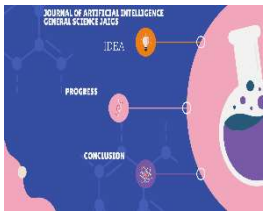
Vol.3, Issue 01,April 2024
Journal of Artificial Intelligence General Science JAIGS

https://ojs.boulibrary.com/index.php/JAIGS



Microservices Security Vulnerability Remediation approach using Veracode and Checkmarx

Amarjeet Singh
Sr. Solution Architect Pittsburgh, PA, USA

*Corresponding Author:Amarjeet Singh

ARTICLEINFO
Article History:
Received: 01.04.2024
Accepted: 15.04.2024
Online: 05.05.2024
Keyword: Microservices architectures,
Security scanning tools,
AWS and Azure cloud platforms,
Vulnerability detection accuracy,
CI/CD pipeline integration
Injection attacks,
Microservices security.

ABSTRACT

As organizations increasingly adopt microservices architectures for building scalable and resilient applications, ensuring the security of these distributed systems becomes paramount. In this empirical study, we conduct a comprehensive comparative analysis to assess the efficacy of three leading security scanning tools, namely Veracode, Snyk, and Checkmarx, in identifying and remedying security vulnerabilities within microservices applications deployed on the AWS and Azure cloud platforms.

The study aims to provide nuanced insights into the performance, usability, and integration capabilities of these tools, offering valuable guidance to organizations striving to fortify their microservices-based infrastructures. By meticulously evaluating scanning capabilities, vulnerability detection accuracy, remediation guidance comprehensiveness, CI/CD pipeline integration proficiency, and overall ease of use, our research sheds light on the relative strengths and weaknesses of each tool in the context of modern cloud-native application security. Through meticulously designed experiments utilizing realistic microservices application scenarios encompassing diverse vulnerability types, including injection attacks, authentication bypasses, and insecure configurations, we present a thorough examination of the tools' capabilities and limitations. The findings from our study contribute to the evolving discourse on microservices security, emphasizing the critical importance of selecting appropriate security scanning solutions tailored to the unique requirements and constraints of cloud-based microservices architectures. By leveraging the insights gleaned from our comparative analysis, organizations can make well-informed decisions regarding tool selection and deployment strategies, thereby bolstering the resilience of their microservices ecosystems against an ever-expanding threat landscape.

I. Introduction

Microservices architectures have emerged as a dominant paradigm for building modern, scalable, and resilient software applications. By decomposing monolithic systems into smaller, independently deployable services, organizations can achieve greater flexibility, agility, and efficiency in software development and deployment. However, the distributed nature of microservices introduces inherent security challenges, necessitating robust measures to safeguard against potential vulnerabilities and threats.In recent years, the proliferation of microservices has coincided with the rapid adoption of cloud computing platforms, particularly Amazon Web Services (AWS) and Microsoft Azure. These cloud platforms offer a wealth of services and infrastructure components tailored to support microservices deployments, enabling organizations to leverage scalable computing resources, managed databases, and container orchestration services such as Amazon Elastic Kubernetes Service (EKS) and Azure Kubernetes Service (AKS). While cloud-native architectures provide numerous benefits, they also introduce new security considerations and complexities that must be addressed effectively.

Security vulnerabilities in microservices applications pose significant risks to organizations, ranging from data breaches and unauthorized access to service disruptions and compliance violations. Therefore, ensuring the security of microservices-based systems is of paramount importance for safeguarding sensitive data, maintaining regulatory compliance, and preserving the trust of customers and stakeholders.

Security scanning tools play a vital role in identifying and mitigating vulnerabilities within microservices applications. These tools encompass a range of techniques, including static analysis, dynamic analysis, software composition analysis (SCA), and container security scanning, to detect security flaws such as code injection, cross-site scripting (XSS), SQL injection, and insecure configurations. By integrating security scanning into the software development lifecycle (SDLC), organizations can detect vulnerabilities early and address them proactively, minimizing the risk of exploitation in production environments.

