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

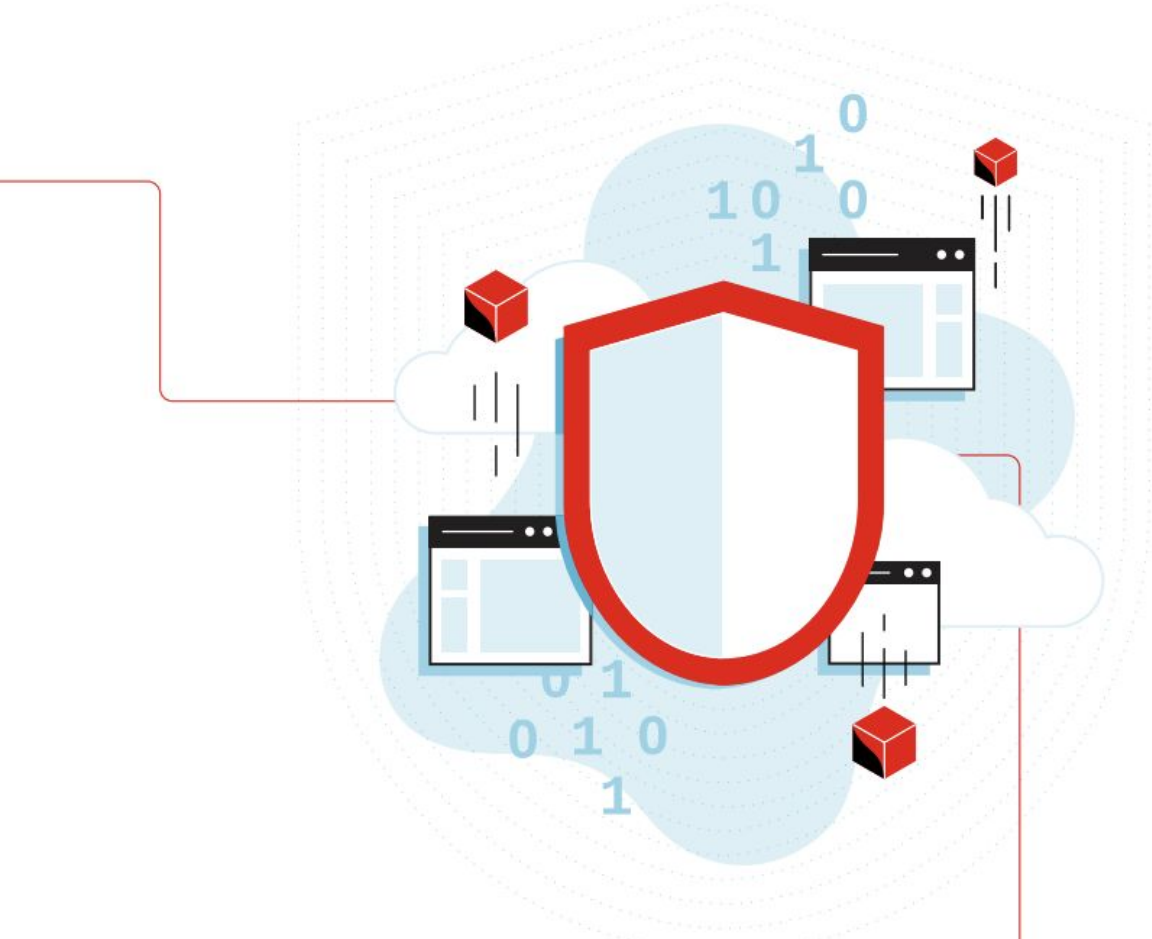
Agenda

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

Red Hat

2022

State of Kubernetes security report



53%

detected Kubernetes security misconfigurations in the last 12 months

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.

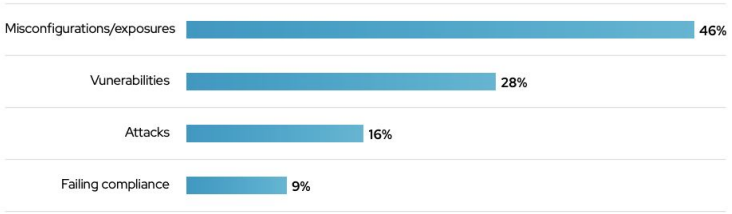
Consequently, respondents worry the most about exposures due to misconfigurations in their container and Kubernetes environments (46%)—nearly three times the level of concern over attacks (16%), with vulnerabilities as the second-leading cause of worry (28%).

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?

