JURNAL LOCUS: Penelitian & Pengabdian
Volume 4 No. 11 November 2025
E-ISSN 2829-7334| P-ISSN 2829-5439
Hompage: https://locus.rivierapublishing.id/index.php/jl

Analysis of the Implementation of DevSecOps Policies and Technology towards Reducing the Number of Vulnerabilities in the Telecommunications Industry: Case Study of MyApps Application at PT XYZ

Dimas Prayogo*, Kalamullah Ramli
Universitas Indonesia
Email: dimas.prayogo@ui.ac.id*

Abstract:
The acceleration of digitalization is highly needed by the telecommunications industry to compete in both national and international markets. To meet this challenge, the telecommunications industry has begun to implement a new approach in application development and deployment, namely by utilizing cloud computing and agile methods. In this case, PT XYZ applies the DevSecOps approach so that each development cycle includes aspects of development speed, security, and operations in an integrated manner. However, in the early stages of implementing the DevSecOps policy, various vulnerabilities were found in the application being developed, in this case the MyApps application. This problem shows that a technology or early detection mechanism is needed to identify vulnerabilities before the application enters the production stage. Therefore, this study was conducted with the aim of reducing the number of vulnerabilities in the MyApps application, thereby enabling safer and more efficient application development. The results of the study showed a reduction in the number of vulnerabilities from SAST by 78.1%, from container scans by 86.7%, and total vulnerabilities by 83.6% in the MyApps application, thus minimizing the risk of cyber attacks in the future.

Keywords: *DevSecOps, Vulnerability, Agile, Cloud Computing*

Corresponding: Dimas Prayogo
E-mail: dimas.prayogo@ui.ac.id

INTRODUCTION

In 2020, the telecommunications industry experienced a significant acceleration in digitalization in response to the need for more efficient and innovative communication services (Amankwah-Amoah et al., 2021; Döhring et al., 2021; Gavrila Gavrila & de Lucas Ancillo, 2021). This transformation includes the integration of digital technologies into every aspect of the business, changing the way services are delivered, managed, and accessed by customers (Subagyo & Ramli, 2022).

One of the strategic adopted is fundamental changes in infrastructure and service development approaches, including migration from on-premise systems to cloud computing and the adoption of more agile software development methodologies such as SDLC (Kumar & Goyal, 2020).

In this context, DevOps (SDLC) emerged as an approach that integrates development and operations to accelerate the software development cycle (Battina, 2021; Byrne & Cevenini, 2022; Kolawole & Fakokunde, 2024). However, security challenges in the DevOps cycle have driven the evolution towards DevSecOps, which emphasizes the integration of security practices early in the development process. DevSecOps ensures that security becomes an

integral part of the software lifecycle, rather than a separate, final consideration (Alonso et al., 2023).

However, the adoption of DevSecOps is not without its challenges. Systematic studies have identified several barriers to implementing DevSecOps, including the need for effective security automation tools and seamless integration into DevOps workflows. In addition, integrating security tools into DevOps often faces difficulties in maintaining the speed and frequency of software deployments (Subagyo & Ramli, 2022).

To address these challenges, technologies such as SAST and Runtime Scan have been proposed as solutions to detect security vulnerabilities early in the development cycle (Andayana, 2023; Effendi & Pribadi, 2021). SAST and Runtime Scan enable the identification of vulnerabilities before software reaches production, thereby improving security without sacrificing development speed (Popentiu-Vladicescu & Albeanu, 2022).

Previous studies have highlighted both the potential and the challenges in implementing DevSecOps effectively. Rajapakse et al. (2022) conducted a systematic review that identified several major barriers to DevSecOps adoption, including the lack of automated security integration, immature tool interoperability, and cultural misalignment between development and security teams. While their study comprehensively mapped these challenges, it did not empirically measure the effectiveness of specific security tools---such as Static Application Security Testing (SAST) and Runtime Container Scanning---in reducing vulnerabilities during real-world implementations. Meanwhile, Li et al. (2023) compared the detection performance of various SAST tools and revealed significant discrepancies in accuracy and coverage. However, their experiment was conducted under controlled laboratory conditions, lacking a holistic assessment of DevSecOps integration in production environments that involve CI/CD pipelines and runtime monitoring.

