Управление рисками и безопасностью

Enhancing Kubernetes Security: The Crucial Role of DevSecOps

Ghadeer Darwesh, Jaafar Hammoud, A.A. Vorobeva

ITMO University, Saint Petersburg, Russia

Abstract. This article highlights the significance of integrating DevSecOps (Development, security and Operations) practices into the research on detecting common attacks in Kubernetes environments. As Kubernetes gains rapid traction as a prominent container orchestration platform, the security challenges associated with containerized applications have grown in magnitude. However, traditional security methodologies often struggle to keep pace with the dynamic and fast-evolving nature of containerized environments, leaving potential vulnerabilities for malicious actors to exploit. By emphasizing the importance of DevSecOps, this article aims to underscore its role in improving the security posture of Kubernetes deployments and promoting a proactive approach to safeguarding containerized applications. The article also discusses key considerations and benefits of implementing DevSecOps in the context of Kubernetes security research.

Ключевые слова: *DevSecOps, Kubernetes security, DevOps, security practices.*

DOI: 10.14357/20790279240309 **EDN: SDRKHO**

Introduction

In recent years, Kubernetes has emerged as the leading container orchestration platform, revolutionizing the deployment and management of containerized applications. As Kubernetes adoption continues to soar, so does the need for robust security measures to protect these environments from emerging threats and vulnerabilities. In response, the concept of DevSecOps (Development, security and Operations) has gained prominence as a vital approach to integrating security practices throughout the entire software development lifecycle.

Kubernetes offers a highly scalable and flexible infrastructure for orchestrating containers, enabling organizations to efficiently manage complex distributed systems [1]. Its popularity stems from its ability to automate application deployment, scaling, and management, while providing features such as load balancing, service discovery, and self-healing. However, as organizations increasingly rely on Kubernetes to power critical applications, ensuring the security and integrity of these deployments becomes paramount.

The dynamic nature of containerized environments introduces unique security challenges. Misconfigurations, vulnerabilities in container images, unauthorized access, and attacks on the Kubernetes control plane are just a few examples of the risks that must be addressed. Therefore, security considerations should be an integral part of Kubernetes deployments, right from the initial development stages through to production.

DevSecOps, a methodology that combines development, operations, and security, aims to embed security practices throughout the software development

lifecycle. It emphasizes the collaboration and communication between development, operations, and security teams, enabling a proactive approach to security rather than a reactive one. By integrating security early and continuously into the development pipeline, organizations can identify and address security vulnerabilities at each stage, reducing the likelihood of security breaches and minimizing their impact [2].

In the context of Kubernetes, DevSecOps plays a crucial role in securing deployments and protecting containerized applications [3]. It involves implementing security measures as code, automating security processes, and incorporating security-focused metrics collection and analysis . DevSecOps helps organizations detect and prevent attacks, ensure compliance with industry regulations, and establish a culture of security awareness and accountability.

This article explores the growing adoption of Kubernetes, the increasing importance of security in Kubernetes environments, and the role of DevSecOps in securing Kubernetes deployments. By emphasizing the need for a proactive security approach and the integration of security practices throughout the development lifecycle, organizations can enhance the overall security posture of their Kubernetes deployments and safeguard their containerized applications against emerging threats.

1. Related work

The related work in the field of DevSecOps and Kubernetes security research spans a wide range of topics, including cultural adoption, practical implementation, emerging trends, and challenges faced during DevSecOps adoption. The following studies provide valuable insights for researchers, practitioners, and organizations seeking to enhance the security posture of their Kubernetes deployments. By leveraging the findings from existing research, researchers can contribute to the advancement of secure and resilient security solutions that safeguard Kubernetes environments and ensure the overall success of DevSecOps practices.

Mary and Ricardo in “A Systematic Literature-Review of DevSecOps” [4] provide a comprehensive analysis of the cultural aspects of DevSecOps adoption. The study highlights the significance of fostering a security-first mindset within organizations and promoting a collaborative and shared responsibility for security among development, operations, and security teams. By synthesizing findings from a wide range of sources, the article emphasizes the role of culture in successful DevSecOps implementation and advocates for continuous learning and knowledge-sharing to build a security-aware organizational culture.

On the practical implementation side, Rahul B. S [5]developing and updating of the latest security methods that tend to give adequate, proficient and productive results in a secured manner. Therefore, to demonstrate the DevSecOps we intend to use the open source tools which can be freely downloaded and used to demonstrate the data security. In order to run the frequent process of integrating and testing for the security of data, we require more resource and time respectively. To overcome this problem here we implementing and integrating security as part of the pipeline process and hence can achieve faster response time and fend off attacks beforehand and accordingly create a secure environment and more protected system.”,”author”:[{“dropping-particle”:””,”family”:”Rahul”,”given”:”S”,”non-dropping-particle”:””,”parse-names”:false,”suffix”:””}],”container-title”:”International Journal of Advance Research”,”id”:”ITEM-1”,”issued”:{“date-parts”:[[“2019”]]},”title”:”Implementation of DevSecOps using Open-Source tools”,”type”:”article-journal”},”uris”:[“http://www.mendeley.com/documents/?uuid=6b3df69e-c99a-3f14-9ff8-06c0e1d95062”]}],”mendeley”:{“formattedCitation”:”[5]”,”plainTextFormattedCitation”:”[5]”,”previouslyFormattedCitation”:”[5]”},”properties”:{“noteIndex”:0},”schema”:”https://github.com/citation-style-language/schema/raw/master/csl-citation.json”} offers valuable insights into integrating security practices within DevOps pipelines, with a specific focus on utilizing open-source solutions in Kubernetes environments. The research presents case studies of organizations that have effectively adopted DevSecOps and leveraged open-source tools and technologies to enhance security. This study serves as a practical guide for organizations seeking to embrace DevSecOps while capitalizing on open-source security solutions, providing real-world examples of successful implementation strategies.

