Security as Culture

A Systematic Literature Review of DevSecOps

Mary Sánchez-Gordón
Department of Computer Sciences
Østfold University College
Halden, Norway
mary.sanchez-gordon@hiof.no

Ricardo Colomo-Palacios
Department of Computer Sciences
Østfold University College
Halden, Norway
ricardo.colomo-palacios@hiof.no

ABSTRACT

DevOps goes beyond automation, continuous integration and delivery processes, since it also encompasses people. In fact, DevOps promotes the collaboration between the development team and the operations team. When security comes into DevOps routines, people play an even more relevant role involving the collaboration between those teams and security team. Moreover, security is especially relevant while developing critical systems where we need to manage goals, risks and evidences. After implementing security into the DevOps toolchain, work only starts. We also need to start with behavioral changes in order to create a security culture. Several authors underlined DevSecOps, as one of the proposals for solving or, at least, minimizing this challenge. However, to date, the characterization of such a culture remains unclear. In this paper, a Systematic Literature Review was carried out to provide a better understanding of this topic from the human factor´s perspective. However it raises the following question: Is DevSecOps going to become mainstream?

CCS CONCEPTS

•Security and privacy~Software and application security • Social and professional topics~Professional topics~Computing profession

KEYWORDS

Security, DevSecOps, Culture, Human factors, Systematic Literature Review.

ACM Reference format:

Mary Sánchez-Gordón and Ricardo Colomo-Palacios. 2020. Security as Culture: A Systematic Literature Review of DevSecOps. In *Proceedings of 1st International Workshop on Engineering and Cybersecurity of Critical Systems (EnCyCriS 2020).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3387940.3392233

ICSEW'20, May 23–29, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7963-2/20/05…$15.00
https://doi.org/10.1145/3387940.3392233

1 Introduction

In the business world, the demand for agility and speed continues to grow. Advancements in technology such as Continuous Engineering, in particular DevOps, allowed some organizations to gain a competitive advantage [11]. However, security concerns have risen because of security breaches, such as massive data breach and leaks, which are forcing organizations worldwide to pay significantly attention to security threats [8]. This is especially true in the context of safety-critical systems, given the possible consequences of security incidents, e.g. loss of life, loss or misuse of sensitive information and major financial loss.

In this scenario, high levels of security integration into DevOps are needed. Thus, the need for security to be integrated in DevOps as DevSecOps was first mentioned in 2012. However, according to "The DevOps Adoption Playbook" [19], DevOps is not designed to maximize speed at the expense of security while "The DevOps Handbook" presents security as a part of DevOps [9]. In spite of the fact that security integration should be a natural part of any DevOps effort, security practices seem to become significant in the higher stages of DevOps implementation [11]. It also seems that the term DevSecOps helps to draw attention to the importance of building security into all aspects of software delivery [7,11]. In fact, there is a growing number of articles that evidence an increasing awareness, recognition and use of DevSecOps approaches [14], even to tackle the technical debt associated with cybersecurity attack tactics [8]. However, it is worth noting that there are other terms used in the industry such as SecDevOps and DevOpsSec.

According to a recent multivocal literature review of DevSecOps [14], DevSecOps is seen as a necessary expansion of DevOps that aims to integrate security controls and processes into the DevOps software development cycle by promoting the collaboration among security teams, development teams and operations teams. Moreover, other previous research on DevSecOps [16], revealed that culture, automation, measurement and sharing (CAMS) are important factors to consider, in similar fashion to DevOps. Thus, an organization cannot just buy or hire its way into DevOps, and the same holds true for DevSecOps. In fact, culture has been recognized as an essential part of both, but DevSecOps emphasizes the importance of creating a security culture [16].

In the light of that, although a secondary study about DevOps culture exists [18], to the best of our knowledge, a study on the cultural side of DevSecOps is not available in the literature. Therefore, this paper aims to bridge this gap by conducting a systematic literature review (SRL) on this topic.