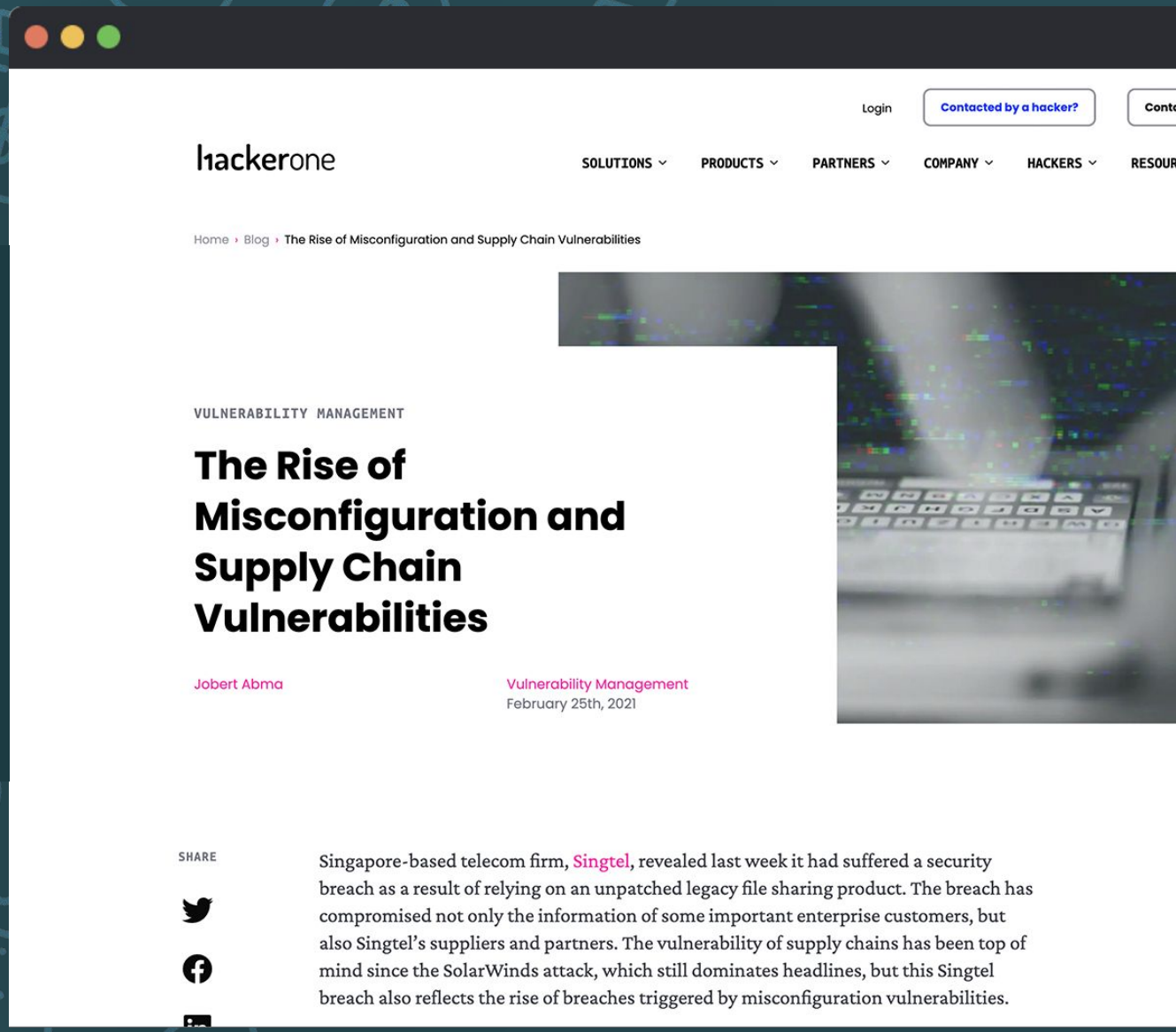


Misconfiguration is a top concern

More than

$150,000

in a few weeks



Azurescape

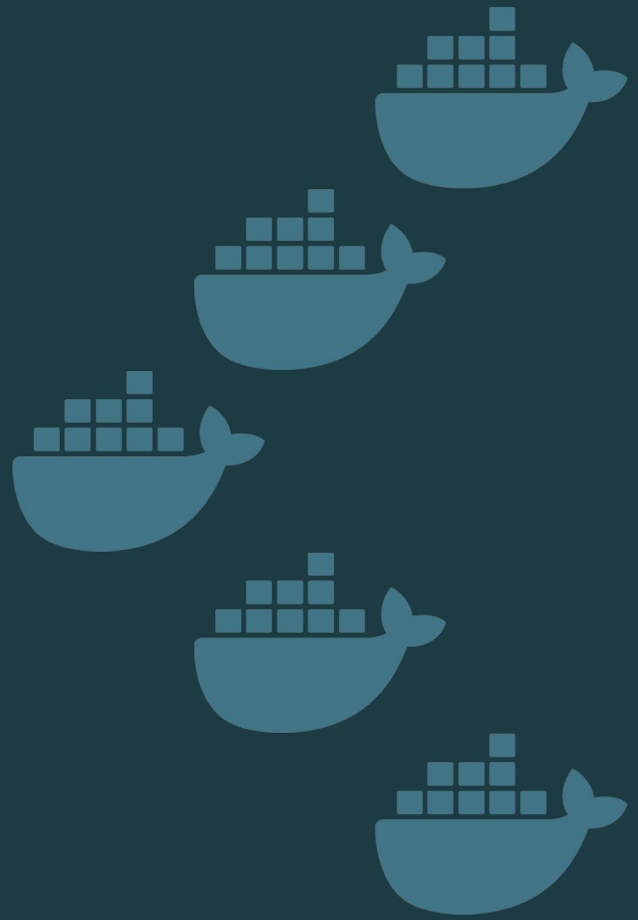
- enabled **cross-account takeover**
- started from a **container escape**
- **not exploited** in the wild*
  *(we think)

Azurescape

- container escape through a two-year-old vulnerability in runC
- code execution on the api-server
- researches 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.”