This study measures the impact of implementing DevSecOps policies and technology on reducing security vulnerabilities in PT XYZ applications (MyApps). The study is divided into 2 research type quantitative and qualitative. Where in qualitative research, the researcher will provide 2 surveys: survey 1 before the application implements DevSecOps policies and technology, and survey 2 after the application implements DevSecOps policies and technology. While for quantitative research, the researcher will conduct 2 scanning phases with 2 types of scanning (SAST Scan, & Runtime Scan), scanning phase 1 before the application implements DevSecOps policies and technology and scanning phase 2 after the application implements DevSecOps policies and technology. The research ended by comparing the results of the three applications and measuring the number of vulnerabilities in each application (Truong & Klein, 2020).

The rest of this work is designed as follows. The literature review, related works, and measuring method are all described in Section II. The technique, data collection, and hypothesis are all described in Section III. After explaining the comparison of the scanning findings and interpreting them into a defined analysis, Section IV examines the assumptions. Section V finishes with a summary of the findings and research recommendations (Sojan et al., 2021). The findings are expected to contribute practically by providing a validated

implementation roadmap and theoretically by enriching the existing body of knowledge on security-driven software development governance in the telecommunications sector.

RESEARCH METHOD

This study involved 5 people from 5 different divisions of PT XYZ. The table below shows the people that the researcher considered as expert comments.

Table 1. List of IT People

Data	Tittle Position	Division
R1	Head of IT Development	IT Development & Digital
R2	Head of IT Operations	IT Operations & Infrastucture
R3	Head of IT Architect	IT Architect & Business
R4	Head of IT Cloud	IT Cloud
R5	Head of IT Security	IT Cybersecurity

This study also uses an applications (MyApps) as case study materials and researchers used 2 different versions of each application. The version before implementing DevSecOps policy and technology, and the version after implementing DevSecOps policy and technology.

Table 2. List of Applications

Application Name	Version
MyApps	7.0.0
	8.0.0

Vulnerability Scanning Tools & Technique

This study uses vulnerability scanning with two main tools, namely Veracode for Static Application Security Testing (SAST) scans at the code stage and Sysdig for runtime scans in the build environment, with both tools applying OWASP Top 10-based techniques to identify security vulnerabilities in applications, and using the Common Vulnerability Scoring System (CVSS) as a metric to assess the severity of the detected vulnerabilities.

Research Scenario

Qualitative Scenario

Survey scenario with questionnaire divided 2 phases:

1. The first phase is carried out before employees receive DevSecOps Policy training. This is to observe the DevSecOps awareness level of XYZ employees before completing the training.
2. The second phase is carried out after employees receive DevSecOps Policy training. This is to observe the DevSecOps awareness level of XYZ employees after completing the training.

The results of the first phase of the survey will be compared with the results of the second phase of the survey to measure the effectiveness of the training provided to employees.

Quantitative Scenario

Vulnerability scan scenario with Veracode and sysdig divided 2 phases in every version of applications:

1. The first phase is carried out before applications integrate and implement DevSecOps technology. This is to observe the total vulnerability of applications before completing the integration and implementation.
2. The second phase is carried out after applications integrate and implement DevSecOps technology. This is to observe the total vulnerability of applications after completing the integration and implementation.

The results of the first phase of the scanning will be compared with the results of the second phase of the scanning to measure the effectiveness of the integration and implementation provided to the applications.

DevSecOps Policy and Technology Training

DevSecOps policy and tehcnology awareness training module was developed in this study that is expected to be able to help to raise the level of DevSecOps awareness among development and operations and could decrease total vulnerability in every application. DevSecOps policy and tehcnology training is given to development and operations in the form of an online module. The material of the module consists of below topics:

1) Introduction of Information Security
2) What is a DevSecOps
3) Type of DevSecOps Technology
4) Implementation DevSecOps Technology policy
5) How to detect and remediate vulnerability findings from DevSecOps Technology

After completing the learning module, employees must be able to complete post-test questions with a minimum score of 80 to pass. If they didn't pass the minimum score, they should retake the post-test until they reach the minimum score.

