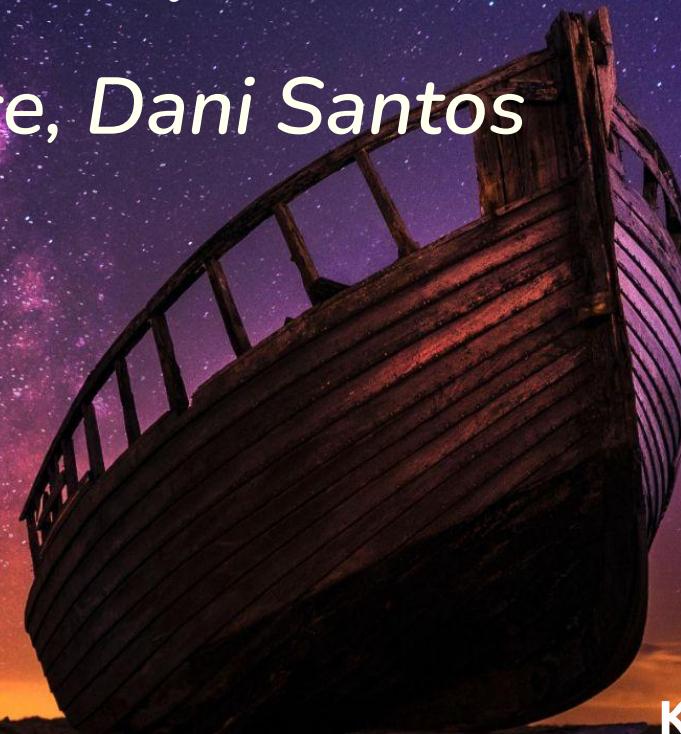


Prevent Embarrassing Cluster Takeovers with This One Simple Trick!

Shane Lawrence, Dani Santos



KubeCon



CloudNativeCon

Europe 2023



Shane Lawrence

Staff Infrastructure Security Engineer
@Shopify



www.lawrence.dev



Dani Santos

Senior Infrastructure Security Engineer
@Shopify



www.danisantoscode.com

Agenda

1. Misconfig: why should we care?
2. Security principles + kubeaudit
3. Demo: the simple trick!
4. kubeaudit @ Shopify
5. Additional resources

2022

State of Kubernetes security report



53%

detected Kubernetes security
misconfigurations
in the last 12 months

Executive summary

Security concerns

Hindering innovation

Container strategies

Responsibility remains decentralized

DevSecOps is seeing mass adoption

Misconfiguration is a top concern

Hybrid cloud deployment

Security use cases

Open source security tools

Tips for better security

About our respondents

Red Hat Advanced Cluster Security for Kubernetes

Respondents worry about misconfigurations above all other security concerns

Kubernetes is a highly customizable container orchestrator, with various configuration options that affect an application's security posture.

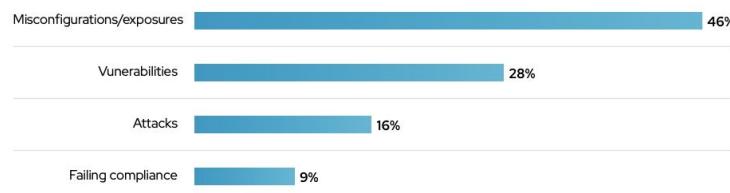
Consequently, respondents worry the most about exposures due to misconfigurations in their container and Kubernetes environments (46%)—nearly three times the level of concern over attacks (16%), with vulnerabilities as the second-leading cause of worry (28%).

Configuration management poses a uniquely difficult challenge for security practitioners. The persons responsible for configuring workloads may not understand security implications of various settings within Kubeneretes. While a host of tools are available for vulnerability scanning of container images, configuration management requires more consideration and will likely be unique to organizations and teams depending on their risk tolerance and level of workload sensitivity.

People may know that they should avoid deploying the Kubernetes dashboard, but configuring a pod's security context or implementing Kubernetes role-based access control (RBAC) are just two examples of more challenging settings that teams need to get right.

