Brought to you by:

SUSE

Zero Trust Container Security

for dummies

A Wiley Brand

Plan your migration to Zero Trust security

Discover the SUSE Security Stack for Zero Trust

Understand Zero Trust use cases

Fei Huang
Glen Kosaka
Tom Callway

SUSE Special Edition

About SUSE

SUSE is a global leader in innovative, reliable, secure enterprise-grade, open-source solutions, relied upon by more than 60 percent of the Fortune 500 to power their mission-critical workloads. We specialize in business-critical Linux, enterprise container management, and edge solutions, and collaborate with partners and communities to empower our customers to innovate everywhere — from the data center to the cloud to the edge and beyond.

SUSE's solutions power everything from autonomous driving to CAT scan and mammogram machines. Our open-source software is embedded inside of air traffic control systems, weather forecasting technologies, trains, and satellites.

- **Business-critical Linux:** The SUSE Linux Enterprise family provides a stable, secure, and well-supported Linux operating system for mission-critical workloads, such as SAP S/4HANA and other solutions.
- **Enterprise container management:** Rancher solutions enable customers to standardize cloud-native workload operations across all devices and landscapes, with end-to-end security meeting the highest standards thanks to SUSE's NeuVector technology.
- **Edge solutions:** Edge offerings bring the best of SUSE's Linux and container technologies together. This is helping SUSE to truly innovate at scale by pushing business applications to where they're needed most.

SUSE puts the "open" back in open source, giving customers the agility to tackle innovation challenges today and the freedom to evolve their strategy and solutions tomorrow. The company employs more than 2,000 people globally. SUSE is listed on the Frankfurt Stock Exchange.

Zero Trust Container Security

SUSE Special Edition

by Fei Huang, Glen Kosaka, and Tom Callway

for dummies®
A Wiley Brand

Zero Trust Container Security For Dummies®, SUSE Special Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: WHILE THE PUBLISHER AND AUTHORS HAVE USED THEIR BEST EFFORTS IN PREPARING THIS WORK, THEY MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES REPRESENTATIVES, WRITTEN SALES MATERIALS OR PROMOTIONAL STATEMENTS FOR THIS WORK. THE FACT THAT AN ORGANIZATION, WEBSITE, OR PRODUCT IS REFERRED TO IN THIS WORK AS A CITATION AND/OR POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE PUBLISHER AND AUTHORS ENDORSE THE INFORMATION OR SERVICES THE ORGANIZATION, WEBSITE, OR PRODUCT MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING PROFESSIONAL SERVICES. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR YOUR SITUATION. YOU SHOULD CONSULT WITH A SPECIALIST WHERE APPROPRIATE. FURTHER, READERS SHOULD BE AWARE THAT WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ. NEITHER THE PUBLISHER NOR AUTHORS SHALL BE LIABLE FOR ANY LOSS OF PROFIT OR ANY OTHER COMMERCIAL DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGES.

ISBN 978-1-119-98210-4 (pbk); ISBN 978-1-119-98211-1 (ebk)

For general information on our other products and services, or how to create a custom *For Dummies* book for your business or organization, please contact our Business Development Department in the U.S. at 877-409-4177, contact info@dummies.biz, or visit www.wiley.com/go/custompub. For information about licensing the *For Dummies* brand for products or services, contact BrandedRights&Licenses@Wiley.com.

Publisher's Acknowledgments

Some of the people who helped bring this book to market include the following:

Project Editor: Elizabeth Kuball

Acquisitions Editor: Ashley Coffey

Editorial Manager: Rev Mengle

Senior Client Account Manager: Matt Cox

Production Editor: Pradesh Kumar

Special Help: Lawrence Miller, Caroline Kvitka

Table of Contents

Introduction

Containers and tools like Kubernetes enable enterprises to automate many aspects of application deployment, providing tremendous business benefits. But these new deployments are just as vulnerable to attacks and exploits from bad actors and insider threats as traditional environments are. Ransomware extortion, cryptojacking, data theft, and service disruption will continue to be used against new, container-based, virtualized environments in both private and public clouds.

To make matters worse, new tools and technologies like Kubernetes and managed container services in the public cloud will themselves be under attack as a gateway into an enterprise's most valuable assets. The recent Kubernetes man-in-the-middle vulnerability and exploit at Tesla are just the first among many against container-technology-based exploits expected to proliferate in the months and years ahead.

Although containers are generally more secure by default than traditional applications, the threat landscape is changing. Attack techniques are evolving, and the sophistication of attackers always matches or outperforms new infrastructure approaches. Bad actors are constantly developing new and novel ways to attack code and infrastructure, including containers.

Zero Trust is based on the security axiom of "Never trust, always verify." Although the concept of Zero Trust security has been around for decades, its practical application to containerized environments is relatively new. In this book, you find out how to get started on the Zero Trust journey for your organization's container environment.

About This Book

Zero Trust Container Security For Dummies, SUSE Special Edition, consists of five chapters that explore the following:

- The need for Zero Trust security in containerized environments (Chapter 1)

- How to migrate to Zero Trust security and establish practical controls in your environment (Chapter 2)
- The SUSE Zero Trust security stack (Chapter 3)
- Key Zero Trust use cases (Chapter 4)
- Ten resources to help you get started with Zero Trust container security (Chapter 5)

Each chapter is written to stand on its own, so if you see a topic that piques your interest, feel free to jump ahead to that chapter. You can read this book in any order that suits you (though we don't recommend upside down or backward).

Foolish Assumptions

It's been said that most assumptions have outlived their uselessness, but we assume a few things nonetheless!

Mainly, we assume that you're interested in learning about the Zero Trust security model and how to adopt it in a containerized environment. Whether you're a chief information security officer (CISO), an IT manager or practitioner, or a DevSecOps professional, this book will help you understand Zero Trust container security and get you started on your journey to Zero Trust.

If any of these assumptions describe you, then this is the book for you! If none of these assumptions describe you, keep reading anyway — it's a great book and after reading it, you'll have complete trust in your knowledge of Zero Trust container security!

Icons Used in This Book

Throughout this book, we occasionally use special icons to call attention to important information. Here's what to expect:

REMEMBER

This icon points out important information you should commit to your nonvolatile memory, your gray matter, or your noggin.

TECHNICAL STUFF

This icon explains the jargon beneath the jargon and is the stuff legends — well, legendary nerds — are made of.

TIP

Tips are appreciated, but never expected, and we sure hope you appreciate these useful nuggets of information.

Beyond the Book

There's only so much we can cover in this short book, so if you find yourself at the end of this book wondering, "Where can I learn more?," go to `https://suse.com`.

IN THIS CHAPTER

- » Recognizing the limitations of traditional perimeter-based security
- » Looking at Zero Trust container security
- » Discovering the seven pillars of a Zero Trust security strategy

Chapter 1
Understanding Why Container Security Is Different

This chapter explores the changing security perimeter and new security challenges in modern cloud-native environments. We explain how Zero Trust security protects containerized application environments at scale with a proactive approach, as well as the seven pillars of a comprehensive Zero Trust security strategy.

The Security Perimeter Is Changing

Cloud computing and the rapid adoption of containers have accelerated digital transformation and the pace of business, while simultaneously introducing new enterprise security challenges.

As digital infrastructure evolves from physical hardware deployed in on-premises data centers to virtualized environments that span data centers, as well as public and private clouds, the traditional security perimeter — separating the "trusted" internal network from the "untrusted" Internet — has all but disappeared. Modern, cloud-native applications have further eroded the security perimeter with microservices architectures that span multicloud environments. Finally, Kubernetes has enabled container orchestration at massive scale, ushering in the era of distributed workloads (see Figure 1-1).

