Strengthening cloud applications: A deep dive into kill chain identification, scoring, and automatic penetration testing

Stefano Simonetto[0009−0009−9778−4019]

Department of Pervasive Systems, University of Twente, Enschede, The Netherlands
s.simonetto@utwente.nl

Abstract. The need to anticipate and defend against potential threats is paramount in cybersecurity. This study addresses two fundamental questions: what attacks can be performed against my system, and how can these attacks be thwarted?
Addressing the first question, this work introduces an innovative method for generating executable attack programs, showcasing the practicality of potential breach scenarios. This approach not only establishes the theoretical vulnerability of a system but also underscores its susceptibility to exploitation.
To respond to the second question, the proposed approach explores a range of mechanisms to counter and thwart the exposed attack strategies. The aim is to use robust and adaptive defensive strategies, leveraging insights from the demonstrated attack programs. These mechanisms encompass proactive measures, such as automatic penetration testing and behavior analysis, and reactive approaches, such as rapid patch deployment and vulnerability prioritization. The resilience of systems against potential breaches can be enhanced by intertwining attack pathways with comprehensive countermeasures, thereby disrupting the adversary's kill chains. This study aims to contribute to the containerized application security deployed in different environments, like the Cloud, Edge, 5G, Internet of Things (IoT), and Industrial IoT (IIoT), by taking these scenarios as a case study.
This research contributes to the evolution of cyber threat analysis through a Design Science Research (DSR) approach, focusing on developing and validating artifacts, tools, and frameworks. Defenders can anticipate, combat, and ultimately mitigate emerging threats in an increasingly complex digital environment by creating tangible attack programs and formulating effective thwarting mechanisms.

Keywords: Vulnerability · Prioritization · Penetration testing · Kill chain

1 Introduction

Cloud application vulnerabilities devastate our digital society, threatening privacy, finances, and critical infrastructure. Businesses must recognize this threat

and safeguard their company from cloud vulnerabilities. A 2021 study by IBM suggests that data breaches caused by cloud security vulnerabilities cost companies an average of $4.8 million to recover [14].
In the past few years, the research community has proposed sophisticated approaches and techniques to enhance automated security testing and promptly identify vulnerabilities before malicious attackers can exploit them.
As a result, many tools are available today to detect vulnerabilities effectively. However, most organizations do not know how to deal with hundreds of vulnerabilities because these tools are prone to produce false positives. The usual behavior is to patch them based on the produced vulnerability's score.
As if the problem wasn't complicated enough, the container orchestration scenario makes the situation more challenging due to the rapid and continuous deployment of new containers and pods in such environments.
In the real world, attackers combine various vulnerabilities to breach systems effectively; thus, analyzing vulnerabilities in combination with each other represents a fundamental step to obtain a realistic "big picture" of their implications. Furthermore, most defensive solutions are reactive, like intrusion detection systems, system calls monitoring, etc. Even if these techniques are well-established, they are affected by scalability problems and are not meant to prevent an attack from happening. The remainder of this paper is organized as follows. Section 2 gives some background information and a taxonomy of fundamental principles that guided the present work. Section 3 presents the research design. Finally, Section 4 presents the conclusions, and Section 5 outlines the principles of open science, which aim to democratize knowledge, increase transparency and collaboration, and enhance scientific research's quality and impact.

2 Background and Taxonomy

This section gives background information and a taxonomy of fundamental principles to understand the research questions and objectives.