2 Research Approach

For this study, we adopted a traditional Systematic Literature Review (SLR) [10,18]. This tool would be adequate to analyze the state of the art of academic literature related to DevSecOps culture. Authors derived a protocol that comprises a research question (RQ), search procedures, and inclusion and exclusion criteria:

RQ: *What is reported on the scientific literature on SecDevOps culture?*

Regarding the searching procedures, we used structured search, i.e., a search was performed out on Google Scholar because it covers all major publisher venues (e.g. Elsevier ScienceDirect, Springer, ACM and IEEE). The following terms related to security into DevOps were selected in order to capture relevant data to answer the research question: "SecDevOps", "DevSecOps", "DevOpsSec" AND "culture". As inclusion criteria (I1), we used the sources addressing the aforementioned terms. The exclusion criteria were: (E1) repeated or duplicate sources (in this case, only the most complete source was considered); (E2) material not written in English or Spanish; and (E3) inaccessible sources. Moreover, a data extraction form was created to capture details from the data sources including bibliographic information and relevant information for answering the RQ.

In the execution stage, we performed the structured search in Jan 2020, collecting the materials by applying the inclusion and exclusion criteria in the searched hits. From the 148 search results, we identified 63 relevant items by reading title and abstract. These sources loosely mention DevSecOps but do not discuss it on a consistent basis for culture. By reading the full-text, 11 items were selected as primary studies. Screening was completed using Covidence software. Moreover, consensus meetings were held to solve disagreements and uncertainty. The extraction process was carried out in such a way that a researcher extracted the relevant data and after that, another researcher reviewed that extraction. Due to the limited number of primary studies, evidence from all types of primary studies was considered.

3 Characterizing DevSecOps culture

The identified attributes were classified based on a categorization scheme for DevOps culture proposed by [18]. However, 2 out of 13 attributes were renamed to best suit our findings: (i) "New personnel and ideas" to "Hiring new personnel"; and (ii) "Improvement cycle" to "Continuous improvement mindset". Although the period for the review was not limited, 11 papers were found within the years 2016 (1), 2017 (1), 2018 (4), and 2019 (5). The research methods were interview (4), focus group (3), and survey (1). Moreover, we identified one experiment, one case study and one solution proposal without validation.

Figure 1 provides an overview of the findings. Overall, this indicates that DevSecOps culture is little explored by the academic community. However, despite the lack of empirical evidence, we believe that our findings provide interesting insights that could be a good starting point for further research on this topic.

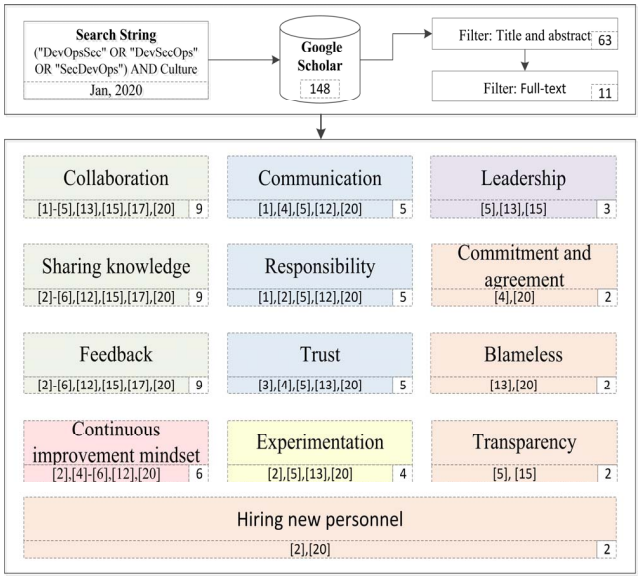

Figure 1: Overview of the categorization scheme for DevSecOps Culture

The findings of this review reveal that the soft side of DevSecOps is not always confessed among practitioners and researchers, which is in line with a previous study of DevOps culture [18]. As a result of the characterization process, we identified 13 attributes of DevSecOps culture which are briefly described below.

Collaboration in DevSecOps refers to the concept of integrating security principles through increased collaboration among the development teams, operations teams, and security teams of an organization [17]. To do so, it is necessary to adopt appropriate culture change and to build an integrated team with members of all the teams [1]. Therefore, DevSecOps is about promoting inclusion and working as a team in order to build secure software [3]. It means continuous collaboration.