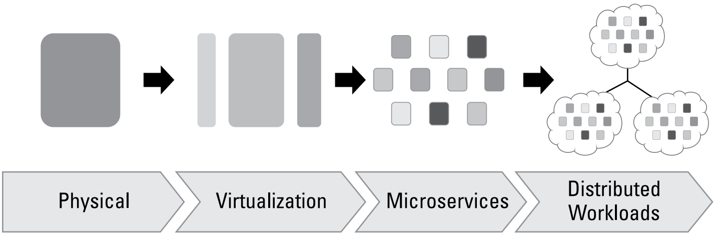


FIGURE 1-1: The traditional security perimeter is changing — and disappearing.

Traditional infrastructure

In a traditional corporate network infrastructure, users inside the network were inherently trusted. They connected to an on-premises data center from the corporate headquarters, or from branch offices via private networks, to run business applications and access sensitive data. Network firewalls were deployed at the security perimeter between the data center and the Internet and were designed to block untrusted Internet traffic from the corporate network.

Other traditional security tools, such as intrusion detection systems (IDSs) and intrusion prevention systems (IPSs), were similarly designed and deployed to inspect this north–south traffic between the Internet and the corporate network and block any unwanted traffic.

Virtualization

Virtualization technologies began gaining popularity more than 20 years ago, beginning with virtual machines and quickly advancing to include virtual storage, virtual networks, and more. As workloads were increasingly virtualized, a key fallacy in traditional perimeter-based security was exposed: The notion of a trusted internal network is fatally flawed.

Traditional security technologies deployed at the network perimeter are blind to traffic between virtual workloads in the data center (east–west) that do not cross an arbitrary security boundary. This blind spot enables attackers who successfully breach traditional perimeter defenses to move freely once inside the corporate network.

REMEMBER

Virtualization is a key enabling technology — if not the key enabling technology — in cloud computing. As enterprises "lift and shift" their traditional infrastructure to the cloud in infrastructure-as-a-service (IaaS) offerings and re-platform or migrate their applications to platform-as-a-service (PaaS) and software-as-a-service (SaaS) solutions, the traditional perimeter-based security model is becoming increasingly ineffective and irrelevant.

Reactive security approaches

Traditional security approaches are reactive in nature and don't scale to protect modern cloud-native applications and containerized environments. Examples of reactive security controls and processes include

- Creating deny list policies
- Identifying threats based on signatures, keywords, and regular expression (regex) matches
- Scanning for known vulnerabilities
- Implementing bad IP and URL lists
- Deploying agent-based protections
- Leveraging zero-day threat intelligence feeds
- Relying on manual configuration and management
- Verifying compliance based on point-in-time snapshots

TECHNICAL STUFF

A *regex* is a text string used in programming languages, such as Java and Perl, to search for, manipulate, and edit text that matches a specified sequence of characters.

The impact of cloud-native applications

Modern cloud-native applications are built on a microservices architecture composed of hundreds or even thousands of discrete services that run in on-premises data centers, as well as in public and private clouds. These microservices are highly dynamic and are often ephemeral, with a life cycle measured in minutes or seconds. Thus, traditional perimeter-based technologies and reactive approaches to security, which are often based on a static location and IP address, can't scale to protect cloud-native applications in an increasingly cloud-native world.

A deep and wide container attack surface

Logical constructs in containers have created a new attack surface that includes container network services, application namespaces, hosts, pod-to-pod communications, sidecars, ingress/egress, and data services.

Kubernetes enables container orchestration and distributed workloads at scale but exponentially increases the container attack surface to include service meshes, cluster-to-cluster communications, multi-cloud and hybrid cloud environments, functions, application programming interfaces (APIs), serverless, and edge computing.

The resulting attack surface extends across the container life cycle — from build and test to staging and production — and includes the following (see Figure 1-2):

- Critical vulnerabilities that can be introduced in the continuous integration/continuous delivery (CI/CD) pipeline at any stage
- Misconfiguration errors in registries, Kubernetes, and container hosts
- New attack surfaces in Kubernetes, Docker, Istio, and other tools
- Inadequate protection in production environments from container exploits and zero-day attacks

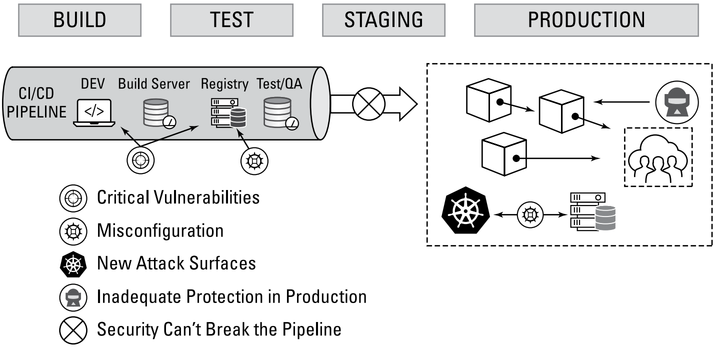


FIGURE 1-2: The container attack surface is deep and wide.

What Is Zero Trust Security?

Zero Trust is based on the concept of "Never trust, always verify." Instead of inherently trusting traffic that is "inside" the network perimeter, Zero Trust security requires every entity on the corporate network to be positively identified and explicitly permitted to perform any action on the network.

A proactive security approach

Zero Trust security takes a proactive approach to security. Instead of relying on reactive, signature-based protections, Zero Trust security is based on a declarative model in which you define acceptable behavior and block everything else. Allow lists are highly proscriptive and minimize the attack surface. Components of a proactive, Zero Trust security approach include the following (see Figure 1-3):

- Allow list policy (untrust by default)
- Live segmentation (and micro-segmentation)
- Behavior learning and locking down
- CI/CD pipeline security and DevSecOps
- Supply-chain security
- Cloud-native deployment and management
- Shared security responsibility
- Policy as code and security automation

- Distributed, lightweight, large-scale, highly available deployments
- Continuous compliance enforcement

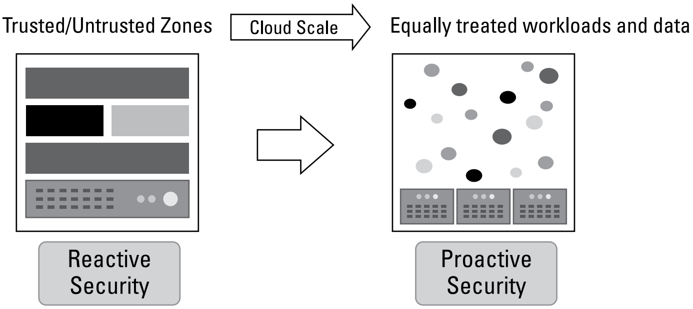


FIGURE 1-3: Comparing reactive and proactive approaches to security.

Full life-cycle security: Defense in depth

A defense-in-depth strategy helps ensure the security of your application development environment across the full life cycle. This strategy addresses both supply-chain security and runtime security (see Figure 1-4):

- **Supply-chain security:** Open-source and proprietary code needs to be secured to prevent unauthorized and potentially vulnerable images from being deployed into your production environment. Key controls include
  - Vulnerability scanning
  - Compliance scanning
  - Admission control
- **Runtime security:** New zero-day and traditional threats need to be detected and prevented in production environments using comprehensive, scalable, and automated controls to prevent runtime attacks. Key controls include
  - Runtime scanning
  - Threat-based controls
  - Zero Trust controls

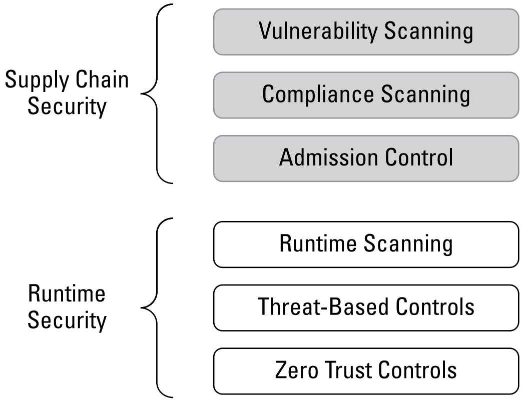


FIGURE 1-4: Defense in depth.

