FUTUREWEI Technologies

Whitepaper

Playbook For Adopting Container Security Solutions

Trademarks and Permissions

FUTUREWEI Technologies and other Futurewei® trademarks are trademarks of Futurewei® Technologies. Huawei trademarks are trademarks of Huawei Technologies Co., Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services, and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services, and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees, or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

FUTUREWEI® TECHNOLOGIES, INC.

Boston Research Center

Address: 111 Speen Street, Suite 114
Framingham, MA 01701
United States of America

Website: http://www.futurewei.com/

Table Of Contents

Executive Summary

- Containers, while promoting innovation, also introduce new security risks. To ensure container security, a comprehensive approach is necessary, combining people, processes, and technology. Investments in training and solutions provide long-term benefits, and proper diligence throughout the container lifecycle is essential.
- Primary goal is to address security concerns related to container technologies and offer practical recommendations for managing these concerns at all stages of container usage, from planning to maintenance.
- **This playbook establishes criteria for evaluating container security solutions, providing a comprehensive perspective on the subject**. Adhering to these criteria would enable IT leaders and architects to effectively address the complexities of container security, ultimately fostering a safer and more efficient containerized environment.
- Leading container security solutions, including Red Hat Advanced Cluster Security, Palo Alto Prisma Cloud, Aqua Security, NeuVector, Anchore, Sysdig Secure, Qualys, and Snyk, are evaluated based on comprehensive criteria.
- Cultivating a culture of security and vigilance within container workflows is vital.

Introduction

The integration of container technologies has profoundly disrupted the established culture and software development methodologies within the organization. With the advent of containers, traditional approaches to development, patching, and system upgrades may no longer be directly applicable, necessitating a willingness among employees to embrace a new model. To ensure a successful transition, it is crucial for the staff to be open to adapting their practices and methodologies.

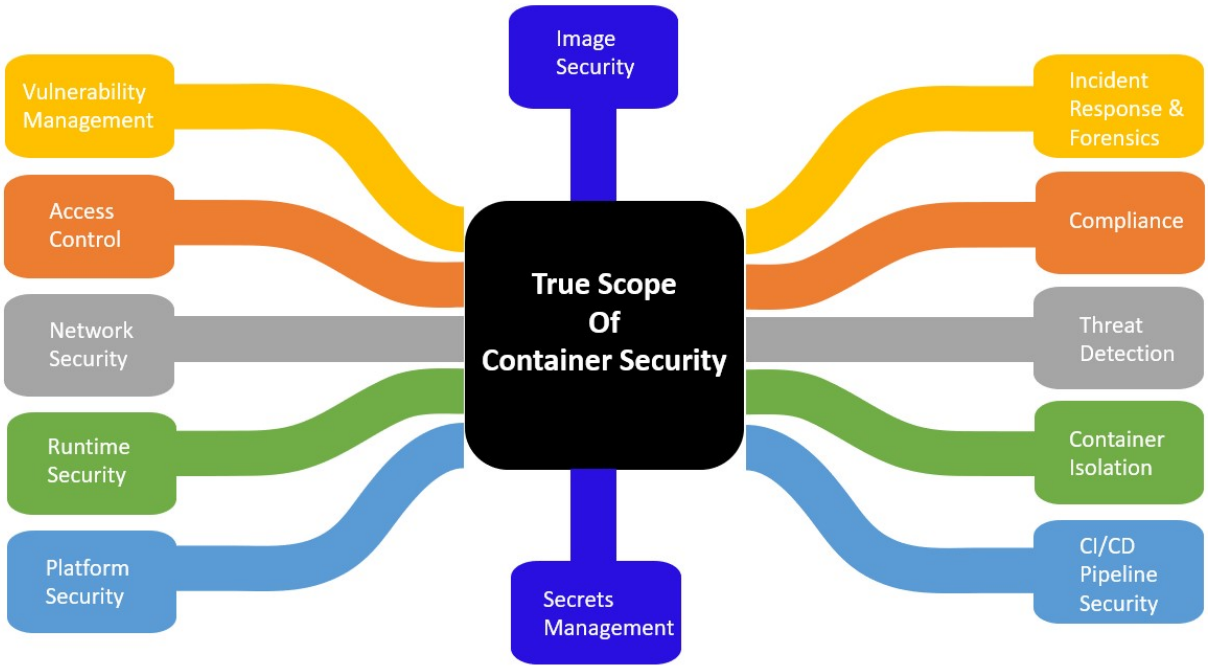
In this guide, we will cover recommended practices for securely building and operating applications within containers, offering essential insights for employees to follow. It is of utmost importance for the organization to reevaluate existing procedures to fully capitalize on the benefits that containers can bring.

The primary objective of this document is to shed light on the security concerns associated with container technologies and provide practical recommendations for effectively addressing these concerns throughout the planning, implementation, and maintenance stages of container usage. By heeding these guidelines, the organization can navigate the challenges and intricacies of container security, fostering a safer and more efficient containerized environment.

Overall, embracing containerization and its accompanying best practices will enable the organization to propel its software development processes to new heights while safeguarding against potential security vulnerabilities.

Scope

Container security has a vast scope, covering a diverse set of practices, tools, and strategies that work together to safeguard every aspect of the containerized application lifecycle. From the initial stages of development to the final deployment, container security ensures comprehensive protection. This includes safeguarding the container images themselves, as well as the runtime environments in which they operate. It also extends to the container orchestration platforms that manage and coordinate containerized applications, ensuring their security and integrity. Moreover, container security encompasses the protection of the underlying infrastructure, which forms the foundation for containerization. By delving deeper into the specific aspects of container security, one can gain a thorough understanding of the measures and techniques employed to maintain the safety and reliability of containerized applications throughout their lifecycle.



1. **Image Security:** Ensuring the security of container images is essential. This involves using only trusted base images, regularly updating images, and scanning for vulnerabilities in the images and their dependencies. Implementing secure image repositories and using digital signatures can also enhance image security.
2. **Vulnerability Management:** Identifying and mitigating vulnerabilities in the software components used within containers is a critical aspect of container security. Regularly scanning and patching container images, as well as monitoring for new vulnerabilities, are essential practices.
3. **Access Control:** Proper access controls must be enforced to limit container access to authorized users and processes. Implementing role-based access control (RBAC) and employing least privilege principles help reduce the attack surface.
4. **Network Security:** Containers communicate over networks, and securing these interactions is crucial. Employing network policies, firewalls, and service mesh helps prevent unauthorized access and potential lateral movement within the infrastructure. Firewall adds more value when placed at the edge of the cluster. If performance is the most important factor, then the firewall can be placed inside the containers themselves. With popular service mesh implementations like Istio and Linkerd, one can use tools and configuration resources to inject sidecar proxies alongside application containers and define policies within the Kubernetes environment.
5. **Runtime Security:** Monitoring container behavior during runtime is vital to detect and respond to security threats in real-time. Runtime

security tools can detect unusual activities, unauthorized access attempts, and potential indicators of compromise.