Project Timeline

We divided the project timeline into 5 activity, 2 phases for survey, 1 phases for DevSecOps training, and 2 phases for scanning. This timeline to make sure every activity could give best result which mean to minimize total vulnerability in the applications. In detail, the project timeline can be seen in Table III below.

Table 3. List of Activities

Week	Activity
Week 1	Vulnerability SAST Scan & Runtime Scan
Week 2	Survey the expert with questionnaire
Week 3	DevSecOps Policy and Technology Training
Week 4	
Week 5	Vulnerability SAST Scan & Runtime Scan
Week 6	Survey the expert with questionnaire

RESULTS AND DISCUSSION

Vulnerability SAST Scan & Runtime Scan Result Before DevSecOps Training

This experiment was conducted by researchers to check the number of vulnerabilities in the MyApps application version 7.0.0. Researchers used the vulnerability scanning method with the OWASP Top 10 approach using the Veracode tool (SAST Scan) to determine the number of vulnerabilities from the code stage and Sysdig tools (Runtime Scan) to determine the number of vulnerabilities from the build stage. At this stage, the developers of the MyApps application have not received training related to DevSecOps, both policy and technology.

SAST scanning shows a dominance of vulnerabilities with a total severity (507 cases), with high findings in SQL Injection and CSLR Injection. While the runtime scan shows a higher number of vulnerabilities overall (862 cases), with significant high severity (216 cases), mainly due to an outdated version of Apache Tomcat and kernel issues. This indicates that the MyApps application has significant vulnerabilities both at the code stage and in the runtime environment, requiring special attention for mitigation, especially for vulnerabilities with high severity.

Below are the results of a vulnerability scan from SAST using Veracode:

Table 4. List of Security Findings

Severity	# Vulnerabilities
Critical	0
High	22
Medium	464
Low	21

Below are the top 3 vulnerabilities from the SAST scan:

Table 5. List of Top 3 Vulnerability

Vulnerability Name	Severity
SQL Injection	High
CSLR Injection	High
Cryotograpic Issue	Medium

Below are the results of a vulnerability scan from a runtime scan using sysdig:

Table 6. List of Vulnerability Runtime

Severity	# Vulnerability
Critical	1
High	216
Medium	645
Low	13

Below are the top 3 vulnerabilities from Runtime scan:

Table 7. List of Top 3 Vulnerability

Vulnerability Name	Severity
Outofdate Apache Tomcate Version	High
Kernel Obsolate	High

Broken SSL	Medium

Survey Results of Experts Before DevSecOps Training

The survey was conducted by giving a questionnaire consisting of 10 questions to 5 respondents who were experts in their fields at PT XYZ.

Table 8. List of Respondent

Name Code	Position
R1	Head of IT Development
R2	Head of IT Operation
R3	Head of IT Architect
R4	Head of IT Cloud
R5	Head IT Security

The following are the results of the questionnaire responses given to five stakeholders at PT XYZ Head of IT Development, Head of IT Operation, Head of IT Infrastructure, Head of IT Cloud, and Head of IT Security. The responses reflect the condition that the current SDLC has not been able to support security effectively and has not succeeded in reducing security gaps in the MyApps application.