From deny list to allow list

A traditional, reactive approach to security requires you to create deny lists and blocking rules. In a Zero Trust model, you explicitly declare the allowed behavior for all your workloads — the network connections that the firewall should enforce and the process and file activity that is allowed in those containers or workloads. Application developers must be able to declare (and update) allowed behaviors in the CI/CD pipeline so they can be programmatically pushed or declared into the production environment.

From pipeline to production

Full life-cycle vulnerability and compliance management is key to a Zero Trust security strategy. This includes continuous scanning and compliance monitoring of open-source and proprietary code, vulnerabilities, compliance issues, Center for Internet Security (CIS) benchmarks, compliance reports, and run-time scanning. Run-time protections include security policy as code, security automation, admission controls, container firewalls, container workload security, and alerts and forensics (see Figure 1-5).

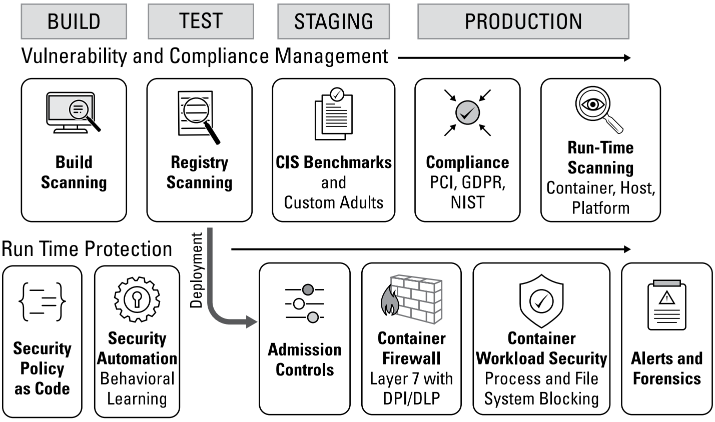

BUILD
TEST
STAGING
PRODUCTION
Vulnerability and Compliance Management
Build Scanning
Registry Scanning
CIS Benchmarks and Custom Adults
Compliance PCI, GDPR, NIST
Run-Time Scanning Container, Host, Platform
Run Time Protection
Deployment
Security Policy as Code
Security Automation Behavioral Learning
Admission Controls
Container Firewall Layer 7 with DPI/DLP
Container Workload Security Process and File System Blocking
Alerts and Forensics

FIGURE 1-5: Full life-cycle container security, from pipeline to production.

The Seven Pillars of a Zero Trust Security Strategy

A comprehensive Zero Trust security strategy protects seven key pillars, or assets, within an organization (see Figure 1-6).

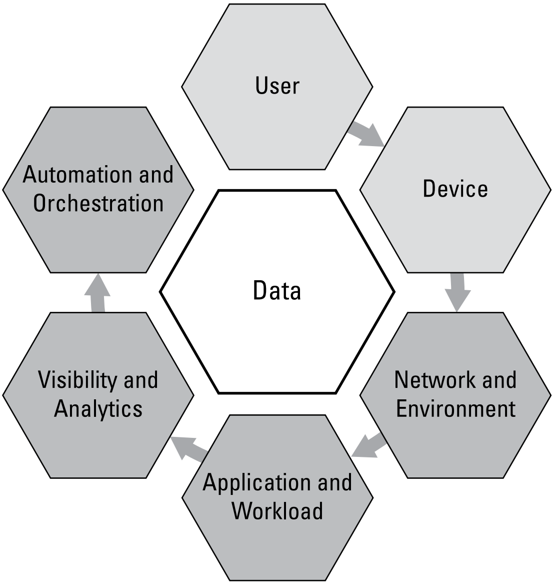


FIGURE 1-6: The seven pillars of Zero Trust security.

User

The first pillar of Zero Trust focuses on the user — specifically, user identity. Zero Trust requires every user to be positively identified before being allowed to connect to the network and access network resources. Identity and access management (IAM) technologies, such as multifactor authentication (MFA) and single sign-on (SSO), enable users to be correctly identified and assigned appropriate permissions based on the principle of least privilege.

Devices

Devices, like users, must also be positively identified and properly authenticated before being allowed to connect to the network and access network resources. Widespread adoption of bring your own device (BYOD) policies and deployment of Internet of Things (IoT) devices has led to a proliferation of devices on the network, which has exponentially increased the enterprise attack surface.

Network and environment

Securing the network and your IT environment in a Zero Trust strategy requires granular segmentation (including micro-segmentation) and deployment of modern next-generation firewalls and cloud security controls.

Application and workload

Zero Trust for applications and workloads must address traditional application infrastructures and workloads, as well as modern cloud-native workloads. As discussed earlier in this chapter, traditional security technologies and approaches are ineffective for protecting modern cloud-native applications and workloads that span multi-cloud and hybrid cloud environments. A Zero Trust, declarative approach to application and workload security is a requirement for keeping security up to date with CI/CD pipelines, which are constantly deploying or updating applications.

Visibility and analytics

Ensuring end-to-end visibility across your digital estate and deploying cloud-native security and management solutions is critical. Likewise, you need to understand and identify normal and malicious behavior in your environment with robust analytics.

Automation and orchestration

Security automation and orchestration enables a Zero Trust security strategy to scale with highly dynamic multi-cloud and hybrid cloud environments. Relying on manual configuration and management of security policies and technologies increases risk due to potential misconfiguration errors and costly delays.

Data

At the center of Zero Trust security is data, which is the most important asset that organizations ultimately must protect. Securing your data requires you to understand the nature of your data, where it's located, and how it's used. Robust data protection includes encryption for data at rest and in transit, as well as data loss prevention technologies.

IN THIS CHAPTER

- **Addressing the seven pillars in four steps**
- **Setting up practical security controls**

Chapter 2 Migrating to Zero Trust Security

This chapter outlines the four steps to implementing Zero Trust security in your organization, as well as some practical controls to help you get started.

Four Steps to Zero Trust

Chapter 1 introduces the seven pillars of a Zero Trust strategy:

- User
- Devices
- Network and environment
- Application and workload
- Visibility and analytics
- Automation and orchestration
- Data

This section explains how to implement these seven pillars using a four-step process in which each successive phase builds on the previous phases (see Figure 2-1).

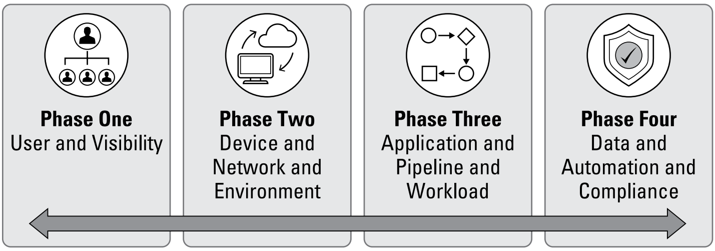

Phase One
User and Visibility
Phase Two
Device and Network and Environment
Phase Three
Application and Pipeline and Workload
Phase Four
Data and Automation and Compliance

FIGURE 2-1: Four steps to Zero Trust security.

Understand user profiles and gain asset visibility

The first phase of a Zero Trust security implementation focuses on getting to know your users and establishing complete, end-to-end visibility across your entire environment, which is referred to as a *protect surface.*

Begin by identifying the critical assets in your protect surface. These assets include your users, devices, networks, applications (software), workloads (hardware), and data. Knowing what assets need to be protected helps you properly define the scope and budget for your project. This information will also help you determine any blind spots where you don't have enough visibility in your environment.

After you've collected your critical asset information, you need to allocate budget and establish a formal project. Implementing Zero Trust is a major undertaking for any organization and will require dedicated resources — including budget and assigned personnel — to be successful.

Next, perform a security assessment to identify any potential gaps and opportunities for improvement. Knowing your current state and envisioning your desired future state will help you determine the steps necessary to achieve your goal and will help your team understand what success looks like.

Create user profiles to identify your user groups based on job functions, access requirements, behavior patterns, and other characteristics. This information will help you implement role-based access controls with least privilege access.