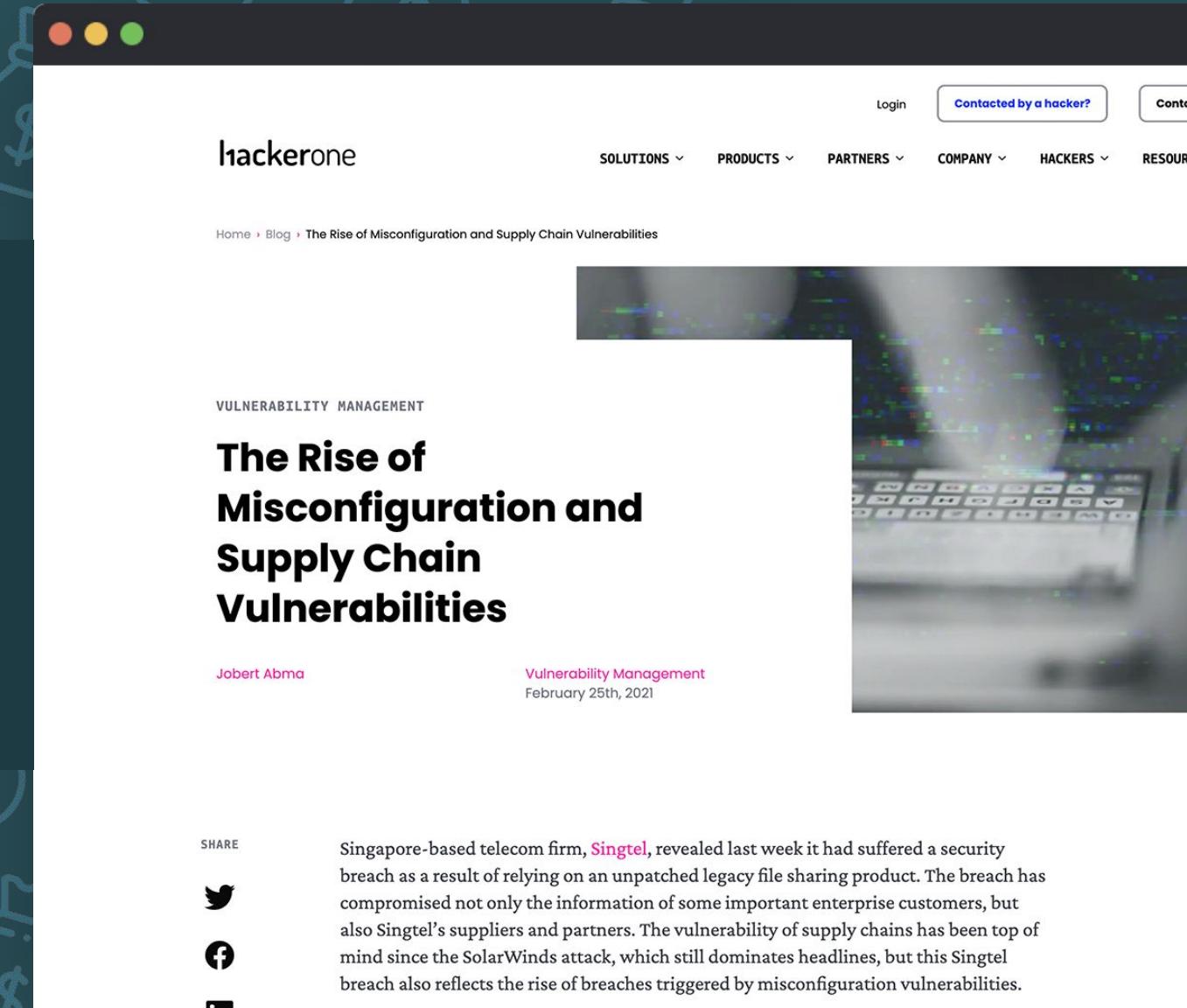
The best way to address this challenge is to automate configuration management as much as possible, so that security tools—rather than humans—provide the guardrails that help developers and DevOps teams configure containers and Kubernetes more securely.

Of the following risks, which one are you most worried about for your container and Kubernetes environments?