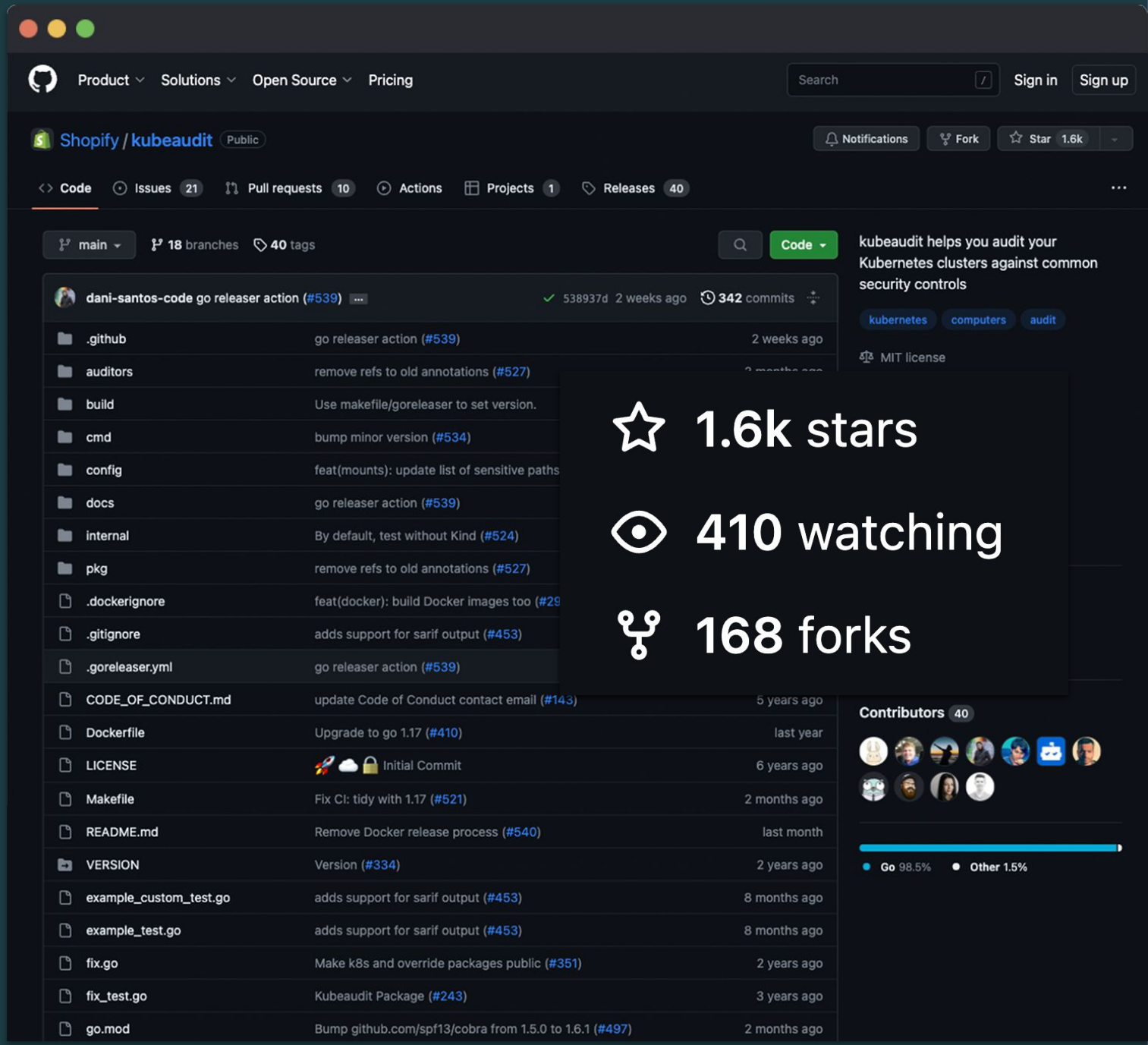
Andrew Martin, ControlPlane
&
Michael Hausenblas, AWS