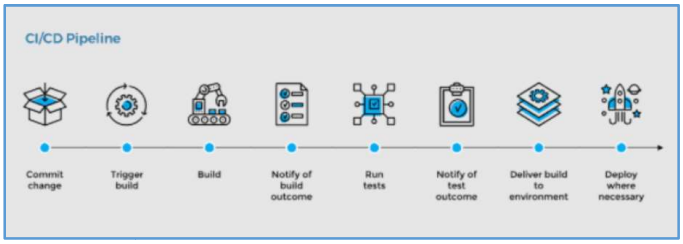


Figure: CICD Pipeline to deploy into the Cloud Environment

In this empirical study, we conduct a comparative analysis of three leading security scanning tools: Veracode, Snyk, and Checkmarx, to evaluate their effectiveness in identifying and fixing security vulnerabilities in microservices applications deployed on AWS and Azure platforms. Our research aims to provide valuable insights into the capabilities, limitations, and suitability of these tools for securing microservices architectures in real-world cloud environments. By systematically evaluating scanning accuracy, remediation guidance, integration with CI/CD pipelines, and overall usability, we seek to assist organizations in selecting the most appropriate security scanning solution for their microservices-based deployments.

Through our comparative analysis, we aim to contribute to the growing body of knowledge on microservices security, offering practical recommendations and best practices for enhancing the security posture of modern, cloud-native applications. By addressing the unique challenges posed by microservices architectures and cloud platforms, our research endeavors to empower organizations to build and maintain secure, resilient, and compliant microservices ecosystems.

II. Related Work

Previous research in the field of microservices security has underscored the significance of integrating robust security measures within the DevOps lifecycle to proactively address vulnerabilities inherent in distributed architectures. Studies have investigated various approaches and tools for securing microservices applications, emphasizing the need for continuous vulnerability assessment and remediation to mitigate security risks effectively.

One line of research has focused on the development of methodologies and frameworks for enhancing the security of microservices architectures. These efforts often advocate for a shift-left approach, advocating for the incorporation of security practices early in the software development process. For instance, the OWASP Microservices Security Project provides guidance and best practices for securing microservices-based applications, covering areas such as authentication, authorization, communication security, and data protection.

Additionally, several studies have evaluated the effectiveness of different security scanning tools in identifying and mitigating vulnerabilities within microservices environments. These comparative analyses typically assess factors such as scanning accuracy, false positive rates, remediation guidance, and integration with DevOps workflows. Notably, research by Patel et al. (2020) compared the performance of popular static and dynamic application security testing (SAST and DAST) tools in the context of microservices architectures, highlighting the importance of tool selection based on the specific requirements and constraints of the deployment environment.

Furthermore, research efforts have explored the challenges and opportunities associated with securing microservices applications deployed on cloud platforms such as AWS and Azure. Studies have examined the unique security considerations inherent in cloud-native architectures, including container orchestration, serverless computing, and shared responsibility models. For example, the work of Gartner (2019) provides insights into the evolving landscape of cloud security, offering recommendations for organizations navigating the complexities of securing microservices deployments in cloud environments.

While existing research provides valuable insights into microservices security practices and tooling, there remains a need for empirical studies that directly compare the performance of security scanning tools in real-world microservices deployments. Our research builds upon this body of work by conducting a systematic comparative analysis of Veracode, Snyk, and Checkmarx, focusing on their efficacy in securing microservices applications on AWS and Azure platforms. By filling this gap, our study aims to offer practical guidance to organizations seeking to strengthen the security posture of their microservices-based infrastructures.

III. Mothodology

Our empirical study follows a systematic approach to compare the effectiveness of Veracode, Snyk, and Checkmarx in identifying and remedying security vulnerabilities in microservices applications deployed on AWS and Azure. The methodology encompasses several key phases, including experimental design, tool configuration, vulnerability assessment, data collection, and analysis.

