Section 1

IMPACT OF CONTAINERIZATION ON SECURITY IN DEVOPS ARCHITECTURE

Bohdan Kosarevskyi
National Aerospace University «Kharkiv Aviation Institute»
Scientific advisor: Dmytro Uzun

Relevance. The integration of containerization into DevOps practices has ushered in a paradigm shift in software development and deployment. As organizations increasingly adopt container orchestration platforms like Kubernetes and Docker, it becomes imperative to scrutinize the impact of this technological evolution on the security landscape within DevOps architectures [1, 2]. This work explores the multifaceted relationship between containerization and security, shedding light on the challenges and opportunities it presents. By examining the nuances of this integration, we aim to provide a comprehensive understanding of how containerization influences security measures in DevOps workflows.

Purpose. The goal of this work is to examine how the advent of containerization technologies, exemplified by Docker, has transformed the landscape of software development, testing, and deployment. Containers, by encapsulating applications and their dependencies, aim to guarantee uniformity across diverse environments. Although this approach enhances agility and scalability, it simultaneously introduces novel security considerations. The primary objective is to initiate a comprehensive reevaluation of security practices within containerized DevOps environments, addressing vulnerabilities intrinsic to this architecture. Through this exploration, we seek to discern the nuanced impact of containerization on security in DevOps workflows.

Main provisions. One notable impact of containerization on DevOps security lies in the isolation of application components. Isolation offered by containers reduces the attack surface, limiting the potential impact of security breaches. However, this containment introduces challenges related to shared kernel vulnerabilities and the need for continuous monitoring to detect and respond to potential threats promptly.

To comprehend the full extent of the impact, a comparative analysis of security measures in traditional and containerized DevOps environments is crucial. Traditional DevOps often relies on virtual machines, each running a full operating system. In contrast, containers share the host system's kernel, making them more lightweight and efficient. This shift presents a trade-off between resource efficiency and security isolation.

Containerized DevOps environments benefit from rapid scaling and resource optimization, but they necessitate enhanced attention to container image security, orchestration vulnerabilities, and the secure configuration of container runtimes.

By contrasting the strengths and weaknesses of traditional and containerized DevOps, organizations can make informed decisions about their security postures.

A critical aspect of the impact of containerization on DevOps security is the identification and mitigation of risks within the continuous integration/continuous deployment (CI/CD) pipeline [3]. The dynamic nature of containerized microservices introduces challenges in maintaining a consistent and secure deployment environment. Security measures must be integrated seamlessly into the CI/CD pipeline, incorporating vulnerability scanning, image signing, and runtime protection to ensure the integrity of the entire software supply chain.

Conclusions. In conclusion, the impact of containerization on security in DevOps architecture is a nuanced and evolving topic that demands careful consideration. While containerization brings unprecedented agility and efficiency, it also introduces a new set of security challenges. The benefits of containers must be balanced with the need for robust isolation and continuous monitoring. A comparative analysis helps organizations make informed decisions about adopting containerization in their DevOps workflows, considering the specific security requirements of their applications.

The future of DevOps security lies in the ability to harness the advantages of containerization while proactively mitigating its inherent risks. Through ongoing research, collaboration, and the implementation of evolving security practices, organizations can navigate the evolving landscape of DevOps with confidence in the security of their containerized workflows.

List of references

1. Pod Security Standards. *Kubernetes*. URL – https://kubernetes.io/docs/concepts/security/pod-security-standards (date of access: 01.11.2023);
2. Security best practices. *Docker docs*. URL – https://docs.docker.com/develop/security-best-practices (date of access: 01.11.2023);
3. Adding Security Into CI/CD Pipeline *Ciklum*. URL – https://www.ciklum.com/blog/adding-security-into-ci-cd-pipeline (date of access: 02.11.2023).

Information about the authors

Bohdan Kosarevskyi, a master's student from the Department of Computer Systems, Networks and Cybersecurity, National Aerospace University «Kharkiv Aviation Institute», b.v.kosarevskyi@student.csn.khai.edu
Dmytro Uzun, PhD, Associate Professor from the Department of Computer Systems, Networks and Cybersecurity, National Aerospace University «Kharkiv Aviation Institute», d.uzun@csn.khai.edu