Fairwinds

WHITE PAPER

Top 5 Kubernetes Security Mistakes You're Probably Making

(and How to Fix Them)

TABLE OF CONTENTS

INTRODUCTION: BASICS OF KUBERNETES SECURITY

Cloud native technology is revolutionizing how organizations develop and deliver applications. Kubernetes is an open-source container orchestration platform that provides critical capabilities for running applications in a 24×7 on-demand world, enabling auto-scaling, auto-recovery, and more. While the benefits of Kubernetes are outstanding, many organizations face challenges when seeking to adopt Kubernetes at scale, because they lack the tools, processes, and experience to successfully launch secure Kubernetes environments.

Many companies struggle with security when adopting Kubernetes. Because Kubernetes and containers present a new approach for deploying applications, operations and security teams question whether the applications and data will be secure when they adopt microservices, containers, and Kubernetes to develop and deploy applications. Many traditional security tools and processes no longer apply, while containers create new security blind spots and new attack surfaces, making visibility across containers and clusters an additional challenge. Similarly, developers must take responsibility for some of the new security challenges, a role they're unaccustomed to and reluctant to embrace.

To adapt to these changes, organizations must shift security left in the development process and give developers visibility into issues so that they can be addressed much earlier in the SDLC. By adopting tools and processes that enable a DevSecOps approach, teams can tightly integrate security throughout the development and deployment process, and securely deploy applications faster. To get started on improving your Kubernetes security, here are five mistakes you're probably making — and the information you need to fix them.

5 MISTAKES YOU'RE PROBABLY MAKING

#1 Granting access to the host node

#2 Assuming operations is aligned with security

#3 Running containers with known vulnerabilities

#4 Expecting security by default using native controls

#5 Moving to production before you're ready

#1 GRANTING ACCESS TO THE HOST NODE

Securing workloads in Kubernetes is an important part of your overall cluster security. As you're getting started, it's tempting to give admin-level access to applications that don't need it because you're trying to understand how things work together and rule out different issues.

Unfortunately, granting access to the host node can increase your risk of attack once you've deployed. When individual application developers over-provision a deployment with root access to get something done, it can lead to negative business impacts that all organizations seek to avoid.

Unfortunately, granting access to the host node can **increase your risk** of attack once you've deployed.

Be cautious with access permissions

To address these common Kubernetes security threats, you need to check configurations in the securityContext attribute for both Kubernetes pods and containers. There are open source tools and configuration validation software options available that can help you review configuration findings, log and analyze historical records of the findings across all clusters, and provide remediation guidance.

Simply gaining visibility into your Kubernetes security posture by auditing workloads and validating configurations for weaknesses, container vulnerabilities, and misconfigured deployments can help. Regardless of which tool you use, make sure to track and prioritize security issues, collaborate across teams, and apply best practices as applications move from development to production.

#2 ASSUMING OPERATIONS IS ALIGNED WITH SECURITY

Kubernetes offers many options for configuration, but if your operations team isn't aligned with security, chances are your Kubernetes deployment isn't secure.

By default, all applications within a single Kubernetes cluster have access to everything else within it, but this level of access isn't necessary — and it presents considerable security risk if your account is compromised. While adopting DevOps helps align development and operations teams, it's essential to bring in security to evaluate potential risks and consider security controls.

Align operations with security

Best practices are evolving rapidly in Kubernetes, and more recent versions of Kubernetes resolve some of the security problems in earlier versions, but that doesn't mean that the security team doesn't need to be involved from the beginning of your Kubernetes deployment.

Involving the security team early on helps ensure critical alignment between teams, creating a DevSecOps environment. To consistently identify and remediate Kubernetes security risk, you need consistent configuration across clusters, container runtime monitoring and scanning, and tools that are tightly integrated into the CI/CD process, including ticketing and assignment workflows for remediation.

#3 RUNNING CONTAINERS WITH KNOWN VULNERABILITIES

Container adoption continues to grow, so it's important for DevOps teams to ensure that the containers in use are secure. Container images offer faster development and deployment, and are the way applications are delivered in cloud-native environments.

An additional concern stems from Common Vulnerabilities and Exposures (CVEs). New CVEs are disclosed regularly, so scanning container images to check every image for vulnerabilities is important. Deploying containers with known vulnerabilities increases the risk that your application will be attacked by cyber criminals exploiting those vulnerabilities.

Deploying containers with known vulnerabilities **increases the risk** that your application will be attacked by cyber criminals exploiting vulnerabilities.

Continually scan and secure containers

Keep your underlying container images up to date for each application and use container scanning tools to check every image for vulnerabilities, then make sure you take new releases and patches and test them to ensure that the changes don't break anything. If possible, test any changes to your container images on internal and staging clusters, rolling out updates slowly and monitoring for problems.