TIP

During this phase, consider enabling multifactor authentication (MFA) for all your users and privileged identity management (PIM) for any administrative accounts.

Harden your devices and network environment

The second phase of a Zero Trust security implementation focuses on devices and your network environment. In this phase, you perform a risk assessment of your overall environment and implement tools to provide deep visibility across your network environment.

Other key activities in this phase include

- Identifying and protecting workload connectivity with encryption
- Implementing egress and ingress traffic control
- Hardening the host environment

Build security into your pipeline and deploy micro-segmentation

The third phase turns your attention to your applications, including your continuous integration/continuous delivery (CI/CD) pipeline and workloads. Characterize the expected behavior of each application to be able to declare the allowed behavior, including network connections, process, and file activity of that application in the production environment. Deploy microsegmentation where possible to ensure protection moves dynamically with your workloads and integrate security into your CI/CD pipeline ("shift left," that is, address security requirements early in the application development life cycle).

TIP

Vulnerability management, risk management, security posture management, and run-time security processes should all be implemented in this phase.

Expand to include data protection, automation, and compliance

Phase four of your Zero Trust security implementation expands your efforts to include data protection, end-to-end security automation, and continuous compliance monitoring. During this phase, you should also begin implementing traditional security tools that have been optimized for container and cloud deployments, such as web application firewalls (WAFs) and data loss prevention (DLP).

Continuous compliance monitoring and auditing help organizations in regulated industries comply with various standards and mandates related to data security and privacy, such as the Payment Card Industry (PCI) Data Security Standards (DSS), Health Insurance Portability and Accountability Act (HIPAA), System and Organization Controls Type 2 (SOC 2), and General Data Protection Regulation (GDPR).

Finally, security automation and analytics should be implemented, including security information and event management (SIEM); security orchestration, automation, and response (SOAR); and extended detection and response (XDR).

Establishing Practical Controls

As you implement Zero Trust security in your organization, you should establish various practical controls to help secure your environment and improve your overall security and risk posture.

Minimize attack surface

Minimizing your attack surface extends to all seven pillars of Zero Trust security, including

- Enforcing least privilege access
- Implementing vulnerability, risk, and posture management
- Establishing "deny by default" policies — all entities are untrusted by default
- Hardening host operating systems and container images by removing all unnecessary packages, libraries, and services

Enforce in real-time

Comprehensive real-time security policy enforcement is essential to the success of Zero Trust security. Deploy technologies that prevent malicious behavior and restrict lateral movement across your environment, inspecting and blocking attacks at network entry points, stopping unauthorized process or file activity in containers, and securing external connections. Sensitive data must also be secured within applications to prevent data exfiltration.

Provide continuous visibility and compliance

Ensure comprehensive visibility across your environment, including deep packet inspection (DPI) to ensure sensitive data is not exfiltrated from your network and malicious traffic is not hiding in encrypted traffic. Implement tools to help ensure continuous compliance with policies and applicable regulatory requirements.

Be transparent and automated

Security should be automated, scalable, and integrated with your CI/CD pipeline and workflows. Leverage analytics services to provide transparency and deep insights into your environment.

IN THIS CHAPTER

- **Monitoring the Zero Trust stack and enforcing policies**
- **Enabling Zero Trust controls and ensuring secure configurations**
- **Delivering security at the operating-system level**
- **Deploying a secure cloud-native hyperconverged infrastructure**
- **Protecting data at rest with encryption**

Chapter 3 Exploring the SUSE Security Stack for Zero Trust

This chapter introduces the full portfolio of solutions in the SUSE Security Stack for Zero Trust, including NeuVector, Rancher, Harvester, SUSE Linux Enterprise Server, and Longhorn.

NeuVector

NeuVector empowers organizations to comprehensively secure their Kubernetes-native applications without compromising business velocity. The NeuVector unified security and compliance platform simplifies and automates security, while providing Zero Trust security for Kubernetes-native applications from pipeline to production.

Profile risk with vulnerability management

NeuVector scans for vulnerabilities across the entire continuous integration/continuous delivery (CI/CD) pipeline, from build to ship to run (see Figure 3-1):

1. Enforce CI/CD security.
2. Prevent vulnerable and unauthorized deployment.
3. Prevent external attacks and data theft.
4. Prevent lateral attack spread.
5. Prevent exploits and malware damage in a container.
6. Assess host/orchestrator security posture and prevent attacks.

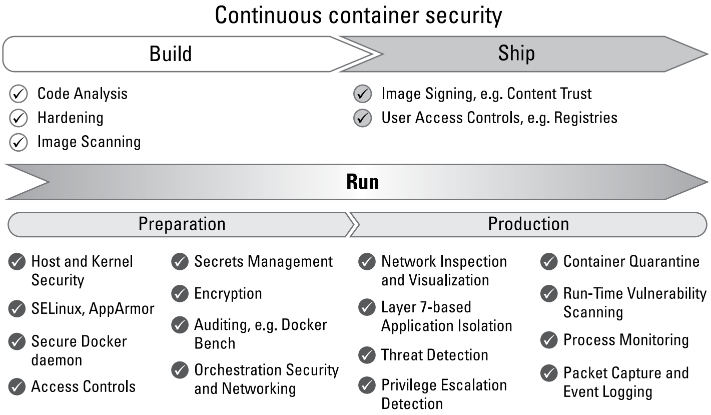

Continuous container security
Build
Ship
Code Analysis
Hardening
Image Scanning
Image Signing, e.g. Content Trust
User Access Controls, e.g. Registries
Run
Preparation
Production
Host and Kernel Security
SELinux, AppArmor
Secure Docker daemon
Access Controls
Secrets Management
Encryption
Auditing, e.g. Docker Bench
Orchestration Security and Networking
Network Inspection and Visualization
Layer 7-based Application Isolation
Threat Detection
Privilege Escalation Detection
Container Quarantine
Run-Time Vulnerability Scanning
Process Monitoring
Packet Capture and Event Logging

FIGURE 3-1: NeuVector delivers full life-cycle container security.

TIP

An example of Zero Trust in the CI/CD pipeline would be to declare trusted production registries and users in admission control rules to block untrusted deployment sources.

NeuVector easily deploys as a container onto virtual machines or bare-metal operating system (OS) environments. The Enforcer container is deployed on each node to protect containers running on it. A Controller container manages the cluster of Enforcers. NeuVector can be managed through the console, representational state transfer (REST) application programming interface (API), or command-line interface (CLI).

Protect data in production

NeuVector discovers normal connections and application container behavior and automatically builds a security policy to protect container-based services. Using process and file system monitoring with Layer 7 network inspection, unauthorized container activity or connections from containers can be blocked without disrupting normal container sessions. Deep packet inspection (DPI) enables real-time identification and blocking of network, packet, zero-day, and application attacks, such as distributed denial-of-service (DDoS) and Domain Name System (DNS).

TIP

An example of Zero Trust would be, for each application, to review and customize, if necessary, the allowed behavior for network connections, process, and file activity, and then lock the application down so any other activity is untrusted.

Security as code for DevOps and DevSecOps

As DevOps teams integrate their toolchain to enable automated deployment of container-based applications, one aspect has always slowed down a modern cloud-native pipeline: security. And although automated vulnerability scanning is now standard practice, creating the security policies to protect application workloads in production has largely been a manual process. The use of Kubernetes custom resource definitions (CRDs) to capture and declare an application security policy early in the pipeline now solves this problem.

In order to "shift left" security, developers can take the initial task of creating not only the application deployment manifest but also the security manifest. The images are built, and automated vulnerability scanning is completed and reviewed by DevOps. After those steps are done, DevOps can test both the deployment manifest and the security manifest and provide feedback to developers on the results (see Figure 3-2).

The DevOps team can then deploy new apps together with the security policy for the apps into the production environment, ensuring that apps are secured as soon as they start running in production.

