faces.

7. Conclusions

Adherence to industry standards and frameworks should be intrinsic to the daily operational activities of any organization. However, the scientific literature lacks a discussion on compliance aspects and how organizations operating in a DevSecOps environment handle them. In this regard, we carried out an SLR on compliance aspects in the context of DevSecOps following the guidelines for systematic literature reviews in software engineering, proposed by Kitchenham and Charters [27]. As a result, ten compliance aspects were identified and grouped into three major categories:

- **Compliance Initiation:** This category covers issues related to defining the desired state of security compliance and specifying the compliance requirements;
- **Compliance Management:** Incorporates all the processes related to the management of compliance such as Automation, Testing and Verification, Validation, Control and Monitoring, Awareness and Training, and Assessment;
- **Compliance Technicalities:** Includes the technicalities that are used for managing compliance in a DevSecOps environment, such as the compliance as code concept and different compliance tools.

As security compliance is becoming a hot topic, researchers should contribute to enhancing the understanding of compliance aspects and their implementation in a DevSecOps environment. In this regard, we identified some future research directions: (i) considering two further perspectives of thinking on compliance: *compliance by design* and *compliance analytics*, (ii) considering the dynamics of the compliance tools and analyzing the usability of compliance tools, (iii) considering the metrics on the performance of each tool and conducting a feasibility analysis on the use of tools in different scenarios, (iv) use of data analytics on compliance controls, and (v) evaluating the harmonization of ten compliance aspects and identifying their challenges.

Author Contributions: Conceptualization, R.C.-P.; methodology, X.R. and M.S.-G.; validation, M.S.-G., V.G. and S.C.; formal analysis, X.R.; resources, X.R. and M.S.-G.; writing—original draft preparation, X.R. and M.S.-G.; writing—review and editing, X.R., M.S.-G., V.G. and S.C. and R.C.-P.; visualization, X.R.; supervision, V.G. and R.C.-P.; funding acquisition, R.C.-P. All authors have read and agreed to the published version of the manuscript.

Funding: This research is partially funded by the Research Council of Norway (RCN) in the INTPART program, under the project "Reinforcing Competence in Cybersecurity of Critical Infrastructures: A Norway-US Partnership (RECYCIN)", with the project number #309911.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abrahams, M.Z.; Langerman, J.J. Compliance at Velocity within a DevOps Environment. In Proceedings of the 2018 Thirteenth International Conference on Digital Information Management (ICDIM), Berlin, Germany, 24–26 September 2018; pp. 94–101.
2. General Data Protection Regulation (GDPR). Available online: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679 (accessed on 22 February 2022).
3. Zeeshan, A.A. Compliance and Security. In *DevSecOps for NET Core: Securing Modern Software Applications*; Zeeshan, A.A., Ed.; Apress: Berkeley, CA, USA, 2020; pp. 265–278. ISBN 978-1-4842-5850-7.
4. California Consumer Privacy Act (CCPA). Available online: https://oag.ca.gov/privacy/ccpa (accessed on 15 March 2022).