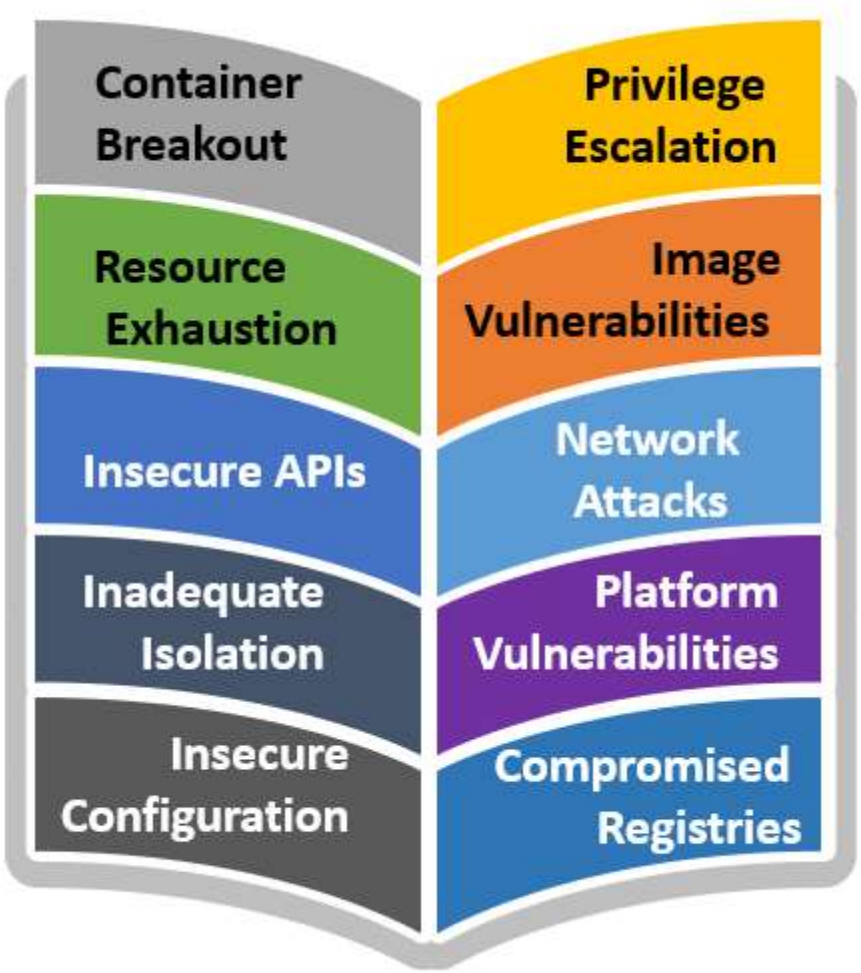
6. **Secrets Management:** Managing sensitive data such as passwords, API keys, and other credentials securely is a fundamental part of container security. Using encryption and secure storage solutions, as well as limiting access to secrets, helps protect critical information.
7. **Orchestration Platform Security:** Container orchestration platforms like Kubernetes play a central role in managing containers. Securing the orchestration platform itself is essential to prevent attacks on the control plane and unauthorized access to sensitive information.
8. **Compliance and Auditing:** Container security should align with relevant security standards and regulatory requirements. Regular auditing and compliance checks ensure that security practices are maintained and followed.
9. **CI/CD Pipeline Security:** Integrating security into the CI/CD pipeline helps identify and address vulnerabilities early in the development process. Secure development practices, code analysis, and continuous security testing are crucial components.
10. **Incident Response and Forensics:** Having a well-defined incident response plan specific to container-related incidents is vital. This includes procedures for containing threats, investigating breaches, and restoring services.

11. **Continuous Monitoring and Threat Detection:** Implementing continuous monitoring and threat detection mechanisms allows for proactive identification of potential security risks and anomalies.

12. **Container Isolation:** Ensuring proper isolation between containers and host systems prevents container escape and protects the underlying infrastructure.

Risks

Containers provide immense benefits like portable workloads, resource efficiency, and scalability. However, as with any technology, they also introduce new security risks that must be managed. While not exhaustive, some key risks include:



1. **Container breakout:** This involves an attacker breaking out of the container and accessing the underlying host infrastructure. It can happen if the container runtime is vulnerable or misconfigured in a way that allows escaping namespaces, capabilities, cgroups, etc.

2. **Privilege escalation:** Containers often run as root or have more privileges than required. This allows an attacker who compromises a container to gain elevated privileges on the host. Privileges should be limited via pod security contexts, read-only filesystems, etc.

3. **Resource exhaustion:** Containers can be used to launch denial of service attacks by consuming excessive compute, memory, storage, or network resources on the host. Resource limits should be set on containers.

4. **Image vulnerabilities:** Images may contain software with known vulnerabilities, insecure default configurations, or malicious injections. Images should be scanned, signed, and obtained from trusted sources only.

5. **Insecure APIs:** Application and Orchestration platform APIs should be secured with auth, encryption, auditing. API endpoints exposed on containers also need proper protection.

6. **Network attacks:** Pods can access each other over the network by default in Kubernetes. Network policies should restrict communication to prevent reconnaissance, lateral movement, etc.

7. **Inadequate isolation:** Shared compute, storage, network resources can allow inter-container attacks and data leaks. Use security contexts, namespaces, filesystem policies to isolate as needed.

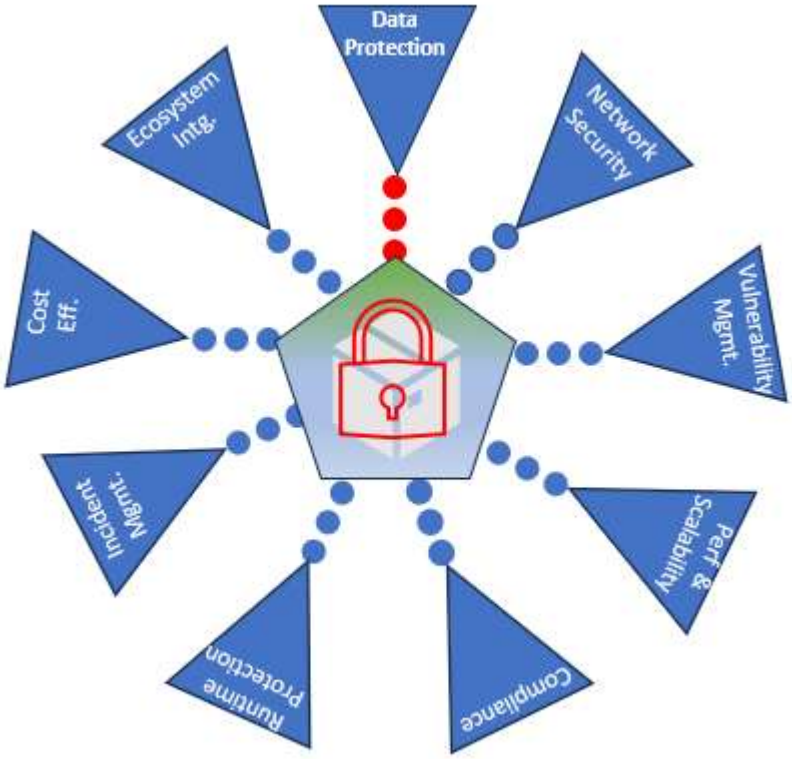
8. **Platform vulnerabilities:** Flaws in the container runtime, OS distro, Kubernetes platform can compromise security controls and container isolation. Timely patching and hardening is important.