Misconfiguration is a top concern

More than
\$150,000
in a few weeks



The screenshot shows a blog post titled "The Rise of Misconfiguration and Supply Chain Vulnerabilities" by Jobert Abma. The post is categorized under "VULNERABILITY MANAGEMENT". It includes social sharing icons for Twitter, Facebook, and LinkedIn. The main content discusses a security breach at Singapore-based telecom firm Singtel due to misconfiguration. To the right of the post is a blurred image of a computer keyboard.

hackerone

SOLUTIONS ▾ PRODUCTS ▾ PARTNERS ▾ COMPANY ▾ HACKERS ▾ RESOUR

Home > Blog > The Rise of Misconfiguration and Supply Chain Vulnerabilities

VULNERABILITY MANAGEMENT

The Rise of Misconfiguration and Supply Chain Vulnerabilities

Jobert Abma

Vulnerability Management

February 25th, 2021

Singapore-based telecom firm, [Singtel](#), revealed last week it had suffered a security breach as a result of relying on an unpatched legacy file sharing product. The breach has compromised not only the information of some important enterprise customers, but also Singtel's suppliers and partners. The vulnerability of supply chains has been top of mind since the SolarWinds attack, which still dominates headlines, but this Singtel breach also reflects the rise of breaches triggered by misconfiguration vulnerabilities.

Azurescape

- enabled **cross-account takeover**
- started from a **container escape**
- not exploited in the wild*
*(we think)



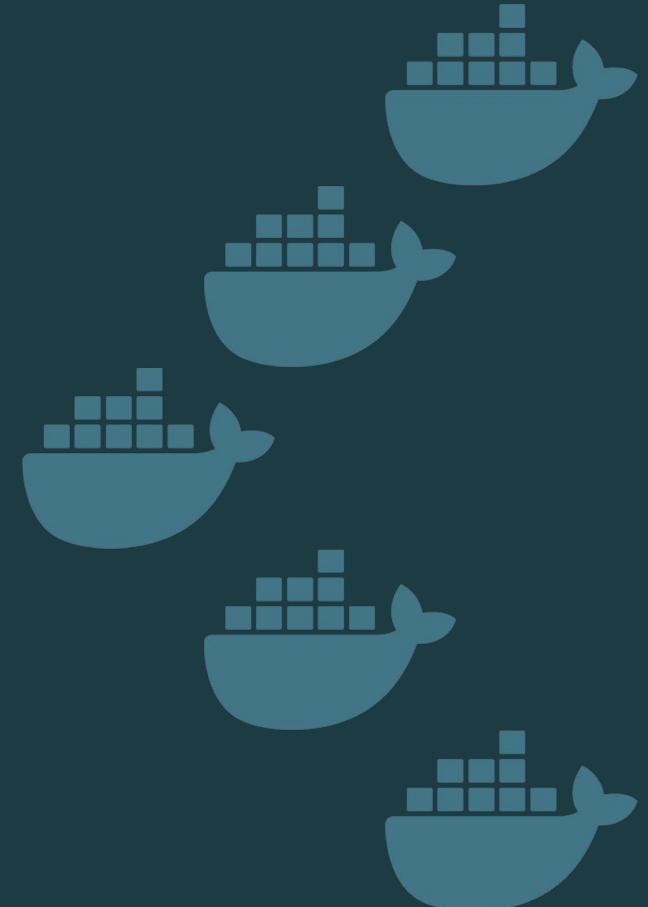
Azurescape



- container escape through a two-year-old vulnerability in runC
- code execution on the api-server
- researchers could execute code on customer containers!

CVE-2019-5736

- vulnerability in container runtime runC
- requires root access to exploit
(runC binary is owned by root)
- affected all Docker containers



“Kubernetes has historically not been security hardened out of the box, and sometimes this may lead to privilege escalation or container breakout.”