Additionally, Runfeng Mao [6] provides a unique perspective by gathering insights from diverse sources such as blogs, whitepapers, and other non-peer-reviewed publications. By exploring emerging trends and challenges in DevSecOps adoption from a variety of sources, this work offers preliminary understandings of the current state of DevSecOps practices. The findings from this research can inform the identification of emerging themes and areas of interest in the field.

Furthermore, “DevSecOps: A Multivocal Literature Review” by Havard Myrbakken and Ricardo Colomo-Palacios [7]. synthesizes multiple perspectives from academic and practitioner sources to present a comprehensive view of DevSecOps practices and their impact on security. This multidimensional review highlights the significance of fostering a collaborative DevSecOps culture across organizations and emphasizes the role of

communication and coordination among development, operations, and security teams. The article underscores the value of integrating security practices throughout the software development lifecycle and emphasizes the need for continuous improvement and adaptation in dynamic and evolving environments.

Finally, "Challenges and Solutions When Adopting DevSecOps: A Systematic Review" by Roshan N. Rajapakse delves into the barriers faced by organizations in adopting DevSecOps practices [8]this trend has presented the challenge of ensuring secure software delivery while maintaining the agility of DevOps. The efforts to integrate security in DevOps have resulted in the DevSecOps paradigm, which is gaining significant interest from both industry and academia. However, the adoption of DevSecOps in practice is proving to be a challenge. Objective: This study aims to systemize the knowledge about the challenges faced by practitioners when adopting DevSecOps and the proposed solutions reported in the literature. We also aim to identify the areas that need further research in the future. Method: We conducted a Systematic Literature Review of 54 peer-reviewed studies. The thematic analysis method was applied to analyze the extracted data. Results: We identified 21 challenges related to adopting DevSecOps, 31 specific solutions, and the mapping between these findings. We also determined key gap areas in this domain by holistically evaluating the available solutions against the challenges. The results of the study were classified into four themes: People, Practices, Tools, and Infrastructure. Our findings demonstrate that tool-related challenges and solutions were the most frequently reported, driven by the need for automation in this paradigm. Shift-left security and continuous security assessment were two key practices recommended for DevSecOps. People-related factors were considered critical for successful DevSecOps adoption but less studied. Conclusions: We highlight the need for developer-centered application security testing tools that target the continuous practices in DevSecOps. More research is needed on how the traditionally manual security practices can be automated to suit rapid software deployment cycles. Finally, achieving a suitable balance between the speed of delivery and security is a significant issue practitioners face in the DevSecOps paradigm.","author":[{"dropping-particle":"","family":"Rajapakse","given":"Roshan N.","non-dropping-particle":"","parse-names":false,"suffix":""},{"dropping-particle":"","family":"Zahedi","given":"Mansooreh","non-dropping-particle":"","parse-names":false,"suffix":""},{"dropping-particle":"","family":"Babar","given":"M. Ali","non-dropping-particle":"","parse-names":false,"suffix":""},{"dropping-particle":"","family":"Shen","given":"Haifeng","non-dropping-particle":"","parse-names":false,"suffix":""}],"container-title":"Information and Software Technology","id":"ITEM-1","issued":{"date-parts":[["2022","1","1"]]},"page":"106700","publisher":"Elsevier","title":"Challenges and solutions when adopting DevSecOps: A systematic review","type":"article-journal","volume":"141"},"uris":["http://www.mendeley.com/documents/?uuid=71d090d2-d3bb-3a23-a6c9-98c2e0eadaf4"]}],"mendeley":{"formattedCitation":"[8]","plainTextFormattedCitation":"[8]","previouslyFormattedCitation":"[8]"},"properties":{"noteIndex":0},"schema":"https://github.com/citation-style-language/schema/raw/master/csl-citation.json"}. The study explores the challenges and limitations that may hinder successful integration of security in the DevOps workflow. By identifying common challenges, the research provides valuable recommendations and mitigation strategies to overcome these obstacles effectively, facilitating the adoption of DevSecOps and enhancing security practices in Kubernetes environments.

2. The Need for DevSecOps in Kubernetes Security Research

Traditional security approaches often struggle to keep pace with the dynamic and distributed nature of Kubernetes environments. Several challenges arise when attempting to secure Kubernetes deployments using conventional security methods:

– **Complexity and Scale:** Kubernetes environments are highly complex, comprising numerous interconnected components, including pods, nodes, clusters, and the control plane [9]we see that most of the cloud-based applications and services often consist of hundreds of micro-services; however, the traditional monolithic pattern is no longer suitable for today's development life-cycle. This is due to the difficulties of maintenance, scale, load balance, and many other factors associated with it. Consequently, people switch their focus on containerization—a lightweight virtualization technology. The saving grace is that it can use machine resources more efficiently than the virtual machine (VM. Traditional security tools and practices may struggle to adequately address the scale and intricacies of these environments.

