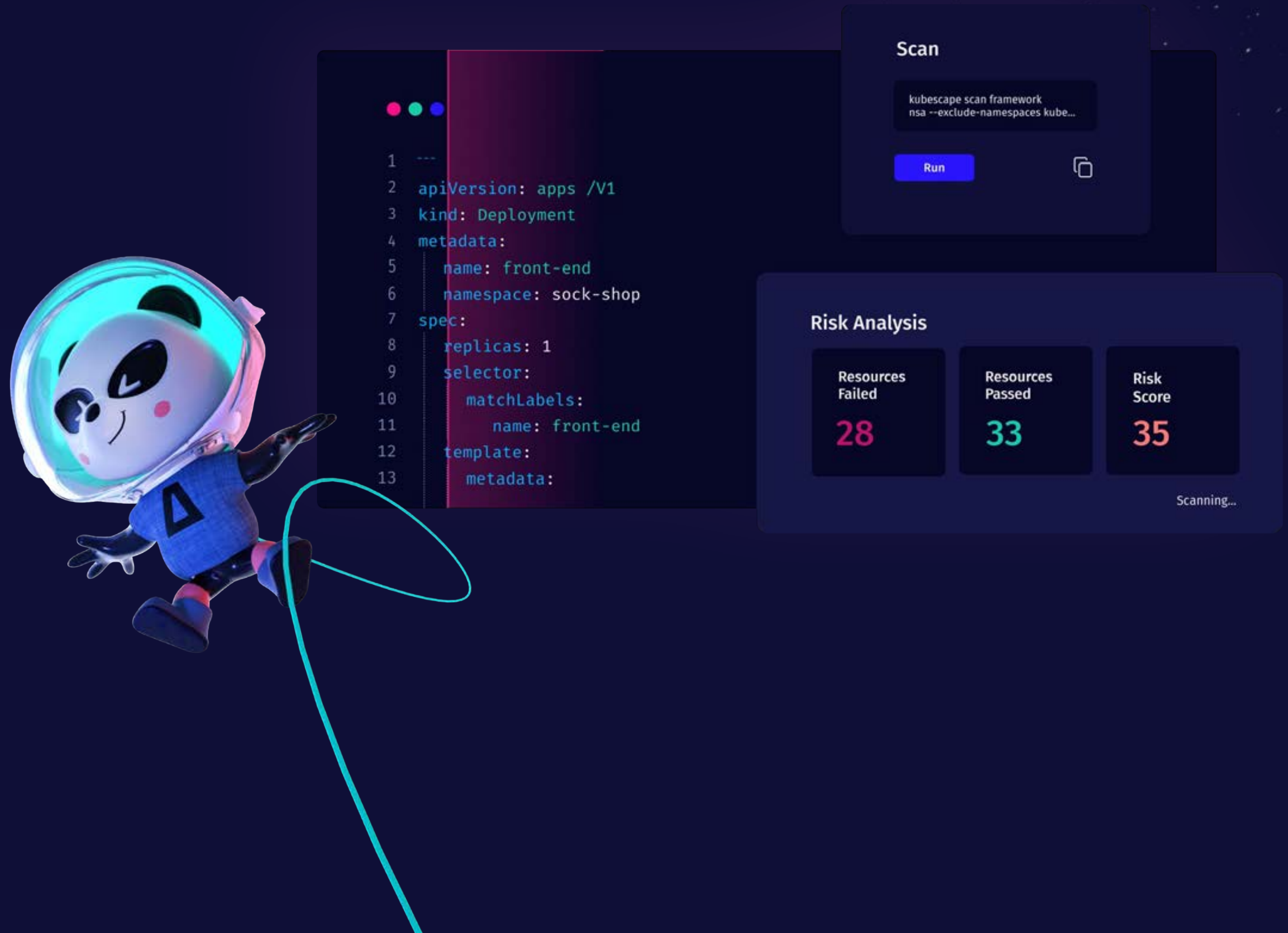
What We've Learned from Scanning 10K+ Kubernetes Clusters



```
---
apiVersion: apps /V1
kind: Deployment
metadata:
  name: front-end
  namespace: sock-shop
spec:
  replicas: 1
  selector:
    matchLabels:
      name: front-end
  template:
    metadata:
```

Scan

```
kubescape scan framework
nsa --exclude-namespaces kube...
```

Risk Analysis

Resources Failed	Resources Passed	Risk Score
28	33	35

Scanning...

ARMO

Rotem Refael
Director of Engineering

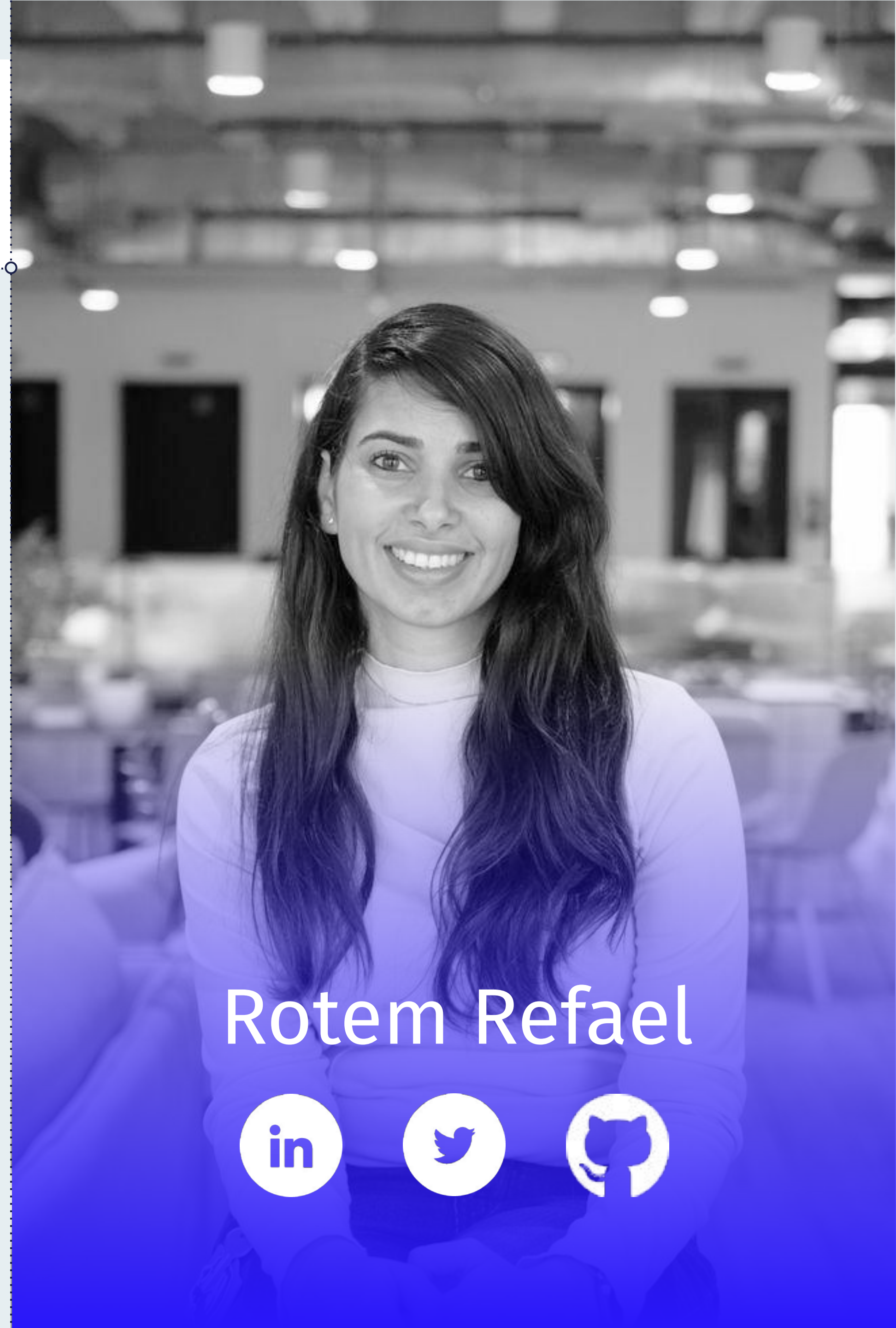
Who am I?

Developer in heart and soul

Devops enthusiast

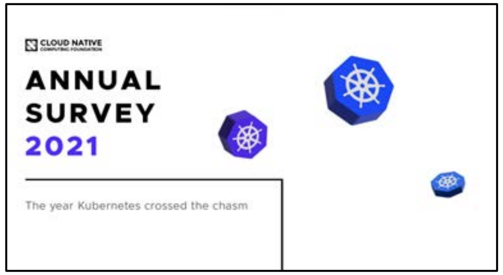
Yoga lover

Basketball fan



Rotem Refael

Kubernetes is the new cloud native operating system



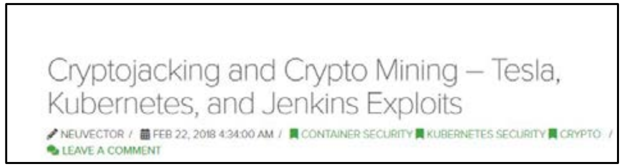
CLOUD NATIVE COMPUTING FOUNDATION

ANNUAL SURVEY 2021

The year Kubernetes crossed the chasm

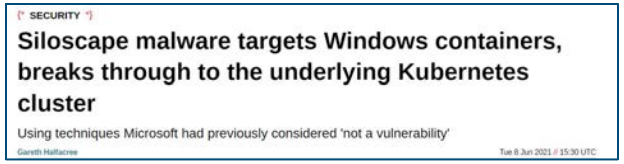
- **96%** of organizations are either using or evaluating Kubernetes
- **5.6 million** developers are using Kubernetes worldwide. That's **31%** of all backend developers
- Kubernetes is increasingly being used in production by companies leveraging managed services and packaged platforms

Kubernetes is no longer "off the radar" for attackers



Cryptojacking and Crypto Mining – Tesla, Kubernetes, and Jenkins Exploits

NEUVECTOR / FEB 22, 2018 4:34:00 AM / CONTAINER SECURITY KUBERNETES SECURITY CRYPTO / LEAVE A COMMENT

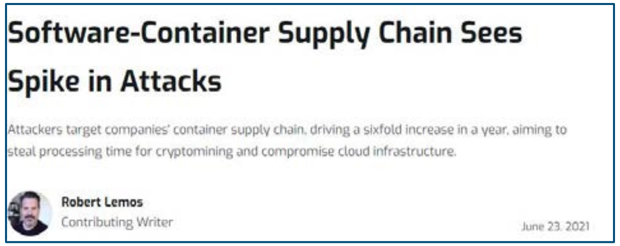


{* SECURITY *}

Siloscape malware targets Windows containers, breaks through to the underlying Kubernetes cluster

Using techniques Microsoft had previously considered 'not a vulnerability'

Gareth Halfacree Tue 8 Jun 2021 // 15:30 UTC



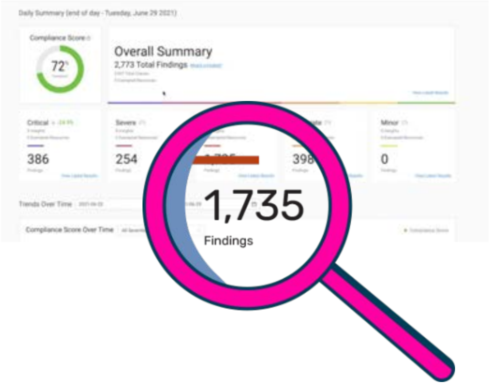
Software-Container Supply Chain Sees Spike in Attacks

Attackers target companies' container supply chain, driving a sixfold increase in a year, aiming to steal processing time for cryptomining and compromise cloud infrastructure.