*Andrew Martin, ControlPlane
&
Michael Hausenblas, AWS*

Security controls

- security contexts
- users & privilege
- capabilities
- network policies
- syscall restrictions

kubeaudit

- fully open source
- developed by Shopify since 2017
- scans Kubernetes manifests & clusters
- identifies misconfigs

The screenshot shows the GitHub repository page for `Shopify / kubeaudit`. The repository is public and has 1.6k stars, 410 watchers, and 168 forks. It contains 18 branches and 40 tags. The code tab is selected, showing a list of recent commits:

File / Action	Description	Date
<code>dani-santos-code go releaser action (#539)</code>	538937d 2 weeks ago	
<code>.github</code>	go releaser action (#539)	
<code>auditors</code>	remove refs to old annotations (#527)	
<code>build</code>	Use makefile/goreleaser to set version.	
<code>cmd</code>	bump minor version (#534)	
<code>config</code>	feat(mounts): update list of sensitive paths	
<code>docs</code>	go releaser action (#539)	
<code>internal</code>	By default, test without Kind (#524)	
<code>pkg</code>	remove refs to old annotations (#527)	
<code>.dockerignore</code>	feat(docker): build Docker images too (#29)	
<code>.gitignore</code>	adds support for sarif output (#453)	
<code>.goreleaser.yml</code>	go releaser action (#539)	
<code>CODE_OF_CONDUCT.md</code>	update Code of Conduct contact email (#143)	
<code>Dockerfile</code>	Upgrade to go 1.17 (#410)	
<code>LICENSE</code>	Initial Commit	
<code>Makefile</code>	Fix CI: tidy with 1.17 (#521)	
<code>README.md</code>	Remove Docker release process (#540)	
<code>VERSION</code>	Version (#334)	
<code>example_custom_test.go</code>	adds support for sarif output (#453)	
<code>example_test.go</code>	adds support for sarif output (#453)	
<code>fix.go</code>	Make k8s and override packages public (#351)	
<code>fix_test.go</code>	Kubeaudit Package (#243)	
<code>go.mod</code>	Bump github.com/spf13/cobra from 1.5.0 to 1.6.1 (#497)	

Contributors: 40

Go: 98.5% • Other 1.5%

kubeaudit helps you audit your Kubernetes clusters against common security controls

kubernetes computers audit

MIT license



bypass

✿ Gaining environment information

✿ DoS the Memory/CPU resources

✿ Hacker container preview

✿ Hidden in layers

✿ RBAC least privileges misconfiguration

✿ KubeAudit - Audit Kubernetes clusters

Welcome

✿ Falco - Runtime security monitoring & detection

✿ Popeye - A Kubernetes cluster sanitizer

✿ Secure Network Boundaries using NSP

✿ Cilium Tetragon - eBPF-based Security Observability and Runtime Enforcement

✿ Securing Kubernetes Clusters using Kyverno Policy Engine

📋 Security Reports

🔴 Teardown

✿ KubeAudit - Audit Kubernetes clusters

✿ Overview

This scenario is very useful in performing Kubernetes security audits and assessments. Here we will learn to run an open-source tool called `kubeaudit` for the Kubernetes cluster and use the results for the further exploitation or fixing of the misconfigurations and vulnerabilities. This is very important and mandates if you are coming from an audit and compliance background in the modern world of containers, Kubernetes, and cloud native ecosystems.



Kubernetes Goat: Scenario diagram WIP

✿ Overview

⚡ The story

🎯 Goal

📝 Hints & Spoilers

💡 Solution & Walkthrough

🌐 Method 1

📌 References

kubeaudit

vs.

Azurescape

kubeaudit checks for:



privileged



running as root



excessive capabilities



Principles

Least Privilege

- non-root
- privileged
- capabilities

Separation of Duties

- hostNetwork
- hostPID
- network policies

“simple trick”

Before

```
! test.yaml U ×  
! test.yaml  
1  apiVersion: v1  
2  kind: Pod  
3  metadata:  
4    name: test  
5    annotations:  
6      | container.apparmor.security.beta.kubernetes.localhos  
7  spec:  
8    containers:  
9      - name: test  
10     | image: busybox:1.28  
11     | command: [ "sh", "-c", "echo 'Hello AppArmor!' && sleep 1h" ]  
12
```