2.1 Architecture

In the ever-growing landscape of software development and applications, technologies such as Docker [1], Kubernetes[5], and cloud providers are emerging as transformational forces. They represent a fundamental shift in how applications are built, managed, and scaled in today's computing environment. Docker, introduced in 2013, revolutionized containerization by packaging applications and their reliance on units called containers. These lightweight, portable containers enable developers to create consistent, isolated environments that run smoothly on any platform, from local development machines to production servers. Kubernetes, often shortened to K8s, emerged in 2014 as an open-source container orchestration platform developed by Google. It supports containerization, which can provide a more efficient and effective way to manage, scale, and deploy containerized applications. K8s abstracts the underlying infrastructure, letting

developers focus on defining desired application states, and the system handles the complex task of managing policies and containers across clusters of machines as depicted in Fig.1. Cloud providers such as Amazon Web Services, Microsoft Azure, Google Cloud Platform, etc., have greatly influenced the way infrastructure and IT resources are provisioned, managed, and utilized [15]. These providers offer a vast array of services, including computing power, storage, networking, and databases, accessible over the Internet on a pay-as-you-go basis. Businesses can leverage these technologies to scale resources on demand, reduce hardware costs, and achieve flexibility and agility in their operations.

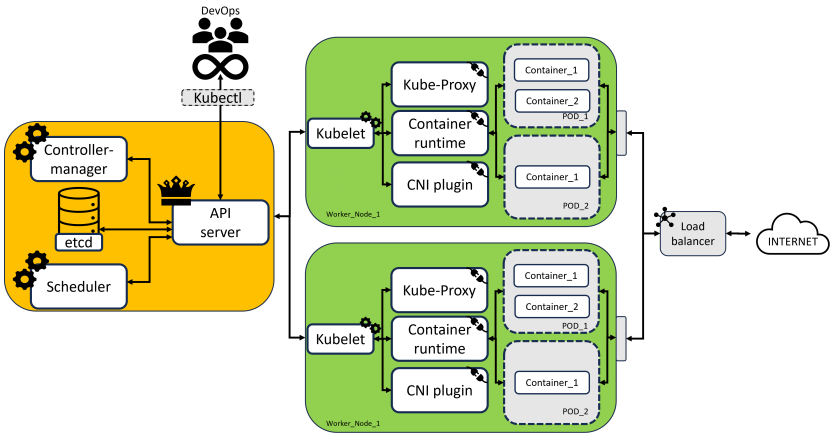

Fig. 1. Kubernetes simple scenario

2.2 Vulnerabilities and frameworks

In the rapidly evolving landscape of cybersecurity, protecting digital assets from potential threats and attacks is a constant challenge. Understanding and addressing vulnerabilities is essential to strengthen systems and networks effectively. Three crucial elements that form the foundation of vulnerability management are Common Vulnerabilities and Exposures (CVEs) [24], Common Weakness Enumeration (CWEs) [23], and misconfigurations [17]. Furthermore, the Common Vulnerability Scoring System (CVSS) [19] serves as a standardized method for assessing and quantifying the severity of identified vulnerabilities.
Understanding adversary behavior is crucial in cybersecurity. Two approaches exist for organizing information about adversarial actions: Common Attack Pattern Enumeration and Classification (CAPEC) and Adversarial Tactics Techniques & Common Knowledge (ATT&CK). Each is tailored for distinct use cases. MITRE ATT&CK [3] is a globally accessible knowledge base of adversary

tactics and techniques based on real-world observations. While CVE, CWE, and CVSS provide essential details about individual vulnerabilities and their severity, the ATT&CK framework comprehensively explains how attackers might exploit them within various cyber attack stages. The ATT&CK knowledge base creates particular threat models and methodologies in cybersecurity products and service communities.
On the other hand, CAPEC [22] focuses on application security and delineates the typical characteristics and methods attackers use to take advantage of recognized weaknesses in cyber-enabled capabilities (e.g., SQL Injection, XSS).

2.3 Effective Vulnerability Analysis

The main challenge for an effective vulnerability analysis and prioritization strategy is considering multiple vulnerabilities and their combined capabilities. In real-world settings, attackers put together ("chain") multiple vulnerabilities to successfully compromise systems. Thus, ranking vulnerabilities individually is insufficient and unrealistic. Hence, certain companies rely on penetration testers to thoroughly assess their security measures to understand the potential kill chains[13]. This concept involves structured phases delineating the attacker's advancement towards accomplishing goals.