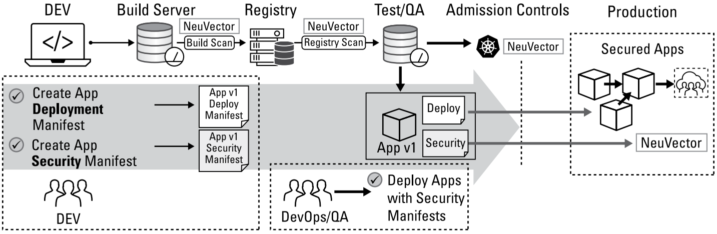


FIGURE 3-2: DevOps teams can "shift left" by creating security as code with NeuVector.

TIP

A Zero Trust example would be for DevOps to create (manually or through automated learning) the security manifest, test, review, and check in this "security policy as code," to be used in any cluster to which the application is deployed.

Rancher

Rancher is a complete container management platform built on Kubernetes. As shown in Figure 3-3, Rancher consists of four primary components:

- A certified Kubernetes distribution, including Rancher Kubernetes Engine (RKE) and K3s, the Cloud Native Computing Foundation (CNCF) lightweight Kubernetes sandbox project
- Consistent cluster operations
- Security/authentication/policy management/governance
- Developer platform services

TECHNICAL STUFF

RKE is a straightforward, lightning-fast Kubernetes installer that works everywhere. RKE is particularly useful in standing up Kubernetes clusters on VMware clusters, bare-metal servers, and virtual machine instances on clouds that don't yet support a Kubernetes service. K3s is packaged as a single binary (approximately 50MB in size) with everything needed to run Kubernetes anywhere, including low-powered Internet of Things (IoT) and edge devices. The binary includes the container runtime and any important host utilities like iptables, socat, and du. The only operating-system dependencies are the Linux kernel itself and proper dev, proc, and sysfs mounts (this is done automatically on all modern Linux distributions).

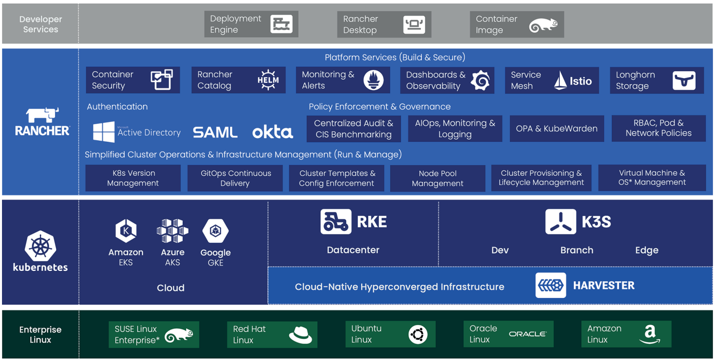

Developer Services
Deployment Engine
Rancher Desktop
Container Image
RANCHER
Platform Services (Build & Secure)
Container Security
Rancher Catalog
HELM
Monitoring & Alerts
Dashboards & Observability
Service Mesh
Istio
Longhorn Storage
Authentication
Active Directory
SAML
okta
Policy Enforcement & Governance
Centralized Audit & CIS Benchmarking
AIOps, Monitoring & Logging
OPA & KubeWarden
RBAC, Pod & Network Policies
Simplified Cluster Operations & Infrastructure Management (Run & Manage)
K8s Version Management
GitOps Continuous Delivery
Cluster Templates & Config Enforcement
Node Pool Management
Cluster Provisioning & Lifecycle Management
Virtual Machine & OS* Management
kubernetes
Amazon EKS
Azure AKS
Google GKE
Cloud
RKE
Datacenter
K3S
Dev
Branch
Edge
Cloud-Native Hyperconverged Infrastructure
HARVESTER
Enterprise Linux
SUSE Linux Enterprise*
Red Hat Linux
Ubuntu Linux
Oracle Linux
ORACLE
Amazon Linux
a

FIGURE 3-3: An overview of Rancher's recipe for production-quality Kubernetes at scale.

Security vulnerabilities in container environments are a well-known issue. Rancher includes features to address key security vulnerabilities and help operators manage risk, including the following:

- **User ID tracking** has been added to audit logs to help users trace events. Rancher now includes the Identity Provider name in both Rancher and Kubernetes audit logs. This helps promote the self-service model of Rancher, giving users clarity to identify different owners of clusters.
- **Image scanning** for common vulnerabilities and exploits (CVEs) is now automated across all images as part of releases, helping users easily determine if there are any major vulnerabilities across images in their cluster. If any critical vulnerabilities are found, Rancher has predetermined actions to help identify, fix, and/or mitigate issues.
- **SLE BCI** (short for SUSE Linux Enterprise Base Container Image) is Rancher's base image for microservices, which allows users access to a secure, open repository.
- **Cluster Templates** allow operators to create, save, and reuse well-tested Kubernetes configurations across their cluster deployments. These templates leverage controls and best practices from the most recent Kubernetes Benchmarks from the Center for Internet Security (CIS). The Cluster Templates feature also includes an option for policy enforcement, which prevents configuration drift and assures that the clusters you deploy do not accidentally introduce security vulnerabilities as you scale.

- **CIS Security Scan** enables security and operations teams to automatically identify configuration errors by comparing their cluster settings with best practice guidance in the CIS Kubernetes Benchmark. When Rancher runs a CIS Security Scan on a cluster, it generates a report showing the results of each test, including a summary with the number of passed, skipped, and failed tests. The report also includes remediation steps for any failed tests.

Harvester

Harvester is a modern, open, interoperable hyperconverged infrastructure (HCI) solution designed to help operators simplify their stack. Built on a foundation of cloud-native solutions, when Harvester is used with Rancher, virtual and container workloads can be easily managed side-by-side, helping enterprises consolidate their infrastructure's complexity and scale using new cloud-native solutions.

Key benefits and features of Harvester include the following:

- **Reducing infrastructure complexity:** Simplify the operations of your stack with Rancher and Harvester. The native integration across both platforms provides operators comprehensive overview to manage both virtual and containerized workloads from core to the edge. Key features include the following:
  - **Deploying Harvester directly from Rancher:** Import and directly manage Harvester clusters directly through the Rancher Virtualization Management page.
  - **Leveraging Rancher's existing features:** Extend operational excellence through Rancher, including authentication and role-based access control (RBAC) for multitenancy support across Harvester clusters.
- **Leveraging modern solutions:** Integrate the latest technology across your stack and futureproof your infrastructure as you scale. Harvester is built on the latest cloud-native technology, including Kubernetes, Longhorn, and Kubevirt. Key features include the following:

  - **Lightweight technology:** Built on cloud-native solutions, Harvester consumes a small footprint of commodity hardware, helping organizations deploy and manage their workloads at the edge.
  - **Reliable and production-ready:** Native integration with Rancher provides a reliable, turnkey enterprise solution to comprehensively manage your infrastructure stack.
  - **Retaining and upskilling talent:** Implement new solutions like Harvester and Rancher to attract, upskill, and retain talent across the technical operational space.
- **Capitalizing on cloud-native economics:** Lower your organization's total cost of ownership (TCO) with Harvester and Rancher. Harvester allows operators to efficiently manage their infrastructure resources without compromising on functionality. Key features include the following:
  - **One hundred percent open-source and free to use without lock-in:** Reduce your reliance on expensive, bulky proprietary HCI solutions. Harvester is free to use with no licensing fees required and can be implemented across any environment from core to the edge.
  - **Not dependent on complex hardware:** Harvester is built on cloud-native solutions and doesn't rely on boutique storage appliances or specialty compute hardware. Instead, Harvester utilizes off-the-shelf commodity hardware components.
  - **Additional support for enterprise peace of mind:** Scale your enterprise infrastructure with confidence. SUSE Priority Support gives customers around-the-clock support from a team of experts when you need it.

TIP

Harvester is built by the same engineering team responsible for successful open-source projects including Rancher, Longhorn, K3s, and RKE. Harvester remains 100 percent open-source with the SUSE engineering team as its principal contributor.