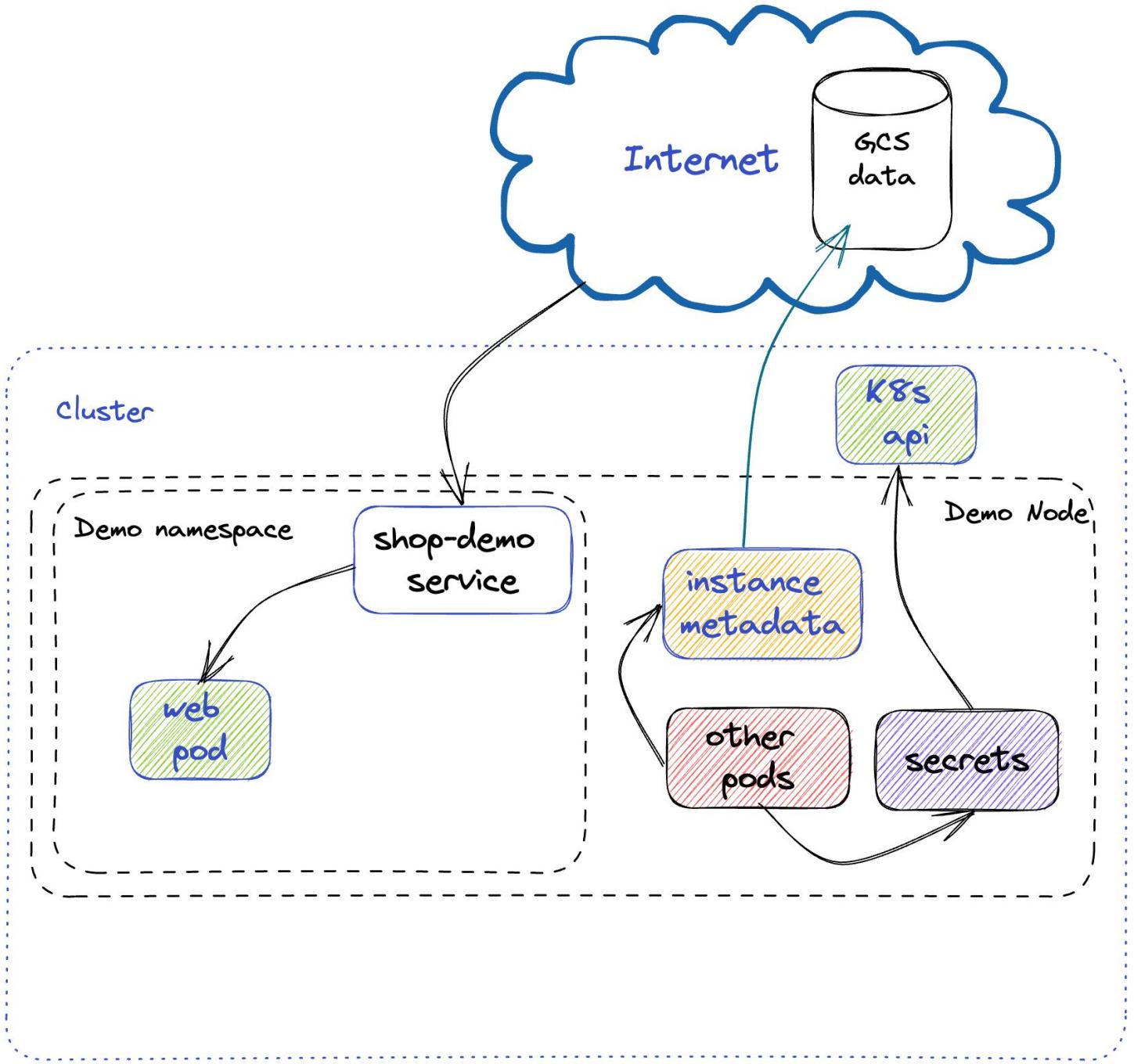
After

```
! test.yaml U ×  
! test.yaml  
1  apiVersion: v1  
2  kind: Pod  
3  metadata:  
4    name: test  
5    annotations:  
6      | container.apparmor.security.beta.kubernetes.localhos  
7  spec:  
8    containers:  
9      - name: test  
10     | image: busybox:1.28  
11     | command: [ "sh", "-c", "echo 'Hello AppArmor!' && sleep 1h" ]  
12  
13  
14   securityContext:  
15     allowPrivilegeEscalation: false  
16     capabilities:  
17       drop:  
18         - ALL  
19     privileged: false  
20     readOnlyRootFilesystem: true  
21     runAsNonRoot: true  
22     automountServiceAccountToken: false  
23   securityContext:  
24     seccompProfile:  
25       type: RuntimeDefault
```



```
kubeaudit autofix -f test.yaml
```