– **Rapid Deployment and Continuous Integration/ Continuous Deployment (CI/CD):** Kubernetes promotes rapid application deployment through CI/CD pipelines, where new code changes are frequently built, tested, and deployed. Traditional security measures, often applied as an afterthought,

may not be able to keep up with the speed of these deployments [10]the number and variety of business requirements are increasingly complex, keeps sustained growth, the process of continuous integration delivery of information systems becomes increasingly complex, the amount of repetitive work is growing. This paper focuses on the continuous integration of specific information systems, a collaborative work scheme for continuous integrated delivery based on Jenkins and Ansible is proposed. Both theory and practice show that continuous integrated delivery cooperative systems can effectively improve the efficiency and quality of continuous integrated delivery of information systems. The effect of the optimization and upgrading of the information system is obvious.","author":[{"dropping-particle":"","family":"Petrochina","given":"Wang Yiran","non-dropping-particle":"","parse-names":false,"suffix":""},{"dropping-particle":"","family":"Petrochina","given":"Zhang Tongyang","non-dropping-particle":"","parse-names":false,"suffix":""},{"dropping-particle":"","family":"Petrochina","given":"Guo Yidong","non-dropping-particle":"","parse-names":false,"suffix":""}],"container-title":"2018 International Conference on Artificial Intelligence and Big Data, ICAIBD 2018","id":"ITEM-1","issued":{"date-parts":[["2018","6","25"]]},"page":"245-249","publisher":"Institute of Electrical and Electronics Engineers Inc.","title":"Design and implementation of continuous integration scheme based on Jenkins and Ansible","type":"article-journal"},"uris":["http://www.mendeley.com/documents/?uuid=2ab94d35-3d8e-3139-83c4-d618e514cbc3"]}],"mendeley":{"formattedCitation":"[10]","plainTextFormattedCitation":"[10]","previouslyFormattedCitation":"[10]"},"properties":{"noteIndex":0},"schema":"https://github.com/citation-style-language/schema/raw/master/csl-citation.json"}.

- **Container Lifecycle Management:** Containers, the building blocks of Kubernetes, have unique security challenges. Ensuring the integrity and security of container images, scanning for vulnerabilities, and enforcing secure configurations across a large number of containers require specialized security considerations that may be lacking in traditional approaches.
- **Dynamic Workloads and Orchestration:** Kubernetes is designed to handle dynamic workloads, including scaling pods and allocating resources based on demand. Traditional security approaches may struggle to adapt to the continuously changing environment and may not effectively protect against attacks targeting the orchestration layer.

Integrating security early in the software development lifecycle is crucial to mitigating risks and addressing vulnerabilities effectively as shown in Figure 1. By incorporating security practices from the early stages of development, organizations can benefit in several ways:

- **Shift-Left Security:** Integrating security early in the development process, often referred to as "shifting left," enables proactive identification and remediation of security vulnerabilities . By conducting security assessments, code reviews, and threat modeling during the development phase, potential risks can be identified and mitigated before they manifest in production environments [11].
- **Cost and Time Efficiency:** Addressing security issues early in the development lifecycle helps reduce the cost and effort associated with fixing vulnerabilities at later stages. Fixing security flaws during development is typically less time-consuming and less disruptive compared to addressing them in production environments.
- **Security by Design:** Incorporating security considerations throughout the development process allows for security to be built into the application architecture and design. By adopting secure coding practices, secure configuration management, and secure deployment strategies, organizations can create a more resilient and secure Kubernetes environment.

DevSecOps offers a comprehensive approach to addressing security gaps and vulnerabilities in Kubernetes deployments. By integrating security practices throughout the software development lifecycle, organizations can effectively tackle the unique security challenges posed by Kubernetes:

- **Automated Security Testing:** DevSecOps promotes the use of automated security testing tools and techniques, such as vulnerability scanning, static code analysis, and penetration testing. These practices help identify security weaknesses and vulnerabilities specific to Kubernetes configurations, container images, and deployed applications.
- **Continuous Monitoring and Incident Response:** DevSecOps emphasizes continuous monitoring of Kubernetes environments, including logging, auditing, and anomaly detection. By implementing robust monitoring solutions, organizations can detect and respond to security incidents promptly, minimizing the impact of potential attacks [1].
- **Infrastructure as Code (IaC):** DevSecOps encourages the use of Infrastructure as Code principles in Kubernetes deployments. By defining infrastructusre and security configurations as code, organizations

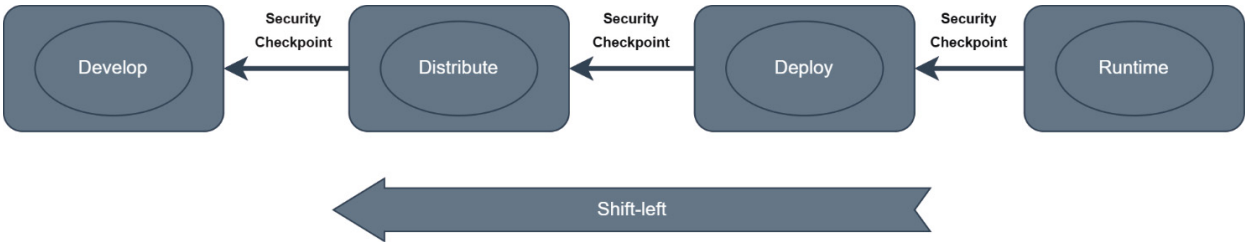

Fig. 1. Security integration in software development life cycle

can enforce consistent and secure deployment practices, reducing the risk of misconfigurations and ensuring compliance with security best practices.

