KUBERNETES SECURITY CHECKLIST

Prepared by HANIM EKEN

https://ie.linkedin.com/in/hanimeken

Kubernetes Security Checklist

Introduction

Kubernetes is a powerful container orchestration platform, but its complexity can introduce security risks if not properly managed. This checklist provides best practices and actionable steps to secure your Kubernetes environment.

1. Cluster Setup

Use the Latest Version

- **Action:** Always use the latest stable version of Kubernetes to benefit from the latest security patches and features.
- **Verification:** Regularly check for updates and upgrade your clusters.

Secure the Kubernetes API Server

- **Action:** Restrict access to the API server using network policies and firewalls.
- **Verification:** Ensure only trusted IP addresses and authenticated users can access the API server.

Enable Role-Based Access Control (RBAC)

- **Action:** Use RBAC to control access to the Kubernetes API and define granular permissions.
- **Verification:** Audit RBAC policies to ensure least privilege principles are applied.

2. Node Security

Use Minimal Base Images

- **Action:** Use minimal base images for your containers to reduce the attack surface.
- **Verification:** Regularly scan container images for vulnerabilities.

Apply Security Patches

- **Action:** Keep the host OS and all software components up to date with the latest security patches.
- **Verification:** Implement automated patch management and regularly review patch status.

Restrict SSH Access

- **Action:** Limit SSH access to Kubernetes nodes and use key-based authentication.
- **Verification:** Regularly audit SSH access logs.

https://ie.linkedin.com/in/hanimeken

3. Network Security

Use Network Policies

- **Action:** Implement Kubernetes Network Policies to control traffic between pods.
- **Verification:** Regularly review and update network policies to reflect current requirements.

Encrypt Traffic

- **Action:** Encrypt all traffic between Kubernetes components using TLS.
- **Verification:** Check that TLS is configured and enabled for all communication channels.

Segregate Network Segments

- **Action:** Use network segmentation to isolate different parts of your application and environment.
- **Verification:** Ensure network segments are properly isolated and monitored.

4. Pod Security

Use Pod Security Policies

- **Action:** Define and enforce Pod Security Policies (PSPs) to control the security configurations of pods.
- **Verification:** Regularly audit PSPs and ensure they align with security best practices.

Limit Resource Usage

- **Action:** Define resource limits and requests for CPU and memory to prevent resource exhaustion attacks.
- **Verification:** Monitor resource usage and adjust limits as necessary.

Run Containers as Non-Root

- **Action:** Configure containers to run as non-root users to minimize potential damage from compromised containers.
- **Verification:** Audit container configurations to ensure they are not running as root.

5. Storage Security

Encrypt Data at Rest

- **Action:** Use encryption to protect sensitive data stored in persistent volumes.
- **Verification:** Ensure encryption is enabled and keys are managed securely.

Use Access Controls

- **Action:** Implement access controls to restrict access to sensitive data.
- **Verification:** Regularly review and update access controls for storage resources.

6. Logging and Monitoring

Enable Logging

- **Action:** Enable logging for all Kubernetes components and applications.
- **Verification:** Ensure logs are collected, stored securely, and monitored for suspicious activity.

Monitor for Anomalies

- **Action:** Implement monitoring and alerting to detect anomalies and potential security incidents.
- **Verification:** Regularly review and update monitoring rules and alerts.

Audit Logs

- **Action:** Enable Kubernetes audit logging to track API requests and changes.
- **Verification:** Regularly review audit logs for suspicious activities.

7. Backup and Disaster Recovery

Regular Backups

- **Action:** Perform regular backups of critical Kubernetes components and application data.
- **Verification:** Test backups regularly to ensure they can be restored successfully.

Disaster Recovery Plan

- **Action:** Develop and test a disaster recovery plan for your Kubernetes environment.
- **Verification:** Conduct regular disaster recovery drills to ensure readiness.

8. Compliance and Governance

Ensure Compliance

- **Action:** Ensure your Kubernetes environment complies with relevant regulatory requirements and industry standards.
- **Verification:** Perform regular compliance audits and address any gaps identified.

Implement Governance Policies

- **Action:** Define and enforce governance policies to manage Kubernetes configurations and operations.
- **Verification:** Use policy management tools to automate and enforce governance.

Conclusion

Securing a Kubernetes environment requires a comprehensive approach that addresses cluster setup, node security, network security, pod security, storage security, logging and monitoring, backup and disaster recovery, and compliance and governance. By following this checklist, organizations can significantly enhance the security of their Kubernetes deployments and reduce the risk of security incidents. Regular audits, continuous monitoring, and staying informed about the latest security best practices are essential to maintaining a secure Kubernetes environment.

HANIM EKEN

https://ie.linkedin.com/in/hanimeken