SUSE Linux Enterprise Server

SUSE Linux Enterprise Server is a modular OS that paves the way for IT transformation in the software-defined era. The modern and modular OS helps simplify your IT environment, modernize your IT infrastructure, and accelerate innovation.

Many organizations today use traditional infrastructure, software-defined infrastructure, or a mix of traditional and software-defined. This leads to a hybrid IT scenario, where different types of IT infrastructure have different technologies, processes, and business drivers. The multimodal design of SUSE Linux Enterprise Server helps organizations transform their IT landscape by bridging traditional and software-defined infrastructure.

REMEMBER

SUSE is committed to delivering best-effort security to its customers and to the open-source community. SUSE engineers promptly react to security incidents, deliver premium-quality security updates, and continuously improve the security-related functionality in SUSE products.

SUSE ensures that compliance standards are applied consistently across your digital estate. The configuration, auditing, and automation features of SUSE Manager make it easy to ensure compliance with internal security policies and external regulations.

Key security features in SUSE Linux Enterprise Server include the following:

- **Data security:** Improved hardware-based data security, using AMD's Secure Encrypted Virtualization (SEV) technology, enables guest virtual machines to run in encrypted memory, helping protect them from memory scrape attacks from the hypervisor.
- **Security standards compliance:** SUSE Linux Enterprise Server is successfully certified after Common Criteria Certification at Evaluation Assurance Level (EAL4+). Multiple cryptography security modules are validated to fulfill the requirements of Federal Information Processing Standards (FIPS) 140-2, including Open Secure Sockets Layer (OpenSSL), Open Secure Shell (OpenSSH) client and server, strongSwan Internet Protocol Security (IPSec)–based virtual private networks (VPNs), the Kernel Crypto API, Mozilla Network Security Services (NSS) Level 2, and libgcrypt.
- **Trusted Platform Module (TPM) 2.0:** Implement hardware-based security with secure cryptoprocessor standard TPM 2.0.
- **Disk encryption:** Protect data at rest without additional software cost. Local and remote disk encryption is supported

using cryptctl for all on-premises, cloud, and hybrid installations and integration via the Enterprise Key Management Interoperability (KMIP) standard.

- **Single sign-on (SSO):** Shibboleth support in SUSE Linux Enterprise Server enables SSO using one identity across different domains for computer networks and web infrastructure.

Longhorn

Longhorn is a cloud-native, lightweight yet powerful distributed storage platform for Kubernetes that can run anywhere. When combined with Rancher, Longhorn makes the deployment of highly available persistent block storage in your Kubernetes environment easy, fast, and reliable.

Key benefits and features of Longhorn include the following:

- **Simplifying your operations:** Orchestrate your storage needs on-demand to eliminate the cross-functional challenges often experienced when provisioning enterprise storage. Key features include the following:
  - **One-click installation and deployment** so teams can easily install Longhorn directly via the Rancher application catalog, using Helm or with kubectl.
  - **A free, intuitive user interface (UI) dashboard** for easy understanding and management of your storage system.
  - **An infrastructure-agnostic solution** that operates the same across on-premises and cloud environments.
- **Maximizing development agility:** Lean engineering teams use Longhorn to easily manage their storage operations while maintaining the agility to build reliable applications. Key features include the following:
  - **One hundred percent open-source software** and a CNCF sandbox project.
  - **Simple and affordable** alternative to overly complex, container storage solutions from legacy vendors.
  - **Production-ready and free to use** with no licensing fees.

  - **Dynamic provisioning of volumes** enables self-service operations.

- **Enabling enterprise-grade disaster recovery:** Longhorn has multiple built-in security features to protect your data from threats, such as malicious attacks, human error, and system failures. Key features include the following:
  - **Create secure policies** and automate your backups within the Longhorn UI to eliminate errors.
  - **Build a resilient disaster recovery system** with features to give users control over cross-cluster replication and granularity over volumes.
  - **Leverage widely available storage formats** for backups, including network file system (NFS) and Amazon Simple Storage Service (S3).

REMEMBER

The SUSE Security Stack for Zero Trust includes the following solutions (see Figure 3-4):

- **NeuVector:** Monitors the Zero Trust stack and enforces workload, orchestrator, and host/OS security policies.
- **KubeWarden:** A flexible, customizable admission controller and policy engine supporting Rego-language-based security rules, with integration into NeuVector.
- **Rancher:** Provides and supports user Zero Trust controls through RBAC integration and SSO, and monitors secure configurations of Rancher and Kubernetes deployments through NeuVector.
- **SUSE Linux:** Provides numerous security capabilities and features at the OS level, including reduced Base Container Image (BCI) versions, supported by secure supply chain, and a hardened reduced footprint image.
- **Harvester:** SUSE's cloud-native hyperconverged infrastructure (HCI) solution leveraging advanced virtual machine isolation technology.
- **Longhorn:** Ensures Zero Trust protections by encrypting data at rest.

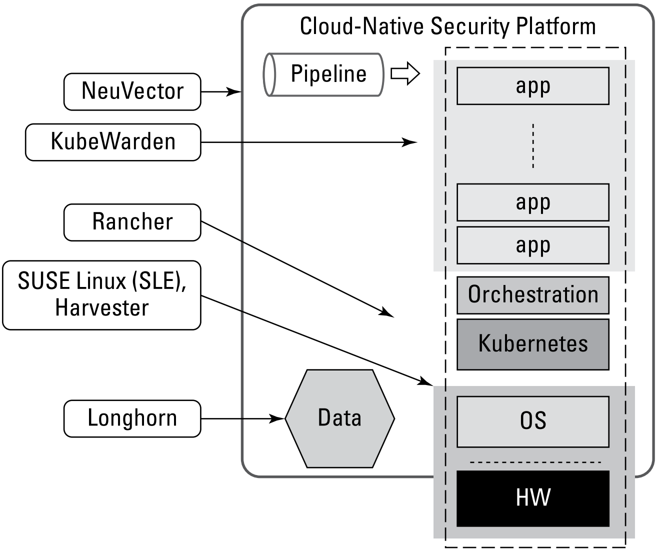


FIGURE 3-4: The SUSE Security Stack for Zero Trust.

IN THIS CHAPTER

- **Integrating vulnerability, risk, and posture management in your pipeline**
- **Protecting your supply chain and ensuring runtime workload security**
- **Achieving and maintaining continuous compliance**
- **Ensuring visibility, threat detection, and protection at scale**

Chapter 4 Looking at Zero Trust Use Cases

This chapter explores several common Zero Trust use cases and illustrates how real-world organizations are successfully addressing these use cases.

Vulnerability, Risk Management, and Security Automation

Implementing a comprehensive vulnerability and compliance program is essential to a Zero Trust strategy. When vulnerability management, risk management, and security posture management are integrated into the pipeline and automated for scanning, auditing, and alerting, enterprises can ensure that misconfigurations or manual security tasks don't inadvertently expose an attack surface that did not previously exist.

AUTOMATING STRONG ADMISSION CONTROLS IN HEALTH-CARE TECH

Preventing vulnerable, noncompliant, or unauthorized deployments is a critical security control for a major U.S.-based multinational health-care technology company.

In addition to criteria based on image scans, such as critical common vulnerabilities and exposures (CVEs) and compliance violations, the security team worked with the DevOps team to develop a set of requirements for all Kubernetes deployments to ensure the security and health of the applications. These requirements included security-related controls, as well as development settings and metadata, including the following:

- Labels to identify the owner of each application (for example, the developer or team)
- Resource requests and limits to ensure that an application did not consume too many resources
- Allow lists for the specific production registries and repositories from which approved images could be deployed

By establishing a combination of Zero Trust controls and deny-list controls (such as critical and high-severity CVEs), the health-care IT company can ensure that each new deployment is clean and approved.

Vulnerability and compliance management teams use image scanning, Center for Internet Security (CIS) benchmarks, and other tools to ensure that vulnerable or noncompliant software doesn't enter the pipeline, progress through it, and get deployed into production.