Robert Lemos
Contributing Writer

June 23, 2021

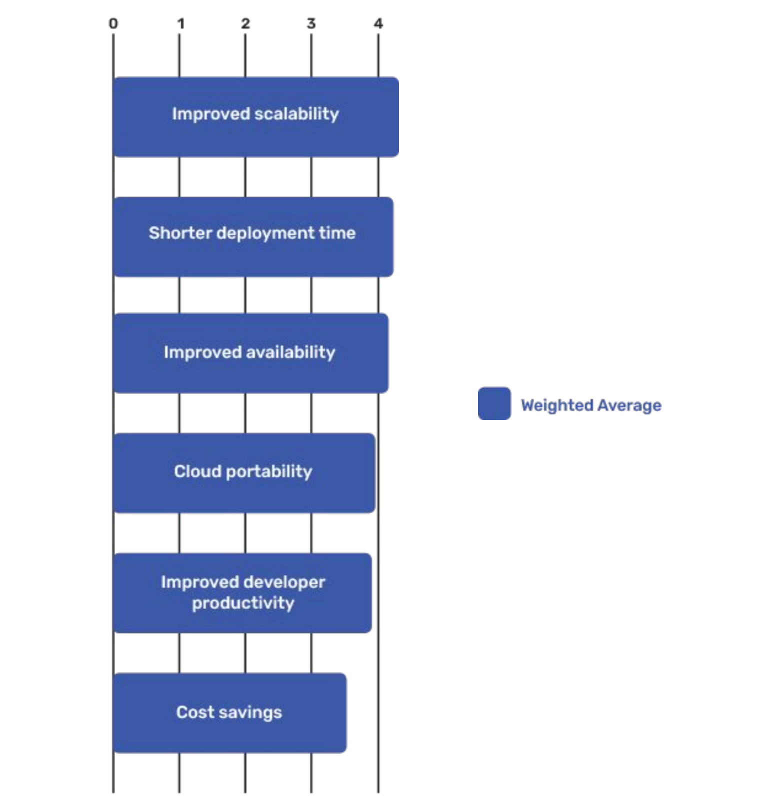
Current security solutions fail to provide "devops" experience



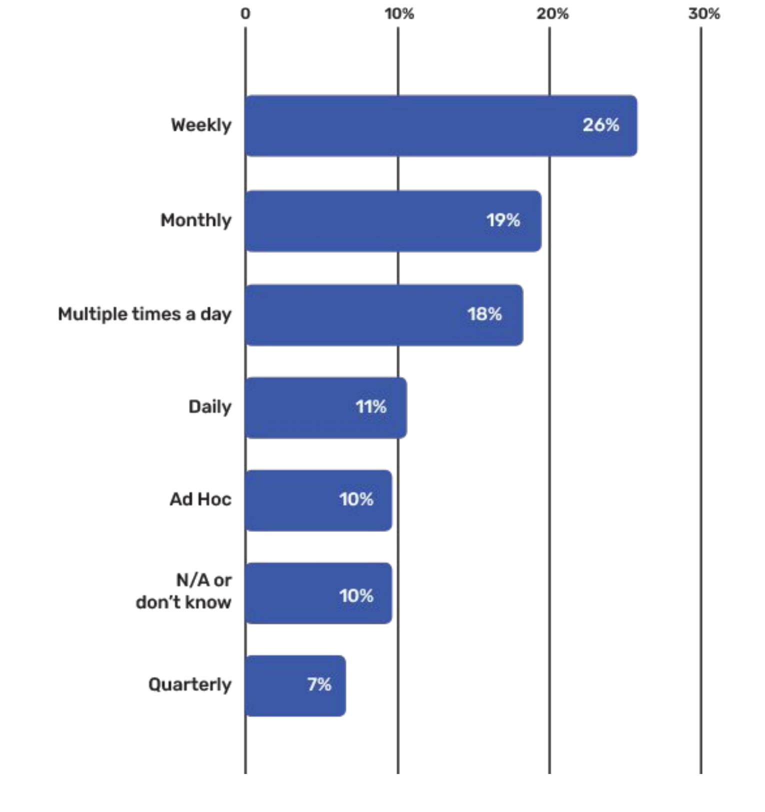
Too many alerts, too much complexity, **diminished security value**

K8s Who, Why and How?

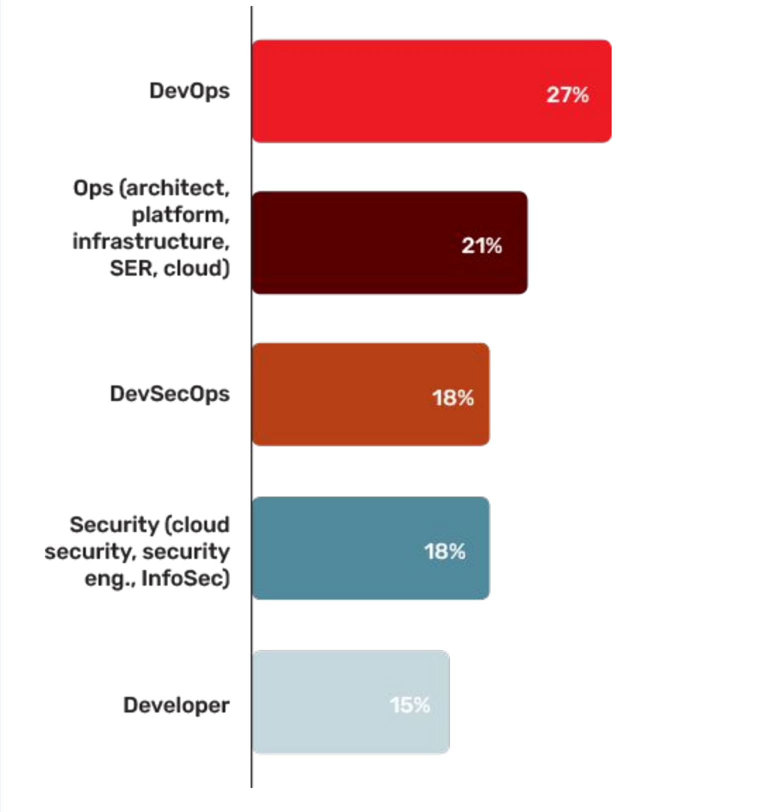
Why are you using K8s?



How often are you release cycles?



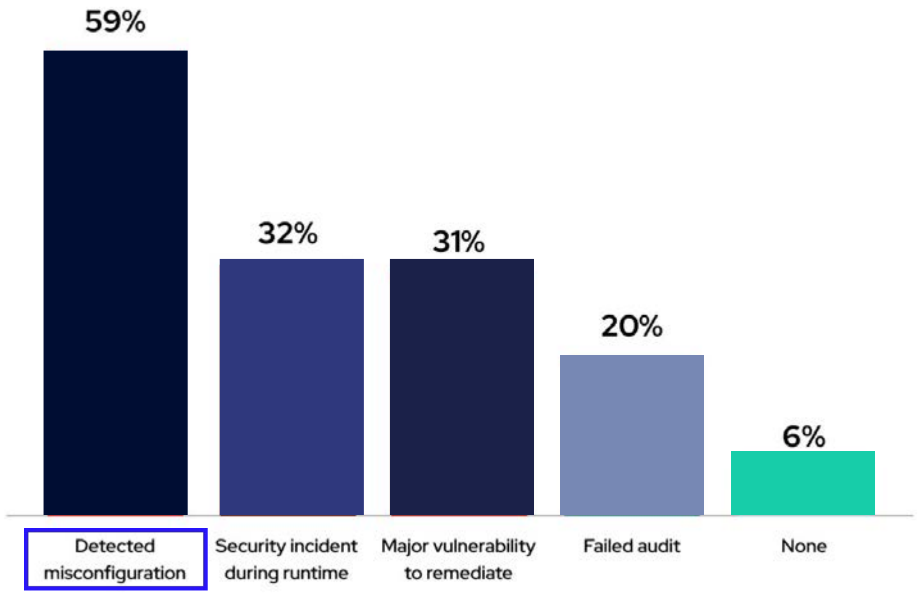
What role at your organization is most responsible for container and Kubernetes security?



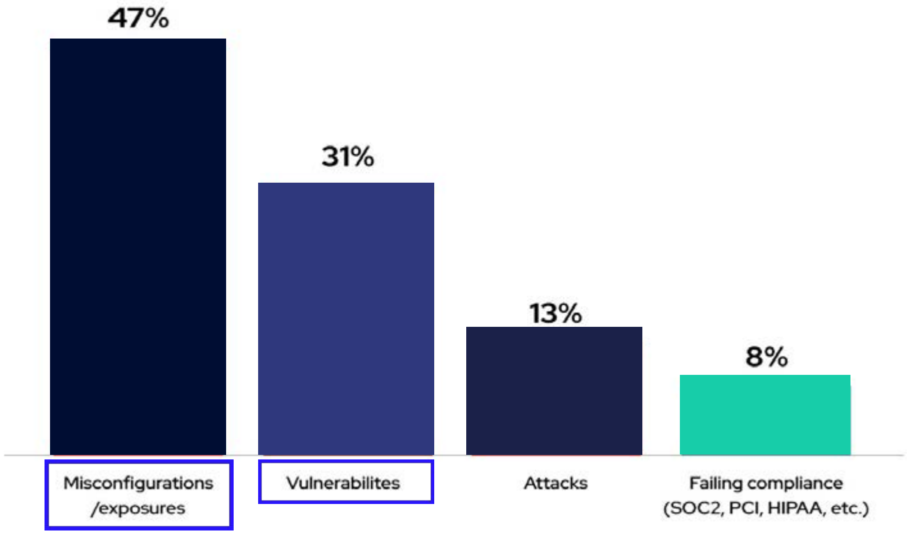
Why do we need K8s security?

Through 2025, more than 99% of cloud breaches will have a root cause of customer misconfigurations or mistakes

In the past 12 months, **what security incidents or issues** related to containers and/or Kubernetes have you experienced?