2.4 Cloud, edge and IoT

As highlighted in the work by Koziolek, H. et al. [16], the dominant approach to software deployment is rapidly shifting towards containerization. Simultaneously, there is a growing fascination with employing container orchestration frameworks, extending beyond conventional data centers to encompass resource-limited hardware like Internet-of-Things devices, edge gateways, and more.
The ongoing effort to extend container orchestration to the edge represents a significant adaptation of containerization technology [8]. While container orchestrators were initially designed for managing cloud-based applications, they are now being applied to edge computing environments. This transition is driven by the need to efficiently manage and deploy containerized applications on resource-constrained devices like IoT devices and Edge gateways. Containers make building, deploying, and maintaining IoT applications easier, even when IoT devices have limited resources to support operating systems. This approach enables the deployment of containerized applications closer to where data is generated, reducing latency and enhancing real-time processing capabilities.
As shown in Fig.2, the containerization, orchestration, and the different use cases add layers to the bare code that needs to run. Thus, in turn, it can bring new specific vulnerabilities and misconfigurations, making the overall scenario more complex.

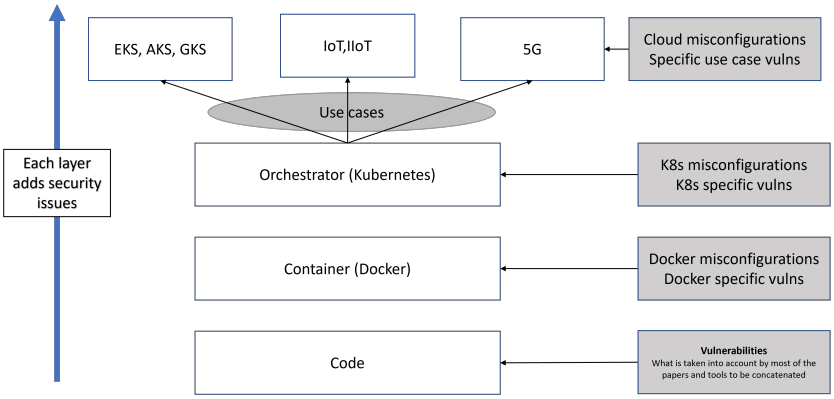

Fig. 2. Complete scenario

3 Research design

This study aims to encourage collaboration and the generation of more resilient solutions for addressing security issues in containerized applications. This section outlines the research objectives and provides an overview of the anticipated outcomes within the research plan.

3.1 Research objective

The research goals can be summarized into three main points.
1. Associating vulnerabilities with the ATT&CK framework while emphasizing kill chains: when considering vulnerability concatenation, the relationship between vulnerabilities and the ATT&CK framework gains depth. This concept illustrates how combining vulnerabilities, CVEs, CWEs, and misconfigurations shapes the ATT&CK matrix. Security becomes more proactive and holistic by recognizing and addressing vulnerabilities in conjunction.
2. Comprehensive security landscape understanding: gain a comprehensive security perspective by analyzing vulnerabilities within the context of the kill chain. Comprehend how these vulnerabilities align, facilitating their prioritization and aggregation for achieving maximal potential impact.
3. Automating penetration testing within container orchestration settings: conduct a study on vulnerability discovery-fix automation processes. Investigate the interplay between vulnerability discovery automation, the ATT&CK framework, and proactive defenses. Utilize insights to automate penetration testing, identifying and addressing vulnerability kill-chains within the container applications environment.

The first objective will provide understanding and reasoning about kill chains and how they can be mapped to the MITRE framework. Once the kill chain has

been recognized, the next step is to prioritize according to the most dangerous kill chain. Finally, the third objective aims to automatically produce kill chains, knowing what a kill chain looks like and how it can be produced in the most lethal way given a certain scenario.

3.2 Research planning

A series of chronological sub-research questions have been formulated to achieve the research objectives. These sub-research questions will be addressed by the conclusion of the Ph.D. program. An overview of the research flow and expected output is provided in Fig. 3.

