Security for Containers and Kubernetes

Learn how to implement robust security measures in containerized environments

Luigi Aversa

www.bpbonline.com

First published: 2023

Published by BPB Online
WeWork
119 Marylebone Road
London NW1 5PU

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-55518-439

www.bpbonline.com

Dedicated to

My beloved wife:

Natalya

&

My Parents ***Rita*** *and* ***Andrea***

About the Author

Luigi Aversa has been working in the tech industry for more than 20 years, playing central roles in numerous projects as a technical leader and security engineer, delivering projects using Linux technologies, and combining DevOps skills with security acumen. Currently, he is a Staff Information Security Engineer at Grail. In the meantime, he successfully got many security certifications in the cyber security and security compliance fields. Furthermore, the author writes technical articles on information security, cyber security and related topics.

About the Reviewer

Werner Dijkerman is a freelance cloud, Kubernetes (certified), and DevOps engineer. He's currently focused on, and working with, cloud-native solutions and tools including AWS, Ansible, Kubernetes, and Terraform. He is also focused on Infrastructure as Code and monitoring the correct "thing" with tools such as Zabbix, Prometheus, and the ELK Stack, with a passion for automating everything and avoiding doing anything that resembles manual work. He is an active reader, comics, non-fictional and IT related books, where he is a Technical reviewer for various books about DevOps, CI/CD and Kubernetes.

Acknowledgement

I want to express my deepest gratitude to my family for their unwavering support throughout this book's writing, especially my wife Natalya. I also want to thank my parents Rita and Andrea for their encouragement and love throughout all my life.

I am also grateful to BPB Publications for their guidance and expertise in bringing this book to fruition. It was a long journey of revising this book, with valuable participation and collaboration of reviewers, technical experts, and editors.

I would also like to acknowledge the valuable contributions of my colleagues and co-worker during many years working in the tech industry and more lately in the information security field, who have taught me so much and provided valuable feedback on my work.

Finally, I would like to thank all the readers who have taken an interest in my book and for their support in making it a reality. Your encouragement has been invaluable.

Preface

Building a secure containerized environment is a complex process that requires a comprehensive understanding of the container stack and the hardware and software infrastructure on which it is built upon. The recent raise of Kubernetes as a container orchestrator solution expands the complexity and the security challenges of the container stack either on premise or in the cloud. Securing both systems requires a good understanding of the threat landscape, reference to the associated risks and knowledge of the powerful tools that have become increasingly popular in the cybersecurity field.

This book is designed to provide a comprehensive guide to understanding security best-practices for containers and Kubernetes. It covers a wide range of topics, including the concepts behind the virtualization of the container stack, advanced topics such as securing container automation and orchestration, and how to secure the Kubernetes cluster for building secure microservices.

Throughout the book, you will learn about the key concepts and techniques needed to secure a container platform from the ground up, including hardware and operating system. You will also learn about best practices and design patterns to apply security best-practices to application and microservices and you will be provided with numerous practical coding examples to help you understand and reproduce the concepts.

This book is intended for security practitioners, DevOps engineers, security engineers, cloud engineers, platform engineers, and cloud architects who play a pivotal role in containerization and Kubernetes deployment. This book is also intended for experienced professionals who want to expand their knowledge with some peculiar security best-practices or techniques in building robust and secure container and Kubernetes stacks.

With this book, you will gain the knowledge and skills to visualize the entire container stack and the security gaps that should be addressed to reduce the attack surface and increase the security posture of your container and orchestrator platforms, but also to obtain hands-on strategies for measuring, analyzing and evaluate the impact of threats and vulnerabilities. I hope you will find this book informative and helpful.

Chapter 1: Containers and Kubernetes Risk Analysis – provides a high level overview of the risks associated with the implementation of the container platform

and the Kubernetes orchestrator, including the risks associated with the underlying hardware and software infrastructure. It also provides a brief overview of the risk associated with container images and container registries as essential components of the container stack.

Chapter 2: Hardware and Host OS Security – presents a detailed overview of the main in-hardware security features that have direct impact on the virtualization technology and therefore on the container stack. It also presents a detailed overview of the main operating system security features that can be leveraged by one or more layer of the container stack including Kubernetes.

Chapter 3: Container Stack Security – provides an overview of the container systems available today to be integrated with Kubernetes, including Docker, and the security best-practices needed to reduce the attack surface and strengthening the network communication in a containerized environment. This chapter presents also a full secure connection section to enable Docker to use TLS communication that can be fully implemented to secure CI/CD systems.

Chapter 4: Securing Container Images and Registries – allows the reader to learn fundamental concepts related to container images and container registries. Image hardening and file configuration are the baseline upon which enabling a sufficient security posture, while vulnerability scanning helps to determine the image life cycle. This chapter also illustrates security strategies to store and retrieve container image either in private or public registries, and how and when applying security methodologies such as SAST or DAST.

Chapter 5: Application Container Security – describes in detail the security aspects of the microservices model by recalling application security frameworks such as OWASP CSVS and NIST SP-800.190. This chapter analyzes security testing models such as SAST, DAST, IAST and RASP and when applying these throughout the phases of the software development life cycle in a containerized environment.