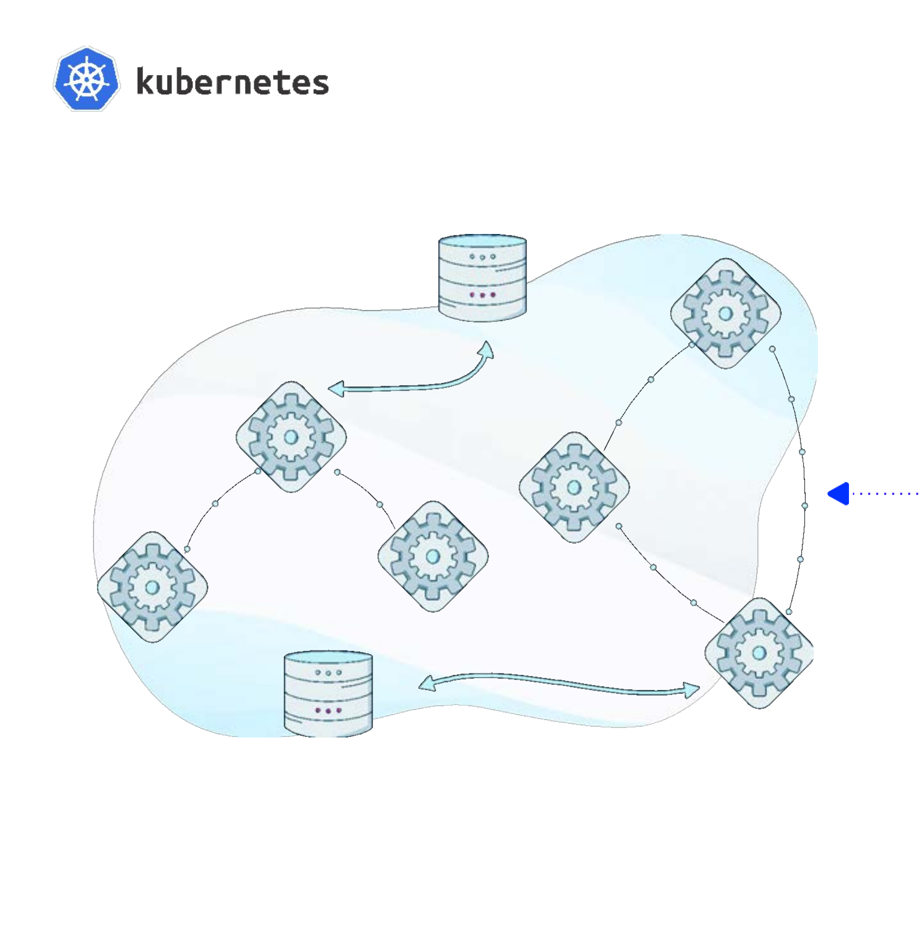
Of the following risks, **which one are you most worried about** for your container and Kubernetes environments?



Protecting Kubernetes – two main paradigms to apply

K8S POSTURE MANAGEMENT

Find known Vulnerabilities
& Misconfigurations



K8S RUN TIME PROTECTION

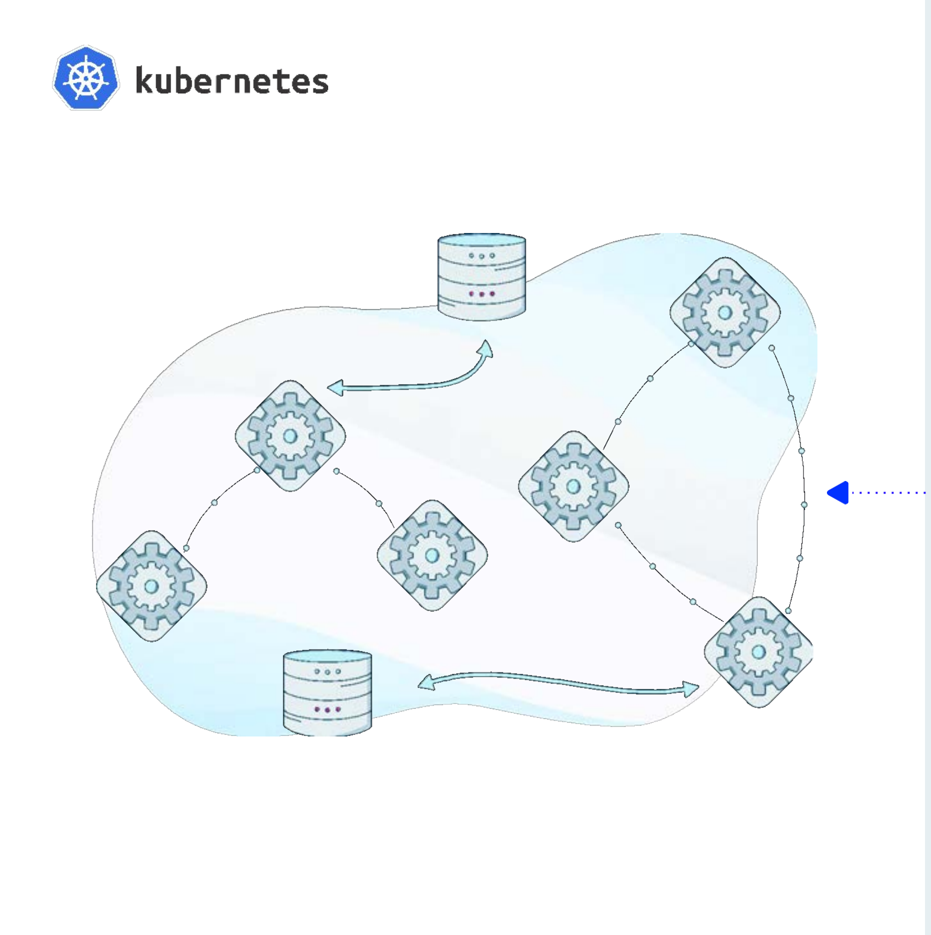
Anomaly Behavioral Analysis
and Network Segmentation

Protecting Kubernetes – two main paradigms to apply

K8S POSTURE MANAGEMENT

Find known Vulnerabilities
& Misconfigurations

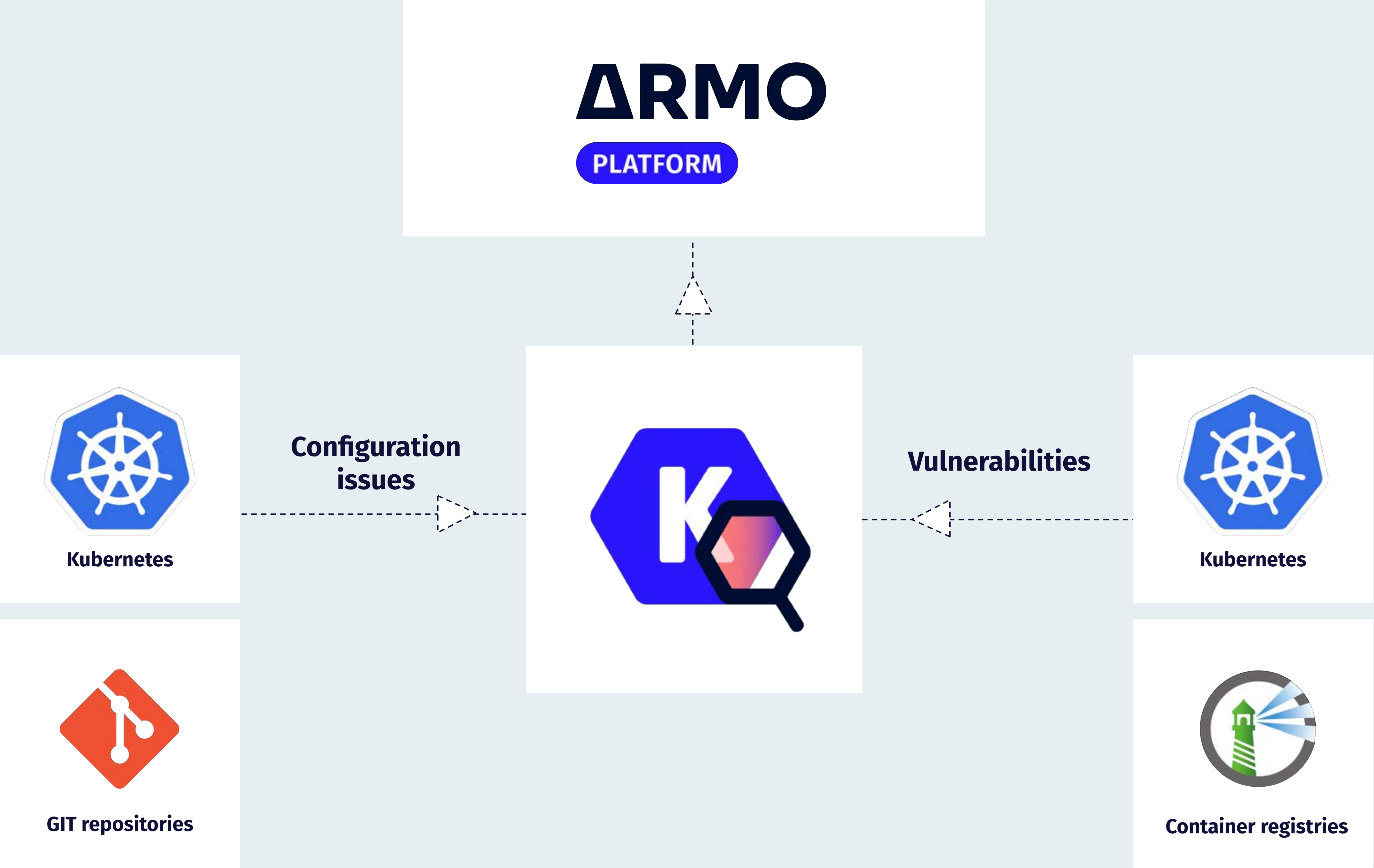
Focused on shrinking the attack surface, can be done early in the CI/CD