– **Collaboration and Communication:** DevSecOps promotes cross-functional collaboration between development, operations, and security teams. This collaboration enables early identification and resolution of security issues, fosters a shared responsibility for security, and facilitates knowledge sharing to stay updated with the latest security threats and mitigation techniques.

By leveraging DevSecOps practices in Kubernetes security research, organizations can address security gaps, proactively mitigate vulnerabilities, and build a secure foundation for their containerized applications. This approach ensures that security is not an afterthought but an integral part of the entire software development lifecycle, enabling organizations to achieve robust and resilient Kubernetes deployments.

3. Key Principles of DevSecOps in Kubernetes Security

DevSecOps in the context of Kubernetes security research is guided by several key principles that help organizations effectively integrate security practices throughout the development lifecycle. These principles ensure collaboration, automation, and the seamless integration of security measures within Kubernetes deployments [12].

– **Collaboration and Communication between Development, Operations, and Security Teams:**

Foster a culture of collaboration and shared responsibility between development, operations, and security teams. This involves breaking down silos and promoting open communication channels for sharing security knowledge, insights, and best practices.

Encourage regular meetings and cross-functional discussions to align security requirements, understand operational constraints, and prioritize security considerations within the Kubernetes environment.

– **Automation of Security Processes and Continuous Monitoring:**

Leverage automation tools and techniques to streamline security processes within Kubernetes. This includes automating security testing, vulnerability scanning, and compliance checks to identify and address security issues in an efficient and consistent manner [13].

Implement continuous monitoring solutions that capture security-related events and activities within the Kubernetes environment. This enables real-time visibility into potential security threats, rapid incident response, and proactive mitigation of risks.

– **Integration of Security Practices throughout the Entire Development Pipeline:**

Integrate security considerations at each stage of the development pipeline, from initial code development to deployment and ongoing operations. This involves incorporating security checks, code reviews, and security testing into the CI/CD process. See Figure 2.

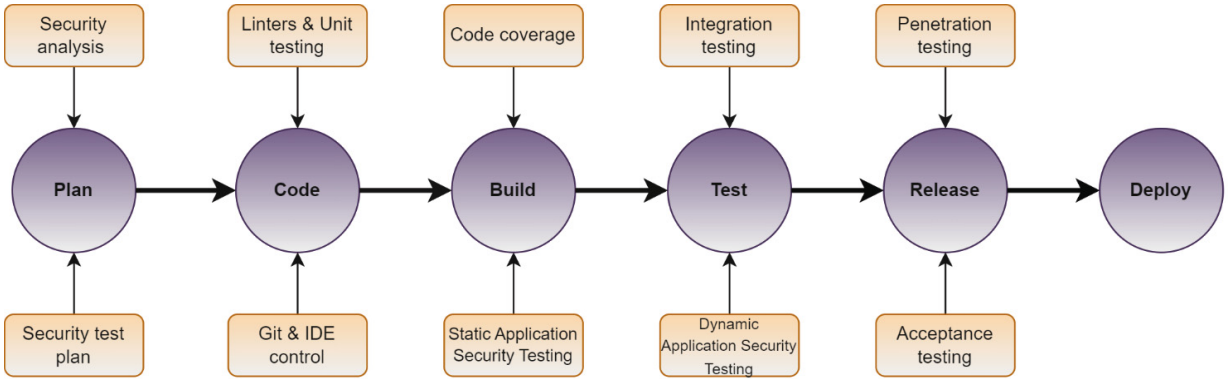

Fig. 2. DevSecOps pipeline

Implement security gates and controls that ensure the adherence to security policies and best practices at every step of the deployment and release process. This ensures that security is not an isolated activity but an integral part of the overall development workflow.

– **Implementing Security as Code to Ensure Consistency and Scalability:**

Treat security configurations and practices as code artifacts, enabling consistency, repeatability, and scalability. Define security policies, configurations, and deployment practices as code, leveraging tools like Infrastructure as Code (IaC) and configuration management tools.

Implement security-focused pipelines and version control systems to track and manage changes to security configurations and ensure their traceability and audibility.

By adhering to these key principles, organizations can establish a robust DevSecOps approach within their Kubernetes security research. This approach promotes collaboration, automates security processes, integrates security throughout the development pipeline, and treats security as a foundational element that ensures the consistency, scalability, and resilience of Kubernetes deployments.

4. Benefits of Adopting DevSecOps in Kubernetes Security Research

Early Detection and Prevention of Security Vulnerabilities and Attacks:

– By integrating security practices throughout the development lifecycle, DevSecOps enables early identification and remediation of security vulnerabilities specific to Kubernetes environments.
– Security testing, vulnerability scanning, and code reviews conducted during the development phase help identify and address security weaknesses before they are deployed into production.
– Continuous monitoring and automated security checks provide real-time visibility into potential security threats, allowing for proactive detection and prevention of attacks.

Increased Visibility and Traceability of Security-Related Events in Kubernetes:

– DevSecOps practices emphasize the implementation of robust monitoring and logging mechanisms within Kubernetes environments.
– Continuous monitoring provides detailed insights into security-related events, allowing for comprehensive visibility into the state of the Kubernetes infrastructure, containerized applications, and user activities.
– Enhanced traceability of security events enables efficient incident response, forensic analysis, and compliance auditing.