Sharing Knowledge is related to education and cross-training for members of all the teams −the development teams, operations teams, and security teams. The aim is to help build security into the DevOps process [12]. In this approach, security automation tools are important, however tools cannot be used without appropriate knowledge, as mentioned practitioners in [20]. In consequence, tools alone are not enough to ensure security but also members of the security team need to share their knowledge.

One way to do that is to have "security champions". Security champions are programmers who have the most security-training on the team and who care about security [20]. Thus, the security team teaches one of the developers about security in order to make him/her care about security [3,6,20], and then, (s)he disseminates the knowledge to the rest of the team [3]. The collaborative culture could be used as a vehicle to promote continuous learning.

Feedback (Continuous and immediate) could solve the problems derived from the lack of an inclusive feedback loop between developers, security professionals, and operations team members [15]. To do so, the security team should explore existing DevOps automation activities and then, security tools should be customized in a way that ensures a short feedback cycle between the security team and the other teams [17]. However, a major challenge identified by practitioners is to give developers good guidance on what they should be looking for in the feedback loop of their DevSecOps process to ensure security [2].

In this sense, some of the security processes take place inline so that the ability to perform an incident response could be really quickly [3]. On the other hand, other security processes can be out-of-band activities, i.e. they can take feedback from the field and route back into the DevOps processes [12]. Moreover, according to practitioners in [5], post-mortems that involve everyone could help to break down silos.

Continuous improvement mindset is the ability to include activities that adequately address not just quality but also security while maintaining the fast pace of delivering code to production [12]. That also implies continuously monitoring the security of applications, as tools, standards and threats evolve [2]. Therefore, measurements need to be applied continuously [20] in order to identify and make incremental improvements [5].

Communication in DevSecOps means that the security team needs to talk to the DevOps team and ask: "How are we going to help you? How are we not going to slow down your process? How are we going to trigger out-of-band activities?" [12]. Apart from that, it is important to eliminate communications and bureaucratic barriers [1]. Moreover, to ensure that communication occurs at the right time and that the delivery ability is continuous, the security team should be involved as early as possible [4]. By shifting security left, the security team can run static analysis in the developer's integrated development environment (IDE) [12]. Then, developers can quickly find issues and check the code. Once the static analysis is finished, the security team can perform software composition analysis to find issues with the libraries or other software components. In most cases, the next step will be to deploy a test or staging environment that allows the application of Dynamic Application Security Testing (DAST) or Interactive Application Security Testing (IAST). At this point, it is worth noting that the right level of automation of these activities is a key enabler of DevSecOps since automated security tests allow processes to be predictable and scalable [20] while making the team more agile.

On the other hand, security champions work as a bridge between developers and other teams. They could be called if a security issue is identified externally to convey the message to the team from one of their own [12,20]. Practitioners also reported that dashboards for different services and warnings could give on instant messaging platform such as Slack, if anything critical happens [20].

Responsibility (personal/mutual) underlines building shared ownership and responsibility for security aspects [1]. That means creating a company culture where security is considered to be everyone's responsibility [5]. If all the team members accept shared responsibility of risks, they become stakeholders in the success of the delivery of software [1]. However, they do not need to become security experts, but their security knowledge should be good enough to be reasonable about it [20].

Trust as foundation of DevSecOps should be built between security team and other teams [5] in order to adopt security practices on a daily basis. The challenge is to build a representative group by identifying the right people at the appropriate time in order to make better decisions [4]. Developers report that they feel attacked by security professionals if they create vulnerable code [13,20]. Even more, developers feel like they are forced to take considerations they do not wish to take [20]. On the other hand, security professionals who are not integrated into the development team are often regarded by development teams with disdain and lack of respect [3]. Insecurities based on lack of security knowledge prevent both developers and security professionals from fully trusting each other [13]. One way to overcome this challenge is to sponsor security champions [3].

Experimentation is needed in DevSecOps due to automated security. It does not seem to be an easy task that likely differs in nature from organization to organization [20]. Indeed, selecting the right tools for the toolchain is a recognized challenge for practitioners in [2]. They agreed that a place to compare tools would be useful and would improve the current practice of trial and error when it comes to tool selection. Moreover, practitioners in [13] state that continual experimentation helps to understand how to use tools. Opportunities to learn from mistakes and to make incremental improvements are also suggested by practitioners in [5].