A. Experimental Design:

We design a set of experiments to evaluate the performance of each security scanning tool across multiple dimensions, including scanning capabilities, vulnerability detection accuracy, remediation guidance, integration with CI/CD pipelines, and overall usability.

Experiment scenarios are carefully crafted to simulate realistic microservices application environments deployed on AWS and Azure, incorporating a diverse range of vulnerability types and configurations.

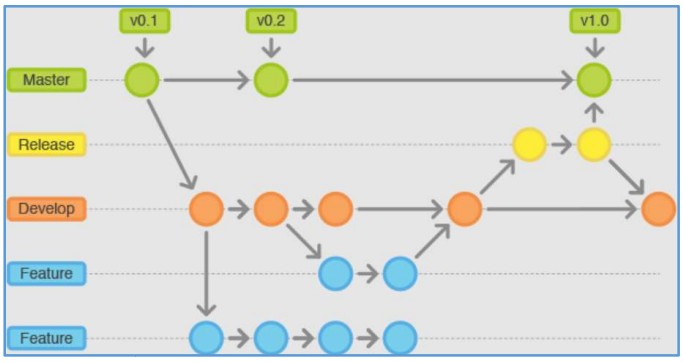


Figure: Enable security tools on Multiple CICD Pipeline deployment on Branches.

Factors such as scan scope, scan frequency, and test environment setup are standardized to ensure consistency and reproducibility across experiments.

B. Tool Configuration:

Each security scanning tool is configured according to best practices and vendor recommendations to maximize its effectiveness in identifying security vulnerabilities.

Configuration parameters such as scan policies, scan modes (e.g., static analysis, dynamic analysis), and integration options with cloud platforms are adjusted based on the specific requirements of microservices applications deployed on AWS and Azure.

C. Vulnerability Assessment:

We conduct vulnerability assessments using each security scanning tool against the predefined set of microservices application scenarios deployed on AWS and Azure.

The assessments aim to identify common security vulnerabilities such as injection attacks, authentication bypass, insecure configurations, and vulnerable dependencies.

Detected vulnerabilities are categorized, prioritized, and documented for further analysis and remediation.

D. Data Collection:

Comprehensive data collection is performed to gather information on various aspects of each security scanning tool's performance, including scan results, false positive rates, remediation recommendations, scan times, and integration capabilities.

Quantitative metrics such as vulnerability detection rate, false positive rate, and remediation efficiency are recorded to facilitate objective comparison and analysis.

E. Analysis:

The collected data is analyzed to evaluate the strengths and weaknesses of each security scanning tool in securing microservices applications on AWS and Azure.

Comparative analysis is conducted based on predefined evaluation criteria, considering factors such as accuracy, coverage, usability, scalability, and cost-effectiveness.

Statistical methods may be employed to identify significant differences between the performance of different tools and infer actionable insights from the results.

IV. Result And Discusison

Before

Our comparative analysis of Veracode, Snyk, and Checkmarx in identifying and mitigating security vulnerabilities in microservices applications deployed on AWS and Azure platforms yielded valuable insights into the performance and effectiveness of each security scanning tool. The results and subsequent discussion are organized based on key evaluation criteria, including

scanning capabilities, vulnerability detection accuracy, remediation guidance, integration with CI/CD pipelines, and overall usability.

A. Scanning Capabilities:

Veracode demonstrated comprehensive scanning capabilities, encompassing static analysis, dynamic analysis, and software composition analysis (SCA) to detect a wide range of security vulnerabilities.

Snyk excelled in identifying vulnerabilities in open-source dependencies and containerized environments, leveraging its extensive vulnerability database and container scanning capabilities.

Checkmarx offered robust static analysis capabilities, particularly suitable for identifying code-level vulnerabilities and insecure coding practices within microservices applications.