K8S RUN TIME PROTECTION

Anomaly Behavioral Analysis
and Network Segmentation

Focused on assuring detection/prevention of attacks when they happen



ARMO

PLATFORM

Configuration issues

Vulnerabilities

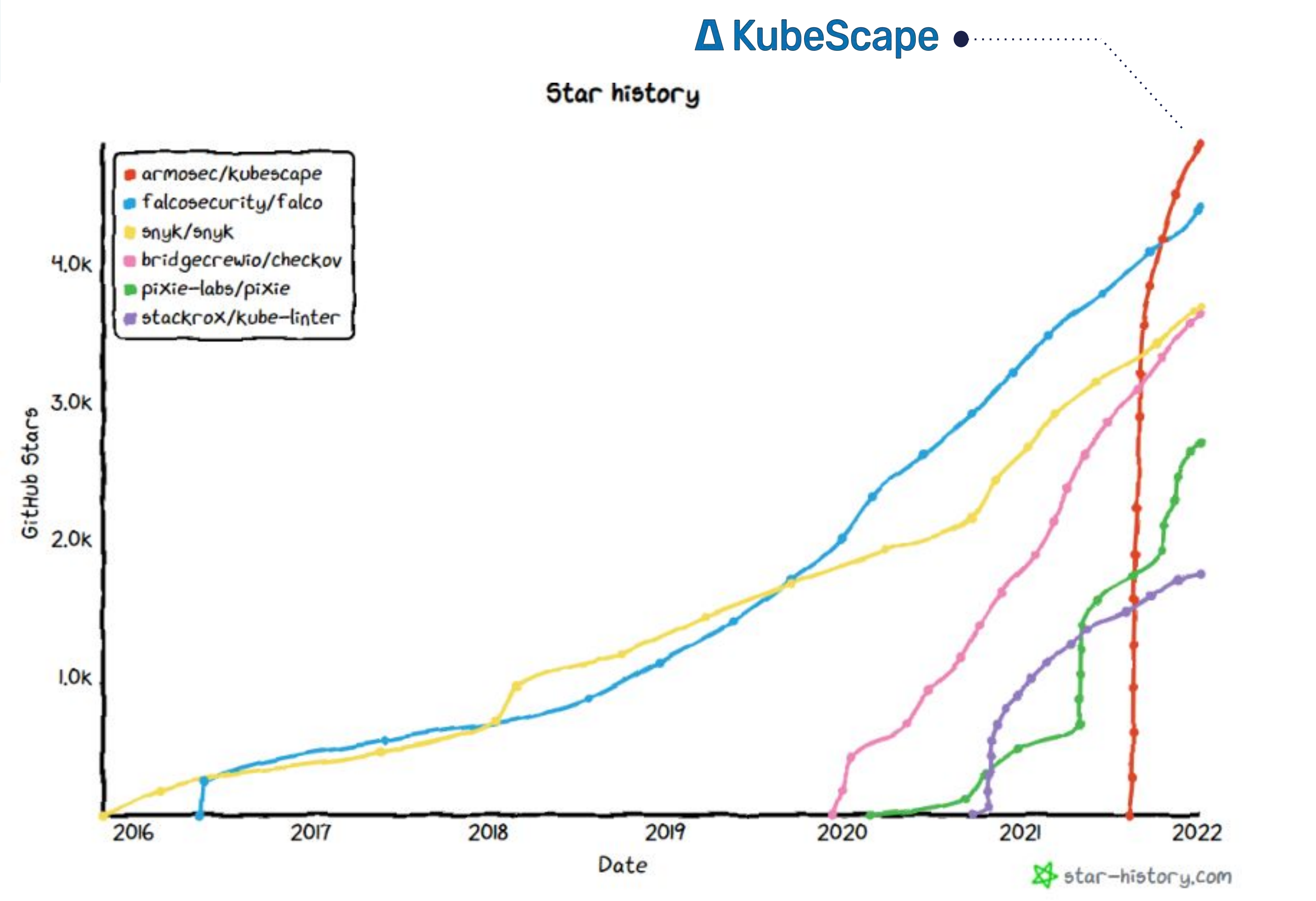
Kubernetes

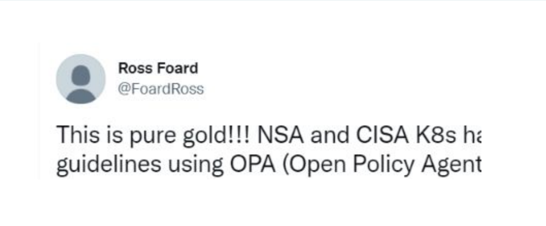
GIT repositories

Kubernetes

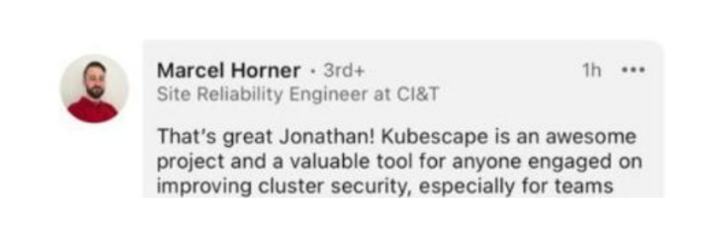
Container registries

Kubescape Is Becoming The Most Popular K8s Security Open-Source tool on Github

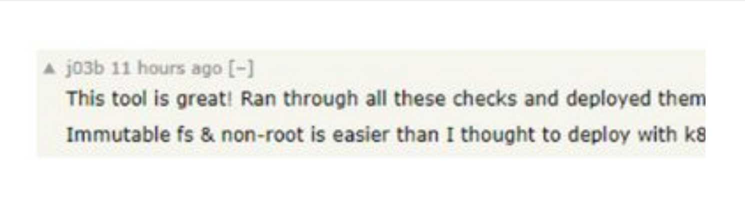



Ross Foard
@FoardRoss

This is pure gold!!! NSA and CISA K8s ha
guidelines using OPA (Open Policy Agent


Marcel Horner · 3rd+ 1h •••
Site Reliability Engineer at CI&T

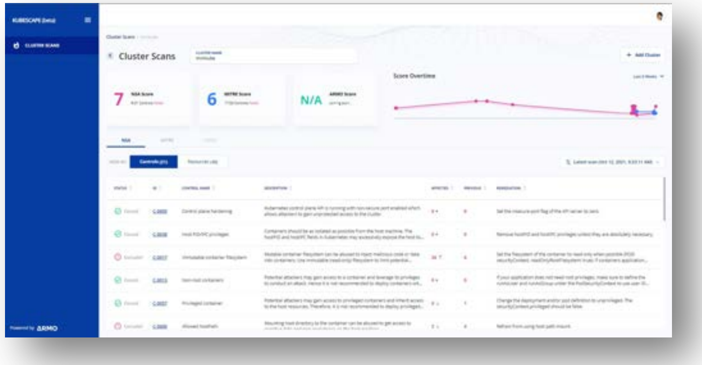
That's great Jonathan! Kubescape is an awesome project and a valuable tool for anyone engaged on improving cluster security, especially for teams


▲ j03b 11 hours ago [–]

This tool is great! Ran through all these checks and deployed them
Immutable fs & non-root is easier than I thought to deploy with k8

A Multidimensional Kubernetes single pane of glass

Risk analysis & Compliance



Cluster Scans

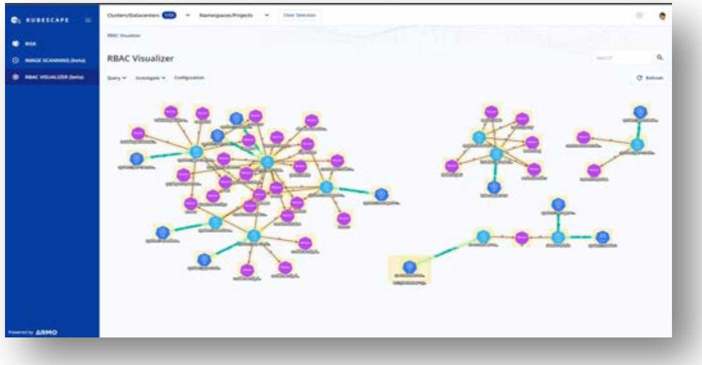
7

6

N/A

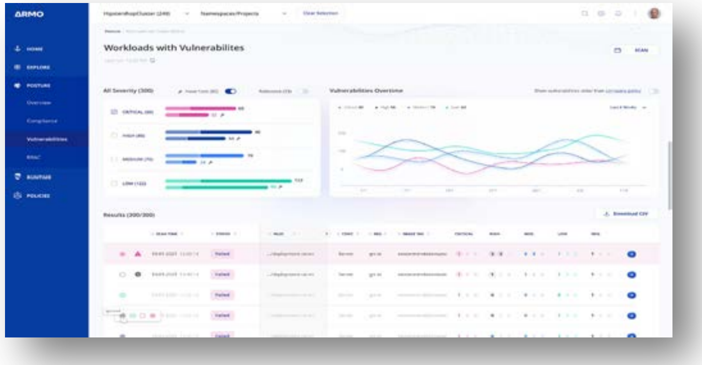
Score Overtime

RBAC Visualizer



RBAC Visualizer

Image scanning



Workloads with Vulnerabilities

Google Kubernetes Engine

Azure Kubernetes Service (AKS)

Amazon EKS

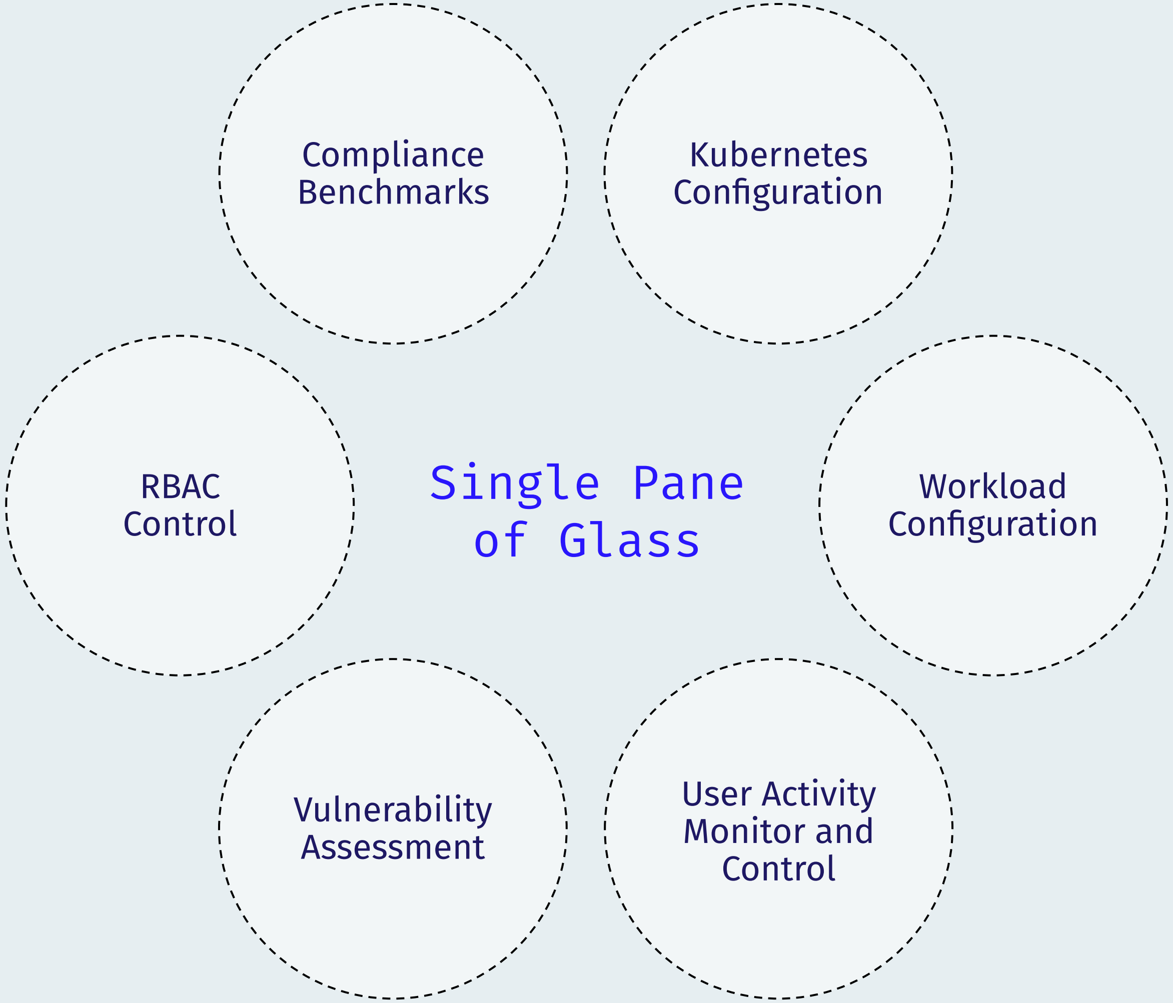
OPENSHIFT

Building Kubernetes Security Single Pane of Glass

- **Define and Enforce Best Practices**
  NSA, MITRE, K8s Best Practices, or create your own custom one