Leadership is also needed [13] not only to encourage cultural change, but also to grow and support the DevSecOps culture [5]. Although, it is worth noting that further leadership actions are required to really enforce DevSecOps as something that needs to be adopted widely across organizations [15].

Commitment and agreement is an interesting aspect in which practitioners reported that a security compromise is not enough to create a security-culture shift [20]. However, the experience report of the implementation of DevSecOps in the context of Systems of Systems (SoS) highlights the importance of encouraging personal commitment of all the staff involved in the project to conform to the policy requirements [4].Additionally, a security champion could help to make sure that security is not overlooked in the process.

Blameless in terms of blameless retrospectives is mentioned in [20]**.** However, practitioners report conflict between security and

development when the first ones criticize the developers` work [20] or they shame developers [13]. That behavior triggers negative emotions and even a minor amount of these emotions can be damaging to a work relationship.

Hiring new personnel that has a technical and procedural understanding of DevSecOps is a major challenge discussed in [2]. In particular, according to the practitioners in [20], it is difficult to find staff who knows how to use static analysis tools and staff who is able to recognize false positives that the tools often produce.

Transparency should be key to DevSecOps enlightenment but the papers in this review do not mention explicitly this attribute. Practitioners in [5] just recommend to build a dependency tree of all components and make sure that teams understand all the dependencies and associated risks in software. While, the development model based on DevSecOps proposed in [15] highlights that it is important to maintain full visibility between all parties within the model.

3 Conclusions and Outlook

Despite the popularity and perceived benefits, software security aspects of DevOps remain a concern for organizations that want to adopt it [11]. To deal with some security issues, culture is an essential element that needs to be adequately addressed in DevSecOps.

The scarce number of studies in this review is a major limitation that reveals DevSecOps culture as an emerging topic that deserves more research, in particular empirical research. Our findings bring detailed insights into the DevSecOps culture. Beyond moving some security practices to an earlier phase of the software lifecycle, DevSecOps culture helps to adopt a different way of working, one that emphasizes cross-team collaboration in the light of security. Therefore, we believe that this review provides a good first overview of DevSecOps culture, but it raises another question as well: Is DevSecOps going to become mainstream?

REFERENCES

[1] Justin F Brunelle, AJ Bognar, Vibha Dhawan, Nicole Gong Parrish, Andrew King, Vidyababu Kuppusamy, and Mano Malayanur. 2018. *Federal Cloud & Data Center Summit Report*. The MITRE Corporation. Retrieved from https://www.mitre.org/sites/default/files/publications/PRS18-2725-1_june2018_federal_cloud__data_center_summit_report.pdf

[2] Justin F Brunelle, Cameron Boozarjomehri, David Hansen, Christine Kim, R Scott Paul, Quang Nguyen, Rock Sabetto, Gavin Schmidt, Mari Spina, Joseph Walter, Katy Warren, Adam Yee, and Tom Suder. 2019. *Federal Cloud & Infrastructure Summit Report*. The MITRE Corporation. Retrieved from https://atarc.org/wp-content/uploads/2019/08/Cloud-White-Paper-Cover-Letter-merged.pdf

[3] Kim Carter. 2017. Francois Raynaud on DevSecOps. *IEEE Softw.* 34, 5 (2017), 93–96. DOI:https://doi.org/10.1109/MS.2017.3571578

[4] Sara Carturan and Denise Goya. 2019. Major Challenges of Systems-of-Systems with Cloud and DevOps – A Financial Experience Report. In *2019 IEEE/ACM 7th International Workshop on Software Engineering for Systems-of-Systems and 13th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems*, Montreal, QC, Canada, 10–17. DOI:https://doi.org/10.1109/SESoS/WDES.2019.00010

[5] Michelle Casagni, Melissa Heeren, Rick Cagle, Richard Eng, Jennifer Flamm, Seth Goldrich, Diane Hanf, Michael Kristan, Justin F Brunelle, Tim Harvey, and Tom Suder. 2018. *Federal DevOps Summit Report*. The MITRE Corporation. Retrieved from https://atarc.org/wp-content/uploads/2019/01/2018-03-01-ATARC-Federal-DevOps-Summit-White-Paper-1.pdf