9. **Insecure configurations:** Misconfigurations of RBAC policies, admission controls, auditing, secrets management, etc. can undermine Kubernetes security. Follow principle of least privilege.

10. **Compromised registries:** Container registries can be hacked to distribute backdoored or malicious images intended for Kubernetes deployments. Sign, scan and verify images.

Product Evaluation Criteria

When evaluating container security software, there are several key metrics to consider. Based on the research I have identified the following fundamental parameters that serve as the basis for evaluating container security products. These metrics play a crucial role in determining the effectiveness and reliability of container security solutions. By considering these parameters, businesses can make informed decisions and select the most suitable container security software for their needs.



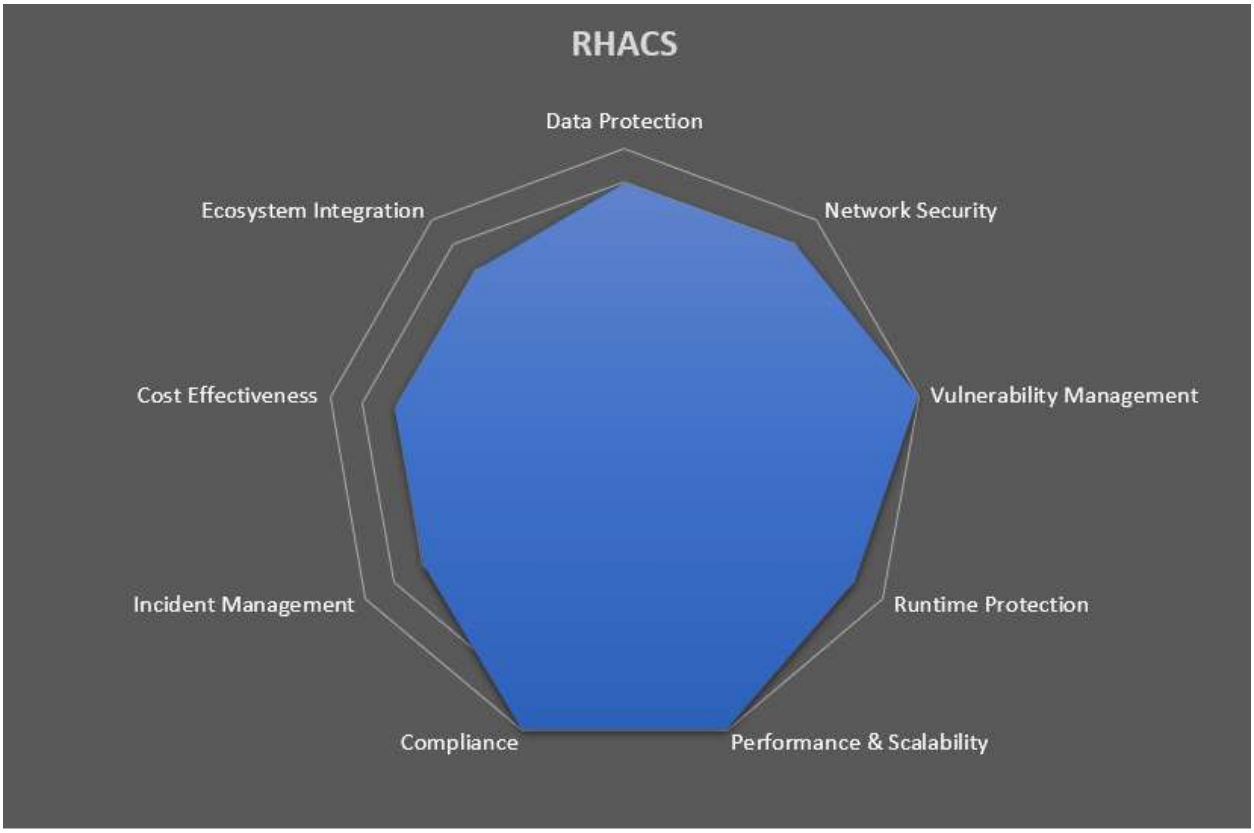
1. **Data Protection** – Does the solution protect the data that is stored in containers, both at rest and in transit? Does it include encrypting data, controlling access to data, and detecting and responding to data breaches.
2. **Network Security** – To what extent does the solution safeguard the container network? This entails the use of firewalls, intrusion detection systems, and encryption to thwart unauthorized access to containers and their data.
3. **Vulnerability management** – Does the solution possess the capability to scan container images and running containers for vulnerabilities in the operating system, libraries, dependencies, and application code? This is crucial for ensuring the security of container images.
4. **Performance & Scalability** - How effectively does the solution scale with the growth of your cluster? Are there limitations on the number of nodes it can manage? Does scaling impact performance?
5. **Compliance** – To what degree does the solution assist organizations in achieving compliance with pertinent regulations such as the General Data Protection Regulation (GDPR) and the Payment Card Industry Data Security Standard (PCI DSS)?
6. **Runtime protection** – Does the solution offer real-time defenses against threats like malware, unauthorized access, anomaly detection, and compliance monitoring for actively running containers?
7. **Incident monitoring & response** – Does the solution provide monitoring for container activities, alerting for potential threats, investigative capabilities for swift incident response, and support for forensic analysis?
8. **Cost-effectiveness** - How cost effective is the solution compared to similar alternatives? Is pricing per-node or flat annually? Does it impose unnecessary costs?
9. **Ecosystem integration** - How seamlessly does the solution integrate with the container ecosystem, including Kubernetes, CI/CD tools, the host operating system, and public cloud platforms?