Chapter 6: Secure Container Monitoring – shows an in-depth analysis of how to secure the container stack at any layer of the container infrastructure. It describes logic and methodologies behind container workloads, alerting and topology visualization; it also illustrates how to ingest metrics into the chosen SIEM, and how to detect a drift in a system behaviour by leveraging machine learning models.

Chapter 7: Kubernetes Hardening – provides a technical overview of the Kubernetes architecture and how to improve the security of the cluster by applying hardening to the control plane and data plane layers. This chapter highlights the importance of securing the network communication within the cluster with

a detailed walkthrough of the security techniques adopted to secure the cluster runtime interface and the PODs. It also provides an overview of the common treats the Kubernetes cluster is exposed to, including the POD escaping technique.

Chapter 8: Kubernetes Orchestration Security – is dedicated to introducing the readers to the fundamental of the orchestration by discussing the complexity of the Kubernetes cluster and how to apply security best-practices in high available scaling environments. This chapter also asserts the importance of securing fundamental components of the cluster such as the internal API system and the admission controller, while highlighting the potential threat the orchestrator is exposed to, and what are the recommended countermeasure.

Chapter 9: Kubernetes Governance – explains how to achieve a robust governance security layer able to cover the main weakness the Kubernetes cluster by leveraging policy engines that can propagate network and security policies. The chapter also describes the admission controller threat model, how to secure resource management and what are the limitation or missing security features affecting the cluster today.

Chapter 10: Kubernetes Cloud Security – is dedicated to describing the security features of the most popular Kubernetes cloud service solutions such as AWS EKS, Azure AKS, and Google GKE. This chapter discusses the shared responsibility models typical of the public cloud providers, and the new 4C security model recently created by the Cloud Native Computing Foundation. This chapter also describes the security peculiarities of less popular Kubernetes services such as OpenShift, Rancher and Tanzu.

Chapter 11: Helm Chart Security – explains best-practices to secure the most popular Kubernetes package manger, how to introduce integrity into external packages handling by the mean of cryptography, and how to establish a circle of trust implementing public keys verification into infrastructure as code tools like Terraform. This chapter also explains how to scan helm charts for vulnerabilities and how to address supply chain security threats by adopting the SBOM model.

Chapter 12: Service Mesh Security – describes in detail the microservices architecture and security benefits that the service mesh brings by running in parallel with the workloads. This chapter explains the container network interface, why the choice to use the Envoy proxy in the service mesh architecture and how to secure it, and delves into the security aspects of the mutual TLS system. This chapter also describes Istio security features, and the zero-trust networking model.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/1i48l0x

The code bundle for the book is also hosted on GitHub at **https://github.com/bpbpublications/Security-for-Containers-and-Kubernetes**. In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

Chapter 1
Containers and Kubernetes Risk Analysis

Introduction

At the time of writing the most popular version control system, **GitHub** hosts nearly 143,000 repositories related to containers with over 23 million commits, while over 106,000 repositories are related to Kubernetes with over 3 million commits. The Kubernetes repository itself hosts nearly 110,000 commits. Those impressive numbers are clearly the sign of an exponential growth that highlights how the microservice age has evolved over the last few years. More than that, it is the sign of how the need to adopt containerized solutions and how to manage them has become prominent across the spectrum of the software development life cycle.

As containers and the use of Kubernetes grow, so does the need to secure the systems. The most common cause of incident is the "known threat": misconfiguration. Almost 70% companies reported a misconfiguration in their containerized environment, making "ignoring the basics" the most common type of vulnerability.

Structure

In this chapter, we will discuss the following topics:

- Host OS Risks
    - Attack Surface

    - System-Level Virtualization
    - Component Vulnerabilities
    - Authentication
    - File System Integrity
- Image Risks
    - Image Vulnerabilities
    - Image Misconfiguration
    - Embedded Secrets
    - Embedded Malware
    - Untrusted Images
- Registry risks
    - Non-secure connections
    - Stales images
    - Authentication and Authorization
- Container Risks
    - Container Runtime
    - Network Traffic
    - The Application Layer
    - Rogue Containers
- Orchestrator Risks
    - Admin Access
    - Unauthorized Access
    - Network Segregation
    - Workload Levels
    - Worker Node Trust

Objectives

This chapter aims to provide a brief but significant overview of the main risks associated with the implementation of containerized solutions, including the technical components often forgotten, especially in agile environments where the DevOps methodology is applied.