- **Identify and Prevent Drifts**
  Continuously, from CI/CD to Production

- **Continuous Env Tightening and attack surface reduction**
  Quick remediation, automatic recommendations, contextual insights



Single Pane of Glass

Compliance Benchmarks

Kubernetes Configuration

RBAC Control

Workload Configuration

Vulnerability Assessment

User Activity Monitor and Control

Dev To Production
Kubernetes Platform

Checkout our Roadmap on GitHub:
https://github.com/kubescape/kubescape/blob/master/docs/roadmap.md

#01 **Configuration**
Open Source.
Free Forever

#02 **Deployment**
Tiered Offering.
Free Tier

#03 **Production**
Tiered Offering.
Free Tier

Live feedback

Live feedback

Check early in the CI/CD

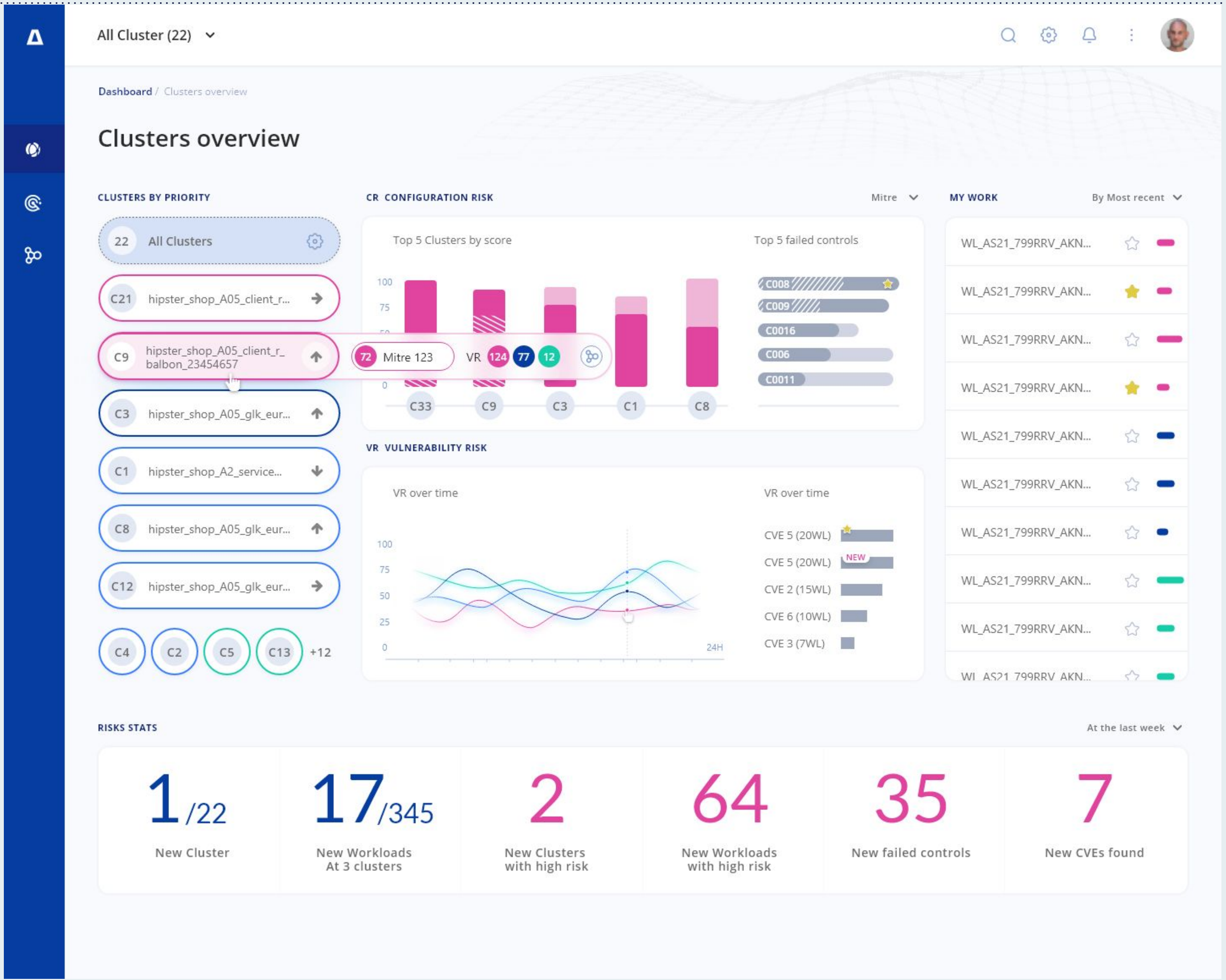
- On demand checks
- CI/CD embedding
- Yamls & cluster (from outside)
- Frameworks

Continues Posture control

- In cluster install
- Always on watching
- Vulnerability scanning
- Admission control
- Audit log analysis
- Live alerts
- Least privilege monitoring
- Native policy enforcement

Runtime Zero-Trust

- Deep observability
- Vulnerability relevancy
- Memory protection
- Secret protection
- Zero Trust network protection
- Identity based data protection
- Service Mesh interoperability



All Cluster (22)

Dashboard / Clusters overview

Clusters overview

CLUSTERS BY PRIORITY

- 22 All Clusters
- C21 hipster_shop_A05_client_r...
- C9 hipster_shop_A05_client_r_balbon_23454657
- C3 hipster_shop_A05_glk_eur...
- C1 hipster_shop_A2_service...
- C8 hipster_shop_A05_glk_eur...
- C12 hipster_shop_A05_glk_eur...
- C4 C2 C5 C13 +12

72 Mitre 123 VR 124 77 12

CR CONFIGURATION RISK Mitre

VR VULNERABILITY RISK

MY WORK By Most recent

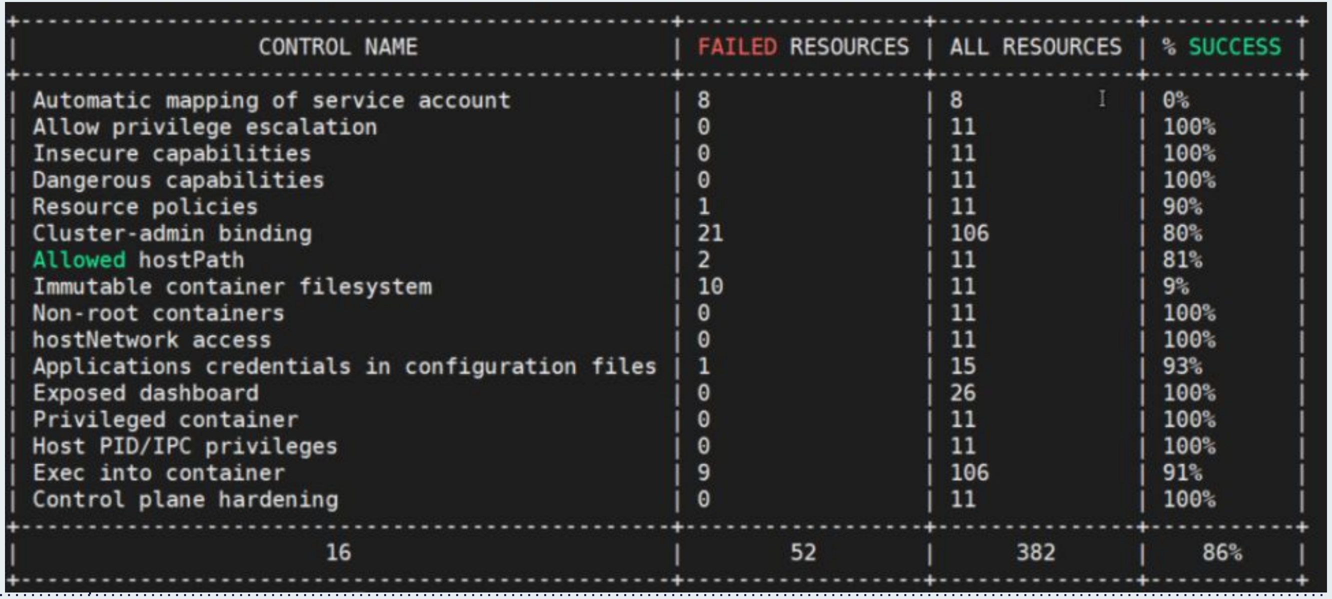
- WL_AS21_799RRV_AKN...
- WL_AS21_799RRV_AKN...
- WL_AS21_799RRV_AKN...
- WL_AS21_799RRV_AKN...
- WL_AS21_799RRV_AKN...
- WL_AS21_799RRV_AKN...
- WL_AS21_799RRV_AKN...
- WL_AS21_799RRV_AKN...
- WL_AS21_799RRV_AKN...
- WL_AS21_799RRV_AKN...

RISKS STATS At the last week

1/22	17/345	2	64	35	7
New Cluster	New Workloads At 3 clusters	New Clusters with high risk	New Workloads with high risk	New failed controls	New CVEs found

Single Pane of Glass Dashboard

3 Min to get your first scan, no in-cluster installation, read only privileges



```
+-----------------------------------------------------+------------------+---------------+-----------+
|                    CONTROL NAME                     | FAILED RESOURCES | ALL RESOURCES | % SUCCESS |
+-----------------------------------------------------+------------------+---------------+-----------+
| Automatic mapping of service account                | 8                | 8             | 0%        |
| Allow privilege escalation                          | 0                | 11            | 100%      |
| Insecure capabilities                               | 0                | 11            | 100%      |
| Dangerous capabilities                              | 0                | 11            | 100%      |
| Resource policies                                   | 1                | 11            | 90%       |
| Cluster-admin binding                               | 21               | 106           | 80%       |
| Allowed hostPath                                    | 2                | 11            | 81%       |
| Immutable container filesystem                      | 10               | 11            | 9%        |
| Non-root containers                                 | 0                | 11            | 100%      |
| hostNetwork access                                  | 0                | 11            | 100%      |
| Applications credentials in configuration files     | 1                | 15            | 93%       |
| Exposed dashboard                                   | 0                | 26            | 100%      |
| Privileged container                                | 0                | 11            | 100%      |
| Host PID/IPC privileges                             | 0                | 11            | 100%      |
| Exec into container                                 | 9                | 106           | 91%       |
| Control plane hardening                             | 0                | 11            | 100%      |
+-----------------------------------------------------+------------------+---------------+-----------+
|                         16                          |        52        |      382      |    86%    |
+-----------------------------------------------------+------------------+---------------+-----------+
```