REMEMBER

The critical link to the production environment is through admission controls, which can enforce Zero Trust deployment policies.

Supply Chain Plus Runtime Workload Security

Application workloads are the fundamental business services and logic that many companies rely on for customer service, account management, financial transactions, and more. These applications have traditional attack surfaces (such as web application interfaces), as well as new ones (such as container orchestration layers and networks).

Whether applications are developed in house, acquired from a third party, or both, the production workloads must be monitored continuously for attacks and exploit attempts must be blocked in real time.

PROTECTING DEV PIPELINES AND PRODUCTION WORKLOADS

A large public energy utility providing electricity to 15 million customers across a service territory of 50,000 square miles needed to apply a Zero Trust security model and securely containerize its applications. The utility runs many third-party applications provided by GE Power, the U.S. Department of Energy, and other service providers, as well as its own custom-developed applications.

Although some suppliers may do their own scanning and have supply-chain security processes in place, highly regulated companies designated as critical infrastructure, such as public utilities, must also scan any software images coming into their repositories. Only after satisfactorily passing these scans can applications be deployed to production.

In the case of this utility company, application deployment occurs across multiple public and private clouds and must be coordinated and secured across all environments. Moving to a Zero Trust security model enables the utility to define and apply network segmentation rules across diverse Kubernetes environments more easily and consistently.

(continued)

(continued)

NeuVector behavioral learning is used in a staging environment to characterize the applications, and network, process, and file rules are exported to declarative YAML files. After verification and testing, these policies can be consistently deployed as Kubernetes custom resource definitions (CRDs) across all Kubernetes clusters before workloads start running, so that Zero Trust workload protections are in place before the workloads are deployed.

Continuous Compliance

Many elements of a Zero Trust strategy address compliance requirements in many industry standards and frameworks such as the Payment Card Industry (PCI) Data Security Standards (DSS), American Institute of Certified Public Accountants (AICPA) System and Organization Controls Type 2 (SOC 2), and others.

Strong, regularly maintained access controls, including identity and access management (IAM) and role-based access controls (RBACs), with detailed user activity logging, are a foundation for many of these requirements. Additionally, vulnerability management; network segmentation; security information and event management (SIEM); and security orchestration, automation, and response (SOAR) monitoring and alerting systems must be in place for a successful compliance audit.

Zero Trust controls for these required security technologies and practices provide a more secure, scalable way to achieve compliance for modern cloud-native pipelines and production environments.

ACHIEVING PCI COMPLIANCE FOR A LARGE AIRLINE

A large European airline with more than 700 aircraft in its fleet is a global provider of transportation and logistics services. The company offers many business travel payment solutions for corporations, including various offerings for business travel payments, accounting, and analysis functions.

The airline is required to perform regular audits on its cloud-based infrastructure to comply with PCI DSS requirements. The airline's travel and expense reports application accesses highly sensitive cardholder information and runs in a mixed cloud environment that includes OpenShift and Microsoft Azure Kubernetes Service (AKS).

Key elements of PCI DSS, including network segmentation and vulnerability management, are critical to demonstrate that the airline has the appropriate security controls, visibility, and processes in place.

The airline selected NeuVector to help it achieve PCI DSS and other compliance requirements, perform vulnerability scanning, and implement segmentation in its containerized environments independent from its Kubernetes platforms. After successfully deploying NeuVector, the airline scaled out its services to 28 locations and countries around the world.

Network Visibility, Threat Detection, and Protection at Scale

Zero Trust controls are most effective for modern cloud-native deployments, but traditional security tools don't work in a Kubernetes environment. For example, traditional firewalls lack the mechanisms to declare allowed behavior rather than block lists, which cannot scale.

SECURING PUBLIC CLOUD PLATFORM-AS-A-SERVICE KUBERNETES SERVICES

A leading public cloud Kubernetes service provider offers infrastructure-as-a-service (IaaS), software-as-a-service (SaaS), and platform-as-a-service (PaaS) solutions — as well as the components that make up these solutions — in public, private, and hybrid cloud delivery models for the financial services industry. The environment is designed to build trust and enable a transparent public cloud ecosystem with the

(continued)

(continued)

specific features for security, compliance, and resilience that financial institutions require.

The cloud provider required traditional network protections to achieve compliance, but it also wanted modern Zero Trust controls to protect against advanced exploits. The cloud provider offers its own managed services on top of its managed Kubernetes clusters, such as financial applications, all of which require network visibility and protection across hundreds of large clusters.

The first step was to implement WAF-type protections for edge clusters where any incoming traffic was inspected for common Open Web Application Security Project (OWASP) threats, such as Structured Query Language (SQL) injection, cross-site scripting (CSS), cross-site request forgery (CSRF), and more. The ability to automate these rules through CRDs was critical to being able to simultaneously manage a large number of deployments. The second step was to deploy Layer 7 segmentation protections, using Zero Trust controls across all internal clusters, to detect and prevent internally originated and propagated attacks. For all clusters, strong Zero Trust egress controls are also required to allow only approved external connections from any cluster.

After thoroughly testing and comparing several possible solutions, the service provider selected NeuVector. Key factors in this decision included the following:

- Network security controls including ingress/egress, WAF, and intrusion detection system (IDS)/intrusion prevention system (IPS) extensions
- Inline blocking capability
- Kubernetes-native architecture that scales in large public cloud environments
- Validated performance in stress tests including thousand-node clusters and boot-storm testing

TECHNICAL STUFF

A cloud-native, application layer (Layer 7) firewall is critical for achieving network visibility and protection within a Kubernetes cluster. A true cloud-native application firewall can implement declarative Zero Trust rules, as well as traditional network threat detection such as web application firewall (WAF) and data loss prevention (DLP) rules using deep packet inspection (DPI).

IN THIS CHAPTER

- **Checking out NeuVector and Rancher**
- **Trying the Open Zero Trust project**
- **Reading the documentation, joining the Slack channel, and networking with the community**
- **Taking a class, attending an event, and signing up for a free webinar**
- **Creating policy as code with Kubewarden and scheduling a demo**

Chapter 5 Ten Resources to Help You Get Started with Zero Trust Container Security

Here are ten great resources to help you get started on your Zero Trust container security journey.

Download and Try NeuVector

NeuVector is one of the first Kubernetes-native, full life-cycle container security platforms for Rancher and the entire cloud-native ecosystem. NeuVector is an enterprise-grade solution that protects many Fortune 500 enterprise customers, government agencies, and midsize to large companies worldwide. Key use cases include the following:

- **Vulnerability management:** Continuously scan throughout the container life cycle from pipeline to production.
- **Compliance:** Automated, audit-ready compliance assessment and reporting for Payment Card Industry (PCI) Data Security Standards (DSS), System and Organization Controls Type 2 (SOC2), and more.
- **Runtime security:** Block known and unknown threats with NeuVector's patented container firewall technology delivering Zero Trust for containers in production and protecting your network, files, and processes.
- **Supply-chain security:** NeuVector covers the entire continuous integration/continuous delivery (CI/CD) pipeline with complete vulnerability management, compliance scanning, and admission controls.
- **Network visibility:** NeuVector provides Layer 7 visibility within and between pods, patented deep packet inspection to detect and block threats, and network mapping.
- **Container segmentation:** Essential for PCI DSS and other requirements, NeuVector creates a virtual wall to keep personal and private information securely isolated on your network.

Download and try NeuVector from your Docker Hub account at `https://hub.docker.com/orgs/neuvector/repositories`. Check out the source code at `https://github.com/neuvector/neuvector`.

Install Rancher and Longhorn

To install Rancher in any Kubernetes cluster, follow the Rancher quick-start guide, available at `https://rancher.com/quick-start`.

After installing Rancher, you can access the Rancher user interface (UI) by opening a browser and going to the hostname or address where you installed Rancher. The UI will guide you through setting up your first cluster.