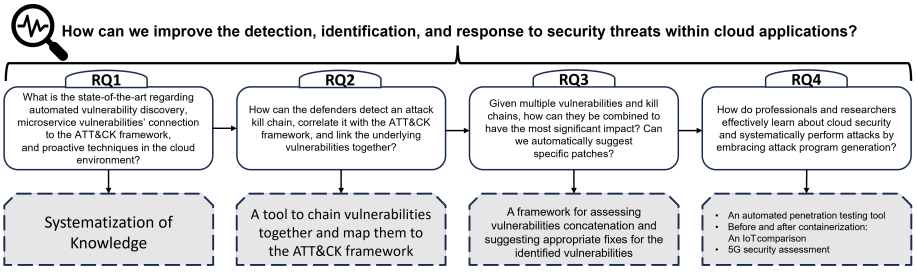

Fig. 3. Schematic research plan

3.3 Research Questions

The research question, which serves as the central query encapsulating the entire research concept and guiding the investigation, can be characterized as follows:

– How can we improve the detection, identification, and response to security threats within cloud applications?

To enhance clarity, the main question should be deconstructed into four sub-questions, each requiring answers about 'What' 'Why' and 'How. The research is still in the early stages. Still, the foundations have been laid, and tools will be developed according to the best technology when tackling the particular problem. The author acknowledges that the 'How' can be better defined, but the DSR framework [12] will be used for understanding, executing, and evaluating.

RQ1 What is the state-of-the-art regarding automated vulnerability discovery, microservice vulnerabilities' connection to the ATT&CK framework, and proactive techniques in the cloud environment?
What: This question aims to provide a comprehensive and structured review of the existing literature, research, and knowledge related to the attacks on cloud

architecture.
Why: The need for Systematization of Knowledge (SoK) in the literature is paramount, and Usenix's recent invitations to provide SoKs have proven highly valuable in assisting the security community in clarifying and contextualizing complex research problems. In this study, underexplored research areas will be identified, methodologies and tools will be thoroughly evaluated, historical context will be provided, existing approaches will be critically assessed regarding strengths and limitations, and potential future research directions with suggested improvements will be outlined.
How: The methodology involves formulating search queries that correspond to the research questions, followed by an extensive review of papers and grey literature to pinpoint deficiencies in the existing literature. The research question aims to conduct a comprehensive review of the state-of-the-art by systematically summarizing and analyzing existing literature to gain insights into automatic vulnerability discovery, exploitation, and patching, the relationship between microservices and the ATT&CK framework, and the use of active and proactive techniques in cloud environments.

RQ2 How can an attack kill chain be identified, matched to the ATT&CK framework, and the underline vulnerabilities be concatenated?
What: This question aims to understand how an attack kill chain can be identified and mapped to the ATT&CK framework.
Why: To the author's best knowledge, the literature does not focus on this mapping between CVEs, CWEs, and misconfiguration regarding the cloud environment. Similarly, only a few articles like [9] and [11] are trying to relate the vulnerabilities generated by a real-world scenario to the ATT&CK framework and automatically retrieve the CVEs, CWEs, and misconfigurations that enable a particular technique/tactic to be performed by the attacker. Minna, F. et al.[18] present a Sok on run-time security for cloud microservices, emphasizing that there is room to improve tools for microservices by adding functionality to correlate exploitation steps to MITRE ATT&CK tactics and CVEs that might be exploited. Furthermore, there are no tools that can automatically find an attack kill chain in this scenario.
How: In the complex environment of Cloud Native applications, carrying out vulnerability management is challenging. Vulnerability discovery has been heavily studied in the past years, to the point where tools like fuzzers and scanners are becoming arguably too good, and we are finding more vulnerabilities than we can properly fix, leading to alert fatigue [21]. This problem occurs when cybersecurity professionals become desensitized after dealing with overwhelming alerts. To fill this gap, this research will focus on designing a tool that can be integrated into the CI/CD pipeline to help developers and security teams check for vulnerabilities that enable a specific tactic in the ATT&CK framework.