5. 201 CMR 17 Standards for the Protection of Personal Information of Residents of the Commonwealth. Available online: https://www.mass.gov/regulations/201-CMR-1700-standards-for-the-protection-of-personal-information-of-residents-of-the-commonwealth (accessed on 15 March 2022).
6. Federal Information Security Management Act of 2002. Available online: https://www.govinfo.gov/content/pkg/PLAW-107publ347/pdf/PLAW-107publ347.pdf (accessed on 22 February 2022).
7. Family Educational Rights and Privacy Act (FERPA). Available online: https://www2.ed.gov/policy/gen/guid/fpco/ferpa/index.html (accessed on 24 February 2022).
8. Sarbanes-Oxley Act of 2002. Available online: https://www.govinfo.gov/content/pkg/PLAW-107publ204/pdf/PLAW-107publ204.pdf (accessed on 22 February 2022).
9. Gramm-Leach-Bliley Act. Available online: https://www.ftc.gov/tips-advice/business-center/privacy-and-security/gramm-leach-bliley-act (accessed on 24 February 2022).
10. Health Insurance Portability And Accountability Act Of 1996. Available online: https://www.govinfo.gov/content/pkg/PLAW-104publ191/pdf/PLAW-104publ191.pdf (accessed on 22 February 2022).
11. National Institute of Standards and Technology. Available online: https://www.nist.gov/ (accessed on 22 February 2022).
12. Official PCI Security Standards Council Site–Verify PCI Compliance, Download Data Security and Credit Card Security Standards. Available online: https://www.pcisecuritystandards.org/document_library (accessed on 22 February 2022).
13. Capability Maturity Model Integration (CMMI) Institute. Available online: https://cmmiinstitute.com/ (accessed on 16 September 2022).
14. COBIT | Control Objectives for Information Technologies. Available online: https://www.isaca.org/resources/cobit (accessed on 23 February 2022).
15. CIS Controls v8 Released | SANS Institute. Available online: https://www.sans.org/blog/cis-controls-v8/ (accessed on 23 February 2022).
16. Facebook Fined €17m for Data Privacy Laws Breach. Available online: https://www.bbc.com/news/articles/cp9yenpgjwzo (accessed on 10 September 2022).
17. Holzhofer, M. Dsgvo-Portal De. Available online: https://www.dsgvo-portal.de/ (accessed on 10 September 2022).
18. Initiative (WAI), W.W.A. WCAG 2 Overview. Available online: https://www.w3.org/WAI/standards-guidelines/wcag/ (accessed on 10 September 2022).
19. Forsgren, N.; Kersten, M. DevOps Metrics. *Commun. ACM* **2018**, *61*, 44–48. [CrossRef]
20. Angermeir, F.; Voggenreiter, M.; Moyón, F.; Mendez, D. Enterprise-Driven Open Source Software: A Case Study on Security Automation. In Proceedings of the 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Madrid, Spain, 25–28 May 2021; pp. 278–287.
21. Ullah, K.W.; Ahmed, A.S.; Ylitalo, J. Towards Building an Automated Security Compliance Tool for the Cloud. In Proceedings of the 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Melbourne, Australia, 16–18 July 2013; pp. 1587–1593.
22. DeLuccia IV, J.; Gallimore, J.; Kim, G.; Miller, B. DevOps Audit Defense Toolkit. Available online: https://itrevolution.com/devops-audit-defense-toolkit/ (accessed on 15 March 2022).
23. Amazon Web Services–Labs. Available online: https://github.com/awslabs (accessed on 13 March 2022).
24. DevSecOps. Available online: https://www.devsecops.org (accessed on 13 March 2022).
25. OpenDevSecOps. Available online: https://github.com/opendevsecops (accessed on 13 March 2022).
26. Rugged Software. Available online: http://ruggedsoftware.org/ (accessed on 13 March 2022).
27. Kitchenham, B.; Charters, S. Guidelines for Performing Systematic Literature Reviews in Software Engineering 2007. Available online: https://www.researchgate.net/profile/Barbara-Kitchenham/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering/links/61712932766c4a211c03a6f7/Guidelines-for-performing-Systematic-Literature-Reviews-in-Software-Engineering.pdf (accessed on 22 February 2022).
28. Editor, C.C. Security–Glossary | CSRC. Available online: https://csrc.nist.gov/glossary/term/security (accessed on 5 October 2021).
29. Marlow, A.T. What's New in ISO 37301:2021 & How It Can Improve Your Compliance Management. Available online: https://emsmastery.com/2021/05/18/whats-new-in-iso-373012021-how-it-can-improve-your-compliance-management/ (accessed on 11 September 2022).
30. Kshetri, N. Big Data's Impact on Privacy, Security and Consumer Welfare. *Telecommun. Policy* **2014**, *38*, 1134–1145. [CrossRef]
31. Santos, C.; Bielova, N.; Matte, C. Are Cookie Banners Indeed Compliant with the Law? Deciphering EU legal requirements on consent and technical means to verify compliance of cookie banners. *arXiv* **2020**, arXiv:1912.07144v2. [CrossRef]
32. Dupont, S.; Ginis, G.; Malacario, M.; Porretti, C.; Maunero, N.; Ponsard, C.; Massonet, P. Incremental Common Criteria Certification Processes Using DevSecOps Practices. In Proceedings of the 2021 IEEE European Symposium on Security and Privacy Workshops (EuroS PW), Vienna, Austria, 6–10 September 2021; pp. 12–23.
33. Julisch, K. Security Compliance: The next Frontier in Security Research. In Proceedings of the 2008 New Security Paradigms Workshop, New York, NY, USA, 22 September 2008; Association for Computing Machinery: New York, NY, USA, 2008; pp. 71–74.
34. Casagni, M.; Heeren, M.; Cagle, R.; Eng, R.; Flamm, J.; Goldrich, S.; Hanf, D.; Kristan, M.; Brunelle, J.F.; Harvey, T.; et al. March 2018 Federal Devops Summit Report. 31. Available online: https://atarc.org/wp-content/uploads/2019/01/2018-03-01-ATARC-Federal-DevOps-Summit-White-Paper-1.pdf (accessed on 3 March 2022).