Product Review

Red Hat Advanced Cluster Security for Kubernetes®

Key Takeaways

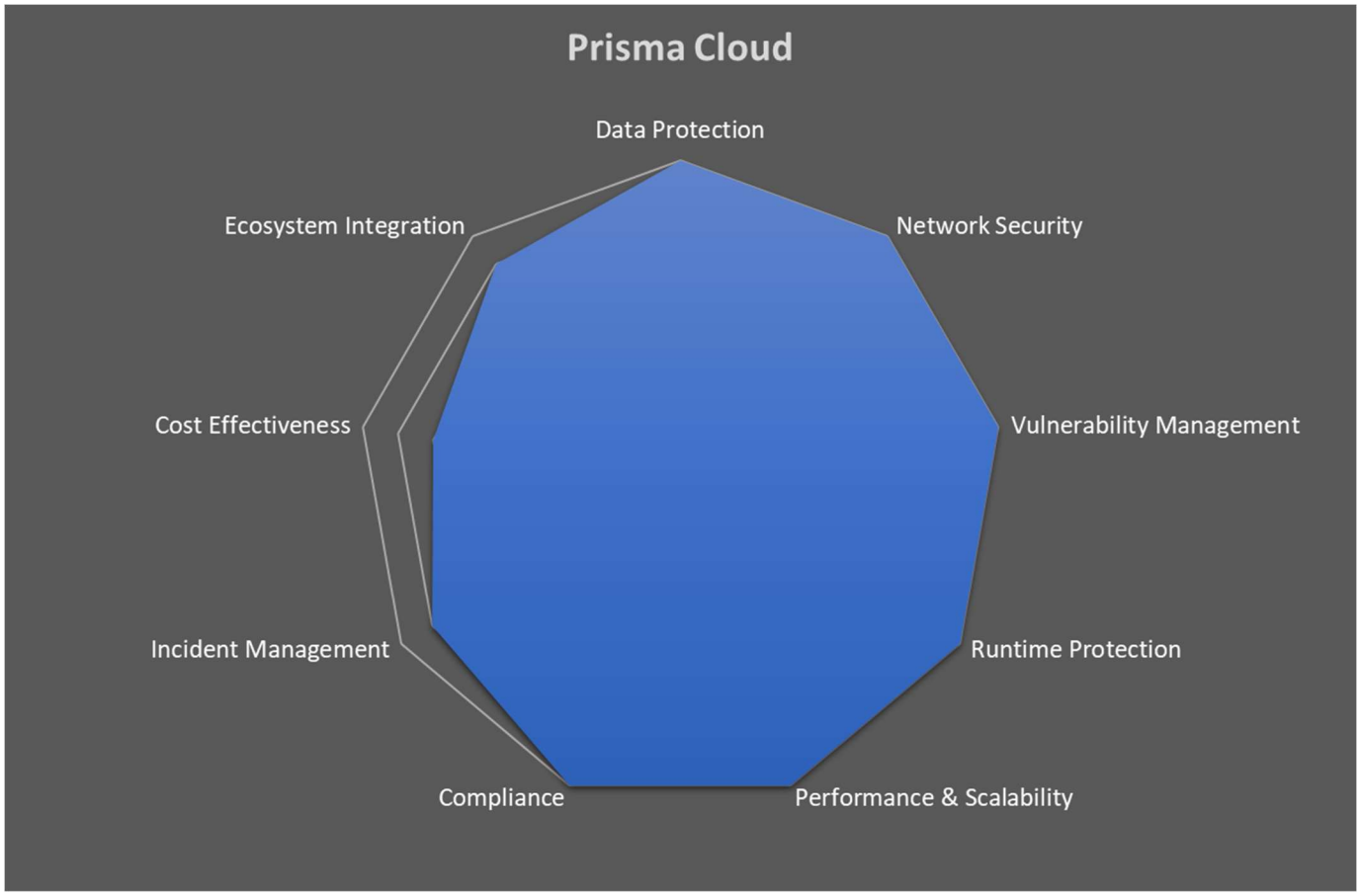
- Kubernetes-native architecture leverages Kube's declarative data and built-in controls for richer context, native enforcement, and continuous hardening
- Leverages the controls built into Kubernetes for policy enforcement, avoiding the operational risks of applying third-party in-line proxying or blocking tools.
- Applies intelligence from runtime behavior to adjust subsequent builds and deployments to continuously monitor and shrink the attack surface.
- Reduces the time and effort needed to implement security and streamlines security analysis, investigation, and remediation using the rich context Kubernetes provides.
- Provides scalability and resiliency native to Kubernetes, avoiding operational conflict and complexity that can result from out-of-band security controls.


RHACS
Data Protection
Network Security
Vulnerability Management
Runtime Protection
Performance & Scalability
Compliance
Incident Management
Cost Effectiveness
Ecosystem Integration

Palo Alto Prisma Cloud®

Key Takeaways

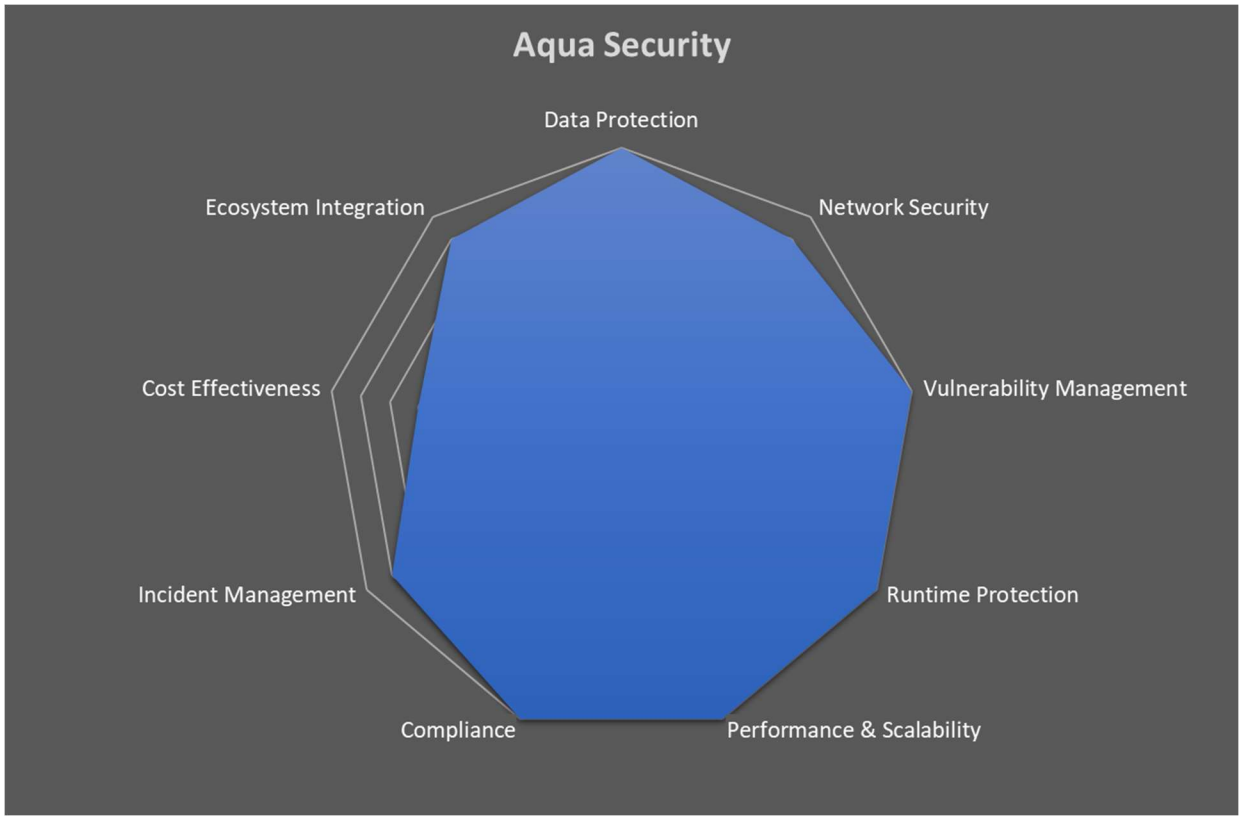
- Scans container images and enforces policies as part of CI/CD workflows.
- Continuously monitors code in repositories and registries.
- Secures both managed and unmanaged runtime environments.
- Combines risk prioritization with runtime protection at scale.
- Support for public and private clouds
- Single console for managed and unmanaged environments
- Full life cycle security for repositories, images, and containers
- Aggregates and prioritizes vulnerabilities continuously in CI/CD pipelines and containers running on hosts or on containers as a service.