Demo Architecture



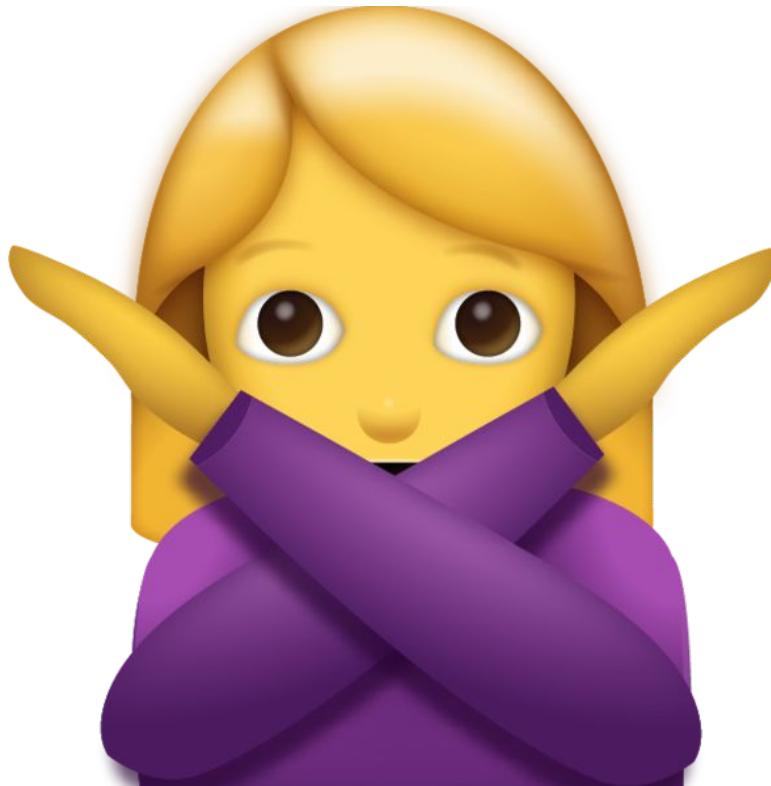
DEMO



Bad news



There's no
simple trick!



Build time: Shifting left

Image scanning

- Snyk
- Trivy
- Anchore
- cloud providers

Code scanning

- static analysis (SAST)
- semgrep
- ...
- kubeaudit!

kubeaudit

@Shopify



Secure defaults

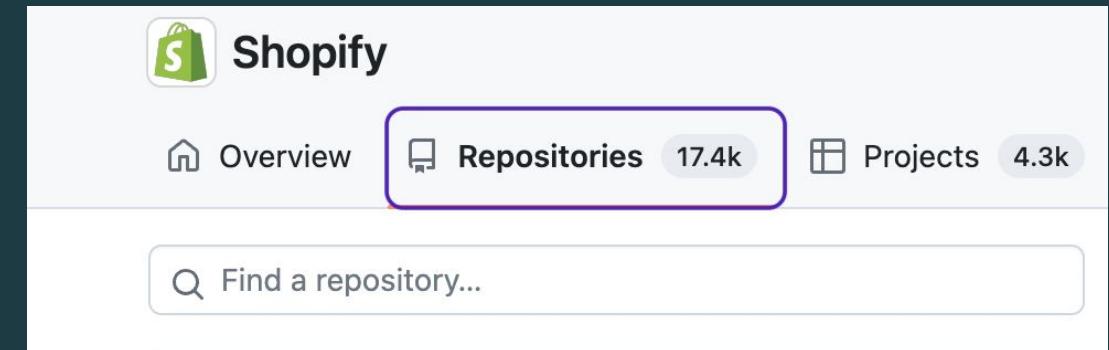
- base kubernetes components
- trust and empower developers
- devs can deviate from the happy path

kubeaudit @Shopify



Thousands of repos

- How do we make sure no misconfig is introduced in new PRs?
- How can we confirm that the risk introduced is acceptable?