35. Daud, M.; Rasiah, R.; George, M.; Asirvatham, D.; Thangiah, G. Bridging The Gap Between Organisational Practices and Cyber Security Compliance: Can Cooperation Promote Compliance in Organisations? *Int. J. Bus. Soc.* **2018**, *19*, 20.
36. Wood, C.C. Policies Alone Do Not Constitute a Sufficient Awareness Effort. *Comput. Fraud. Secur.* **1997**, *1997*, 14–19. [CrossRef]
37. ISO/IEC 27001:2013 Information Technology—Security Techniques—Information Security Management Systems—Requirements. Available online: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/05/45/54534.html (accessed on 23 February 2022).
38. ISO 27001 Annex A.18–Compliance. Available online: https://www.isms.online/iso-27001/annex-a-18-compliance/ (accessed on 23 February 2022).
39. Hsu, T.H.-C. *Hands-On Security in DevOps: Ensure Continuous Security, Deployment, and Delivery with DevSecOps*; Packt Publishing Ltd.: Birmingham, UK, 2018; ISBN 978-1-78899-241-1.
40. McGraw, G. Software Security. *IEEE Secur. Priv.* **2004**, *2*, 80–83. [CrossRef]
41. Raynaud, F. DevSecOps Whitepaper. Available online: https://pdfcoffee.com/devsecops-whitepaper-pdf-free.html (accessed on 20 February 2022).
42. OWASP Top Ten Web Application Security Risks | OWASP. Available online: https://owasp.org/www-project-top-ten/ (accessed on 23 February 2022).
43. Chatterjee, R. Security in DevOps and Automation. In *Red Hat and IT Security: With Red Hat Ansible, Red Hat OpenShift, and Red Hat Security Auditing*; Chatterjee, R., Ed.; Apress: Berkeley, CA, USA, 2021; pp. 65–104. ISBN 978-1-4842-6434-8.
44. Sun, X.; Cheng, Y.; Qu, X.; Li, H. Design and Implementation of Security Test Pipeline Based on DevSecOps. In Proceedings of the 2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 18–20 June 2021; Volume 4, pp. 532–535.
45. MacDonald, N.; Head, I. DevSecOps: How to Seamlessly Integrate Security Into DevOps. Gartner Research 2016. Available online: https://www.gartner.com/en/documents/3463417 (accessed on 22 February 2022).
46. Riungu-Kalliosaari, L.; Mäkinen, S.; Lwakatare, L.E.; Tiihonen, J.; Männistö, T. DevOps Adoption Benefits and Challenges in Practice: A Case Study. In *Product-Focused Software Process Improvement*; Abrahamsson, P., Jedlitschka, A., Nguyen Duc, A., Felderer, M., Amasaki, S., Mikkonen, T., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2016; Volume 10027, pp. 590–597. ISBN 978-3-319-49093-9.
47. Carturan, S.B.O.G.; Goya, D.H. A Systems-of-Systems Security Framework for Requirements Definition in Cloud Environment. In Proceedings of the 13th European Conference on Software Architecture–ECSA '19, Paris, France, 9–13 September 2019; ACM Press: Paris, France, 2019; Volume 2, pp. 235–240.