Prisma Cloud
Data Protection
Network Security
Vulnerability Management
Runtime Protection
Performance & Scalability
Compliance
Incident Management
Cost Effectiveness
Ecosystem Integration

Aqua Security®

Key Takeaways

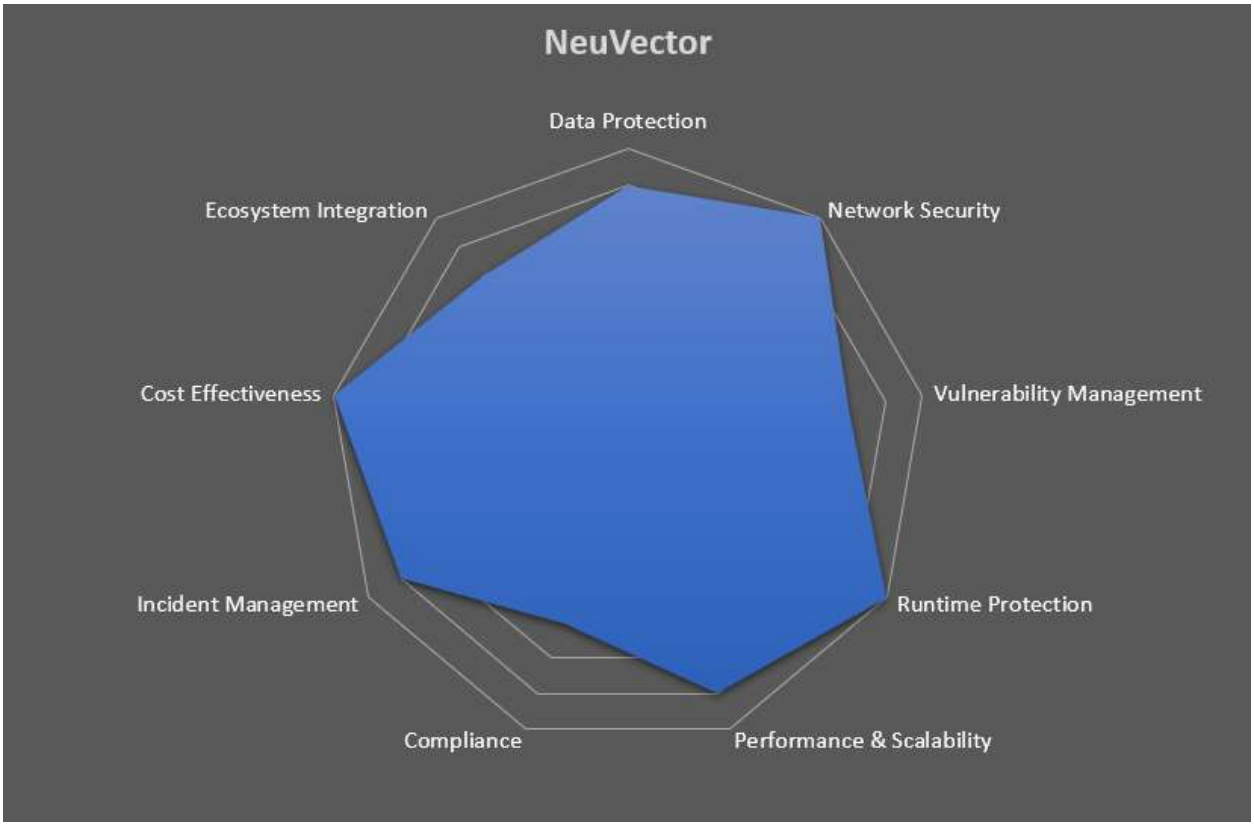
- Kubernetes Security Posture Management (KSPM) and Kubernetes runtime protection provide policy-driven life cycle protection and compliance for K8s applications.
- Dynamic map of running K8s clusters that highlights and rates Kubernetes security risks.
- Real-time visibility into namespaces, deployments, nodes (hosts), containers and the images they came from, as well as network connections between and within namespaces.
- Powered by Open Policy Agent (OPA), new Kubernetes Assurance Policies apply dozens of rules or add custom rules using Rego expressions.
- Works in conjunction with Aqua's Image Assurance Policies to prevent the deployment of unsafe and non-compliant workloads.
- Discover malware hidden in open-source packages and third-party images, preventing attacks on container-based applications.
- Analyzes images before they arrive in a secure isolated sandboxed environment, examining and tracing behavioral anomalies
- Static and dynamic scanning to create flexible image assurance policies that determine which images would be allowed to progress through the development pipeline and run in clusters or hosts.