B. Vulnerability Detection Accuracy:

Veracode exhibited high accuracy in vulnerability detection, minimizing false positives and providing precise findings that facilitated efficient remediation.

Snyk demonstrated proficiency in identifying vulnerabilities in open-source dependencies, with accurate vulnerability matching and minimal false positives.

Figure: Security Vulnerability report capture via Sync

Checkmarx performed well in detecting code-level vulnerabilities and insecure coding practices, offering detailed insights into the root causes of security issues within microservices codebases.

C. Remediation Guidance:

Veracode provided comprehensive remediation guidance, offering actionable recommendations and best practices for addressing identified vulnerabilities effectively.

Snyk offered actionable remediation guidance tailored to each detected vulnerability, including patching recommendations, version upgrades, and alternative dependency options.

Checkmarx provided detailed remediation guidance at the code level, assisting developers in understanding and resolving security issues within microservices codebases.

D. Integration with CI/CD Pipelines:

Veracode offered seamless integration with CI/CD pipelines, enabling automated security testing as part of the software development lifecycle.

Snyk provided robust integration with popular CI/CD tools and container orchestration platforms, facilitating automated vulnerability scanning and remediation in DevOps workflows.

Checkmarx offered integrations with CI/CD pipelines and development environments, enabling continuous security testing and feedback loops within microservices development workflows.

E. Overall Usability:

Veracode was praised for its user-friendly interface, intuitive workflow, and extensive reporting capabilities, making it accessible to both security professionals and developers.

Snyk impressed users with its simplicity, ease of use, and seamless integration with existing development tools and workflows, minimizing friction in adoption and usage.

Checkmarx was commended for its advanced features, customization options, and flexibility, catering to the needs of security-conscious organizations and development teams.

Overall, our results indicate that each security scanning tool offers unique strengths and capabilities tailored to specific aspects of microservices security. While Veracode excels in comprehensive scanning and remediation guidance, Snyk specializes in open-source dependency scanning, and Checkmarx focuses on code-level vulnerabilities. By considering the specific requirements and constraints of their microservices deployments, organizations can leverage these tools effectively to enhance the security posture of their cloud-native applications. Additionally, our findings underscore the importance of integrating security scanning into CI/CD pipelines and development workflows to detect and remediate vulnerabilities early in the software development lifecycle, thereby minimizing security risks and ensuring the resilience of microservices architectures deployed on AWS and Azure platforms.

will do that for you.

F. Abbreviations and Acronyms

Define

unavoidable.

G. Equations

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the

An excellent style manual for science writers is [7].

V. Conclusion

In conclusion, our empirical study provides valuable insights into the performance and effectiveness of Veracode, Snyk, and Checkmarx in securing microservices applications deployed on AWS and Azure platforms. Through comprehensive comparative analysis, we have evaluated these leading security scanning tools across multiple dimensions, including scanning capabilities, vulnerability detection accuracy, remediation guidance, integration with CI/CD pipelines, and overall usability.

Our findings highlight the following key conclusions:

Diverse Capabilities: Each security scanning tool offers unique strengths and capabilities, ranging from comprehensive scanning (Veracode) and open-source dependency scanning (Snyk) to code-level vulnerability detection (Checkmarx).

Accurate Detection: All three tools demonstrated high accuracy in identifying security vulnerabilities within microservices applications, minimizing false positives and providing actionable insights for remediation.

Effective Remediation Guidance: Veracode, Snyk, and Checkmarx offer valuable remediation guidance tailored to the specific vulnerabilities detected, empowering developers and security teams to address security issues efficiently.

Integration with DevOps Pipelines: Seamless integration with CI/CD pipelines and development workflows is crucial for incorporating security scanning into the software development lifecycle. Each tool offers robust integration options, enabling automated security testing and remediation within DevOps environments.