Discovering and resolving vulnerabilities early in the SDLC both reduces your risk and the costs in terms of time and effort to remediate any vulnerabilities. Scanning image layers against CVEs in CI/CD and in the production environment, tracking known vulnerabilities in containers, and prioritizing findings by severity helps your developers and compliance teams decide where to focus remediation efforts to reduce risk of a security incident.

Kubernetes does not offer security by default, so **plan to build a strong security profile** from the beginning of your deployment.

#4 EXPECTING SECURITY BY DEFAULT USING NATIVE CONTROLS

Kubernetes offers native security features, but it's a complex platform that consists of multiple components. And because Kubernetes was designed for portability, enabling workload migration and overall flexibility, it can be customized to fit many different deployment scenarios.

For all its flexibility, Kubernetes is also predictable. Kubernetes clusters typically listen on a range of defined ports, which makes it easier for cybercriminals to identify and attack those clusters. By default, all applications within a single Kubernetes cluster have permission to access the entire network within the cluster. And if you don't spend time crafting a well-defined role-based access control (RBAC) policy, you might find your cluster deleting resources, exposing secrets, or mining bitcoin. Kubernetes does not offer security by default, so plan to build a strong security profile from the beginning of your deployment.

Implement security policies and controls

While handling configurations properly can be confusing, it's important to set access permissions for your clusters according to the principle of least privilege. If an application only needs to view logs, limit its access so it can only view logs. Setting network policy allows you to control communications between different parts of a cluster, which can help you to restrict an attacker to one workload rather than spreading through an entire cluster. Network policy can also manage cluster ingress and egress to manage internal and external traffic appropriately and limit the potential attack surface of your applications.

Similarly, RBAC can help you grant fine-grained permissions to access different resources on the cluster, also according to the principle of least privilege. While it's tempting to give admin-level access to simple applications because it's the easiest way to deploy a new application or provision a new user, thoughtful Kubernetes RBAC rules reduce the potential damage if an account is compromised. Together, network policy and RBAC can help you implement effective security policies and controls to improve your Kubernetes security posture.

#5 MOVING TO PRODUCTION BEFORE YOU'RE READY

Kubernetes is a complex platform composed of multiple different components, including an API server, a scheduler, controllers, agents, and a key value store.

While you're understandably excited to get your application up and running, moving through your deployment configuration too quickly leads to security gaps. Many organizations think they're ready to deploy without putting resource requests and limits in place, but these omissions can lead to a denial of service attack due to a misconfigured deployment.

Kubernetes is a complex platform composed of multiple different components, including an API server, a scheduler, controllers, agents, and a key value store. It also includes a container runtime for executing containers, a persistent storage solution, a logging tool, and operating systems to power each node. Each component comes with a set of vulnerabilities and each has its own security needs, which is why it's important to carefully consider the security of each component before deploying to a production environment.

Slow down — and deploy securely

Kubernetes includes some robust built-in security tooling and benefits from a broad ecosystem of open source and commercial solutions that can help you harden your clusters. Thinking through your security strategy — and testing and reviewing it regularly — won't slow down your deployment in the long run. Instead, it'll help you move faster, because a strong security profile prevents a wide range of time-consuming, damaging security problems.

To secure your Kubernetes deployment, it's important to automate security at scale and ensure visibility across multiple teams, clusters, and tenancy. Kubernetes-native solutions that integrate security validation checks into your CI/CD pipeline help to shift security left in the SDLC, minimizing risks and helping your teams to remediate vulnerabilities early on, keeping your applications secure and available.

CONCLUSION

Kubernetes and containers present new opportunities for developing and deploying applications at scale, maximizing efficiency and availability. Kubernetes is also extremely flexible and complex, presenting new challenges for development, operations, and security teams.

To deliver on the promise of Kubernetes, it's important to embrace Kubernetes' built-in security features and prioritize best practices. Visibility across clouds, clusters, and teams enables security and DevOps to identify and remediate Kubernetes security risk, while creating and automatically enforcing security policies ensures reduced risks for greater peace of mind.

Fairwinds

WHY FAIRWINDS

Fairwinds is your trusted partner for Kubernetes security, policy and governance. With Fairwinds, customers ship cloud-native applications faster, more cost effectively, and with less risk. We provide a unified view between dev, sec, and ops, removing friction between those teams with software that simplifies complexity. Fairwinds Insights is built on Kubernetes expertise and integrates our leading open source tools to help you save time, reduce risk, and deploy with confidence.

WWW.FAIRWINDS.COM