48. Mohan, V.; ben Othmane, L.; Kres, A. BP: Security Concerns and Best Practices for Automation of Software Deployment Processes: An Industrial Case Study. In Proceedings of the 2018 IEEE Cybersecurity Development (SecDev), Cambridge, MA, USA, 30 September 2018–2 October 2018; pp. 21–28.
49. Carter, K. Francois Raynaud on DevSecOps. *IEEE Softw.* **2017**, *34*, 93–96. [CrossRef]
50. Kumar, R.; Goyal, R. Modeling Continuous Security: A Conceptual Model for Automated DevSecOps Using Open-Source Software over Cloud (ADOC). *Comput. Secur.* **2020**, *97*, 101967. [CrossRef]
51. Yasar, H. *Overcoming DevSecOps Challenges: A Practical Guide for All Stakeholders*; Carnegie-Mellon Univ: Pittsburgh, PA, USA, 2020.
52. Rahman, A.A.U.; Williams, L. Software Security in DevOps: Synthesizing Practitioners' Perceptions and Practices. In Proceedings of the 2016 IEEE/ACM International Workshop on Continuous Software Evolution and Delivery (CSED), Austin, TX, USA, 14–15 May 2016; pp. 70–76.
53. Sen, A. DevOps, DevSecOps, AIOPS- Paradigms to IT Operations. In *Proceedings of the Evolving Technologies for Computing, Communication and Smart World*; Singh, P.K., Noor, A., Kolekar, M.H., Tanwar, S., Bhatnagar, R.K., Khanna, S., Eds.; Springer: Singapore, 2021; pp. 211–221.
54. Mohan, V.; Othmane, L.B. SecDevOps: Is It a Marketing Buzzword?–Mapping Research on Security in DevOps. In Proceedings of the 2016 11th International Conference on Availability, Reliability and Security (ARES), Salzburg, Austria, 31 August–2 September 2016; pp. 542–547.
55. Lie, M.F.; Sánchez-Gordón, M.; Colomo-Palacios, R. DevOps in an ISO 13485 Regulated Environment: A Multivocal Literature Review. In Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Bari, Italy, 5 October 2020; ACM: Bari, Italy, 2020; pp. 1–11.
56. Lee, J.; Kang, K.; Shim, C. Devsecops for Small and Medium-Sized Enterprises: A Systematic Literature Review. *SSRN* **2022**. [CrossRef]
57. Rajapakse, R.N.; Zahedi, M.; Babar, M.A.; Shen, H. Challenges and Solutions When Adopting DevSecOps: A Systematic Review. *arXiv* **2021**, arXiv:2103.08266. [CrossRef]
58. Rafi, S.; Yu, W.; Akbar, M.A. Towards a Hypothetical Framework to Secure DevOps Adoption: Grounded Theory Approach. In Proceedings of the Evaluation and Assessment in Software Engineering, New York, NY, USA, 15 April 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 457–462.
59. Rios, E.; Iturbe, E.; Larrucea, X.; Rak, M.; Mallouli, W.; Dominiak, J.; Muntés, V.; Matthews, P.; Gonzalez, L. Service Level Agreement-Based GDPR Compliance and Security Assurance in(Multi)Cloud-Based Systems. *IET Softw.* **2019**, *13*, 213–222. [CrossRef]