[6] Rebecca Deck. 2019. Adapting AppSec to a DevOps World. Retrieved from https://pdfs.semanticscholar.org/74c3/ce0f45a4624b9a0d67051d8ea305c3b8be78.pdf

[7] Shamayel M. Farooqui. 2018. Conclusion: The New Era. In *Enterprise DevOps Framework: Transforming IT Operations*, Shamayel M. Farooqui (ed.). Apress, Berkeley, CA, 107–117. DOI:https://doi.org/10.1007/978-1-4842-3612-3_10

[8] Clemente Izurieta and Mary Prouty. 2019. Leveraging SecDevOps to Tackle the Technical Debt Associated with Cybersecurity Attack Tactics. In *2019 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 33–37. DOI:https://doi.org/10.1109/TechDebt.2019.00012

[9] Gene Kim, Patrick Debois, John Willis, and Jez Humble. 2016. *The DevOps Handbook: How to create world-class agility, reliability, & security in technology organizations* (First edition ed.). IT Revolution Press, LLC, Portland, OR.

[10] Barbara Kitchenham and S. Charters. 2007. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. School of Computer Science and Mathematics, Keele University.

[11] Andi Mann, Michael Stahnke, Alanna Brow, and Nigel Kersten. 2019. *2019 State of DevOps Report*. PuppetLabs. CircleCI and Splunk. Retrieved from https://puppet.com/resources/report/state-of-devops-report/

[12] Steve Mansfield-Devine. 2018. DevOps: finding room for security. *Netw. Secur.* 2018, 7 (July 2018), 15–20. DOI:https://doi.org/10.1016/S1353-4858(18)30070-9

[13] Gary McGraw. 2018. Silver Bullet Talks with Tanya Janca. *IEEE Secur. Priv.* 16, 5 (September 2018), 7–11. DOI:https://doi.org/10.1109/MSP.2018.3761705

[14] Havard Myrbakken and Ricardo Colomo-Palacios. 2017. DevSecOps: A Multivocal Literature Review. In *International Conference on Software Process Improvement and Capability Determination*, Springer, 17–29.

[15] Jessica Nguyen and Marc Dupuis. 2019. Closing the Feedback Loop Between UX Design, Software Development, Security Engineering, and Operations. In *Proceedings of the 20th Annual SIG Conference on Information Technology Education - SIGITE '19*, ACM Press, Tacoma, WA, USA, 93–98. DOI:https://doi.org/10.1145/3349266.3351420

[16] Pulasthi Perera, Roshali Silva, and Indika Perera. 2017. Improve software quality through practicing DevOps. In *2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, 1–6. DOI:https://doi.org/10.1109/ICTER.2017.8257807

[17] Akond Ashfaque Ur Rahman and Laurie Williams. 2016. Software Security in DevOps: Synthesizing Practitioners' Perceptions and Practices. In *2016 IEEE/ACM International Workshop on Continuous Software Evolution and Delivery (CSED)*, 70–76. DOI:https://doi.org/10.1109/CSED.2016.021

[18] Mary Sánchez-Gordón and Ricardo Colomo-Palacios. 2018. Characterizing DevOps Culture: A Systematic Literature Review. In *Software Process Improvement and Capability Determination* (Communications in Computer and Information Science), Springer International Publishing, Cham, 3–15. DOI:https://doi.org/10.1007/978-3-030-00623-5_1

[19] Sanjeev Sharma. 2017. *The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise | Wiley*. John Wiley & Sons Inc. Retrieved January 7, 2020 from https://www.wiley.com/en-us/The+DevOps+Adoption+Playbook%3A+A+Guide+to+Adopting+DevOps+in+a+Multi+Speed+IT+Enterprise-p-9781119310761

[20] Nora Tomas, Jingyue Li, and Huang Huang. 2019. An Empirical Study on Culture, Automation, Measurement, and Sharing of DevSecOps. In *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, 1–8. DOI:https://doi.org/10.1109/CyberSecPODS.2019.8884935