

The Kubernetes Common Configuration Scoring System

KCCSS

Julien Sobrier

Introduction



Julien Sobrier

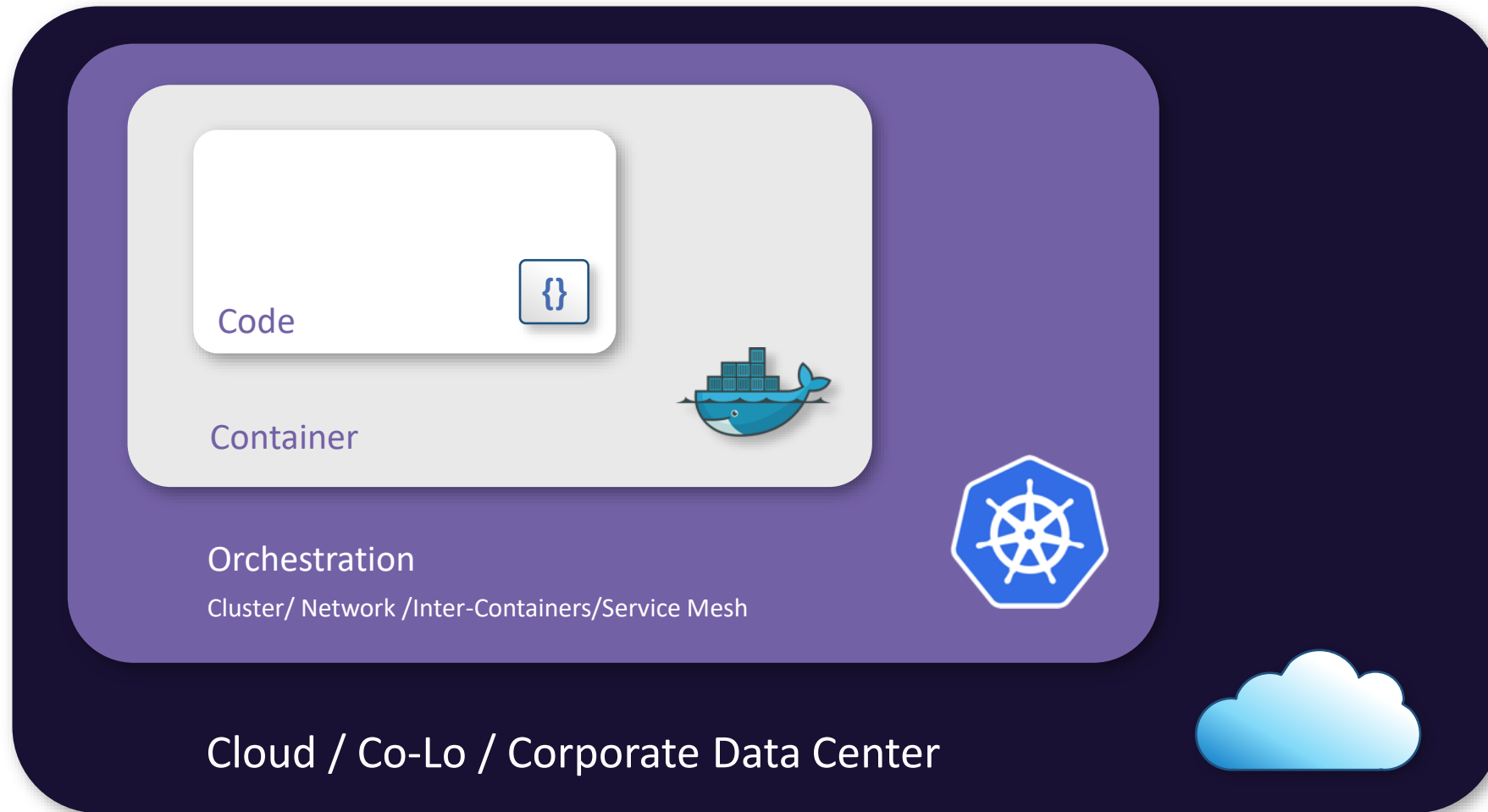
Product Line Manager,
VMWare