Less than 3 Min to get your first scan

API Based with read-only Privileges

Get Started: GitHub https://github.com/armosec/kubescape

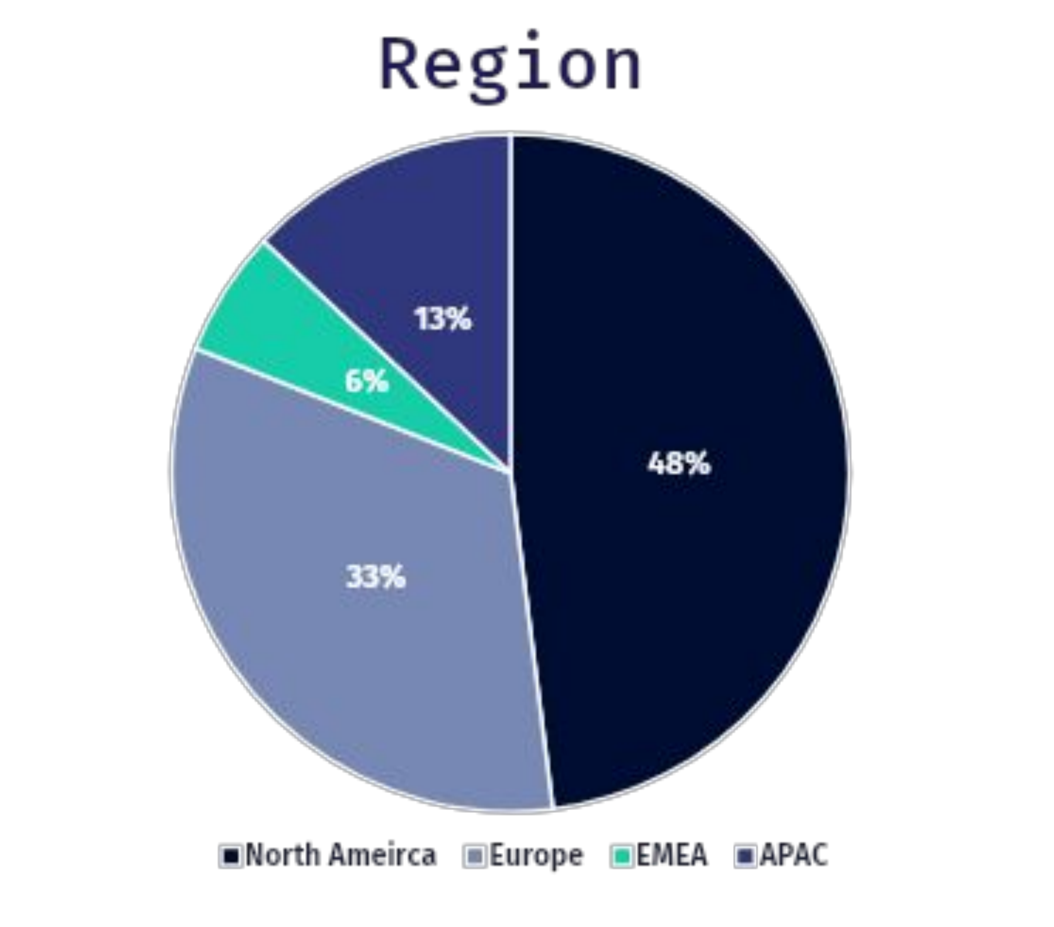
Kubescape

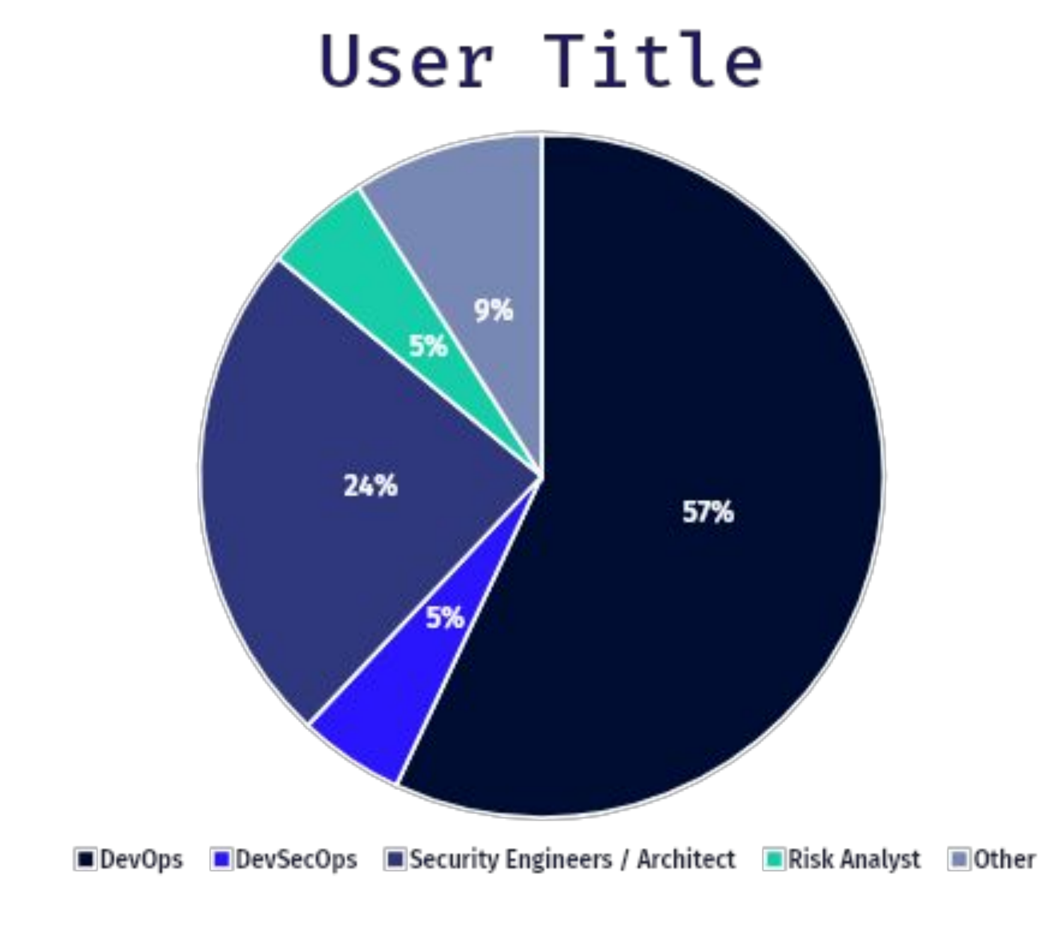
by ARMO

Research Results:

What have we learned from scanning over {10,000 K8s clusters}

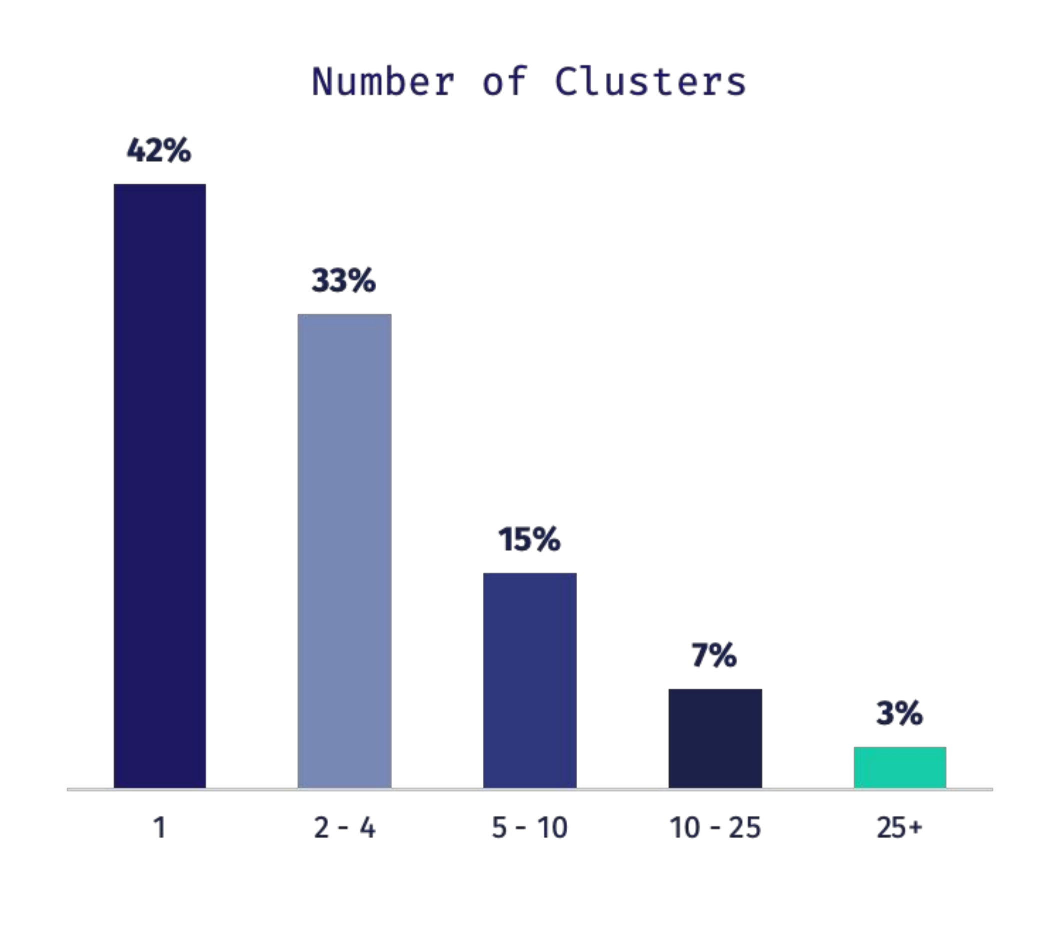
Data-Set Overview

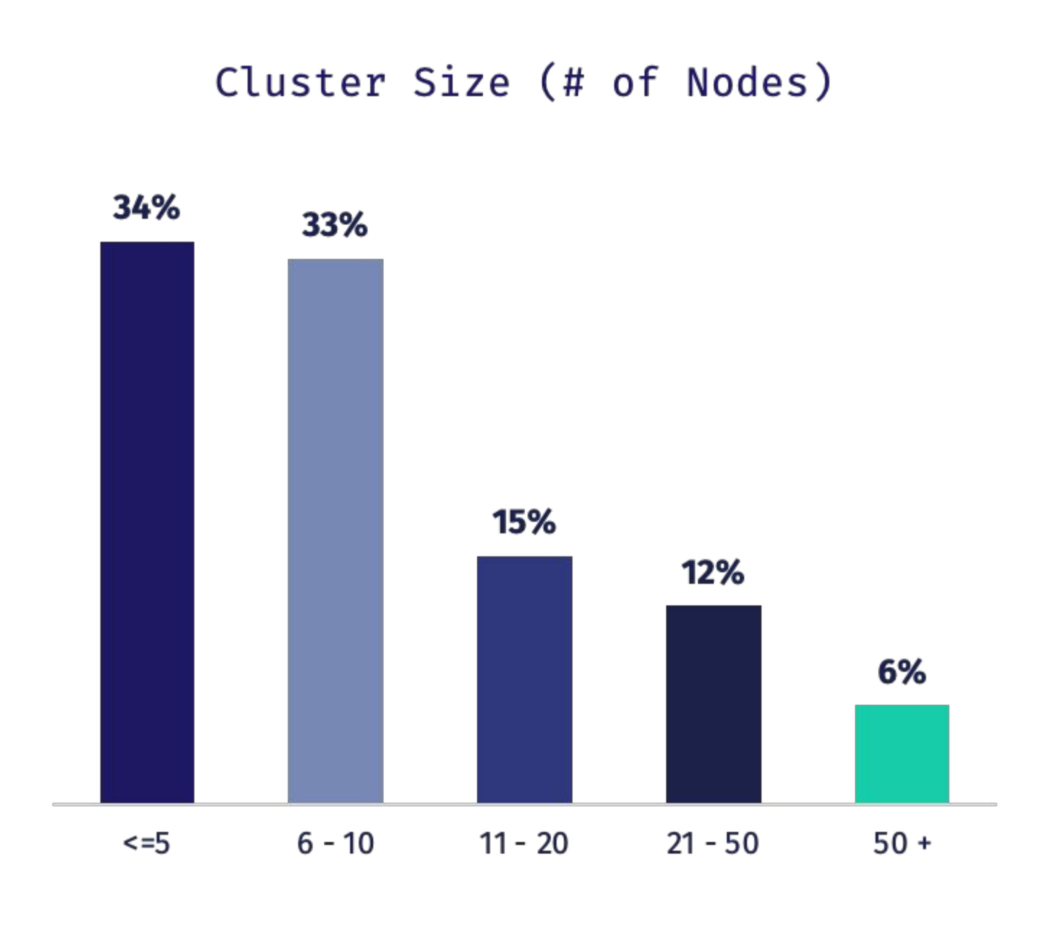




Data-Set Overview

Kubescape by ARMO

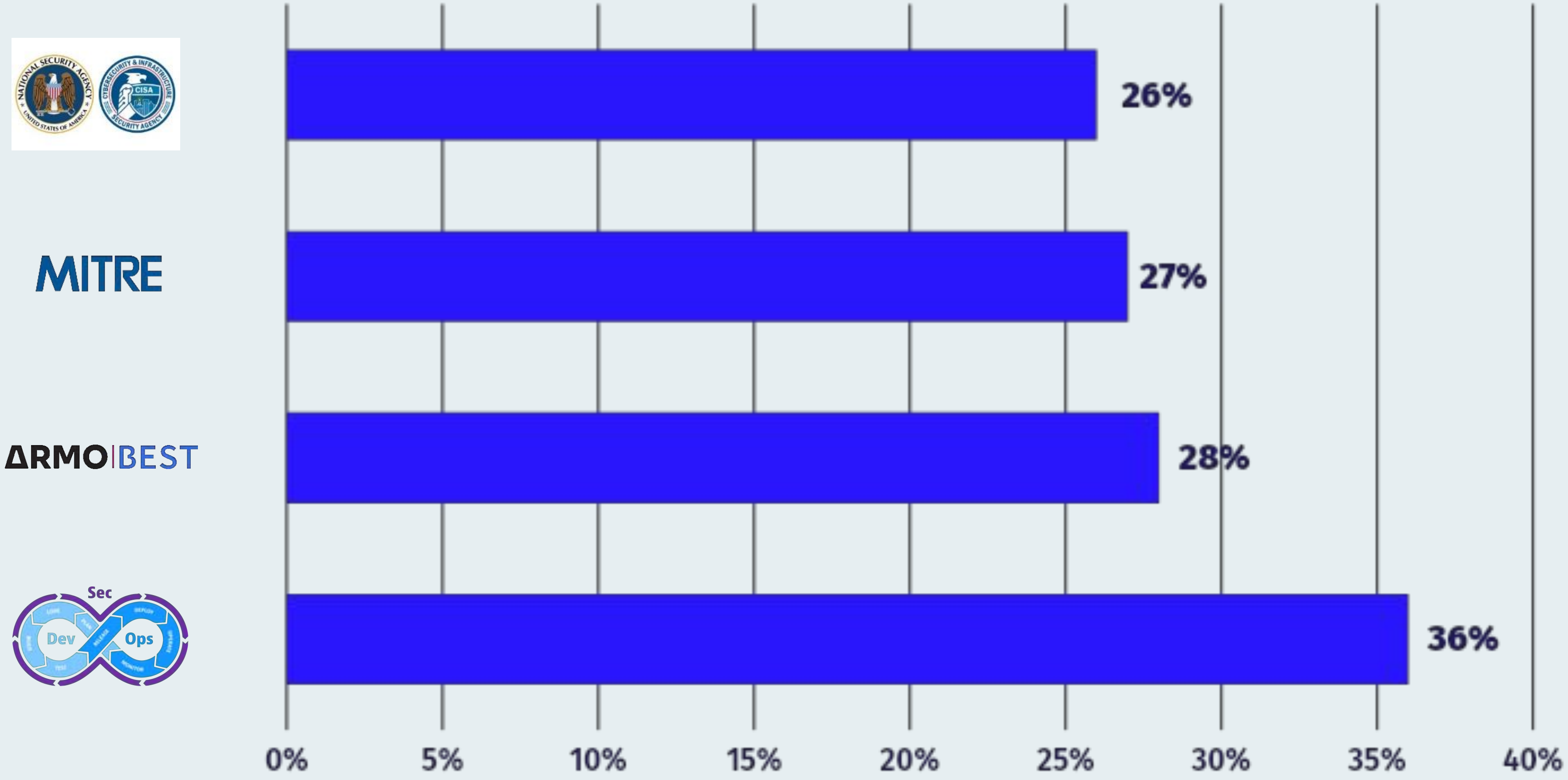




What have we learned from scanning over 10,000 clusters?

Kubescape by ARMO

Average risk score per framework



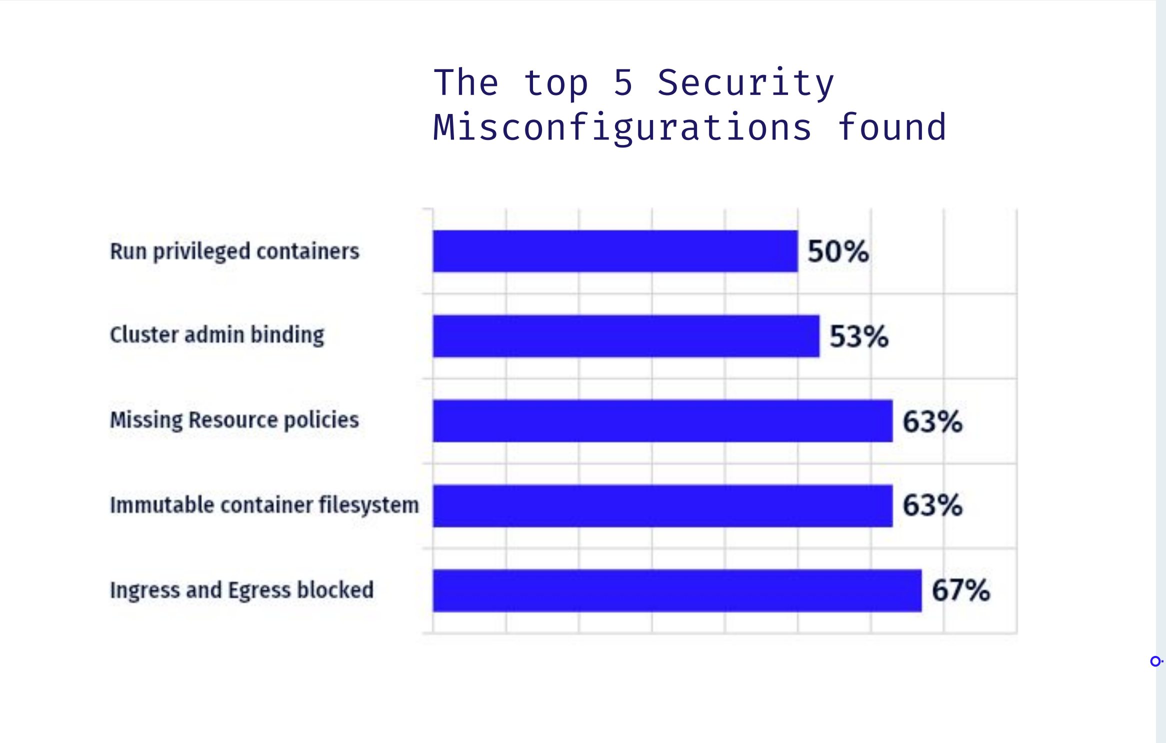
BEST PRACTICE

KEEP YOUR SCORE BELOW 30

Risk Score above 60+ puts you in the worsts 5%

Risk Score below 10 puts you in the best 10%

What have we learned from scanning over 10,000 clusters?