Security controls

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

kubeaudit

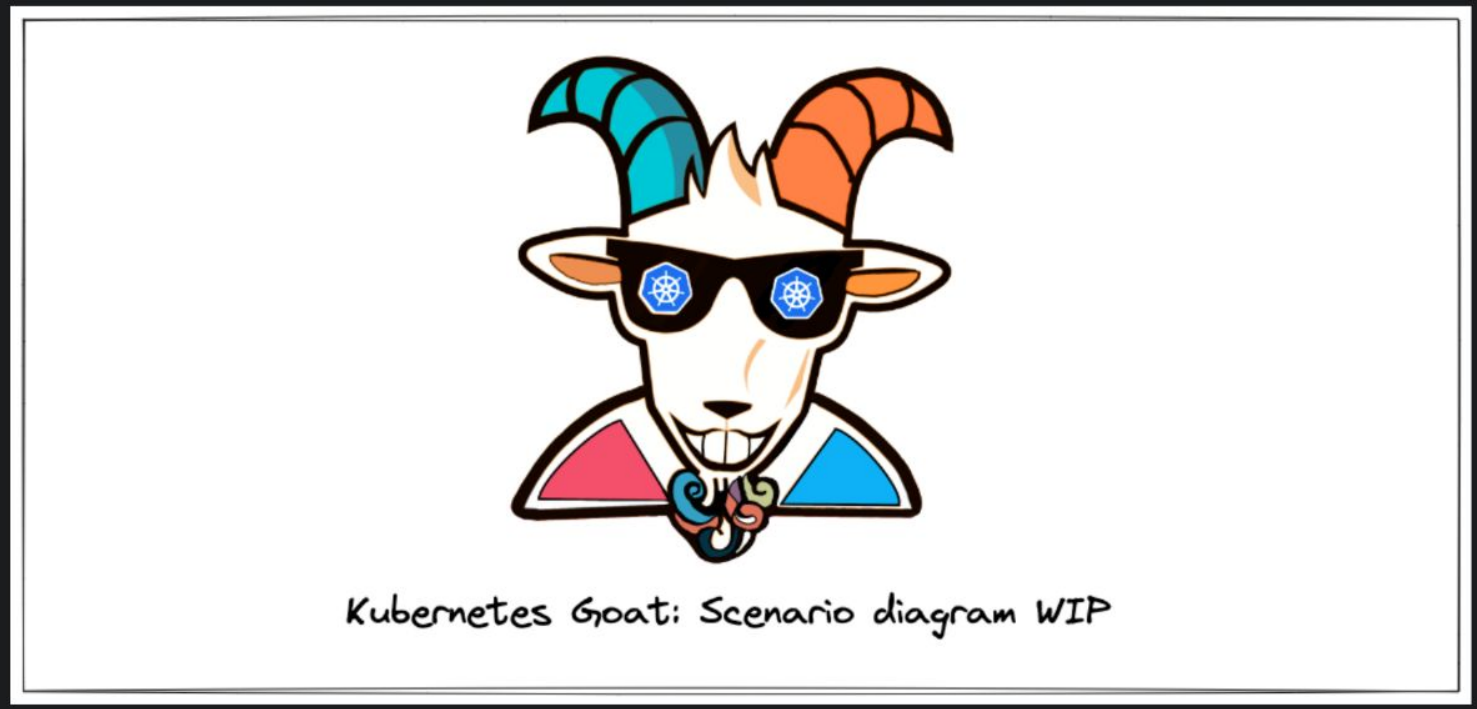
- fully open source
- developed by Shopify since 2017
- scans Kubernetes manifests & clusters
- identifies misconfigs



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.



kubeaudit vs. Azurescape

kubeaudit checks for:

- 😱 privileged
- 😱 running as root
- 😱 excessive capabilities

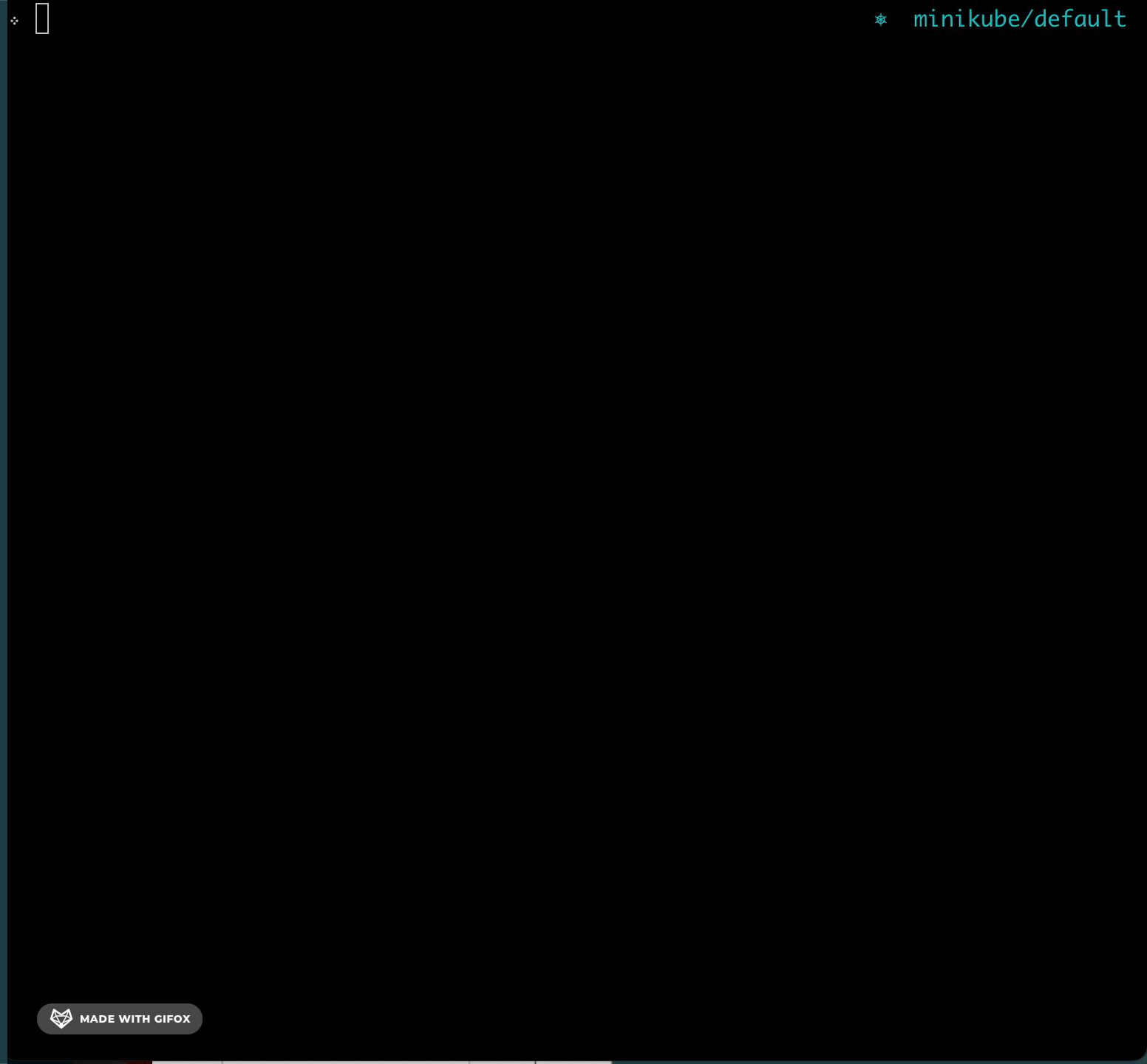
Principles

Least Privilege

- non-root
- privileged
- capabilities

Separation of Duties

- hostNework
- hostPID
- network policies

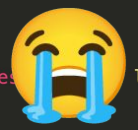


“simple trick”

Before

test.yaml U

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: test
  annotations:
    container.apparmor.security.beta.kubernetes
spec:
  containers:
  - name: test
    image: busybox:1.28
    command: [ "sh", "-c", "echo 'Hello AppArmor!' && sleep 1h" ]
```



localhos

After

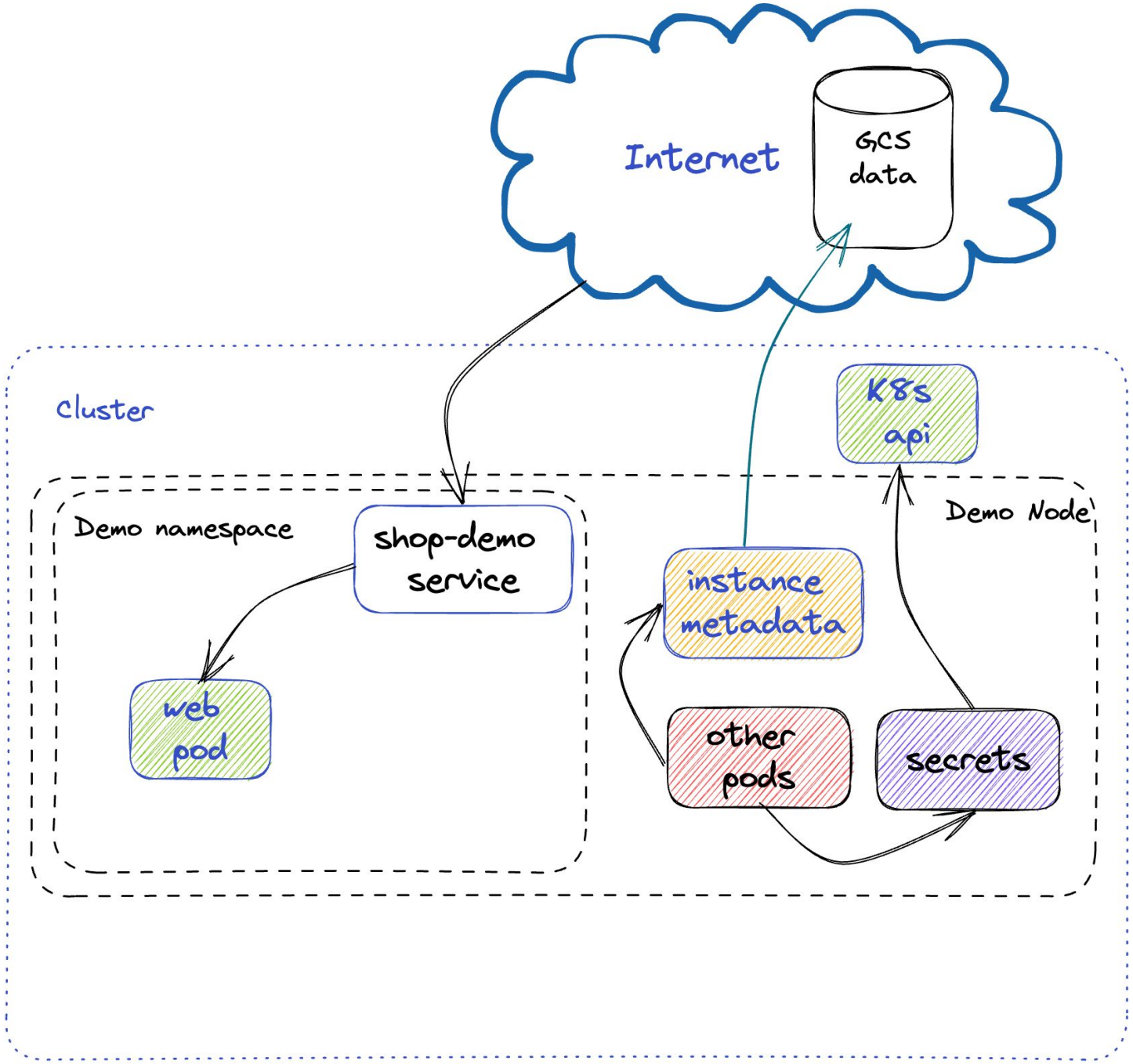
test.yaml U

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: test
  annotations:
    container.apparmor.security.beta.kubernetes.
spec:
  containers:
    - name: test
      image: busybox:1.28
      command: ["sh", "-c", "echo 'Hello AppArmor!' && sleep 1h"]
      resources: {}
      securityContext:
        allowPrivilegeEscalation: false
        capabilities:
          drop:
            - ALL
        privileged: false
        readOnlyRootFilesystem: true
        runAsNonRoot: true
  automountServiceAccountToken: false
  securityContext:
    seccompProfile:
      type: RuntimeDefault
```