Table 9. List of Respondent Answer

Question	Head of IT Development	Head of IT Operations	Head of IT Architect	Head of IT Cloud	Head of IT Security
How effective is PT XYZ's current SDLC in integrating secure coding practices (e.g., threat modeling or secure coding) at each stage of application development?	Less effective	Not effective	Less effective	Not effective	Less effective
Does PT XYZ's SDLC include a formal process to identify and mitigate security vulnerabilities such as SQL Injection or XSS during the planning and design phases?	No, only ad-hoc	No formal process	No, only ad-hoc	No, only ad-hoc	No, only ad-hoc
How often do developers receive security training to recognize and prevent application vulnerabilities?	Never	Never	Every 12 months	Every 12 months	Every 12 months
Does PT XYZ's SDLC pipeline use automated mitigation tools such as SAST (Static Application Security Testing) to detect	No, only manual mitigation	No, only manual	No, only manual	No, only manual	No, only manual

Question	Head of IT Development	Head of IT Operations	Head of IT Architect	Head of IT Cloud	Head of IT Security
code-level vulnerabilities?					
How quickly does the development team handle critical vulnerabilities (e.g., Remote Code Execution) detected during development or production?	Within 1 month	Within 1 month	No clear process	Within 1 month	Within 1 month
Does PT XYZ's SDLC have a vulnerability remediation policy based on risk level (e.g., CVSS score)?	Yes, but not consistent	No policy	In development	In development	In development
How often are code review and security checks performed during the SDLC?	Only during major releases	Only during major releases	Never	Only during major releases	Only during major releases
Does PT XYZ's SDLC support runtime mitigation and protection tools to reduce production environment vulnerabilities?	No runtime mitigation	No runtime mitigation	No runtime mitigation	No runtime mitigation	No runtime mitigation
Does PT XYZ's SDLC ensure compliance with security standards such as OWASP Top 10 or ISO 27001 during application development?	During adjustment process	During adjustment process	During adjustment process	No compliance process	No compliance process
What are the main challenges in PT XYZ's current SDLC that hinder effective security vulnerability reduction during application development?	Lack of developer security training leads to delayed detection of SQL Injection, XSS, and similar issues.	Limited collaboration between security and development teams causes delayed response to critical vulnerabilities.	Lack of security-focused architecture guidance causes vulnerabilities not being mitigated early.	Infrastructure does not support automated scanning, and the team lacks tools for misconfiguration detection.	Cloud environment not aligned with security practices and lacks integration with tools like SAST or Container Security.

The responses from the five stakeholders indicate that the current SDLC at PT XYZ is not yet able to support security effectively. The majority of respondents (80%) stated that the integration of security practices in the SDLC is "less effective" or "ineffective". The absence of a formal process for vulnerability mitigation (80%) and minimal security training (80% stated "never") are the main factors. Vulnerability scans such as SAST and Runtime are not widely used (80% reported "not at all" or "only manual"), and there is no policy for prioritizing fixes (80% stated "no policy"). Security code reviews are rarely conducted (60% "never"), and there is no WAF/runtime protection mitigation (80% "no mitigation"). Compliance with standards such as OWASP Top 10 is also minimal (80% "no compliance"). The main

challenges include lack of training, weak team collaboration, and lack of security automation, which is in line with the initial findings of the study where the total vulnerability from SAST scans was 507 cases and from runtime scans was 862 cases (Wang et al., 2022)..

Vulnerability SAST Scan & Runtime Scan Result After DevSecOps Training

The researcher again running SAST scan and runtime scan after the training has been given to the stakeholder team. At this time researcher using MyApps application version 8.0.0. Researcher also use the same tools and the method like previous experiment. The result of the scan indicates that total vulnerability decreases compared to the previous experiment.

SAST scan results show there is no high severity findings with total severity (92 cases) and from Sysdig scan result show there is no critical findings and for high findings is decrease from previous scan with total security (116 cases). This indicates that the MyApps application has significantly reduce vulnerabilities both at the code stage and in the runtime environment.