Improved Incident Response and Mitigation Capabilities:

– DevSecOps promotes a proactive approach to incident response by integrating security monitoring, automated alerting, and incident management processes within the Kubernetes environment.
– Real-time monitoring and automated incident response mechanisms enable swift detection and mitigation of security incidents, reducing the impact of potential breaches.
– Collaboration between development, operations, and security teams facilitates faster incident response, enabling coordinated efforts to resolve security issues effectively.

By adopting DevSecOps in Kubernetes security research, organizations can leverage these benefits to enhance the overall security posture of their Kubernetes deployments. Early detection and prevention of security vulnerabilities, increased visibility and traceability of security events, improved incident response capabilities, risk reduction, and compliance facilitation collectively contribute to creating a more secure and resilient Kubernetes environment.

5. Implementing DevSecOps in the Research Framework

To effectively implement DevSecOps principles in the research framework focused on Kubernetes security and attack detection, the following steps can be followed:

Integrating Security Practices within the Attack Detection Model:

– Incorporate security considerations directly into the design and implementation of the attack detection model.
– Define security-specific features and indicators that can help identify potential attacks or anomalies in the Kubernetes environment.
– Ensure that the model is trained and tested using datasets that include various attack scenarios and security-relevant patterns.

Incorporating Security-Focused Metrics Collection and Analysis:

– Define and collect security-focused metrics related to the Kubernetes environment, such as resource utilization, network traffic, container behavior, and API usage.
– Analyze these metrics to detect any abnormal patterns or deviations from expected behavior, which may indicate security threats or vulnerabilities.
– Leverage data visualization techniques to provide meaningful insights into the security posture of

the Kubernetes environment and facilitate decision-making.

Leveraging Automated Security Testing and Vulnerability Scanning Tools:

– Integrate automated security testing tools and vulnerability scanners into the research framework to identify potential weaknesses and vulnerabilities in the Kubernetes deployment.
– Use these tools to regularly scan container images, Kubernetes configurations, and the underlying infrastructure for known security issues.
– Incorporate the findings from these security tests into the research pipeline to assess the impact on the attack detection model and identify areas for improvement.

Collaboration and Communication:

– Foster collaboration between the research team, security experts, and operations personnel to ensure a holistic approach to Kubernetes security.
– Engage in regular discussions and knowledge-sharing sessions to align research objectives with security goals and best practices.
– Encourage open communication channels to promptly address security concerns, share findings, and implement security enhancements within the research framework.

By implementing these DevSecOps practices, the research framework can benefit from the integration of security into the attack detection model, the inclusion of security-focused metrics, the use of automated security testing tools, and the establishment of a feedback loop for continuous improvement. This approach enables a proactive and comprehensive approach to Kubernetes security in the research process and contributes to the development of effective security solutions for Kubernetes environments.

6. Best Practices for DevSecOps in Kubernetes Security Research

By following the best practices, organizations engaged in Kubernetes security research can enhance the effectiveness and efficiency of their DevSecOps approach. The following practices facilitate the identification and mitigation of security risks, ensure secure configuration and access controls, enable continuous monitoring and analysis, and promote regular security assessments and audits to maintain a robust security posture in Kubernetes deployments [14].

Conducting Threat Modeling and Risk Assessments Specific to Kubernetes:

– Perform threat modeling exercises to identify potential security risks and vulnerabilities specific to Kubernetes deployments. Consider factors such as container security, network security, and access controls.
– Conduct regular risk assessments to evaluate the impact and likelihood of identified threats, prioritize them based on risk levels, and implement appropriate security measures.

Ensuring Secure Configuration and Hardening of Kubernetes Clusters:

– Implement secure configuration practices for Kubernetes clusters, including proper network segmentation, secure API server settings, and restricted access to sensitive resources.
– Follow Kubernetes hardening guides and security best practices to eliminate common security misconfigurations and reduce attack surfaces.
– Regularly review and update cluster configurations to address emerging security concerns and align with industry standards.

Implementing Identity and Access Management Controls:

– Establish strong identity and access management controls for Kubernetes environments.
– Implement robust authentication mechanisms, such as multi-factor authentication (MFA), and enforce the principle of least privilege to limit access to sensitive Kubernetes resources.
– Implement Role-Based Access Control (RBAC) policies to ensure that only authorized users have appropriate permissions to interact with Kubernetes resources [15]s.

Continuous Monitoring and Log Analysis for Security Events and Anomalies:

– Implement a comprehensive monitoring solution that captures and analyzes security events, logs, and metrics from Kubernetes clusters.
– Continuously monitor for suspicious activities, unauthorized access attempts, and abnormal behavior within the Kubernetes environment.
– Utilize log analysis and anomaly detection techniques to identify potential security incidents or indicators of compromise.

Regular Security Assessments and Audits:

– Conduct regular security assessments and penetration testing to evaluate the effectiveness of security controls within Kubernetes deployments.
– Perform vulnerability scanning and security testing of container images and Kubernetes configurations.
– Conduct periodic security audits to validate compliance with security policies, industry regulations, and best practices.

7. Challenges and Mitigation Strategies

Implementing DevSecOps in the research framework for Kubernetes security may encounter certain

challenges. However, with appropriate mitigation strategies, these challenges can be overcome effectively:

Overcoming Resistance to Change and Fostering a Security Culture:

Challenge: Resistance to change and lack of awareness or understanding of the importance of security practices can hinder the adoption of DevSecOps.