localhos

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

Demo Architecture



DEMO

Bad news

☹️

Build time: **Shifting left**

Image scanning

- Snyk
- Trivy
- Anchore
- cloud providers

Code scanning

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

kubeaudit @Shopify

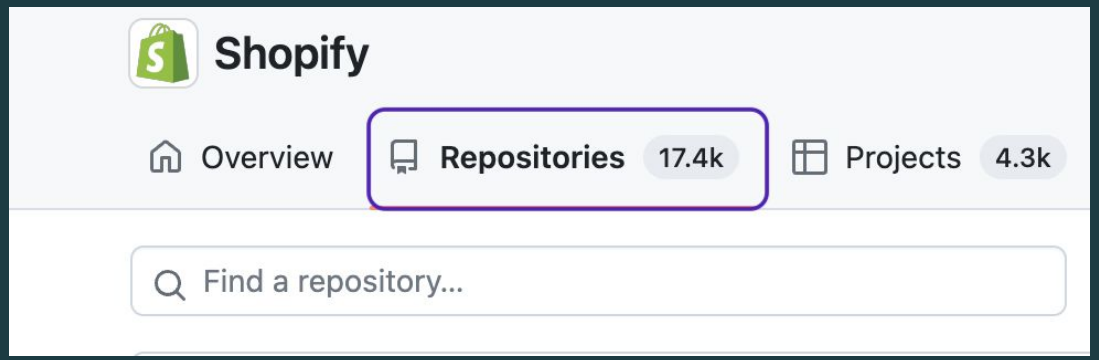
Secure defaults

- 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

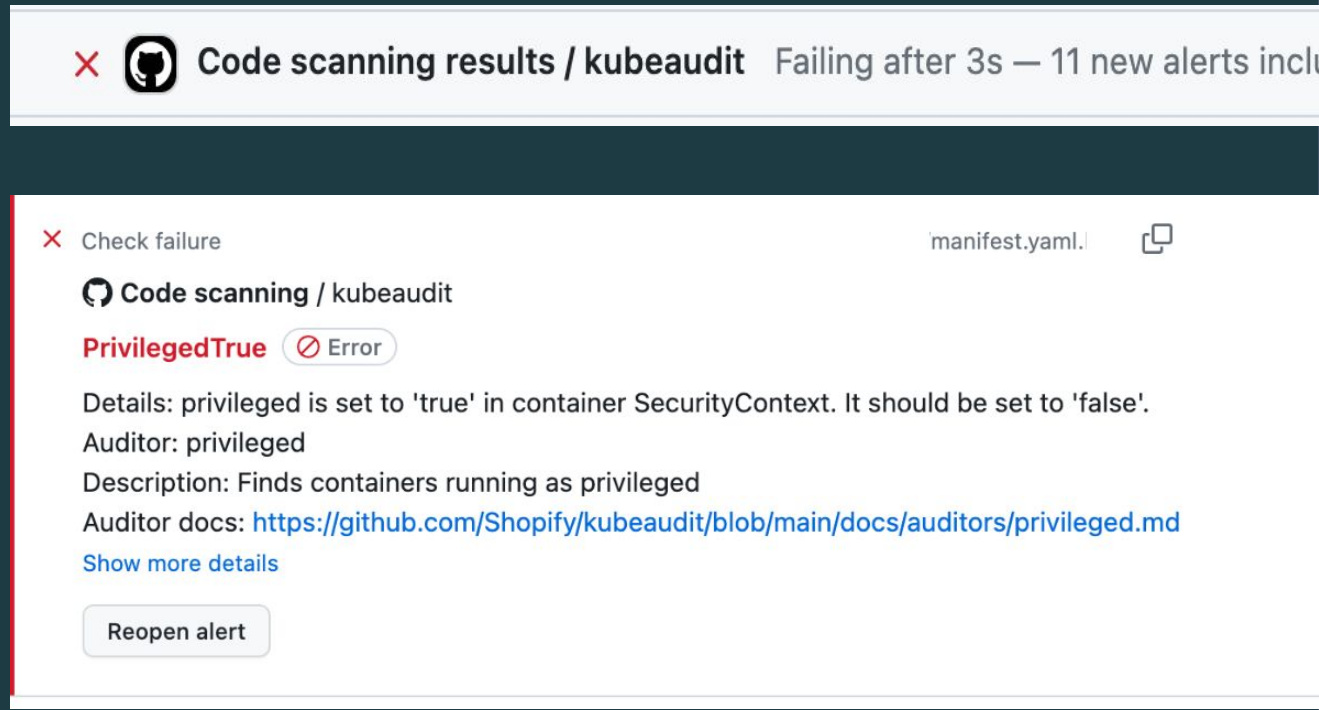