RQ3 Given multiple vulnerabilities and kill chains, how can they be combined to have the biggest impact? Can we automatically suggest specific patches?

What: This question aims to develop an automated vulnerability assessment system that utilizes real-world vulnerabilities and kill chains to propose effective, context-aware fixes.
Why: A primary gap in the literature is the limited availability of valuable datasets for vulnerability analysis. Emphasizing the concatenation is essential because assessing vulnerabilities solely based on their individual severity is both inadequate and impractical. Once the kill chains are identified, the next open problem in the literature is to propose context-aware patches automatically.
How: Creating a framework for the novel score-based rule for assessing vulnerabilities in real-world scenarios, particularly focusing on the vulnerabilities concatenation and the blast radii. Introducing a novel technique enhances the assessment of vulnerabilities in real-world scenarios, focusing on understanding how vulnerabilities concatenate. It intends to develop a score-based rule focusing on blast radii by employing a metric such as the CVSS environmental score to evaluate vulnerabilities. Subsequently, the research aims to leverage generative AI or other approaches to suggest appropriate fixes for the identified vulnerabilities.

RQ4 How do professionals and researchers effectively learn about cloud security and systematically perform attacks in this domain?
What: This research question aims to create an automated penetration testing tool that unifies established vulnerable scenarios, incorporates potential automated pen testing solutions, and integrates prevalent kill chains. Furthermore, it sheds light on the interplay between IoT/IIoT and 5G regarding their security implications within containerization.
Why: Due to the lack of repositories where researchers and security teams can experiment with security practices in the cloud application environments, efforts will be made to establish a unified repository for vulnerable scenarios to streamline the availability and access to useful and on-purpose vulnerable scenarios. A significant emphasis within this work will be on achieving the effective compilation of kill chains to provide an exhaustive comprehension of security threats because only [7] is trying to model the chaining process in such a scenario.
It is crucial to prioritize security and portability integration in IoT/IIoT scenarios, but formalization in this field is weak. Furthermore, the security aspect of container orchestration in the intricate setting of 5G networks will be examined, as this specific concern has not been covered in existing literature.
How: This research aims to create an automated penetration testing service by improving consolidated known vulnerable scenarios like [2], exploring solutions for automating pen testing, and combining effective kill chains. As highlighted in [10], today's attacks are not fine-tuned to microservices architecture, so there is room for improvement. The primary objective is to gather various Kubernetes security tools, such as Pirates [4], Kube-hunter [6], Kubeaudit [20], and others, and integrate their functionalities.
Comparing IoT/IIoT security before and after containerization and validating the results using a formal method such as fault trees.

Leveraging the previously mentioned tools to highlight the issues introduced by the container orchestration in 5G (RAN and core) and spot new attack paths that are introduced.

4 Conclusions

In conclusion, this research will dig into the vulnerability concatenation realm, aligning it with the ATT&CK framework and emphasizing the crucial aspect of kill chains. The concept of vulnerability concatenation, encompassing vulnerabilities, CVEs, CWEs, and misconfigurations, provides a deeper understanding of the intricate nature of cyber threats. Doing so equips developers and security teams to proactively guard against attacks across the entire kill chain, recognizing and addressing the compounding impact of exploiting multiple weaknesses together. Moreover, this research underscores the importance of gaining a comprehensive security perspective by analyzing vulnerabilities in the context of the kill chain. This approach aids in prioritizing and aggregating vulnerabilities based on their alignment with specific phases, thereby maximizing their potential impact. Such an approach is particularly beneficial in the IoT/IIoT scenario.

5 Open Science Principles

In line with open science principles, this research is committed to promoting accessibility and transparency. The study will openly share the results and tools developed with the scientific community and the public, promoting a collaborative and inclusive approach to cybersecurity research. The primary platform for sharing code and outcomes will be GitHub, whereas the research papers will be open-access. This will allow others to build on the findings and contribute to the advancement of the field.