Product Line Manager at VMWare

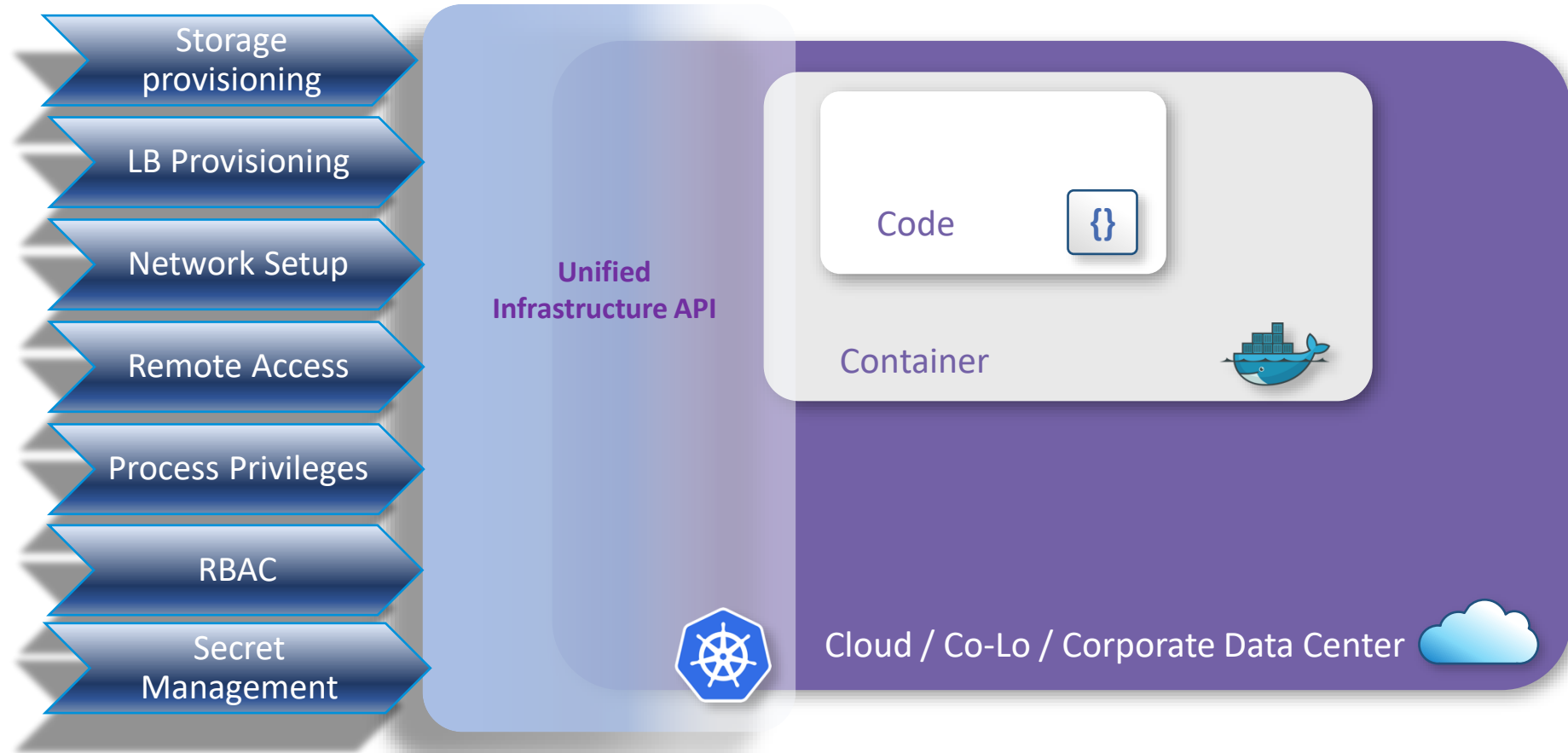
10+ years as Security Researcher

jsobrier@vmware.com

Kubernetes is a container orchestrator ...