Aqua Security
Data Protection
Network Security
Vulnerability Management
Runtime Protection
Performance & Scalability
Compliance
Incident Management
Cost Effectiveness
Ecosystem Integration

Neuvector®

Key Takeaways

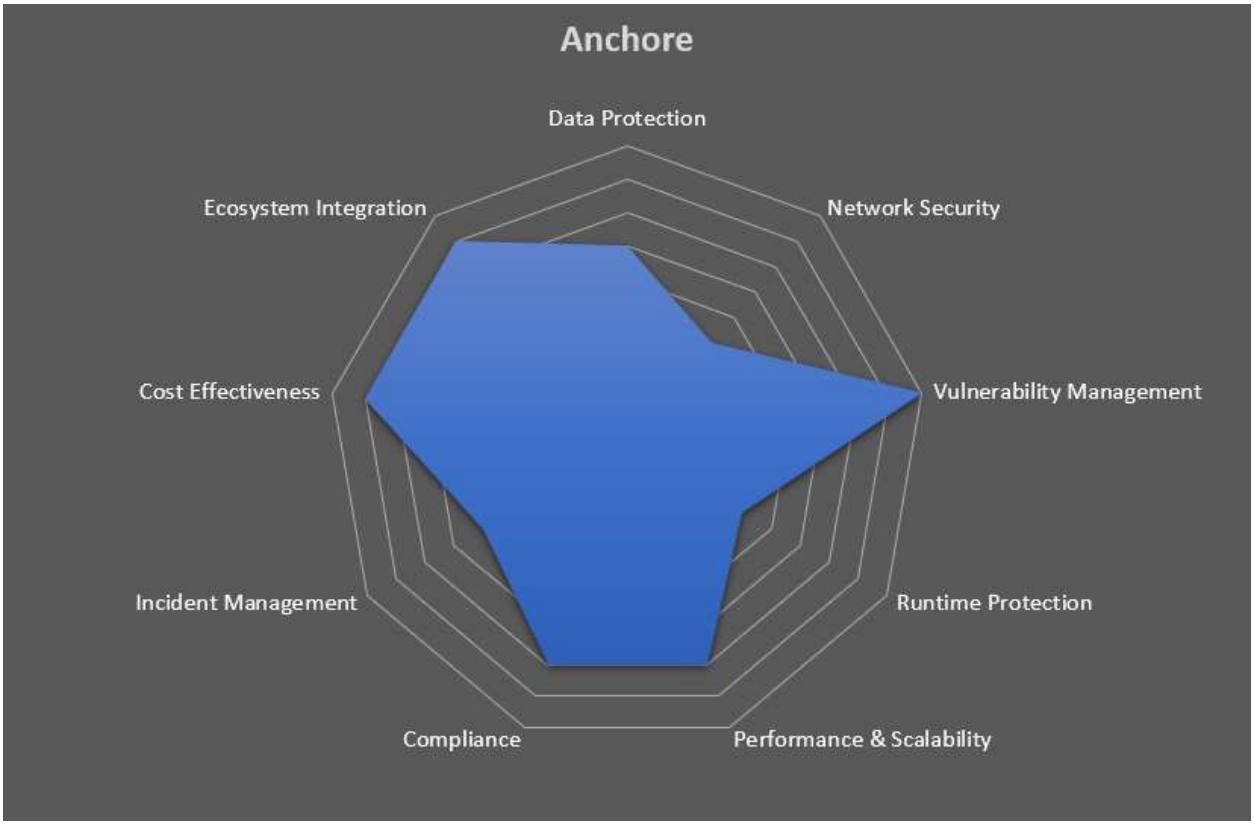
- NeuVector can be deployed through the Rancher catalog.
- Pushes security capabilities across the entire cloud-native footprint.
- Ensures container security using network inspection.
- NeuVector will be available as an add-on to SUSE Rancher
- Supports the full range of Kubernetes management products in addition to Rancher and including OpenShift, Mirantis, and Tanzu.
- Scans for vulnerabilities during the entire CI/CD pipeline
- Uses the Jenkins plug-in to scan during build, monitor images in registries, and run automated tests for security compliance.
- Prevents deployment of vulnerable images with admission control, and monitor production containers.
- Deploys as a container onto virtual machines or bare metal OS environments.
- Scanning and admission control during build, test, and deployment
- Layer 7 container firewall.


NeuVector
Data Protection
Network Security
Vulnerability Management
Runtime Protection
Performance & Scalability
Compliance
Incident Management
Cost Effectiveness
Ecosystem Integration

Anchore®

Key Takeaways

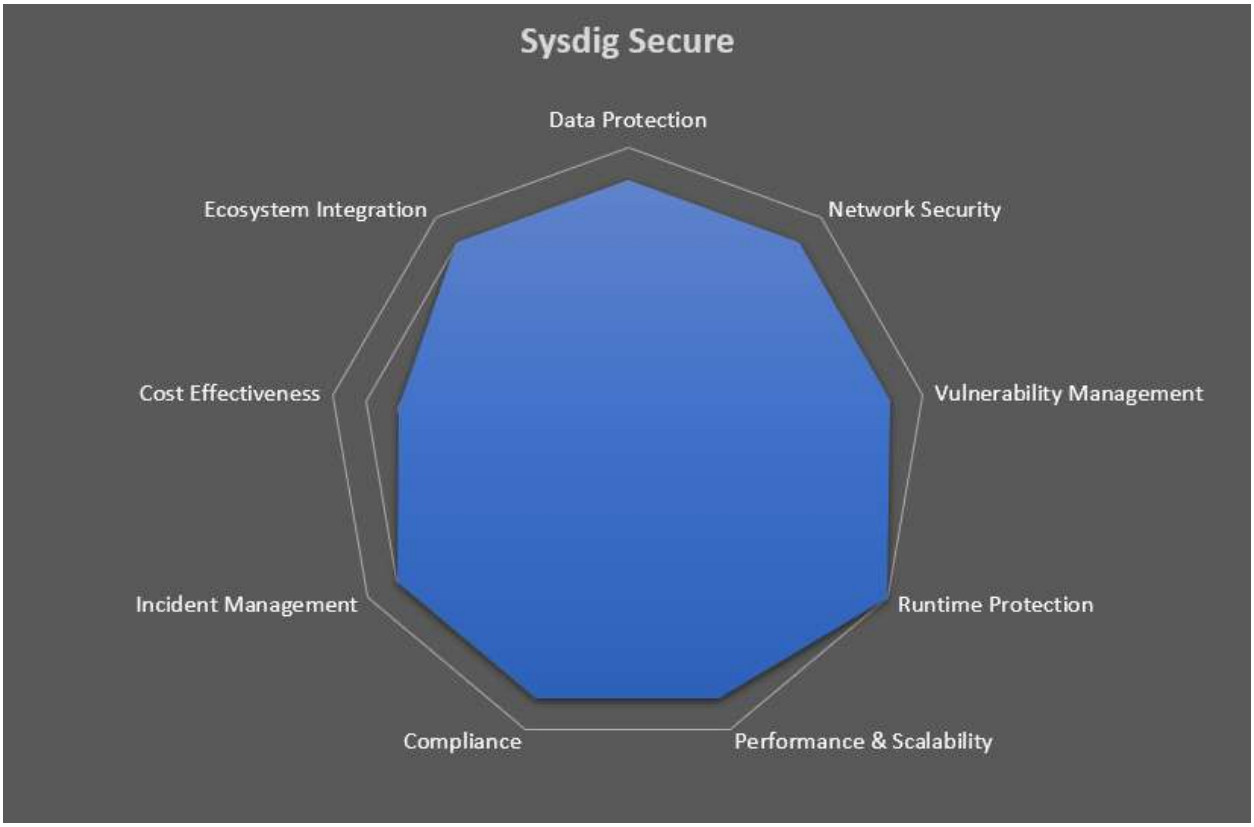
- Software supply chain management built with the software bill of materials (SBOM) to identify upstream dependencies in source code repositories and monitor for SBOM drift that can indicate malware or compromised software.
- Automatically generate SBOMs for all software produced and verify SBOMs for software consumed (both open source and proprietary)
- Use SBOM data to continuously assess security and compliance risks before and after deployment.
- Continuously monitor software applications for new or zero-day vulnerabilities that arise.
- Extends scanning for dependencies to include source code repositories in addition to support for container scanning through CI/CD, registries, or Kubernetes.
- Container registries include Harbor, Quay, JFrog, and DockerHub as well as offerings from AWS, Azure, and Google
- Reports and evaluations can be accessed using the command-line interface (CLI) or Anchore Enterprise UI, and webhooks can trigger action in other systems.
- Open-source tools for image inspection and vulnerability scanning perform analysis of container workloads.
- Ensures no secrets are present in images such as passwords and API keys.
- Identifies non-OS third-party libraries, including Node.js NPM, Ruby GEM, Python PIP, DotNet, and Java archives.