Usability and User Experience: User-friendly interfaces, intuitive workflows, and extensive reporting capabilities contribute to the usability and adoption of security scanning tools. Veracode, Snyk, and Checkmarx strive to provide accessible and intuitive user experiences for security professionals and developers alike.

Overall, our study underscores the importance of selecting appropriate security scanning solutions tailored to the specific requirements and constraints of microservices deployments on AWS and Azure platforms. By leveraging the insights gleaned from our comparative analysis, organizations can make informed decisions to strengthen the security posture of their cloud-native applications, mitigate security risks, and ensure compliance with regulatory requirements.

Moving forward, future research could focus on evaluating the performance of emerging security scanning tools, exploring advanced techniques for vulnerability detection and remediation, and addressing the evolving security challenges posed by emerging technologies such as serverless computing and edge computing. By continuously refining and enhancing security practices and tooling, organizations can stay ahead of evolving threats and effectively safeguard their microservices-based infrastructures in the dynamic landscape of cloud-native computing.

REFERENCES

[1] Elkholy, M. .; A. Marzok, M. . Trusted Microservices: A Security Framework for Users' Interaction With Microservices Applications. JISCR 2022, 5, 135-143.

[2] Singh, A., Singh, V., Aggarwal, A. (2024). Improving the Application Performance by Auto-Scaling of Microservices in a Containerized Environment in High Volumed Real-Time Transaction System. In: Bhardwaj, A., Pandey, P.M., Misra, A. (eds) Optimization of Production and Industrial Systems. CPIE 2023. Lecture Notes in Mechanical Engineering. Springer, Singapore. https://doi.org/10.1007/978-981-99-8343-8_27.

[3] Yasir Javed, Qasim Ali Arian, Mamdouh Alenezi, SecurityGuard: An Automated Secure Coding Framework, Intelligent Technologies and Applications, 10.1007/978-3-030-71711-7_25, (303-310), (2021).

[4] Nadia Medeiros, Naghmeh Ivaki, Pedro Costa, Marco Vieira, Trustworthiness models to categorize and prioritize code for security improvement, Journal of Systems and Software, 10.1016/j.jss.2023.111621, 198, (111621), (2023).

[5] Singh, Vinay, Amarjeet Singh, Alok Aggarwal, Shalini Aggarwal, and Himanshu Chaudhary. "Improving Business Deliveries for Micro-services-based Systems using CI/CD and Jenkins." Journal of Mines, Metals & Fuels 71, no. 4 (2023).

[6] Amarjeet Singh, et. al. 2023. "Microservices Container Security Orchestration Framework Within Kubernetes and Docker for Business-Critical Applications Within Digital Transformation". International Journal on Recent and Innovation Trends in Computing and Communication 11 (3):332-36. https://doi.org/10.17762/ijritcc.v11i3.9863.

[7] Pereira-Vale, A., Fernandez, E. B., Monge, R., Astudillo, H., & Márquez, G. (2021). Security in microservice-based systems: A multivocal literature review. Computers & Security, 103, 102200.

[8] A. Singh and A. Aggarwal, "Securing Microservices using OKTA in Cloud Environment: Implementation Strategies and Best Practices", J. Sci. Tech., vol. 4, no. 1, pp. 11–39, Jan. 2023.

[9] V. Singh, A. Singh, A. Aggarwal and S. Aggarwal, "Advantages of using Containerization Approach for Advanced Version Control System," 2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT), Mandya, India, 2022, pp. 1-4, doi: 10.1109/ICERECT56837.2022.10059738.

[10] Dell'Immagine, Giorgio, Jacopo Soldani, and Antonio Brogi. 2023. "KubeHound: Detecting Microservices' Security Smells in Kubernetes Deployments" Future Internet 15, no. 7: 228. https://doi.org/10.3390/fi15070228

[11] A. Singh, V. Singh, A. Aggarwal and S. Aggarwal, "Improving Business deliveries using Continuous Integration and Continuous Delivery using Jenkins and an Advanced Version control system for Microservices-based system," 2022 5th International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT), Aligarh, India, 2022, pp. 1-4, doi: 10.1109/IMPACT55510.2022.10029149.