Mitigation: Educate and raise awareness among team members about the significance of security in Kubernetes deployments. Promote a security-first mindset and foster a culture of shared responsibility for security [16]in particular of security standards. A comprehensive overview of this scattered field is still missing and we know little about how to achieve security compliance in agile software development. Existing secondary studies (mapping studies and literature reviews. Provide training and resources to help team members understand the value and benefits of DevSecOps.

Balancing Security and Operational Requirements in Kubernetes Deployments:

Challenge: Striking a balance between implementing robust security measures and maintaining operational efficiency can be challenging.

Mitigation: Collaborate closely with operations teams to understand their requirements and constraints. Conduct risk assessments to identify critical security needs and prioritize them based on operational impact. Continuously monitor and refine security measures to optimize the balance between security and operational requirements.

Ensuring Compatibility of DevSecOps Practices with Research Objectives:

Challenge: Research objectives may have specific requirements that need to be aligned with DevSecOps practices.

Mitigation: Analyze the research objectives and identify areas where DevSecOps practices can be seamlessly integrated. Adapt DevSecOps processes to accommodate the specific needs of the research while ensuring that security considerations are not compromised. Seek guidance from domain experts and security practitioners to find the right balance.

Addressing Resource Constraints and Scalability Concerns:

Challenge: Limited resources, such as budget, personnel, or infrastructure, can pose challenges to implementing comprehensive DevSecOps practices.

Mitigation: Prioritize security activities based on risk assessments and available resources. Automate security processes, such as testing and monitoring, to reduce the manual effort required. Leverage cloud-based security services or consider outsourcing certain security tasks to optimize resource utilization. Focus on scalability by designing the research framework in a modular and extensible manner.

It is important to note that challenges may vary based on the specific context and organization. Regularly assess the effectiveness of the mitigation strategies and adjust them as needed to overcome emerging challenges. By addressing these challenges proactively, the research framework can benefit from the enhanced security provided by DevSecOps practices, enabling more reliable and secure Kubernetes deployments.

Conclusion

In conclusion, DevSecOps plays a crucial role in ensuring the security and resilience of Kubernetes environments. This article has emphasized the importance of integrating security practices early in the software development lifecycle and highlighted the benefits and best practices of adopting DevSecOps in Kubernetes security research.

The adoption of DevSecOps brings several significant advantages. It enables early detection and prevention of security vulnerabilities and attacks, increases visibility and traceability of security events, improves incident response capabilities, reduces security risks and associated costs, and facilitates compliance with industry regulations and standards. By incorporating collaboration, automation, and security as code principles, organizations can establish a robust security posture in their Kubernetes deployments.

To fully embrace DevSecOps, it is essential for organizations and researchers to overcome challenges such as resistance to change, balancing security and operational requirements, ensuring compatibility with research objectives, and addressing resource constraints. By fostering a security culture, aligning security practices with operational needs, adapting DevSecOps to research goals, and optimizing resource utilization, these challenges can be effectively mitigated.

Looking ahead, the future of Kubernetes security research lies in exploring further advancements in DevSecOps practices. Areas for further research and development include the refinement of machine learning-based attack detection models, the integration of advanced threat intelligence and anomaly detection techniques, and the continuous evolution of automated security testing tools. Additionally, researching ways to enhance the scalability and resource efficiency of DevSecOps in Kubernetes environments will be crucial as organizations increasingly adopt container orchestration platforms.

In conclusion, embracing DevSecOps is vital to enhance the security and resilience of Kubernetes envi-

ronments. By implementing the discussed benefits and best practices, organizations can effectively protect their Kubernetes deployments, mitigate security risks, and drive innovation in the field of Kubernetes security research. By continuously improving and expanding DevSecOps practices, researchers can contribute to the advancement of secure and reliable Kubernetes ecosystems for the benefit of the entire industry.