Below are the results of a vulnerability scan from SAST using Veracode:

Table 10. SAST result after training

Severity	# Vulnerabilities
Critical	0
High	0
Medium	81
Low	11

Below are the top 3 vulnerabilities from the SAST scan:

Table 1. Top 3 Vulnerability after training

Vulnerability Name	Severity
Authorization Issue	Medium
CSLR Injection	Medium
Cryotograpic Issue	Medium

Below are the results of a vulnerability scan from a runtime scan using sysdig:

Table 2. Sysdig result after training

Severity	# Vulnerability
Critical	0
High	81
Medium	34
Low	1

Below are the top 3 vulnerabilities from Runtime scan:

Table 3. Top 3 vulnerability after training

Vulnerability Name	Severity
Outofdate OpenSSL Version	High
Kernel Obsolete	High
Obsolete Library	Medium

Survey Results of Experts After DevSecOps Training

The survey was conducted by giving a questionnaire consisting of 10 questions to 5 respondents who were experts in their fields at PT XYZ.

Table 4. Respondent Survey

Name Code	Position
R1	Head of IT Development
R2	Head of IT Operation
R3	Head of IT Architect
R4	Head of IT Cloud
R5	Head IT Security

The following are the results of the questionnaire responses given to five stakeholders at PT XYZ Head of IT Development, Head of IT Operation, Head of IT Infrastructure, Head of IT Cloud, and Head of IT Security. The responses reflect the condition that the DevSecOps has been able to support security effectively and has succeeded in reducing security gaps in the MyApps application.

Table 5. Survey result after training

Question	Head of IT Development	Head of IT Operations	Head of IT Architect	Head of IT Cloud	Head of IT Security
How effective is PT XYZ's DevSecOps policy in integrating SAST scan (Veracode) into the CI/CD pipeline to detect vulnerabilities such as SQL Injection or XSS during the code development phase?	Very effective	Quite effective	Quite effective	Very effective	Quite effective
Is the runtime scan (Sysdig) fully implemented in the production environment to monitor misconfiguration vulnerabilities, server dependencies, or real-time security threats?	Yes, fully implemented	Yes, fully implemented	Yes, fully implemented	Yes, fully implemented	Yes, fully implemented
How often does the application development team receive training related to SAST scan (Veracode) interpretation to address vulnerabilities such as OWASP Top 10?	Every 3 months	Every 6 months	Every 6 months	Every 3 months	Every 3 months
Does PT XYZ's DevSecOps policy ensure that SAST scan (Veracode) results are routinely analyzed and followed up to reduce vulnerabilities at the code stage?	Yes, regularly	Yes, regularly	Yes, regularly	Yes, regularly	Yes, regularly
How fast does the MyABC team handle	Within 24 hours	Within 1 week	Within 1 week	Within 24 hours	Within 24 hours

Question	Head of IT Development	Head of IT Operations	Head of IT Architect	Head of IT Cloud	Head of IT Security
critical vulnerabilities (e.g., Remote Code Execution, high CVSS score) detected through SAST or runtime scans?					
Does PT XYZ's DevSecOps policy use CVSS scoring to prioritize vulnerability remediation found in SAST or runtime scans (Sysdig)?	Yes, with clear priority levels	Yes, with clear priority levels	Yes, with clear priority levels	Yes, with clear priority levels	Yes, with clear priority levels
How often are SAST scan (Veracode) configurations updated to ensure detection of relevant vulnerabilities aligned with OWASP Top 10 and PT XYZ's applications?	Every code change	Regularly (weekly/monthly)	Regularly (weekly/monthly)	Every code change	Every code change
Is Sysdig configured in the production environment to detect and mitigate real-time vulnerabilities such as misconfigurations or dependency exploits?	Yes, with complete configuration	Yes, with complete configuration	Yes, with complete configuration	Yes, with complete configuration	Yes, with complete configuration
Does PT XYZ's DevSecOps policy ensure compliance with OWASP Top 10 standards based on results from SAST (Veracode) and runtime scans (Sysdig)?	Yes, fully compliant	Yes, fully compliant	Yes, fully compliant	Yes, fully compliant	Yes, fully compliant
What are the key supporting factors in PT XYZ's DevSecOps implementation, especially SAST (Veracode) and runtime scan (Sysdig), that help reduce security risks in application development?	Routine developer training enables early detection and fixing of vulnerabilities like SQL Injection or XSS via SAST.	Runtime automation with Sysdig enhances real-time monitoring of container misconfigurations.	Infrastructure team collaboration with IT architects improves Sysdig configuration for vulnerability mitigation.	Integration of SAST and runtime scan in the cloud-native pipeline supports consistent OWASP Top 10 detection.	CVSS scoring prioritization accelerates remediation of critical security vulnerabilities.