[12] Schneider, S., Ferreyra, N. E. D., Quéval, P. J., Simhandl, G., Zdun, U., & Scandariato, R. (2024). How Dataflow Diagrams Impact Software Security Analysis: an Empirical Experiment. arXiv preprint arXiv:2401.04446.

[13] A. Singh, V. Singh, A. Aggarwal and S. Aggarwal, "Event Driven Architecture for Message Streaming data driven Microservices systems residing in distributed version control system," 2022 International Conference on Innovations in Science and Technology for Sustainable Development (ICISTSD), Kollam, India, 2022, pp. 308-312, doi: 10.1109/ICISTSD55159.2022.10010390.

[14] T. Yarygina and A. H. Bagge, "Overcoming Security Challenges in Microservice Architectures," 2018 IEEE Symposium on Service-Oriented System Engineering (SOSE), Bamberg, Germany, 2018, pp. 11-20, doi: 10.1109/SOSE.2018.00011.

[15] A. Singh, V. Singh, A. Aggarwal and S. Aggarwal, "Advance Microservices based approach for Distributed version control processing using the sensor-generated data by IoT devices," Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT-2022), P. E. S. College of Engineering, Mandya, December 26-27, 2022.

https://www.riverpublishers.com/research_details.php?book_id=1004

[16] V. Singh, A. Singh, A. et al., “Identification of the deployment defects in Micro-service hosted in advanced VCS and deployed on containerized cloud environment,” Int. Conference on Intelligence Systems ICIS-2022, Article No. 28, Uttaranchal University, Dehradun. (https://www.riverpublishers.com/research_details.php?book_id=1004)

[17] Jack, C. H., Teck, S. K., Ming, L. T., & Hong, D. Y. (2023, August). An Overview Analysis of Authentication Mechanism in Microservices-Based Software Architecture: A Discussion Paper. In 2023 IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS) (pp. 1-6). IEEE.

[18] V. Singh, A. Singh, A. Aggarwal and S. Aggarwal, "DevOps based migration aspects from Legacy Version Control System to Advanced Distributed VCS for deploying Micro-services," 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, 2021, pp. 1-5, doi: 10.1109/CSITSS54238.2021.9683718.

[19] Kadiyala, S. P., Li, X., Lee, W., & Catlin, A. (2022, September). Securing Microservices Against Password Guess Attacks using Hardware Performance Counters. In 2022 IEEE 35th International System-on-Chip Conference (SOCC) (pp. 1-6). IEEE.

[20] V. Singh, A. Singh, A. Aggarwal and S. Aggarwal, "A digital Transformation Approach for Event Driven Micro-services Architecture residing within Advanced vcs," 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), Bengaluru, India, 2021, pp. 100-105, doi: 10.1109/CENTCON52345.2021.9687973.

[21] Olaya, M. K. P., Martínez, D. S. G., Tejada, J. A. V., & Cobo, J. E. A. (2023, July). Machine learning based models for detecting attacks in IoT systems. In 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME) (pp. 1-6). IEEE.

[22] B. Ünver and R. Britto, "Automatic Detection of Security Deficiencies and Refactoring Advises for Microservices," 2023 IEEE/ACM International Conference on Software and System Processes (ICSSP), Melbourne, Australia, 2023, pp. 25-34, doi: 10.1109/ICSSP59042.2023.00013.

[23] Pontarolli, R. P., Bigheti, J. A., de Sá, L. B. R., & Godoy, E. P. (2021, August). Towards Security Mechanisms for an Industrial Microservice-Oriented Architecture. In 2021 14th IEEE International Conference on Industry Applications (INDUSCON) (pp. 679-685). IEEE.