

Kubernetes

Security and stuff

LINUS ÖSTBERG
linus@scilifelab.uu.se

SciLifeLab Data Centre



WHO AM I?



Linus Östberg

SciLifeLab Data Centre

(soon Conoa AB)



talavis



linus@oestberg.dev



WHERE TO START?

- ▶ Risk assessment
- ▶ Threat modelling
- ▶ Laws
- ▶ Requirements from organisations

CIA

- ▶ Confidentiality
- ▶ Integrity
- ▶ Availability

FUNDAMENTALS

- ▶ Defence in depth
- ▶ Least privilege
- ▶ Zero trust
- ▶ Separation of duties

CLOUD-NATIVE SECURITY

- ▶ Code
- ▶ Containers
- ▶ Clusters
- ▶ Cloud (on-premise)

CODE

- ▶ Security starts in the application
- ▶ Secure software development

SUPPLY CHAIN SECURITY

- ▶ Trusting third-party libraries
 - Have you analysed the code of all third-party libraries before use?
 - Did you verify their integrity?
- ▶ Do you keep all libraries up-to-date?

THIRD-PARTY SECURITY VULNERABILITIES

- ▶ Automated warnings about security vulnerabilities
- ▶ Snyk
- ▶ Trivy
- ▶ Dependabot

TASK: SCANNING WITH TRIVY

- ▶ <https://github.com/ScilifelabDataCentre/lunch-menu>
- ▶ Are there any known vulnerabilities?
- ▶ Use `trivy`, `snyk`, or any other scanner
- ▶ Hints:
 - `requirements.txt`
 - `yarn.lock`
 - `trivy fs`

What is a container?

What are the differences between a container
and a virtual machine?

CONTAINERS

- ▶ Namespaced processes
- ▶ OCI
 - Image
 - Runtime
 - Distribution

CONTAINER RUNTIMES

- ▶ Docker
- ▶ Containerd
- ▶ CRI-O
- ▶ Gvisor
- ▶ Kata
- ▶ Firecracker

- ▶ Containers share the kernel
- ▶ Very sensitive data should be in different vms/clusters

- ▶ Containers share the kernel
- ▶ Very sensitive data should be in different vms/clusters

ROOT == ROOT != ROOT

- ▶ Container user == host user
 - User namespaces exist, but have limited support
- ▶ Capabilities
- ▶ hostUsers
- ▶ Privileged container == danger

- ▶ Run containers using hardened runtimes
- ▶ Kata containers (virtualisation)
- ▶ gVisor (sandbox)

TASK: USE GVISOR

- ▶ <https://killercoda.com/killer-shell-cks/scenario/sandbox-gvisor>

CONTAINER DRIFT

- ▶ Preventing containers from doing unintended actions

SECCOMP, APPARMOR, SELINUX

- ▶ Security frameworks to add extra security
- ▶ Ubuntu: seccomp, apparmor
- ▶ Red Hat: seccomp, selinux
- ▶ Define a security profile to be used with a container
- ▶ Optimal security: define a specialised profile for each container
- ▶ Using the "runtime default" is better than nothing

TASK: USING APPARMOR

- ▶ <https://killercode.com/killer-shell-cks/scenario/apparmor>

SECURITY VULNERABILITIES

- ▶ Scan the packaged software in container images
- ▶ E.g. Trivy

TASK: IMAGE AND CODE SCANNING

- ▶ `https://github.com/ScilifelabDataCentre/lunch-menu`
- ▶ Do the latest container images contain any known vulnerabilities?
- ▶ Use Trivy or any other scanner
- ▶ `trivy image container:label`

MALICIOUS COMPLIANCE

- ▶ <https://www.youtube.com/watch?v=9weGi0csBZM>
- ▶ Possible to trick the vulnerability scanners
 - Remove package management files
 - Symlinks
 - Multi-stage builds

BUILDING CONTAINERS

- ▶ Minimise
 - Minimal base
 - Remove unused binaries
 - Squash layers
- ▶ Never include secrets during the build steps
- ▶ Automate
- ▶ Do not run as root
- ▶ Immutable

KUBERNETES

- ▶ Container orchestration
- ▶ Designed to be flexible
- ▶ Declare wanted state
 - Reconciliation loop