Comparison of Results Before and After Implementing DevSecOps Policies and Technologies

A comparison of the results of the first and second stage vulnerability assessments can be seen in the table below:

Table 6. Comparison vulnerability

Simulation	SAST Result	Container Scan Result	Total Vulnerability
Phase I	517	875	1392
Phase II	113	116	229

Based on the data processing results displayed in Figure 6, there is a significant decrease in the number of vulnerabilities after the DevSecOps approach was implemented. The results of the scan using the Static Application Security Testing (SAST) method showed a decrease from 517 vulnerabilities in the initial stage to 113 vulnerabilities after the intervention, which means a decrease of 78.1%. Meanwhile, the results of the container scan using Sysdig also showed a positive trend, with the number of vulnerabilities decreasing from 875 to 116, or equivalent to a decrease of 86.7%. Overall, the total vulnerabilities from the two scanning methods decreased from 1,392 to 229, reflecting a total decrease in vulnerabilities of 83.6%. This finding strengthens the evidence that DevSecOps integration, including the systematic use of SAST and container scanning, is effective in reducing potential security risks in developed applications (Dupont et al., 2021; Rahman & Parnin, 2023).

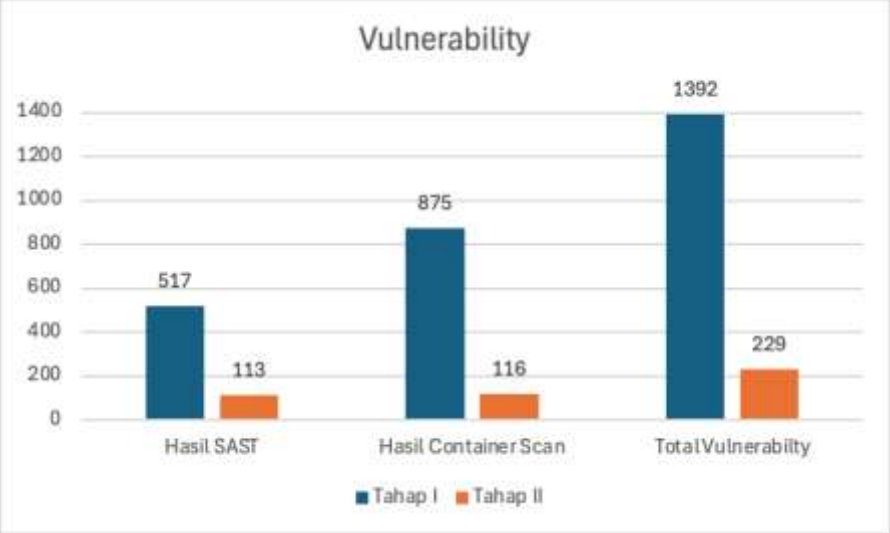

Figure 1. percentage vulnerability before after

CONCLUSION

The analysis highlights the essential role of DevSecOps in tackling security challenges within the rapidly digitalizing telecommunications industry by integrating security throughout the software development life cycle. Initial scans uncovered 507 SAST and 862 runtime vulnerabilities, including 22 high and 1 critical issues, revealing the importance of awareness programs. Following targeted education, there was an 83.6% overall reduction in vulnerabilities, with SAST vulnerabilities decreasing by 78.1% and container scan issues by 86.7%. The study recommends conducting regular SAST and container scans alongside quarterly DevSecOps training sessions. Future research should involve larger, more diverse samples and incorporate penetration testing to achieve optimal vulnerability reduction.