It is important to raise awareness regarding security concerns!

```
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

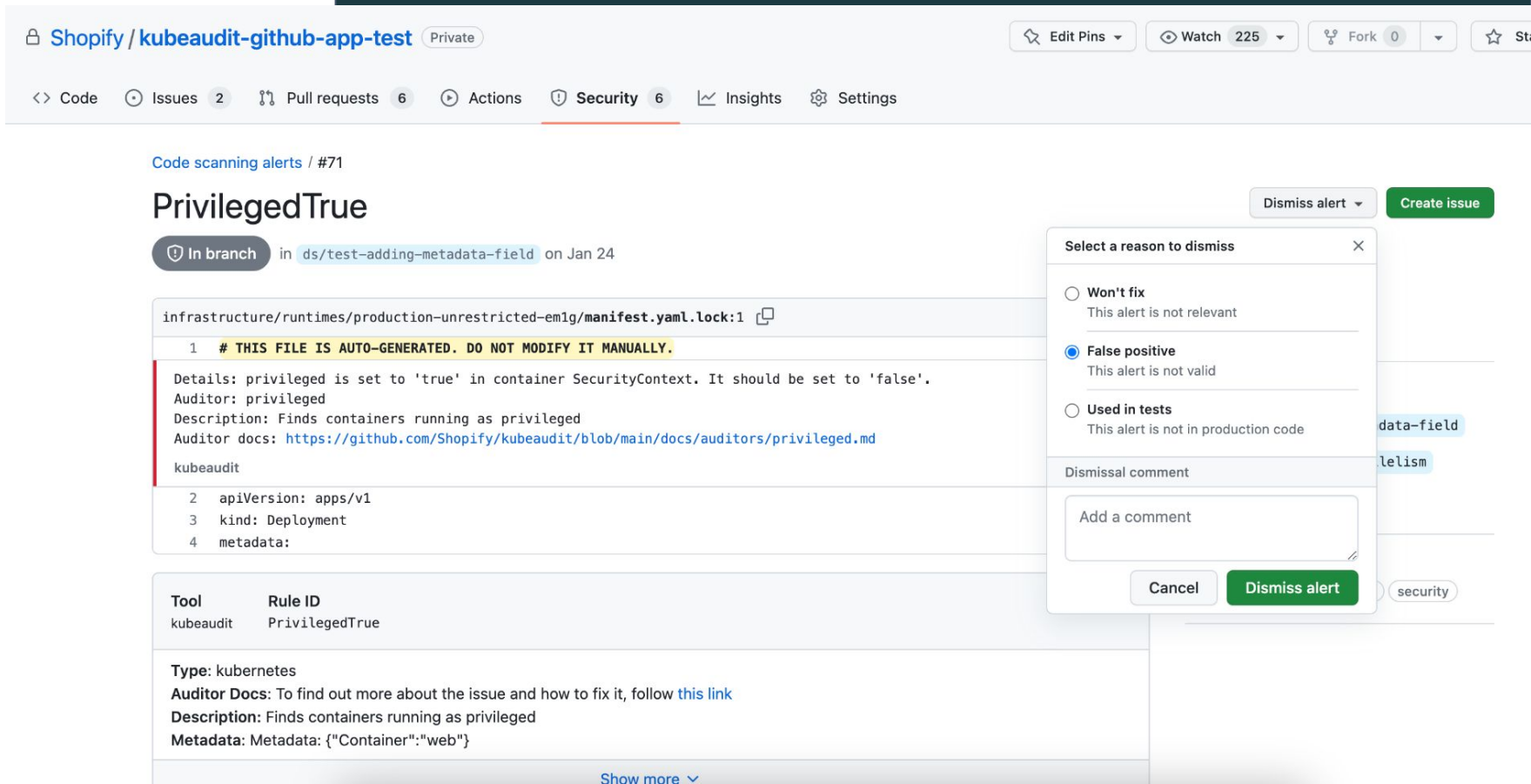
SARIF reports

Github Code Scanning integration



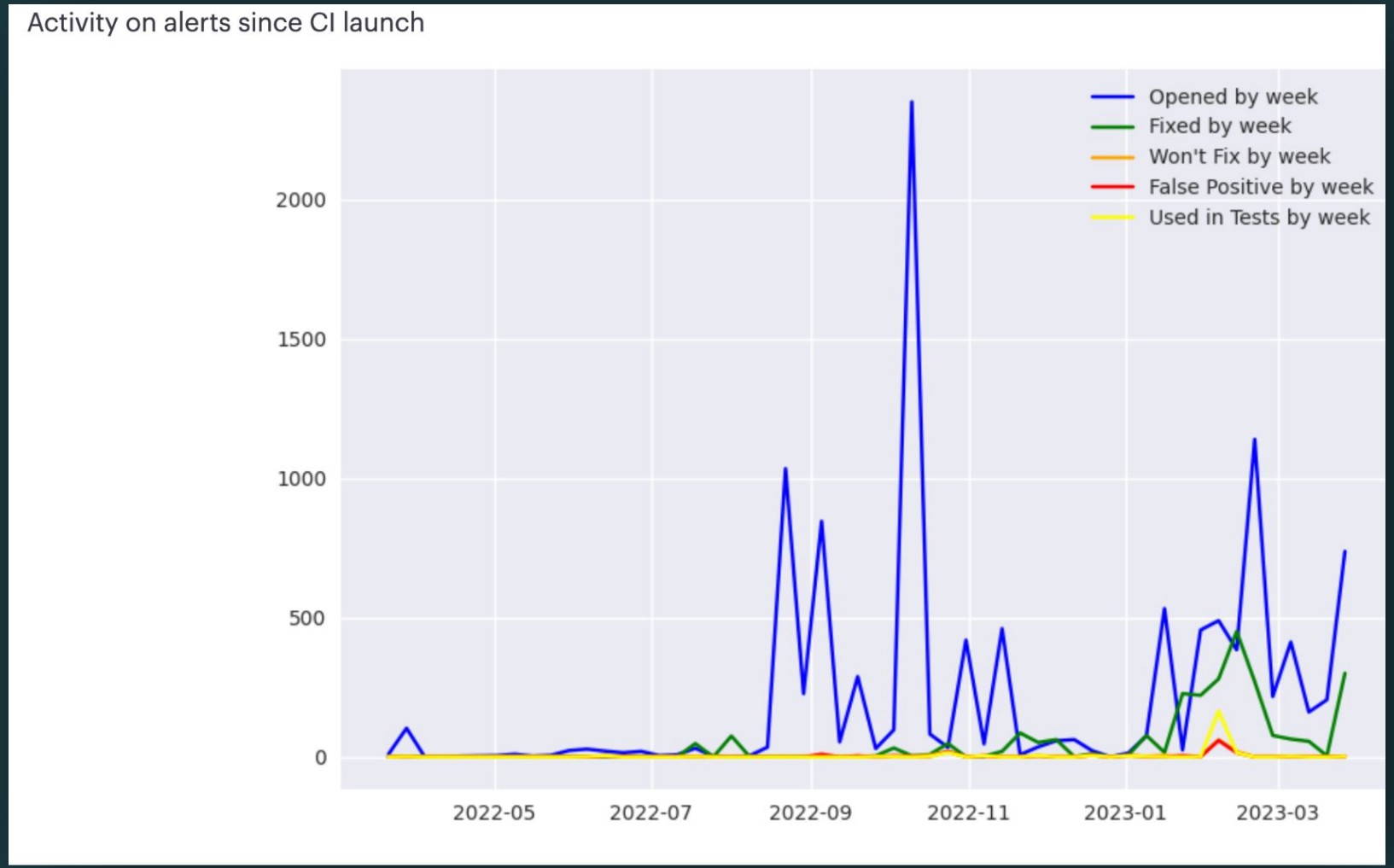
kubeaudit @Shopify

Opting out



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	
							Access Kubernetes dashboard	
							Access tiller endpoint	

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