Mitigating Security Attacks in Kubernetes Manifests for Security Best Practices Violation

Shazibul Islam Shamim*
mshamim42@tntech.edu
Tennessee Technological University
Cookeville, Tennessee, USA

ABSTRACT

Kubernetes is an open-source software system that helps practitioners in automatically deploying, scaling, and managing containerized applications. Information technology (IT) organizations, such as IBM, Spotify, and Capital One, use Kubernetes to manage their containers and reported benefits in the deployment process. However, recent security breaches and survey results among practitioners suggest that Kubernetes deployment can be vulnerable to attacks due to misconfiguration and not following security best practices. This research explores how malicious users can perform potential security exploits from the violations of Kubernetes security best practices. We explore how attacks can be conducted such as denial of service attacks against one of the security best practices violations in Kubernetes manifests. In addition, we are exploring potential exploits in the Kubernetes cluster to propose mitigation strategies to secure the Kubernetes cluster.

CCS CONCEPTS

• **Security and privacy → Software security engineering**.

KEYWORDS

software security, secure software engineering, kubernetes, devops, devsecops, security, attacks, attack mitigation, denial of service, dos, security policies, security practices, cloud computing, misconfiguration, manifests, configuration, compromised user, container

ACM Reference Format:
Shazibul Islam Shamim. 2021. Mitigating Security Attacks in Kubernetes Manifests for Security Best Practices Violation. In *Proceedings of the 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '21), August 23–28, 2021, Athens, Greece.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3468264.3473495

1 RESEARCH PROBLEM AND MOTIVATION

Kubernetes is an open-source software system that helps practitioners in automatically deploying, scaling, and managing containerized applications [7]. Practitioners report benefits as Kubernetes has removed the burden of repetitive manual processes in container deployment and speed up the deployment process. Organizations such as the United States Department of Defense use Kubernetes to manage their deployment and have reduced their release time from three to eight months to one week [2, 5].

Despite the reported benefits, recent surveys show that security is one of the primary concerns for practitioners. The survey result from the StackRox [9] suggests that 44% organizations delay their deployment for security concerns. The result also demonstrates that 94% of the organizations have faced at least one security incident in the last 12 months, among which 69% security issues are misconfiguration related [3]. The Cloud Native Computing Foundation (CNCF) survey [3] result shows that 32% practitioners among 1,324 survey participants consider that security is their primary challenge in Kubernetes deployment. Recent incidents of security breaches provide legitimacy of the practitioners concern. For example, Tesla experienced a malicious attack in 2018 for not adhering to security best practices. Tesla's Kubernetes console was not password-protected, and AWS credentials were exposed in a container[4].

Prior research has conducted a systematic literature study of grey literature such as blog posts, tutorials, videos, white papers, and identified security best practices for Kubernetes [8]. We also found that researchers conducted a study on open-source software(OSS) repositories and identified commits for updating security-related defects in Kubernetes manifests[1]. The presence of security defect-related commits in OSS repositories indicates that the practitioners may violate the Kubernetes security best practices in OSS repositories which can lead to malicious attacks. We are exploring how the malicious users can generate malicious attacks due to security best practices violation.

2 BACKGROUND

Kubernetes is an open-source software system that helps practitioners in automatically deploying, scaling, and managing containerized applications [7]. A Kubernetes installation is also referred to as a Kubernetes cluster [7]. A Kubernetes cluster contains two types of worker machines called nodes. One type of node is called the control plane node, and another type of node is called the worker node, as shown in Figure 1.

A control plane node has the following four components: 'kube-api-server', 'etcd', 'kube-controller-manager', and 'kube-scheduler'. The 'kube-api-server' controls the functionality of Kubernetes cluster through application program interface (API). The 'kube-api-server' orchestrates all the operations within the Kubernetes cluster. The 'kube-controller-manager' is a component on the control plane that watches the state of the cluster through the 'kube-api-server' and changes the current state towards the desired state.

*Adviser: Akond Rahman, Tennessee Technological University

ESEC/FSE '21, August 23–28, 2021, Athens, Greece

ACM ISBN 978-1-4503-8562-6/21/08…$15.00
https://doi.org/10.1145/3468264.3473495

The 'kube-scheduler' is the component in the control plane responsible for scheduling pods across multiple nodes. The 'etcd' is a key-value-based database, stores all configuration information for the Kubernetes cluster. Users use a command-line tool 'Kubectl' to communicate with the 'kube-apiserver' in the control plane.

The worker nodes host the applications that run on Kubernetes [7]. The worker node has the following components: 'kube-proxy', 'kubelet' and 'pod'. 'kube-proxy' maintains the network rules on nodes. 'kubelet' is an agent that ensures containers are running inside a pod. The pod is the smallest Kubernetes entity, which includes at least one active container. A container is a standard software unit that packages the code and associated dependencies to run in any computing environment [7]

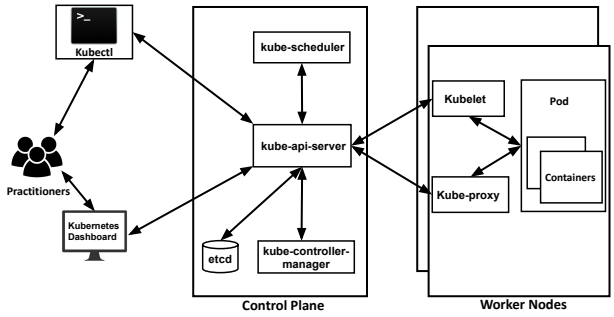

Figure 1: Kubernetes Architecture Overview

Kubernetes provides support for declarative programming [6] to specify Kubernetes cluster configuration. Practitioners use configuration files such as YAML files, also known as Kubernetes manifests, to configure the Kubernetes cluster.