References

1. *Darwesh G., Hammoud J. and Vorobeva A.A.* "A novel approach to feature collection for anomaly detection in Kubernetes environment and agent for metrics collection from Kubernetes nodes," Sci. Tech. J. Inf. Technol. Mech. Opt., vol. 23, no. 3, pp. 538–546, Jun. 2023, doi: 10.17586/2226-1494-2023-23-3-538-546.
2. *Gomes K.* "The Importance of DevSecOps," Honor. Capstones, May 2018, Accessed: Jul. 23, 2023. [Online]. Available: https://huskiecommons.lib.niu.edu/studentengagement-honorscapstones/1214
3. *Prates L., Faustino J., Silva M. and Pereira R.* "DevSecOps metrics," Lect. Notes Bus. Inf. Process., vol. 359, pp. 77–90, 2019, doi: 10.1007/978-3-030-29608-7_7/COVER.
4. *Sánchez-Gordón M. and Colomo-Palacios R.* "Security as Culture: A Systematic Literature Review of DevSecOps," Proc. - 2020 IEEE/ACM 42nd Int. Conf. Softw. Eng. Work. ICSEW 2020, pp. 266–269, Jun. 2020, doi: 10.1145/3387940.3392233.
5. *Rahul S.* "Implementation of DevSecOps using Open-Source tools," Int. J. Adv. Res., 2019, Accessed: Jul. 22, 2023. [Online]. Available: www.IJARIIT.com
6. *Mao R. et al.* "Preliminary Findings about DevSecOps from Grey Literature," Proc. - 2020 IEEE 20th Int. Conf. Softw. Qual. Reliab. Secur. QRS 2020, pp. 450–457, Dec. 2020, doi: 10.1109/QRS51102.2020.00064.
7. *Myrbakken H. and Colomo-Palacios R.* "DevSecOps: A multivocal literature review," Commun. Comput. Inf. Sci., vol. 770, pp. 17–29, 2017, doi: 10.1007/978-3-319-67383-7_2/COVER.
8. *Rajapakse R.N., Zahedi M., Babar M.A. and Shen H.* "Challenges and solutions when adopting DevSecOps: A systematic review," Inf. Softw. Technol., vol. 141, p. 106700, Jan. 2022, doi: 10.1016/J.INFSOF.2021.106700.
9. *Mondal S.K., Pan R., Kabir H.M.D., Tian T. and Dai H.N.* "Kubernetes in IT administration and serverless computing: An empirical study and research challenges," J. Supercomput., vol. 78, no. 2, pp. 2937–2987, Feb. 2022, doi: 10.1007/s11227-021-03982-3.
10. *Petrochina W.Y., Petrochina Z.T. and Petrochina G.Y.* "Design and implementation of continuous integration scheme based on Jenkins and Ansible," 2018 Int. Conf. Artif. Intell. Big Data, ICAIBD 2018, pp. 245–249, Jun. 2018, doi: 10.1109/ICAIBD.2018.8396203.
11. *Lombardi F. and Fanton A.* "From DevOps to DevSecOps is not enough. CyberDevOps: an extreme shifting-left architecture to bring cybersecurity within software security lifecycle pipeline," Softw. Qual. J., vol. 31, no. 2, pp. 619–654, Jun. 2023, doi: 10.1007/S11219-023-09619-3/METRICS.
12. *Mohan V. and Ben Othmane L.* "SecDevOps: Is it a marketing buzzword? Mapping research on security in DevOps," Proc. - 2016 11th Int. Conf. Availability, Reliab. Secur. ARES 2016, pp. 542–547, Dec. 2016, doi: 10.1109/ARES.2016.92.
13. *Mburano B. and Si W.* "Evaluation of web vulnerability scanners based on OWASP benchmark," 26th Int. Conf. Syst. Eng. ICSEng 2018 - Proc., Feb. 2019, doi: 10.1109/ICSENG.2018.8638176.
14. *Darwesh G., Hammoud J. and Vorobeva A.A.* "SECURITY IN KUBERNETES: BEST PRACTICES AND SECURITY ANALYSIS," J. Ural Fed. Dist. Inf. Secur., vol. 22, no. 2, 2022, doi: 10.14529/SECUR220209.
15. *Shamim S.I.* "Mitigating security attacks in kubernetes manifests for security best practices violation," in ESEC/FSE 2021 - Proceedings of the 29th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Association for Computing Machinery, Inc, Aug. 2021, pp. 1689–1690. doi: 10.1145/3468264.3473495.
16. *Moyon F., Almeida P., Riofrio D., Mendez D. and Kalinowski M.* "Security Compliance in Agile Software Development: A Systematic Mapping Study," Proc. - 46th Euromicro Conf. Softw. Eng. Adv. Appl. SEAA 2020, pp. 413–420, Aug. 2020, doi: 10.1109/SEAA51224.2020.00073.

Ghadeer Darwesh. ITMO University, Saint Petersburg, Russian Federation, PhD Student, https://orcid.org/0000-0003-1116-9410, ghadeerdarwesh32@gmail.com

Jaafar Hammoud. ITMO University, Saint Petersburg, Russian Federation, PhD, https://orcid.org/0000-0002-2033-0838, hammoudgj@gmail.com

Vorobeva Alisa A. ITMO University, Saint Petersburg, 197101, Russian Federation, PhD, Associate Professor, https://orcid.org/0000-0001-6691-6167, alice_w@mail.ru

Повышение безопасности Kubernetes: решающая роль DevSecOps

Гадир Дарвиш, Джафар Хаммуд, А.А. Воробьева
Университет ИТМО, г. Санкт-Петербург, Россия

Аннотация. В статье подчеркивается важность интеграции практик DevSecOps (разработка, безопасность и эксплуатация) в исследования по обнаружению распространенных атак в средах Kubernetes. По мере того, как Kubernetes быстро становится популярной платформой оркестровки контейнеров, проблемы безопасности, связанные с контейнерными приложениями, значительно возросли. Однако традиционные методологии безопасности часто с трудом успевают за динамичной и быстро развивающейся природой контейнерных сред, оставляя потенциальные уязвимости для использования злоумышленниками. Подчеркивая важность DevSecOps, статья призвана подчеркнуть его роль в повышении уровня безопасности развертываний Kubernetes и продвижении упреждающего подхода к защите контейнерных приложений. В статье также обсуждаются ключевые аспекты и преимущества внедрения DevSecOps в контексте исследований безопасности Kubernetes.