Security Awareness

It is important to raise awareness regarding security concerns!

```
117      capabilities:  
118          add:  
119              - CHOWN  
120              - DAC_OVERRIDE  
121              - KILL  
122              - SETGID  
123              - SETUID  
124          drop:  
125              - ALL  
126      privileged: true  
127      readOnlyRootFilesystem: true  
128      runAsNonRoot: false
```

kubeaudit @Shopify



SARIF reports

Github Code Scanning integration

Code scanning results / kubeaudit Failing after 3s — 11 new alerts inclu

Check failure manifest.yaml.

Code scanning / kubeaudit

PrivilegedTrue Error

Details: privileged is set to 'true' in container SecurityContext. It should be set to 'false'.
Auditor: privileged
Description: Finds containers running as privileged
Auditor docs: <https://github.com/Shopify/kubeaudit/blob/main/docs/auditors/privileged.md>
[Show more details](#)

kubeaudit @Shopify



Opting out

Shopify / kubeaudit-github-app-test · Private

Code Issues 2 Pull requests 6 Actions Security 6 Insights Settings

Code scanning alerts / #71

PrivilegedTrue

In branch in ds/test-adding-metadata-field on Jan 24

infrastructure/runtimes/production-unrestricted-em1g/manifest.yaml.lock:1

```
1 # THIS FILE IS AUTO-GENERATED. DO NOT MODIFY IT MANUALLY.
```

Details: privileged is set to 'true' in container SecurityContext. It should be set to 'false'.
Auditor: privileged
Description: Finds containers running as privileged
Auditor docs: <https://github.com/Shopify/kubeaudit/blob/main/docs/auditors/privileged.md>

kubeaudit

```
2 apiVersion: apps/v1
3 kind: Deployment
4 metadata:
```

Tool	Rule ID
kubeaudit	PrivilegedTrue

Type: kubernetes
Auditor Docs: To find out more about the issue and how to fix it, follow [this link](#)
Description: Finds containers running as privileged
Metadata: Metadata: {"Container": "web"}