... and much more



30+ security configuration settings

Isolation

- share PID
- share IPC
- host Path
- Network MiTM
- Network sniffing

Network Exposure

- Load Balancer
- share Host network
- kubectl port-forward
- Ingress policy
- Egress policy

Integrity

- Writeable file system
- host paths
- Privileged

K8s API

- Create/delete pods
- Access secrets

Capabilities

- Run As Root
- Privileged
- CAP_SYS_ADMIN
- CAP_NET_RAW

Immutability

- Seccomp
- SELinux
- AppArmor

Service Mesh

- Encryption
- Workload identity
- L7 Network policy

30+ security configuration settings

Isolation

- share PID
- share IPC
- host Path
- Network MiTM
- Network sniffing

Integrity

- Writeable file system
- host paths

Immutability

- Seccomp
- SELinux
- AppArmor
- Read-Only FS

Network Exposure

- share Host network
- kubelet port forward

Load Balancer

Egress policy

Ingress policy

K8s API

Privileged

Create/delete pods

Access security

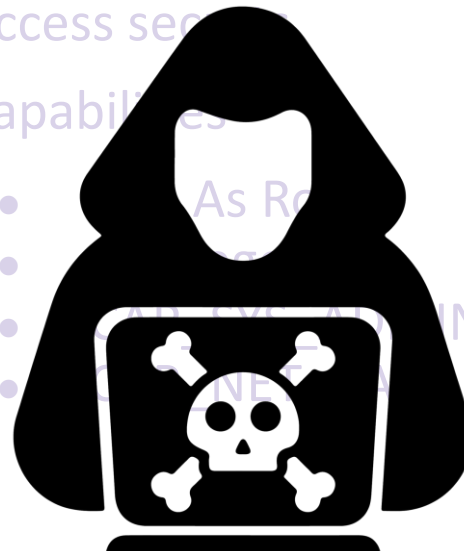
Capabilities

- As Role
-
-
-

Service Mesh

Encryption

- Workload identity
- L7 Network policy



30+ security configuration settings

Isolation

- share PID
- share IPC
- host Path

Integrity

Writeable file system

- host paths
- Privileged

Immutability

- Seccomp
- SELinux
- AppArmor

Network MiTM

Network sniffing

Network Exposure

kubectl port-forward
share Host network

Capabilities

- Run As Root
- Privileged
- CAP_SYS_ADMIN

CAP_NET_RAW

Service Mesh

Encryption

- Workload identity
- L7 Network policy

Risky?
Don't know



Goal for risk framework

Give a risk score to the workload

Explain where the risk comes from and what it is

Show how to remediate the risk

Risk ^	Name	Kind	Namespace	Domain
9	webserver-for-tests	Deployment	webserver2	cloud:aws-west
8	analysis-kafka	StatefulSet	analysis	cloud:aws-west
5	analytics	StatefulSet	analysis	cloud:aws-west
5	analytics	StatefulSet	development	cloud:aws-west
5	apache	StatefulSet	vm	cloud:aws-west
5	backend	StatefulSet	development	cloud:aws-west
5	baltimore	StatefulSet	inventory	cloud:aws-west

Existing risk frameworks

CVSS: score vulnerabilities

Impact of the risk: Availability,
Confidentiality, Integrity

Scope of the risk (blast radius)

Exploitability, attack vector

CCSS: CVSS applied to configuration

CCE: check list of configuration settings



NIST

**National Institute of
Standards and Technology**

U.S. Department of Commerce



KCCSS

List of rules (like CCE)

Same description of rules as CVSS

Applies to configurations settings (like CCSS)