REFERENCES

Alonso, J., Piliszek, R., & Cankar, M. (2023). *Embracing IaC through the DevSecOps philosophy: Concepts, challenges, and a reference framework. IEEE Software, 40*(1), 56–62. https://doi.org/10.1109/MS.2022.3212194

Amankwah-Amoah, J., Khan, Z., Wood, G., & Knight, G. (2021). *COVID-19 and digitalization: The great acceleration. Journal of Business Research, 136,* 602–611.

Andayana, M. N. D. (2023). *Evaluation of the implementation of the Family Hope Program (PKH) in poverty alleviation. Integration: Journal of Social Sciences and Culture, 1*(4), 147–159. https://doi.org/10.38142/ijssc.v1i4.130

Battina, D. S. (2021). *The challenges and mitigation strategies of using DevOps during software development. International Journal of Creative Research Thoughts (IJCRT),* ISSN 2320–2882.

Byrne, K., & Cevenini, A. (2022). *Aligning DevOps concepts with agile models of the software development life cycle (SDLC) in pursuit of continuous regulatory compliance. Conference on Innovative Technologies in Intelligent Systems and Industrial Applications,* 359–374.

Döhring, B., Hristov, A., Maier, C., Roeger, W., & Thum-Thysen, A. (2021). *COVID-19 acceleration in digitalisation, aggregate productivity growth and the functional income distribution. International Economics and Economic Policy, 18*(3), 571–604.

Dupont, S., Mouton, S., De Oliveira, A. S., Lekens, T., Costante, E., Merlo, A., & Valenza, F. (2021). *Incremental common criteria certification processes using DevSecOps practices.* In *Proceedings - 2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW 2021)* (pp. 12–23). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/EuroSPW54576.2021.00009

Effendi, G. N., & Pribadi, U. (2021). *The effect of leadership style on the implementation of artificial intelligence in government services. IOP Conference Series: Earth and Environmental Science, 717*(1). https://doi.org/10.1088/1755-1315/717/1/012018

Gavrila Gavrila, S., & de Lucas Ancillo, A. (2021). *COVID-19 as an entrepreneurship, innovation, digitization and digitalization accelerator: Spanish Internet domains registration analysis. British Food Journal, 123*(10), 3358–3390.

Kolawole, I., & Fakokunde, A. (2024). *Improving software development with continuous integration and deployment for agile DevOps in engineering practices. International Journal of Computer Applications Technology and Research, 14*(1), 25–39.

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

Popentiu-Vladicescu, F., & Albeanu, G. (2022). *Increasing SoS dependability by DevSecOps.* In *ICETECC 2022 - International Conference on Emerging Technologies in Electronics, Computing and Communication.* Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICETECC56662.2022.10069468

Rahman, A., & Parnin, C. (2023). *Detecting and characterizing propagation of security weaknesses in Puppet-based infrastructure management. IEEE Transactions on Software Engineering, 49*(6), 3536–3553. https://doi.org/10.1109/TSE.2023.3265962

Sojan, A., Rajan, R., & Kuvaja, P. (2021). *Monitoring solution for cloud-native DevSecOps.* In *Proceedings - 2021 IEEE 6th International Conference on Smart Cloud (SmartCloud 2021)* (pp. 125–131). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/SmartCloud52277.2021.00029

Subagyo, E. P., & Ramli, K. (2022). *Analyzing the impact of information security awareness training to the employees of Telco Company XYZ. Budapest International Research and Critics Institute (BIRCI-Journal): Humanities and Social Sciences, 5*(2). https://doi.org/10.33258/birci.v5i2.4666

Truong, H. L., & Klein, P. (2020). *DevOps contract for assuring execution of IoT microservices*

in the edge. Internet of Things, 9, Article 100150. https://doi.org/10.1016/j.iot.2019.100150

Wang, Z., Guo, G., Liu, C., & Zhu, W. (2022). *Research on railway DevSecOps system construction based on "People–Process–Technology."* In *Proceedings - 2022 2nd International Signal Processing, Communications and Engineering Management Conference (ISPCEM 2022)* (pp. 19–23). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ISPCEM57418.2022.00010