Like K3s, Longhorn started life as a Rancher Labs project but was donated to the CNCF in October 2019. To find out more about Longhorn, visit `https://longhorn.io` or read the Longhorn technical documentation at `https://longhorn.io/docs`.

Get Open Zero Trust

SUSE is known for its strong commitment to the open-source community. The company has contributed the Open Zero Trust (OZT) project to the Cloud Native Computing Foundation (CNCF). Open Zero Trust is the upstream project for NeuVector. It includes all the core technologies and functions being validated and used by customers worldwide. SUSE will actively work with the CNCF and the open-source community to grow the project and improve security for the Kubernetes ecosystem.

Visit `https://openzerotrust.io` and `https://github.com/openzt` to learn more.

Read the Documentation

Okay, you may not like hearing "Just ask for directions" or "Read the manual," but great open-source projects typically have great technical documentation, and Rancher and NeuVector are no different.

Topics on the Rancher documentation site range from Rancher 2.x and Rancher Kubernetes Engine (RKE) to K3s (lightweight Kubernetes) and more.

The NeuVector documentation site covers everything from deploying NeuVector 5.x on Kubernetes or using Helm Charts or OpenShift Operator to popular topics such as CI/CD automated scanning, Security Policy as Code, enterprise authentication and single sign-on (SSO), and more.

Get started at `https://rancher.com/docs` and `https://open-docs.neuvector.com`.

Join the Slack Channel and Check Out the Rancher YouTube Channel

Join the Rancher Users Slack channel to ask questions and learn from other community members, share your experiences using Rancher, and stay up-to-date on future events and training sessions. Check out the NeuVector-security channel at `https://slack.rancher.io`.

If you're just getting started and you don't have much time to read documentation, check out Rancher's huge library of recorded meetups, Master Classes, Office Hours videos, and introductory training on Rancher's YouTube channel at `http://youtube.com/c/rancher`.

Learn with Rancher Academy

SUSE's global, diverse community of aspiring and accomplished cloud-native practitioners — including application developers and testers, release and DevOps engineers, and application and infrastructure operators — is eager to learn but doesn't have time to waste. That's where Rancher Academy comes in. This free learning platform can help you quickly develop the practical skills and knowledge you need to confidently deliver cloud-native applications. You'll find a range of offerings, from introductory classes on Kubernetes, Rancher, and security to multipart classes to get you up and running with Rancher, K3s, and other technologies. A wide variety of master classes is also available, where you can do a deep dive into a topic.

Get details on Rancher Academy at `https://suse.com/community`.

Attend an Event

Rancher Virtual Rodeos are free, in-depth online workshops designed to give DevOps and IT teams the hands-on skills they need to deploy and manage Kubernetes everywhere. The content is delivered by Rancher's technical experts and aims to educate anyone interested in learning how to use containers or Kubernetes.

Find all the latest on-demand and live events, virtual conferences and meetups, online meetups, rodeos, and more taking place online and in your region at `https://suse.com/events`.

Sign Up for a Cloud Native Computing Foundation Webinar

The Cloud Native Computing Foundation (CNCF) regularly runs free webinars covering a variety of topics, including:

- Application and development
- CI/CD
- Customizing and extending Kubernetes
- Machine learning and data
- Observability
- Security identity and policy
- Serverless
- Service mesh
- Storage

Sign up for the next CNCF webinar at `www.cncf.io/webinars`.

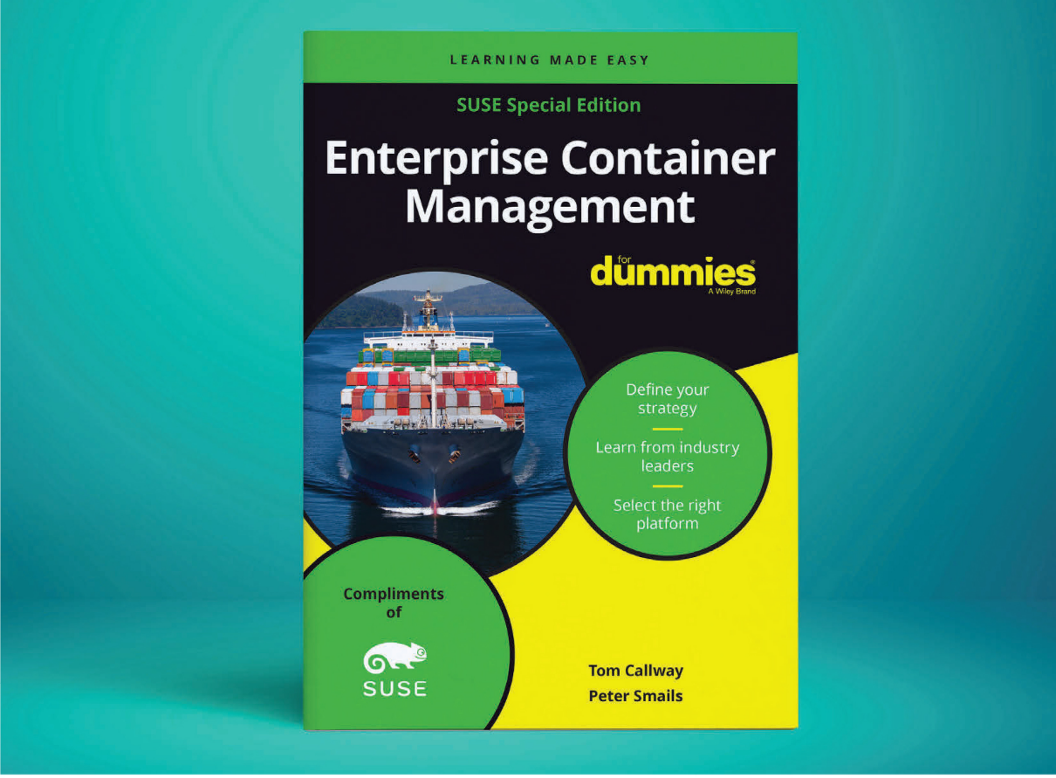
Download Kubewarden

Kubewarden is a policy engine for Kubernetes that simplifies the adoption of policy-as-code. You can write policies in your favorite programming language, compile them in WebAssembly, and distribute them using container registries. Policies are portable and can be integrated into CI/CD pipelines.

Learn more and download Kubewarden at `www.kubewarden.io`.

Schedule a Demo

Discuss your unique business and technical requirements with a SUSE expert. Go to `www.suse.com/contact` to get in touch with the SUSE sales team.



Build an enterprise-grade Kubernetes environment

As enterprise applications become more complex, development and operations teams need a tool to orchestrate that complexity. Kubernetes is that tool, allowing enterprises to deploy, scale and manage containerized applications anywhere. This guide walks you through the process of adopting an enterprise container management platform. It helps you to:

- Assess your progress on the Kubernetes journey
- Identify the best platform for your use case
- Learn from real-world use cases
- See how Rancher can help you become more agile

Download your copy today at more.suse.com/ECM4Dummies

Brought to you by SUSE

Start your Zero Trust container security journey

The traditional security perimeter is changing, and the new container attack surface is deep and wide. This changing landscape requires a proactive, defense-in-depth Zero Trust security strategy. As the widespread adoption of containers and Kubernetes continues to accelerate application deployment velocity, security automation also becomes ever more critical. *Zero Trust Container Security For Dummies* will guide you on your journey and introduce you to the SUSE Security Stack for Zero Trust to help you secure your container environment.

Inside…

- Understand the container attack surface
- Build security into your CI/CD pipeline
- Use a declarative process to protect workloads
- Enable Zero Trust protection at scale
- Ensure continuous compliance

SUSE

Fei Huang is vice president of security strategy at SUSE. **Glen Kosaka** is head of product security at SUSE. **Tom Callway** is senior director of product marketing at SUSE.

Go to Dummies.com™
for videos, step-by-step photos,
how-to articles, or to shop!

ISBN: 978-1-119-98210-4
Not For Resale

9 781119 982104

for dummies®
A Wiley Brand

WILEY END USER LICENSE AGREEMENT