60. Steffens, A.; Lichter, H.; Moscher, M. Towards Data-Driven Continuous Compliance Testing. 7. Available online: CSE2018_preface(ceur-ws.org) (accessed on 1 February 2022).
61. Moyón, F.; Soares, R.; Pinto-Albuquerque, M.; Mendez, D.; Beckers, K. Integration of Security Standards in DevOps Pipelines: An Industry Case Study. In *Proceedings of the Product-Focused Software Process Improvement*; Morisio, M., Torchiano, M., Jedlitschka, A., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 434–452.
62. Vadlamudi, S.; Sam, J. A Novel Approach to Onboarding Secure Cloud-Native Acquisitions into Enterprise Solutions. In Proceedings of the 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), Bengaluru, India, 22–24 November 2021; Volume 1, pp. 228–233.
63. Jimenez-Maggiora, G.A.; Bruschi, S.; Qiu, H.; So, J.-S.; Aisen, P.S. ATRI EDC: A Novel Cloud-Native Remote Data Capture System for Large Multicenter Alzheimer's Disease and Alzheimer's Disease-Related Dementias Clinical Trials. *JAMIA Open* **2022**, *5*, ooab119. [CrossRef]
64. Peldszus, S.; Bürger, J.; Kehrer, T.; Jürjens, J. Ontology-Driven Evolution of Software Security. *Data Knowl. Eng.* **2021**, *134*, 101907. [CrossRef]
65. Shameem Ahamed, W.S.; Zavarsky, P.; Swar, B. Security Audit of Docker Container Images in Cloud Architecture. In Proceedings of the 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC), Jalandhar, India, 21–23 May 2021; pp. 202–207.
66. Shields, M. The Secret to Achieving a Faster ATO. Available online: https://pyramidsystems.com/wp-content/uploads/2020/09/Pyramid-ATO-ebook-1.pdf (accessed on 10 March 2022).
67. Desai, R.; Nisha, T.N. Best Practices for Ensuring Security in DevOps: A Case Study Approach. *J. Phys. Conf. Ser.* **2021**, *1964*, 042045. [CrossRef]
68. Preidel, C.; Borrmann, A. Towards Code Compliance Checking On The Basis Of A Visual Programming Language. 20. Available online: 2016_25.content.01707.pdf(itcon.org) (accessed on 22 February 2022).
69. Kim, S.H.; Yang, K.H.; Park, S. An Integrative Behavioral Model of Information Security Policy Compliance. *Sci. World J.* **2014**, *2014*, 1–12. [CrossRef] [PubMed]
70. Automation and the DevOps Workflow. Available online: https://www.chef.io/docs/default-source/legacy/automation-and-the-devops-workflow.pdf (accessed on 23 February 2022).
71. Kryptowire®: Mobile Security Company. Available online: https://www.kryptowire.com/ (accessed on 12 April 2022).
72. Compliance in a DevOps Culture. Available online: https://martinfowler.com/articles/devops-compliance.html (accessed on 22 October 2022).
73. Myrbakken, H.; Colomo-Palacios, R. DevSecOps: A Multivocal Literature Review. In *Proceedings of the Software Process Improvement and Capability Determination*; Mas, A., Mesquida, A., O'Connor, R.V., Rout, T., Dorling, A., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 17–29.
74. MUSA Project Website. Available online: https://www.musa-project.eu/ (accessed on 23 February 2022).
75. Dynamic Application Security Testing: DAST Basics. Available online: https://www.whitesourcesoftware.com/resources/blog/dast-dynamic-application-security-testing/ (accessed on 4 March 2022).
76. OpenSCAP Portal. Available online: https://www.open-scap.org/ (accessed on 22 February 2022).
77. Building End-to-End AWS DevSecOps CI/CD Pipeline with Open Source SCA, SAST and DAST Tools. Available online: https://aws.amazon.com/blogs/devops/building-end-to-end-aws-devsecops-ci-cd-pipeline-with-open-source-sca-sast-and-dast-tools/ (accessed on 4 March 2022).
78. Secure Standardized Logging–AWS CloudTrail–Amazon Web Services. Available online: https://aws.amazon.com/cloudtrail/ (accessed on 4 March 2022).
79. Standards, E. EN IEC 62443-4-1. Available online: https://www.en-standard.eu/csn-en-iec-62443-4-1-security-for-industrial-automation-and-control-systems-part-4-1-secure-product-development-lifecycle-requirements/ (accessed on 23 February 2022).
80. MacDonald, N.; Haight, C. DevOpsSec: Creating the Agile Triangle. Available online: https://www.gartner.com/en/documents/1896617 (accessed on 25 October 2022).