Anchore
Data Protection
Network Security
Vulnerability Management
Runtime Protection
Performance & Scalability
Compliance
Incident Management
Cost Effectiveness
Ecosystem Integration

Sysdig Secure®

Key Takeaways

- A new Drift Control feature helps teams to detect, prevent, and speed up incident response for containers that were modified in production, also known as container drift.
- Malware and crypto mining detection with threat intelligence feeds from Proofpoint Emerging Threats (ET) Intelligence and the Sysdig Threat Research Team
- Digs directly into compromised or suspicious containers with on-demand secured shell access and investigates the blocked executable and communications.
- Automates scanning locally in continuous integration and continuous deployment (CI/CD) pipelines and registries without images leaving the environment.
- Visualizes network communication between pods, services, and applications inside Kubernetes.
- Conducts incident response using granular data with Kubernetes and cloud context and forwards events to SIEM tools like Splunk, QRadar, AWS Security Hub
- Continuously validates cloud security posture and meets compliance standards (e.g., NIST 800-53, SOC2, and PCI) and internal mandates.


Sysdig Secure
Data Protection
Network Security
Vulnerability Management
Runtime Protection
Performance & Scalability
Compliance
Incident Management
Cost Effectiveness
Ecosystem Integration

Qualys®

Key Takeaways

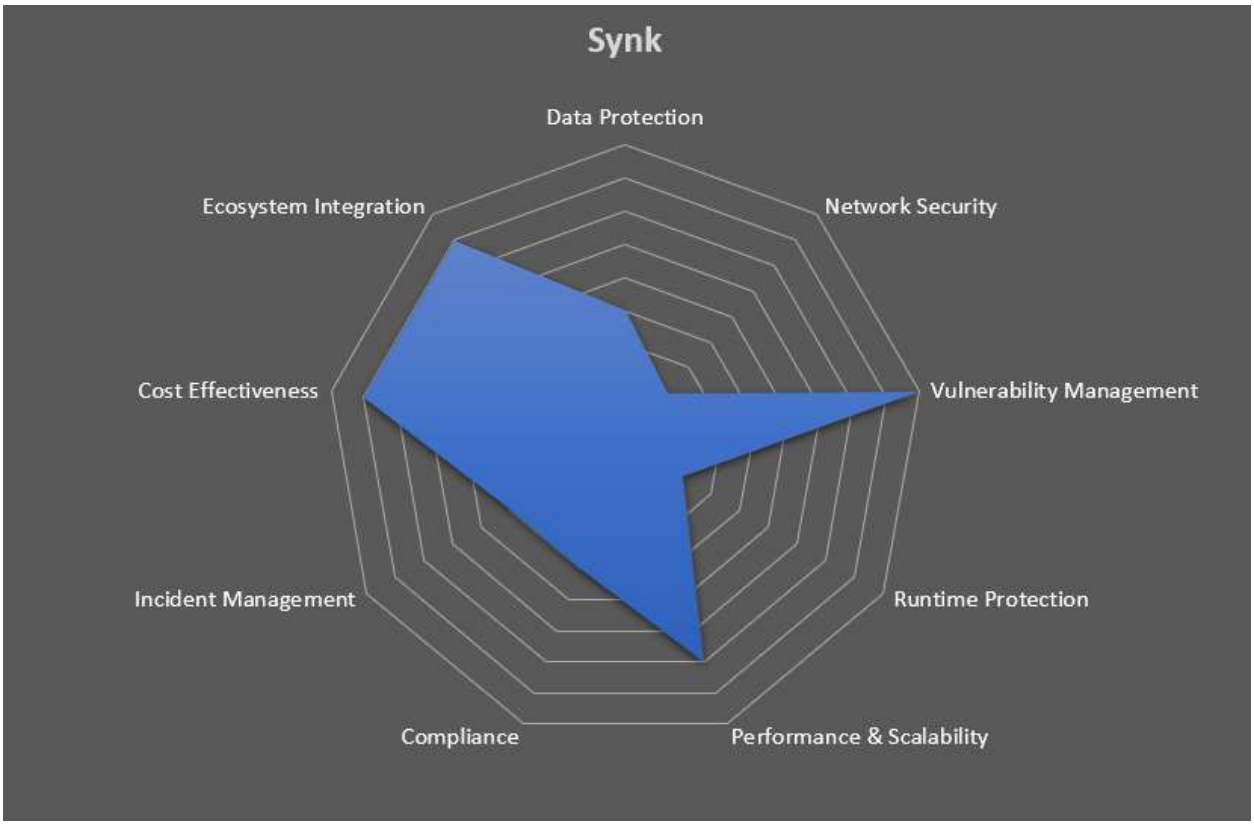
- Secures containers whether on-premises and in the cloud.
- Enforce policies to block the use of images that have specific vulnerabilities or that have vulnerabilities above a certain severity threshold.
- Vulnerability detection and remediation in the DevOps pipeline by deploying plugins like Jenkins or Bamboo or via REST APIs
- Search for images that have high-severity vulnerabilities, unapproved packages, and older or test release tags.
- Container Runtime Security (CRS) offers visibility into running containers as well as the ability to enforce policies that govern behavior.
- Centralized discovery and tracking for containers and images.
- View metadata for containers and images including labels, tags, installed software, and layers.
- Coverage of Linux OS distributions to container-centric OSs, applications, and programming languages