Ключевые слова: *DevSecOps, безопасность Kubernetes, DevOps, методы обеспечения безопасности.*
DOI: 10.14357/20790279240309 **EDN: SDRKHO**

Литература

1. *Дарвиш Г., Хаммуд Дж., Воробьева А.А.* Новый подход к сбору признаков для обнаружения аномалий в среде Kubernetes и агент для сбора метрик с узлов Kubernetes. Тех. Ж. Инф. Технол. Мех. Опт. 2023. Вып. 23. С. 538–546. DOI: 10.17586/2226-1494-2023-23-3-538-546.
2. *Гомес К.* «Важность DevSecOps», Честь. Capstones, май 2018 г., доступ: 23 июля 2023 г. [Онлайн]. Доступно: https://huskiecommons.lib.niu.edu/studentengagement-honorscapstones/1214.
3. *Пратес Л., Фаустино Дж., Сильва М. и Перейра Р.* «Метрики DevSecOps», Lect. Примечания Автобус. Инф. Процесс. 2019. Вып. 359. С. 77–90. DOI: 10.1007/978-3-030-29608-7_7/COVER.
4. *Санчес-Гордон М. и Р. Коломо-Паласиос М.* «Безопасность как культура: систематический обзор литературы по DevSecOps», Proc. - 42-я Международная конференция IEEE/ACM 2020 г. Конф. Программное обеспечение англ. Работа. ICSEW. 2020. С. 266–269. DOI: 10.1145/3387940.3392233.
5. *Рахул С.* «Внедрение DevSecOps с использованием инструментов с открытым исходным кодом», Int. Дж. Адв. Res., 2019, доступ: 22 июля 2023 г. [Онлайн]. Доступно: www.IJARIIT.com.
6. *Мао Р. и др.* «Предварительные выводы о DevSecOps из серой литературы», Proc. - 20-й Международный IEEE 2020 г. Конф. Программное обеспечение Квал. Надежный. Безопасность. QRS. 2020. С. 450–457. DOI: 10.1109/QRS51102.2020.00064.
7. *Мирбаккен Х. и Коломо-Паласиос Р.* «DevSecOps: многоголосый обзор литературы», Commun. Вычислить. Инф. наук. 2017. Т. 770. С. 17–29. DOI: 10.1007/978-3-319-67383-7_2/COVER.
8. *Раджапаксе Р.Н., Захеди М., Бабар М.А. и Шен Х.* «Проблемы и решения при внедрении DevSecOps: систематический обзор», Инф. Программное обеспечение Технол. 2022. Вып. 141. С. 106700. DOI: 10.1016/J.INFSOF.2021.106700.
9. *Мондал С.К., Пан Р., Кабир Х.М.Д., Тиан Т. и Дай Х.Н.* «Kubernetes в ИТ-администрировании и бессерверных вычислениях: эмпирическое исследование и исследовательские задачи», J. Supercomput. 2022. Вып. 78, № 2. С. 2937–2987. DOI: 10.1007/s11227-021-03982-3.
10. *Петрокина В.Ю., Петрокина З.Т. и Петрчина Г.Ю.* «Проектирование и реализация схемы непрерывной интеграции на основе Jenkins и Ansible», 2018 Int. Конф. Артиф. Интел. Боль-

шие данные, ICAIBD. 2018. С. 245–249. DOI: 10.1109/ICAIBD.2018.8396203.
11. *Ломбарди Ф. и Фэнтон А.* «От DevOps к DevSecOps недостаточно. CyberDevOps: крайне сдвинутая влево архитектура, обеспечивающая кибербезопасность в рамках жизненного цикла безопасности программного обеспечения», Softw. Квал. Дж. 2023. Вып. 31. № 2. С. 619–654. DOI: 10.1007/S11219-023-09619-3/METRICS.
12. *Мохан В. и Отман Л. Бен.* «SecDevOps: это модное маркетинговое словечко? Картирование исследований по безопасности в DevOps», Учеб. - 2016 11-й Международный. Конф. Доступность, Надежность. Безопасность. ARES. 2016. С. 542–547. DOI: 10.1109/ARES.2016.92.
13. *Мбурано Б. и Си В.* «Оценка сканеров веб-уязвимостей на основе эталонного теста OWASP», 26-й международный конгресс. Конф. Сист. англ. ICSEng. 2018. Proc. DOI: 10.1109/ICSENG.2018.8638176.
14. *Дарвиш Дж., Хаммуд Дж., Воробьева А.А.* Безопасность в kubernetes: лучшие практики и анализ безопасности // Журнал Уральской федерации. Расст. Инф. Безопасность. 2022. Т. 22, № 2. DOI: 10.14529/SECUR220209.
15. *Шамим С.И.* «Снижение атак на безопасность в манифестах Kubernetes для нарушения лучших практик безопасности», в ESEC / FSE 2021 – Материалы 29-го совместного заседания ACM, Европейской конференции по разработке программного обеспечения и симпозиума по основам программной инженерии, Association for Computing Machinery, Inc. 2021. С. 1689–1690. DOI: 10.1145/3468264.3473495.
16. *Мойон Ф., Алмейда П., Риофрио Д., Мендес Д. и Калиновски М.* «Соответствие требованиям безопасности при гибкой разработке программного обеспечения: систематическое картографическое исследование», Proc. - 46-я Евромикро Конференция. Программное обеспечение англ. Адв. Прил. SEAA. 2020. С. 413–420. DOI: 10.1109/SEAA51224.2020.00073.

Гадир Дарвиш. Университет ИТМО, г. Санкт-Петербург, Россия. Аспирант.
E-mail: ghadeerdarwesh32@gmail.com

Джафар Хаммуд. Университет ИТМО, г. Санкт-Петербург, Россия. Кандидат наук. https://orcid.org/0000-0002-2033-0838. E-mail: hammoudgj@gmail.com

Воробьева Алиса Андреевна. Университет ИТМО, г. Санкт-Петербург, Россия.Доцент. https://orcid.org/0000-0001-6691-6167. E-mail: alice_w@mail.ru