RUNNING WITH SCISSORS

<https://www.youtube.com/watch?v=ltrV-Qmh3oY>

OWASP KUBERNETES TOP TEN

[HTTPS://OWASP.ORG/WWW-PROJECT-KUBERNETES-TOP-TEN/](https://owasp.org/www-project-kubernetes-top-ten/)

- K01 Insecure Workload Configurations
- K02 Supply Chain Vulnerabilities
- K03 Overly Permissive RBAC Configurations
- K04 Lack of Centralized Policy Enforcement
- K05 Inadequate Logging and Monitoring
- K06 Broken Authentication Mechanisms
- K07 Missing Network Segmentation Controls
- K08 Secrets Management Failures
- K09 Misconfigured Cluster Components
- K10 Outdated and Vulnerable Kubernetes Components

K10: OUTDATED AND VULNERABLE KUBERNETES COMPONENTS

- ▶ Keep Kubernetes updated
- ▶ Three most recent minor releases:
 - 1.28
 - 1.27
 - 1.26
- ▶ ~1 year support
- ▶ Distributions may be supported longer

K09: MISCONFIGURED CLUSTER COMPONENTS

- ▶ CIS Benchmarks (kube-bench)

K08: SECRETS MANAGEMENT FAILURES

- ▶ Encrypt secrets
- ▶ Vault
- ▶ Sealed secrets etc

K07: MISSING NETWORK SEGMENTATION CONTROLS

- ▶ Zero trust
- ▶ Network policies

NETWORK POLICIES

- ▶ <https://editor.networkpolicy.io/>
- ▶ Default: deny-all for namespace
- ▶ Minimise access

K06: BROKEN AUTHENTICATION MECHANISMS

- ▶ certificate = forever
- ▶ Service account tokens
- ▶ use MFA if possible

K05: INADEQUATE LOGGING AND MONITORING

- ▶ Audit logs, logs in external system
- ▶ Falco etc for monitoring
- ▶ Fluentbit etc
- ▶ Loki

K04: LACK OF CENTRALIZED POLICY ENFORCEMENT

- ▶ OPA Gatekeeper, Kyverno
- ▶ “Policies in git”

TASK: USING KYVERNO

<https://killercoda.com/kyverno/scenario/intro>

- ▶ GitOps — Argo
- ▶ Infrastructure as code
- ▶ Ansible
- ▶ “Replace one node every month”

K03: OVERLY PERMISSIVE RBAC CONFIGURATIONS

- ▶ Least Privilege
- ▶ Service account and user RBAC permissions
- ▶ Limit use of ClusterRoleBinding
- ▶ Not everyone needs admin permissions

K02: SUPPLY CHAIN VULNERABILITIES

- ▶ Security vulnerabilities in third-party libraries
- ▶ Insecure/malicious images
- ▶ Container signing
- ▶ Image minimisation

SUPPLY CHAIN VULNERABILITIES

- ▶ Signing
- ▶ Do not use untrusted images

Signing

K01: INSECURE WORKLOAD CONFIGURATIONS

- ▶ Scanners: kube-score, kubesecc, kubeaudit, snyk ...
- ▶ Pod Security Standards
- ▶ Pod admission: kyverno, Open Policy Agent Gatekeeper

POD SECURITY STANDARDS

- ▶ <https://kubernetes.io/docs/concepts/security/pod-security-standards/>
- ▶ **Apply to a namespace**
- ▶ `pod-security.kubernetes.io/<MODE>: <LEVEL>`
- ▶ `pod-security.kubernetes.io/<MODE>-version: <VERSION>`
- ▶ **Enforce, audit, warn**

TASK: CREATE A HARDENED DEPLOYMENT

- ▶ Make a deployment of `ghcr.io/scilifelabdatacentre/menu-backend:latest`
- ▶ Create a new namespace and apply the pod security standard restricted to it
- ▶ Update your deployment to allow deployment in the created namespace

HOST SECURITY

- ▶ Hardening
- ▶ CIS Benchmark
- ▶ Minimisation
- ▶ Firewalls

COMPLIANCE

- ▶ Not just a checklist
- ▶ Aid to make your systems more secure

- ▶ Least Privilege
- ▶ Defence in depth (layered security)
- ▶ Zero Trust