NEW: aggregates all risks into a single workload risk

NEW: specific to Kubernetes

- tools/Helm
- K-1-Privileged.yaml
- K-10-HostPathRO.yaml
- K-11-CAP_SYS_ADMIN.yaml
- K-12-ExternalLoadBalancer.yaml
- K-13-NodePort.yaml
- K-14-IngressController.yaml
- K-15-SharedHostPort.yaml
- K-16-ShareHostNetwork.yaml
- K-17-ShareHostPID.yaml
- K-18-ShareHostIPC.yaml
- K-2-RunningAsRoot.yaml
- K-3-AllowPrivilegeEscalation.yaml
- K-4-CAP_NET_RAW.yaml
- K-6-UnmaskedProcMount.yaml
- K-7-AllowedUnsafeSysctls.yaml
- K-8-CPUMemoryQuota.yaml
- K-9-HostPathRW.yaml

The rules

Risk

- Impact on Availability
- Confidentiality
- Integrity
- Exploitability
- Attack Vector
- Scope

MEDIUM RISK
Workload is exposed through a shared host network [show less](#)

LOW INTEGRITY IMPACT
Services open to the Internet may be used to access unprotected services (move laterally) by leveraging remote code vulnerabilities, vulnerable third-party libraries or vulnerable OS services

HIGH CONFIDENTIALITY IMPACT
This allows the network to listen to the loopback interface and sniff the traffic to and from other pods. This setting also allows workloads to bind their listening IP address to the host IP, making the service accessible from other networks and/or from the Internet

HIGH AVAILABILITY IMPACT
Accidental exposure to the Internet can make the workload susceptible to DoS attacks from random attackers


LOW EXPLOITABILITY | **REMOTE** ATTACK VECTOR | **HOST** SCOPE
Fairly unlikely to be exploited | Remotely exploitable | Impact the node

DESCRIPTION
This Security Context setting allows the workload to share the same network namespace as the host

The rules

Remediation

- Lower exiting risk
- Impact on Availability
- Confidentiality
- Integrity
- Attack Vector
- Scope

**REMEDIATION**
Workload is instrumented by Octarine[show less](#)

LOW **INTEGRITY IMPACT**
The strong identity provided by an Octarine and/or Istio service mesh prevents rogue containers from impersonating trusted workloads

HIGH **CONFIDENTIALITY IMPACT**
Service meshes, such as Istio and Octarine, provide encryption of network traffic, as well as strong identity, which prevents network sniffing and Man-in-the-Middle (MiTM) attacks

LOW **AVAILABILITY IMPACT**
Service meshes, such as Istio and Octarine, can detect and stop abnormal increases in network activities and network errors

REMOTE **ATTACK VECTOR** **CONTAINER** **SCOPE**
Mitigate Remote Risks Protect the container

DESCRIPTION
The Istio and Octarine service mesh encrypts all internal network activities with a mutual TLS connection and uses certificates to provide strong identity to all workloads, which greatly reduces the potential attack surface

Formulas

1. Rate each risk

- 0 (low) to 10 (high)
- Similar to CVSS formula
- Base Impact score = $f(\text{Availability, Confidentiality, Integrity})$
- Impact score = $f(\text{scope, Base Impact score})$
- Exploitability score = $f(\text{Attack Vector, Exploitability})$
- Rule score = Impact score + Exploitability score

Formulas

2. Workload score

- 0 (low) to 10 (high)
- Brand new
- Working on improved version

Scores = Max(Attack Vector \cap Scope)

Workload score = $\sqrt{(\text{Score1}^2 + \text{Score2}^2 + \dots)}$

Formulas

Remediations

For each risk, match remediation with the same attack vector & scope
Lower corresponding risk impact

Example:

Risk:	C:H/I:H/A:H
Remediation:	C:L/I:H/A:N
Final risk:	C:M/I:L/A:H

kube-scan

- KCCSS should be used by tools to run the risk score on your workloads
- Kube-scan: open-source workload configuration scanner using KCCSS
 - Install the kube-scan container in your cluster
 - Scan your running workloads
 - See the results through the Web UI

Demo

Further work

- Better matching of remediations and risks
- Improved formula to rate the workload risk
- Additional rules around RBAC
- More references: CIS Benchmark, MITRE Attack framework, etc
- Additional tools to explore KCCSS

github

KCCSS: <https://github.com/octarinesec/kccss/>

kube-scan: <https://github.com/octarinesec/kube-scan>



Thank You