Enhancing Cloud Native Security: Analysis of Kubernetes Vulnerabilities

Heekyung Shin, Jiwon Ock, and Seongmin Kim*

Sungshin Women's University, Seoul, Republic of Korea
{220224009, 220224011, sm.kim}@sungshin.ac.kr

Keywords: Kubernetes, MEC, CNFs,

1 Introduction

The new paradigm of containerized environments has been rapidly integrated with mobile edge cloud (MEC) for agile management and deployment of cloud-native mobile applications [1, 2]. Evolved from Virtual Network Function (VNF), Cloud Native Network Function (CNF) leverages container technology to process and deploy applications through network capabilities efficiently. At its core, containers package applications along with the necessary libraries and binaries. This packaging approach ensures reliable execution, addressing unexpected issues that may arise from environmental changes. CNFs are designed as containerized applications, managed within the Kubernetes platform. Notably, Kubernetes, developed by Google, has become a de-facto standard for orchestrating CNFs. According to the survey, 96% of cloud companies have adopted Kubernetes for this purpose [3]. However, the increasing adoption of Kubernetes to manage cloud-native applications in the network, deploying CNFs, and expanding their presence has brought security concerns to the forefront. As the complexity grows due to integrating various cloud-native components and heterogeneous containers, the potential attack surfaces for security breaches expand [4]. Hence, it is essential to conduct vulnerability analysis research aimed at fortifying Kubernetes' security within the context of cloud-native environments. This paper conducts an analysis of security vulnerabilities in Kubernetes, focusing on the 4Cs: Cloud, Clusters, Containers, and Code. The analysis is based on real-world attack scenarios within a vulnerable cluster environment known as Kubernetes goat. The insights gained from this research aim to enhance security practices for deploying mobile applications while securely utilizing CNFs within cloud-native MEC environments.

2 Case studies on Kubernetes Attack Scenario

Cloud Native Security in Perspective of the 4Cs. In the context of cloud-native security [5], it is possible to consider the application of a Defense-in-depth model by subdividing it into four layers: Cloud, Cluster, Container, and Code. The Code layer addresses application code vulnerabilities; the Container layer enhances container security by accounting for dependencies; the Cluster layer safeguards Kubernetes workloads through configuration and application vulnerability management; and the Cloud layer ensures foundational security across cloud providers and infrastructure.

Vulnerability analysis. This study simulates Kubernetes Goat with 11 attack scenarios and analyzes security vulnerabilities, mapping to vulnerabilities in the 4C categories. Kubernetes

The 7th International Conference on Mobile Internet Security (MobiSec'23), Dec. 19-21, 2023, Okinawa, Japan, Article No.P-18
*Corresponding author: Sungshin Women's University, Seoul, 02844, Republic of Korea, Email: sm.kim@sungshin.ac.kr

Goat [6] implements attack scenarios to intentionally construct vulnerable Kubernetes cluster environments. In the analysis of this particular attack scenario, the code layer revealed application vulnerabilities due to Server-Side Request Forgery (SSRF) attacks which are frequently exploited in the cloud [4]. A critical vulnerability was identified in the container layer, resulting in unauthorized access to the host system, attributed to vulnerable components within Docker, which serves as the container runtime engine. Furthermore, in the cluster layer, instances were identified where improper configuration of Kubernetes policies led to the misuse of unauthorized access privileges. It was observed in the cloud layer that significant consumption of Kubernetes infrastructure resources could lead to cases of Denial of Service (DoS) attacks that may impact resource availability. In Table 1, Kubernetes vulnerabilities are summarized based on refined 4C metrics.

	Vulnerabilities	**Attack Scenario**
Code	Exploiting SSRF vulnerabilities in application code	SSRF in the Kubernetes(K8S) world
Container	Access the host system by including a host path in the Docker filesystem	Container escape to the host system
	Exploit Docker container sockets to access the host system	DIND(docker-in-docker) exploitation
	Exposing sensitive information contained in Dockerfile	Hidden in layers
	Exposing environment variables in the Docker private registry	Attacking private registry
	Upload a container image with a crypto miner to use cluster resources	Analyzing crypto miner container
Cluster	Granting access to unexposed services due to Kubernetes configuration errors	NodePort exposed services
	Multi-tenant environment is exploited, exposing critical resources internally	Kubernetes namespaces bypass
	RBAC permissions are exploited on the Kubernetes API server	RBAC least privileges misconfiguration
	Deny traffic due to incorrect network policy configuration in a Kubernetes cluster	Secure Network Boundaries using NSP
Cloud	Lack of Resource Management Configuration in Pods Impacting Kubernetes Infrastructure Security	DoS the Memory/CPU resources

Table 1: Analysis of potential security vulnerabilities in each of the 4C layers

Discussion. We analyze vulnerabilities in a Kubernetes simulation environment and compile a list of known issues intending to aid cloud environment developers in creating secure cloud-native environments. It's also recognized that additional measures are needed to address vulnerabilities within Kubernetes, and CNF should adhere to the latest security recommendations from Kubernetes. Our follow-up study aims to implement scenarios regarding MEC components and propose a framework to mitigate attacks. This framework ensures network security, establishes network stability, and ensures safety in the face of potential threats to 5G MEC components.

Acknowledgement: This work was partly supported by the NRF grant funded by the Korean government (MSIT) No. 2021R1G1A1006326, a KIAT grant funded by the Korean Government (MOTIE) (P0008703), and the MSIT under the ICAN program (No. IITP-2022-RS-2022-00156310) supervised by the IITP.

References

[1] Y.Y. Shin, J.S. Shin, C.H. Park, and J.G. Park. ebpf technology trends for networking and security in cloud-native. *Electronics and Telecommunications Trends*, 37(5), 2022.

[2] Sofiane Imadali and Ayoub Bousselmi. Cloud native 5g virtual network functions: Design principles and use cases. In *2018 IEEE 8th International Symposium on Cloud and Service Computing (SC2)*, pages 91–96, 2018.

[3] Cloud Native Computing Foundation. CNCF Annual Survey 2021. https://www.cncf.io/reports/cncf-annual-survey-2021/ [Online].

[4] Andrew Martin and Michael Hausenblas. *Hacking Kubernetes.* " O'Reilly Media, Inc.", 2021.

[5] Kubernetes. Cloud Native Security. https://kubernetes.io/ko/docs/concepts/security/overview/ [Online].

[6] Madhu Akula, Vickie Li, and Travis Cotton. Practical kubernetes security learning using kubernetes goat. 2021.