Host OS risks

First and foremost, what is it a host, and why it is an important part of risk analyzing? A host OS is the software that interacts with the underlying hardware, and it represents the first layer of security we should look at from the software standpoint. In *Chapter 2, Hardware and Host OS Security*, we will also look at the hardware layer and its intrinsic bond with the operating system. Container and orchestrator technologies have surfaced along with the adoption of DevOps practices that attempt to improve the integration between building and running applications; as a result, the Host OS or operating system is something that is often overlooked due to the shift in focus. Many readers are already familiar with the difference between the deployment of applications within containers and virtual machines, but it is helpful recalling the difference visually in *Figure 1.1, Virtual Machines and Containers Structure*, facilitating the understanding of the risk this section aim to address. Refer to the following figure:

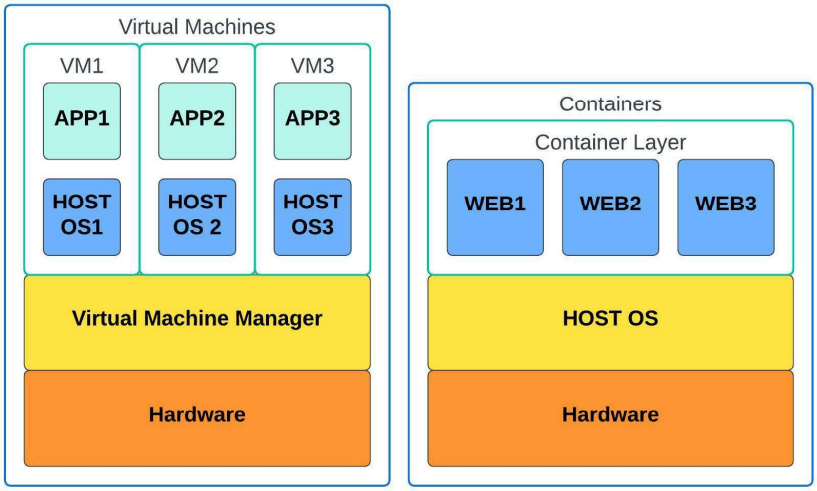

***Figure 1.1**: Virtual Machines and Containers Structure*

Figure 1.1, Virtual Machines and Containers Structure shows that regardless of the deployment methodology, the operating system is a crucial component of that deployment, except for some dedicated Cloud services like AWS ECS or Azure container instances where the burden of maintaining the underneath OS layer shifts back to the Cloud provider.

Both approaches allow multiple applications to share the same hardware infrastructure, but while the virtual machines use a hypervisor that provides hardware-abstraction via a virtual machine manager, the containers approach allows multiple applications to "share" the same operating system. From the security perspective, the hypervisor is also responsible for providing hardware-level isolation across virtual machines,

while the container service is responsible for enabling hardware-level resources for running containers.

The thoughts about Cloud Managed Services would include a wider argumentation that is not the objective of this chapter, so it is deferred to *Chapter 10, Kubernetes Cloud Security*, for a deeper analysis.

Attack surface

An operating system has an attack surface as much as any other platform or system. The extension of the attack surface is strictly connected to the type of operating system and to the technical philosophy behind it. A Linux desktop distro would potentially have a wider attack surface than a Linux server minimal distro, and a Windows 11 system would potentially have a wider attack surface than a Windows Nano server system.

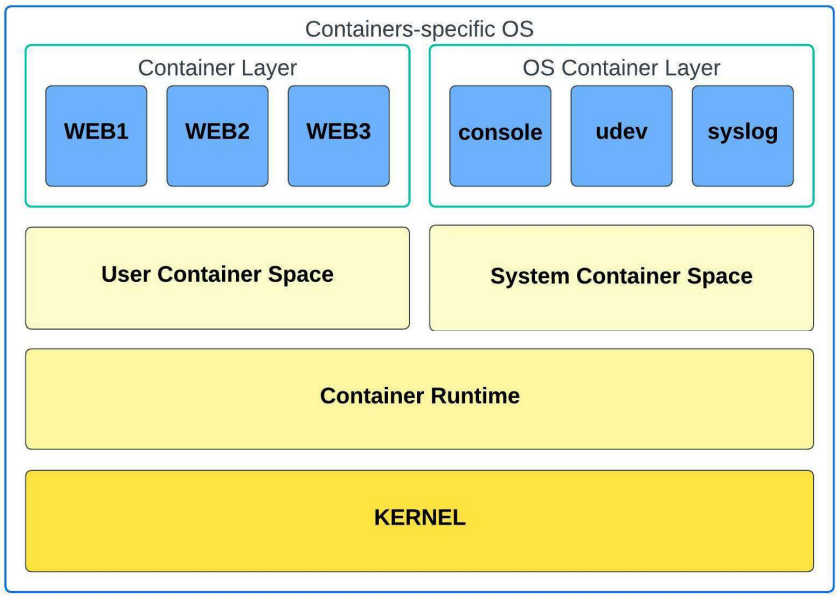

***Figure 1.2**: Container-specific OS*

There are essentially two types of Host OSes: ***General-purpose OSes***, such as Ubuntu, openSUSE Leap, and RedHat Enterprise Linux; and ***Container-specific OSes***, such as CoreOS Container Linux (now Fedora CoreOS), openSUSE Leap Micro, and RancherOS. The former category is the Host OS as we know it, typically used in any known application environment, while the latter has been specifically designed to have a minimalistic approach to run containers. In some cases, such as **openSUSE MicroOS**, **RancherOS** or **Clear Linux**, the Host OS itself is a containerized abstraction of the operating system, capable of providing atomic updates via rolling release