Qualys
Data Protection
Network Security
Vulnerability Management
Runtime Protection
Performance & Scalability
Compliance
Incident Management
Cost Effectiveness
Ecosystem Integration

Snyk®

Key Takeaways

- Scales security capabilities by enabling developers to eliminate vulnerabilities by upgrading to a more secure base image or rebuilding when the base image is outdated.
- Focuses attention on the highest priority issues instead of taking one by one.
- Uses risk signals like exploit maturity and insecure workload configuration to help teams cut through the typical noise of container vulnerability reports.
- Detects and monitors open-source dependencies as part of the container scan.
- Detects vulnerable dependencies during coding to avoid future fixing efforts and save development time.
- Scans pull requests before merging.
- Tests projects directly from the repository and monitors them daily for new vulnerabilities.
- Prevents new vulnerabilities from passing through the build process by adding an automated Snyk test to the CI/CD


Synk
Data Protection
Network Security
Vulnerability Management
Runtime Protection
Performance & Scalability
Compliance
Incident Management
Cost Effectiveness
Ecosystem Integration

Conclusion

This paper has outlined some of the leading container security solutions available today. With container adoption accelerating, organizations must make informed choices to protect their environments. However, solutions alone cannot guarantee security. Developers should consistently apply security best practices during the application design and build processes to mitigate risks proactively.

As container ecosystems grow in complexity, vulnerabilities can emerge across the technology stack. Containers connect multiple components - host OS, orchestrator, registries, CI/CD pipelines. Each represents a potential attack surface. Flaws anywhere along this chain can expose the entire system. Thus, holistic security measures are essential.

Organizations should strategically invest in container-native security solutions. Capabilities like image scanning, runtime protection, and micro segmentation help reduce risks. Still, tools and container security solutions are only part of the equation. To maximize container security, organizations need to foster a culture of security, ensure developers and ops teams have appropriate skills and knowledge, and implement strong policies and controls across the container lifecycle. With proper vigilance, containers can be leveraged securely at scale to drive digital transformation.

References

https://peerspot.com/products/comparisons/qualys-vmdr_vs_red-hat-advanced-cluster-security-for-kubernetes

https://docs.openshift.com/acs/3.66/operating/evaluate-security-risks.html

https://peerspot.com/products/red-hat-advanced-cluster-security-for-kubernetes-reviews

https://redhat.com/en/technologies/cloud-computing/openshift/advanced-cluster-security-kubernetes

https://redhat.com/en/resources/advanced-cluster-security-for-kubernetes-datasheet

https://docs.openshift.com/acs/3.68/operating/manage-vulnerabilities.html

https://access.redhat.com/documentation/en-us/red_hat_advanced_cluster_security_for_kubernetes/3.70/html/operating/examine-images-for-vulnerabilities

https://g2.com/products/red-hat-advanced-cluster-security-for-kubernetes/reviews

https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin-compute/deployment_patterns/performance_planning

https://peerspot.com/products/prisma-cloud-by-palo-alto-networks-reviews

https://paloaltonetworks.com/blog/prisma-cloud/leader-gigaom-radar-cspm-2022

https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin-compute/compliance

https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin/prisma-cloud-compliance

https://docs.paloaltonetworks.com/content/techdocs/en_US/prisma/prisma-cloud/prisma-cloud-admin/prisma-cloud-compliance.html

https://gartner.com/reviews/market/cloud-workload-protection-platforms/vendor/palo-alto-networks/product/prisma-cloud

https://g2.com/products/palo-alto-networks-prisma-cloud/reviews

https://trustradius.com/products/palo-alto-networks-prisma-cloud/reviews?qs=pros-and-cons

https://aquasec.com/news/aqua-security-unveils-out-of-the-box-runtime-protection

https://support.aquasec.com/support/solutions/articles/16000120194-vulnerability-severity-and-score

https://aquasec.com/trust/product-privacy-policy

https://securityscorecard.com/security-rating/aquasec.com

https://peerspot.com/products/aqua-cloud-security-platform-reviews

https://g2.com/products/aqua-security/reviews

https://aquasec.com/news/runtime-protection-zero-day-vulnerabilities-containers

https://aquasec.com/about-us/news

https://gartner.com/reviews/market/cloud-workload-protection-platforms/vendor/aqua-security/product/aqua-cloud-native-security-platform

https://blog.sonatype.com/neuvector

https://blog.neuvector.com/article/container-risk-score

https://neuvector.com/why-neuvector/use-cases/compliance

https://neuvector.com/solutions/container-compliance-auditing-solutions

https://neuvector.com

https://suse.com/products/neuvector

https://suse.com/neuvector

https://suse.com/products/neuvector

https://g2.com/products/neuvector/reviews

https://qualysguard.qg2.apps.qualys.com/qwebhelp/fo_portal/knowledgebase/severity_levels.htm

https://securityscorecard.com/security-rating/qualys.com

https://qualys.com/scanning-accuracy

https://qualys.com/apps/policy-compliance

https://gartner.com/reviews/market/vulnerability-assessment/vendor/qualys/product/qualys-vulnerability-management-detection-and-response-vmdr

https://qualys.com/company/privacy

https://trustradius.com/products/qualys-cloud-platform/reviews

https://sysdig.com/press-releases/sysdig-named-a-top-10-security-provider-by-g2-reviewers

https://sysdig.com/ecosystem

https://docs.sysdig.com/en/docs/sysdig-secure/network

https://sysdig.com/solutions/vulnerability-management

https://gartner.com/reviews/market/cloud-native-application-protection-platforms/vendor/sysdig/product/sysdig-secure

https://docs.sysdig.com/en/docs/sysdig-secure/vulnerabilities

https://gartner.com/reviews/market/cloud-workload-protection-platforms/vendor/sysdig/product/sysdig-secure

https://g2.com/products/sysdig-sysdig-secure/reviews

Introducing open source security runtime monitoring | Snyk

https://snyk.io/blog/introducing-open-source-security-runtime-monitoring

https://snyk.io/blog/scoring-security-vulnerabilities-101-introducing-cvss-for-cve

https://docs.snyk.io/more-info/how-snyk-handles-your-data

https://gartner.com/reviews/market/application-security-testing/vendor/snyk/product/snyk-code