3 APPROACH

In prior research, the researchers have performed a systematic literature study from grey literature and identify the security best practices for Kubernetes [8]. The researchers have described 11 types of security best practices to secure the Kubernetes cluster [8]. The researchers have recommended applying security best practices such as implementing Kubernetes-specific security policies, defining CPU, memory, and request limit for container deployment, avoiding default namespace by separating namespaces, etc [8].

In this research, we explore how the violation of security best practices can lead to potential exploits inside the Kubernetes cluster. We identify that if a malicious attacker gets some access inside the Kubernetes cluster, the attacker can perform potential exploits in the Kubernetes cluster. The malicious attacker can be any compromised user or over-privileged Kubernetes user inside the Kubernetes cluster. We configured a minimal Kubernetes cluster with two nodes, a control plane, and a worker node, to explore potential exploits.

We identify that a malicious user can get root access to the container running inside the pod due to misconfigured pod security policy. To explore the potential exploit, we deploy a container with root privilege access inside our Kubernetes cluster. We find that a malicious user with a compromised credential can access the container as a root user. The malicious user can get access to other containers in the Kubernetes cluster as well. We also find that not defining the CPU/memory resource limit for container deployment can lead to a denial of service attack. By default, Kubernetes starts running pod with unbounded resource limit. We identify that once a malicious get access inside the container, the malicious user can install and run malicious applications. Such applications can consume all the available CPU and memory resources for the node in Kubernetes. Whenever a Kubernetes user deploys a container inside the Kubernetes cluster, it is deployed in the 'default' namespace if a separate namespace is not defined. A namespace is a virtual cluster in the node that creates logical isolation to separate teams or application groups. We observe that one team can access sensitive applications such as databases of other team existing in the 'default' namespace. We hypothesize that we can build threat models for each of the attackers due to security practices violations and propose a prevention strategy to mitigate the possible attacks in the Kubernetes cluster.

4 CONCLUSION AND FUTURE WORK

Kubernetes is becoming an attractive choice for maintaining containers for organizations and practitioners. Securing Kubernetes installation requires more attention as default configurations Kubernetes is often insecure. In this work, we build a minimal Kubernetes cluster to explore potential exploits. We explore how the Kubernetes security best practices violation can lead to potential exploit by malicious users. In our future work, we will use repositories from GitHub and focus on how these violations appear in the OSS repositories. We will also explore more attacks due to security practices violation and propose prevention and mitigation strategies for securing Kubernetes cluster.

ACKNOWLEDGMENTS

We thank the PASER group at Tennessee Technological University for their valuable feedback. This research was partially funded by the U.S. National Science Foundation (NSF) award # 2026869.

REFERENCES

[1] Dibyendu Brinto Bose, Akond Rahman, and Md Shazibul Islam Shamim. 2021. 'Under-reported' Security Defects in Kubernetes Manifests. In *EnCyCriS 2021*. IEEE.
[2] Cloud Native Computing Foundation 2020. With Kubernetes, the U.S. Department of Defense Is Enabling DevSecOps on F-16s and Battleships. https://www.cncf.io/case-study/dod/
[3] Cloud Native Computing Foundation 2020. CNCF SURVEY 2020. https://www.cncf.io/wp-content/uploads/2020/11/CNCF_Survey_Report_2020.pdf
[4] Dan Goodin 2018. Tesla cloud resources are hacked to run cryptocurrency-mining malware. https://arstechnica.com/information-technology/2018/02/tesla-cloud-resources-are-hacked-to-run-cryptocurrency-mining-malware/
[5] Kubernetes 2020. Kubernetes User Case Studies. https://kubernetes.io/case-studies/
[6] John W Lloyd. 1994. Practical Advtanages of Declarative Programming.. In *GULP-PRODE (1)*. 18–30.
[7] S. Miles. 2020. *Kubernetes: A Step-By-Step Guide For Beginners To Build, Manage, Develop, and Intelligently Deploy Applications By Using Kubernetes (2020 Edition)*. Independently Published. https://books.google.com/books?id=M4VmzQEACAAJ
[8] Md Shazibul Islam Shamim, Bhuiyan, Farzana Ahamed, and Akond Rahman. 2020. XI Commandments of Kubernetes Security: A Systematization of Knowledge Related to Kubernetes Security Practices. In *2020 IEEE Secure Development (SecDev)*. IEEE, 58–64.
[9] Stackrox 2020. Stackrox Kubernetes Security Report 2020. https://www.stackrox.com/post/2020/02/5-surprising-findings-from-stackroxs-latest-kubernetes-security-report/