Show more ▾

Select a reason to dismiss

Won't fix
This alert is not relevant

False positive
This alert is not valid

Used in tests
This alert is not in production code

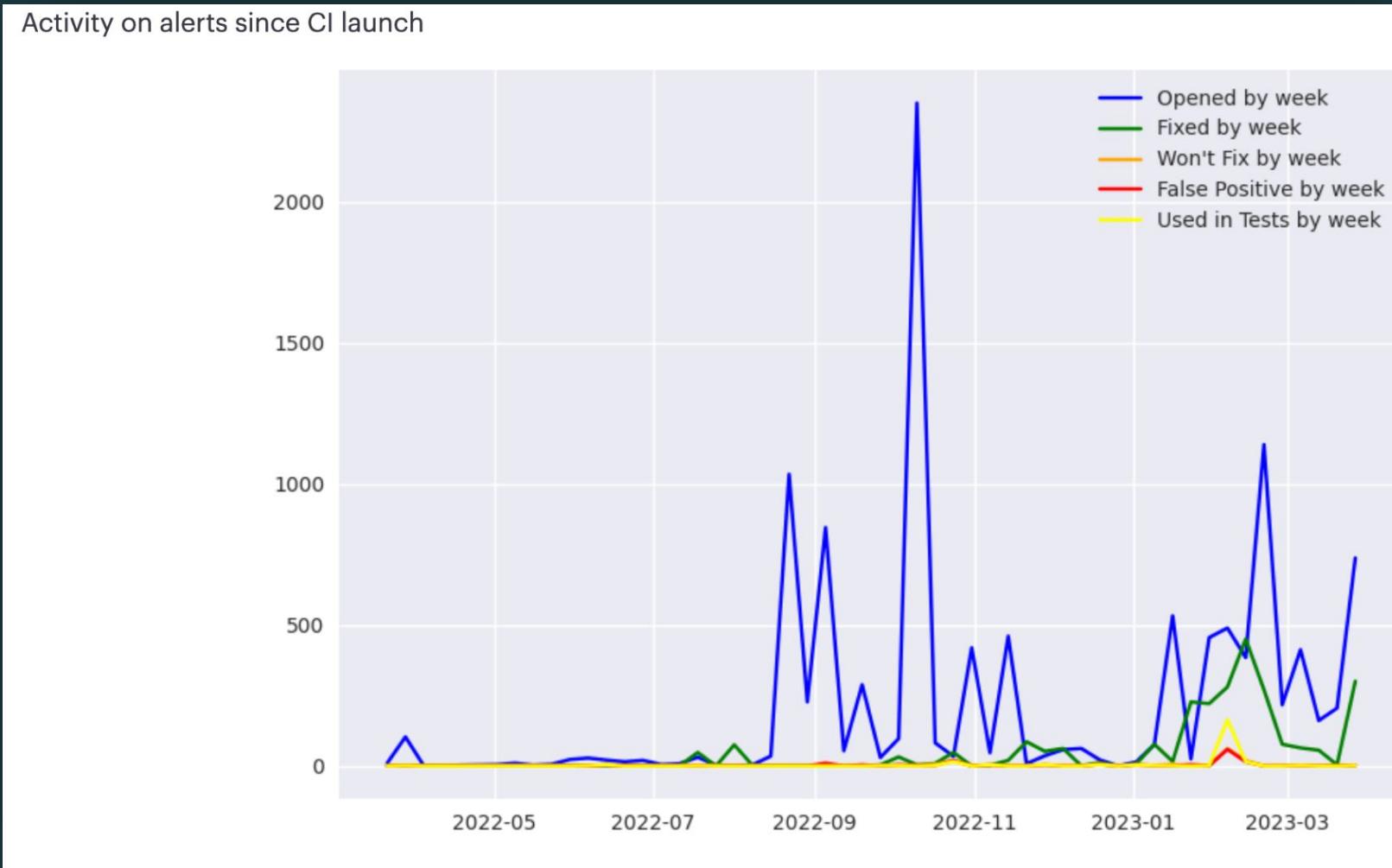
Dismissal comment

Add a comment

Cancel Dismiss alert

Monitoring

- alert metrics
- fine tuning
- Avoid: *The Boy Who Cried Wolf*



Build-time

- secure Docker configuration
- minimal container images

Deploy-time

- admission controllers
- security contexts & policies

Run-time

- host hardening
- network security

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Impact
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access the K8S API server	Access cloud resources	Data Destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account	Resource Hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access container service account	Network mapping	Cluster internal networking	Denial of service
Application vulnerability	Application exploit (RCE)		Access cloud resources	Connect from Proxy server	Applications credentials in configuration files	Access Kubernetes dashboard	Applications credentials in configuration files	
Exposed Dashboard	SSH server running inside container					Instance Metadata API	Writable volume mounts on the host	

MITRE ATT&CK® Framework

Conclusion

- 👉 Harden your k8s clusters
- 👉 Raise security awareness in your org!
- 👉 Contribute to kubeaudit ❤️

Thank you!

Shane

 @shaneplawrence



 www.lawrence.dev

Dani

 @danisantoscode



 www.danisantoscode.com



Scan to rate our talk.

Resources

- Kubeaudit: <https://github.com/Shopify/kubeaudit>
- Demo repo: https://github.com/dani-santos-code/kubecon_2023_prevent_cluster_takeover
- Kubernetes Goat: <https://madhuakula.com/kubernetes-goat/>
- Redhat State of Kubernetes Security Report (2022): <https://www.redhat.com/en/resources/state-kubernetes-security-report>
- "What you Need to Know About Azurescape": (Palo Alto Networks blog post): <https://www.paloaltonetworks.com/blog/2021/09/azurescape/>
- "Breaking out of Docker via runC – Explaining CVE-2019-5736" (Unit 42): <https://unit42.paloaltonetworks.com/breaking-docker-via-runc-explaining-cve-2019-5736/>
- "The Rise of Misconfiguration and Supply Chain Vulnerabilities" (Hackerone- Jobert Abma): <https://www.hackerone.com/vulnerability-management/rise-misconfiguration-and-supply-chain-vulnerabilities>
- "The Truth About False Positives in Security" (The Hacker News): <https://thehackernews.com/2022/08/the-truth-about-false-positives-in-security.html>
- "Hacking Kubernetes: Threat-Driven Analysis and Defense" by Andrew Martin and Michael Hausenblas
- "The Kubernetes Book" by Nigel Poulton and Pushkar Joglekar
- "Kubernetes Security and Observability: A Holistic Approach to Securing Containers and Cloud Native Applications" by Brendan Creane and Amit Gupta