10 Steps to Secure Kubernetes

www.thecyphere.com

info@thecyphere.com

What are Kubernetes Environments?

Kubernetes Environments are orchestrated container platforms that manage, deploy and scale containerised applications across clusters of nodes.

10 Steps to Secure Kubernetes

01. Enable Kubernetes RBAC

06. Use Process Whitelisting

02. Use Third-Party Authentication for API Server

07. Turn on Audit Logging

03. Proctect etcd with TLS, Firewall & Encryption

08. Keep Kubernetes Version Updated

04. Isolate Kubernetes Nodes

09. Implement Pod Security Policies

05. Monitor Network Traffic to Limit Communication

10. Spread Awareness & Educate Your Team

01:
Enable Kubernetes RBAC

- Disable Attribute-Based-Access-Control (ABAC) and enable Role-Based-Access-Control (RBAC) to control API access and permissions.

- Use RBAC to grant namespace-specific permissions for safer access, and avoid cluster-wide privileges.

02:
Use Third-Party Authentication for API Server

- Integrate Kubernetes with third-party authentication for extra security and multi-factor authentication.
- Avoid user management at the API server level, consider implementing OAuth 2.0 connectors.

03:
Protect *etcd* with TLS, Firewall & Encryption

- Secure **etcd** to prevent unauthorised access and potential cluster control by isolating it and implementing firewall protection.

- Encrypt **etcd** data at rest with **kube-apiserver's --encryption-provider-config**, specifying encryption provider and secret keys for enhanced security.

04:
Isolate Kubernetes Nodes

- Isolate Kubernetes nodes on a distinct network, shielding them from public and corporate networks for enhanced security.

- Ensure separation of control and data traffic to prevent exposure. An ingress controller with master node-specific port access via ACL is recommended.

05: Monitor Network Traffic to Limit Communication

- Monitor containerised application network against Kubernetes policies for interaction patterns and anomaly detection.

- Enhance security by identifying unused network policies, and comparing allowed/disallowed traffic.

06:
Use Process Whitelisting

- Use process whitelisting to detect unusual processes. Build a whitelist from typical behaviour to bolster security.

- Runtime process analysis can be difficult at times; explore automated tools that can help detect and identify irregularities within processes.

07:
Turn on Audits & Logging

- Enable audit logs, and monitor for Forbidden authentication failures as potential credential misuse.

- Use **--audit-policy-file** for **kube-apiserver** logs, and choose logging levels: None, Metadata, Request, RequestResponse.

08: Keep Kubernetes Version Updated

- Keep Kubernetes up-to-date by running the latest version.
- Prioritise timely and regular updates and security patch management.

09: Implement Pod Security Policies

- Define pod security policies to establish strict container runtime security, limiting privileged access and capabilities.

- Limit metadata exposure to pods, reducing the information available to potential attackers.

10:
Spread Awareness & Educate your Team

- Equip your team with knowledge about securing Kubernetes clusters.
- Cultivate a culture where each team member considers Kubernetes security as their responsibility.

Are you planning for your IT health check?

Contact Us

info@thecyphere.com

If you find it useful, follow for more updates and share ❤