6 Acknowledgment

The author likes to thank his supervisors, H.G. Peter Bosch, Paul J.M. Havinga, and Willem Jonker, for their contributions to this line of work.

References

1. Docker website. https://www.docker.com/, accessed on April 12, 2026
2. Kubernetes goat. https://github.com/madhuakula/kubernetes-goat, accessed on April 12, 2026
3. Matrix - Enterprise | MITRE ATT&CK. https://attack.mitre.org/matrices/ enterprise/containers/, accessed on April 12, 2026
4. Peirates. https://github.com/inguardians/peirates, accessed on April 12, 2026

5. Production-grade container orchestration, https://kubernetes.io/, accessed on April 12, 2026
6. Aquasecurity: Kube-hunter (2023), https://github.com/aquasecurity/kube-hunter
7. Blaise, A., Rebecchi, F.: Stay at the helm: secure kubernetes deployments via graph generation and attack reconstruction. In: 2022 IEEE 15th International Conference on Cloud Computing (CLOUD). pp. 59–69 (2022). https://doi.org/10.1109/CLOUD55607.2022.00022
8. Goethals, T., De Turck, F., Volckaert, B.: Fledge: Kubernetes compatible container orchestration on low-resource edge devices. In: International Conference on Internet of Vehicles. pp. 174–189. Springer (2019)
9. Grigorescu, O., Nica, A., Dascalu, M., Rughinis, R.: Cve2att&ck: Bert-based mapping of cves to mitre att&ck techniques. Algorithms **15**(9), 314 (2022)
10. Gupta, C., van Ede, T., Continella, A.: Honeykube: Designing and deploying a microservices-based web honeypot. In: SecWeb 2023 (2023)
11. Hemberg, E., Kelly, J., Shlapentokh-Rothman, M., Reinstadler, B., Xu, K., Rutar, N., O'Reilly, U.M.: Linking threat tactics, techniques, and patterns with defensive weaknesses, vulnerabilities and affected platform configurations for cyber hunting. arXiv preprint arXiv:2010.00533 (2020)
12. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS quarterly pp. 75–105 (2004)
13. Hutchins, E.M., Cloppert, M.J., Amin, R.M., et al.: Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Leading Issues in Information Warfare & Security Research **1**(1), 80 (2011)
14. IBM Security: Cost of a data breach - a view from the cloud 2021. https://www.ibm.com/downloads/cas/JDALZGKJ (2021)
15. Kaushik, P., Rao, A.M., Singh, D.P., Vashisht, S., Gupta, S.: Cloud computing and comparison based on service and performance between amazon aws, microsoft azure, and google cloud. In: 2021 International Conference on Technological Advancements and Innovations (ICTAI). pp. 268–273. IEEE (2021)
16. Koziolek, H., Eskandani, N.: Lightweight kubernetes distributions: A performance comparison of microk8s, k3s, k0s, and microshift. In: Proceedings of the 2023 ACM/SPEC International Conference on Performance Engineering (2023)
17. Loureiro, S.: Security misconfigurations and how to prevent them. Network Security **2021**(5), 13–16 (2021)
18. Minna, F., Massacci, F.: Sok: run-time security for cloud microservices. are we there yet?. Computers & Security p. 103119 (2023)
19. National Institute of Standards and Technology (NIST): National Vulnerability Database. https://nvd.nist.gov/vuln-metrics/cvss, accessed on April 12, 2026
20. Shopify: kubeaudit (4 2023), GitHub https://github.com/Shopify/kubeaudit
21. Simonetto, S., Bosch, P.: Are we reasoning about cloud application vulnerabilities in the right way? In: 8th IEEE European Symposium on Security and Privacy (2023)
22. The MITRE Corporation: Common attack pattern enumeration and classification. Website, https://capec.mitre.org/, accessed on April 12, 2026
23. The MITRE Corporation: Common Weakness Enumeration (CWE). https://cwe.mitre.org/, accessed on April 12, 2026
24. The MITRE Corporation: CVE. https://cve.mitre.org/, accessed on April 12, 2026