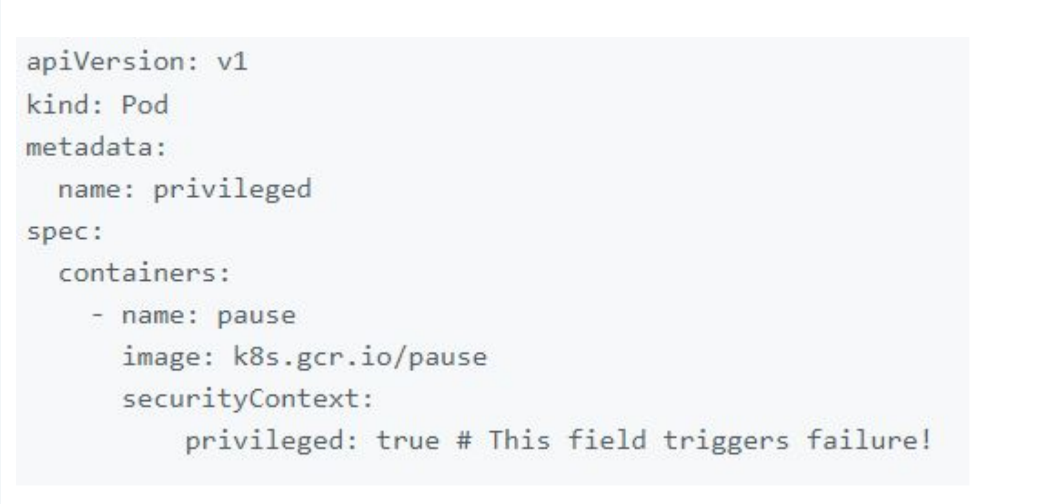


100% of clusters had misconfiguration in them

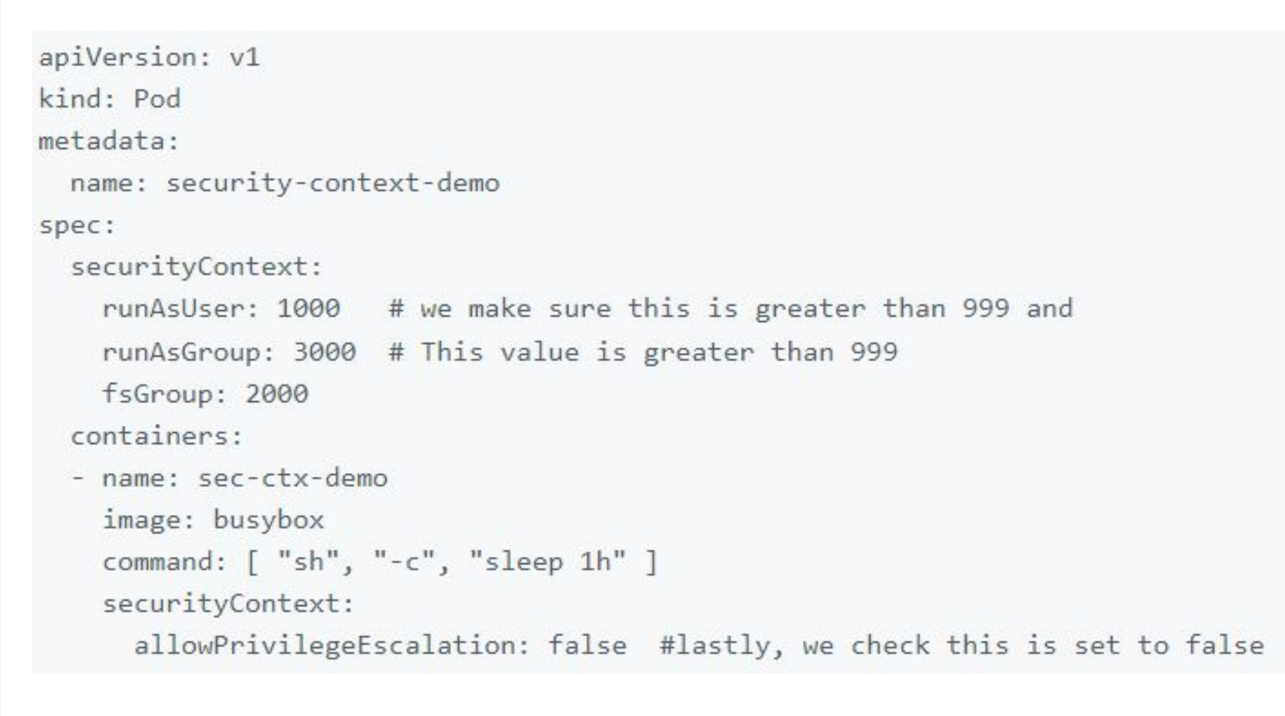
65% of cluster had at least one high severity misconfiguration

50% of clusters had 14 or more failed controls

Stop running Privileged Containers...



```yaml
apiVersion: v1
kind: Pod
metadata:
  name: privileged
spec:
  containers:
    - name: pause
      image: k8s.gcr.io/pause
      securityContext:
          privileged: true # This field triggers failure!
```

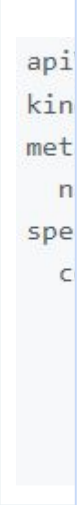


```yaml
apiVersion: v1
kind: Pod
metadata:
  name: security-context-demo
spec:
  securityContext:
    runAsUser: 1000   # we make sure this is greater than 999 and
    runAsGroup: 3000  # This value is greater than 999
    fsGroup: 2000
  containers:
  - name: sec-ctx-demo
    image: busybox
    command: [ "sh", "-c", "sleep 1h" ]
    securityContext:
      allowPrivilegeEscalation: false  #lastly, we check this is set to false
```

Basically cancels the entire container isolation concept

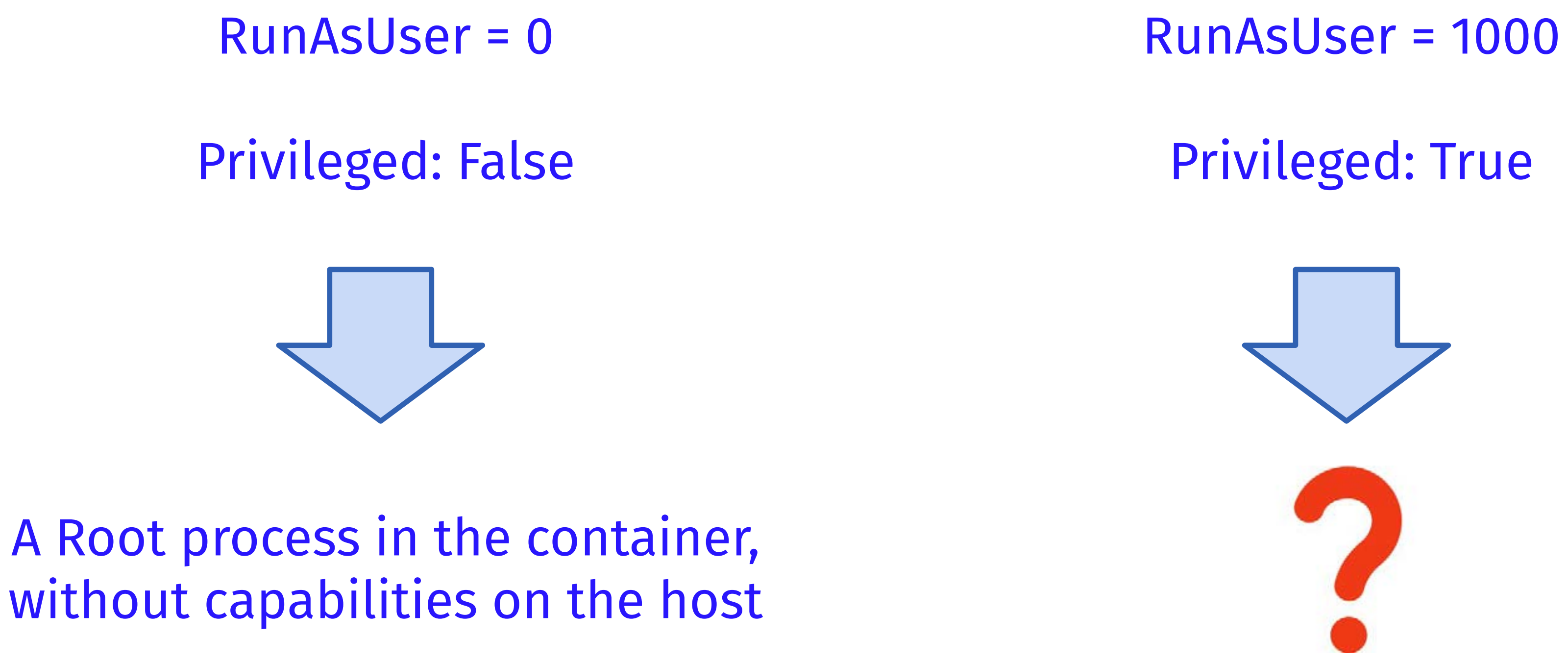
Leaves you exposed to kernel vulnerabilities

Stop running Privileged Containers...



```
api
kin
met
  n
spe
  c
```

<<POP QUIZ>>



RunAsUser = 0

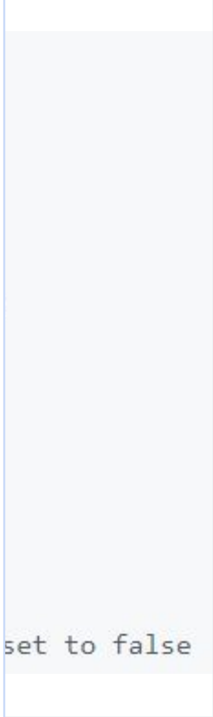
Privileged: False

A Root process in the container, without capabilities on the host

RunAsUser = 1000

Privileged: True

?

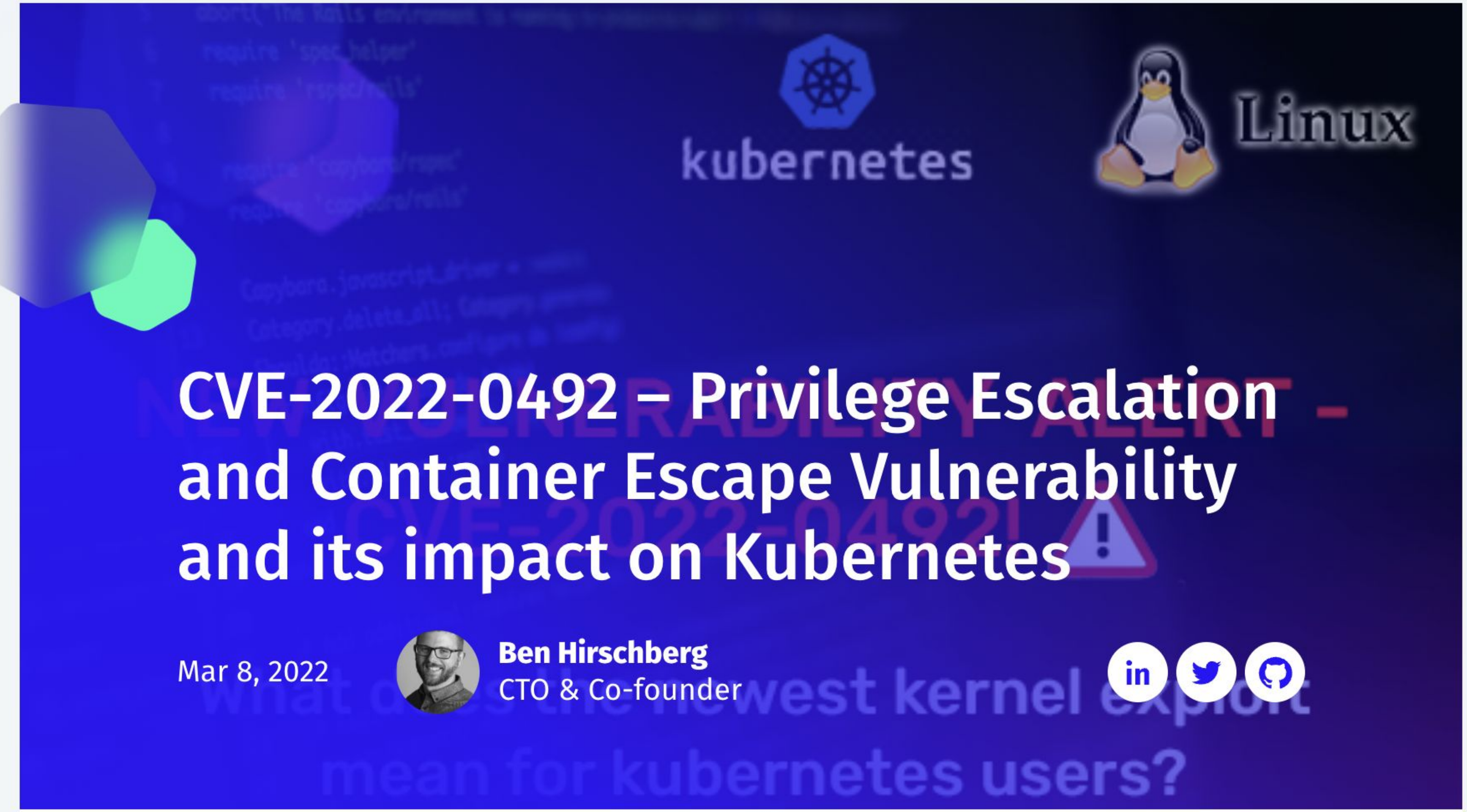


```
set to false
```

container isolation concept

vulnerabilities

Stop running Privileged Containers...



kubernetes
Linux
NEW VULNERABILITY ALERT - CVE-2022-0492!
What does the newest kernel exploit mean for kubernetes users?

CVE-2022-0492 – Privilege Escalation and Container Escape Vulnerability and its impact on Kubernetes

Mar 8, 2022

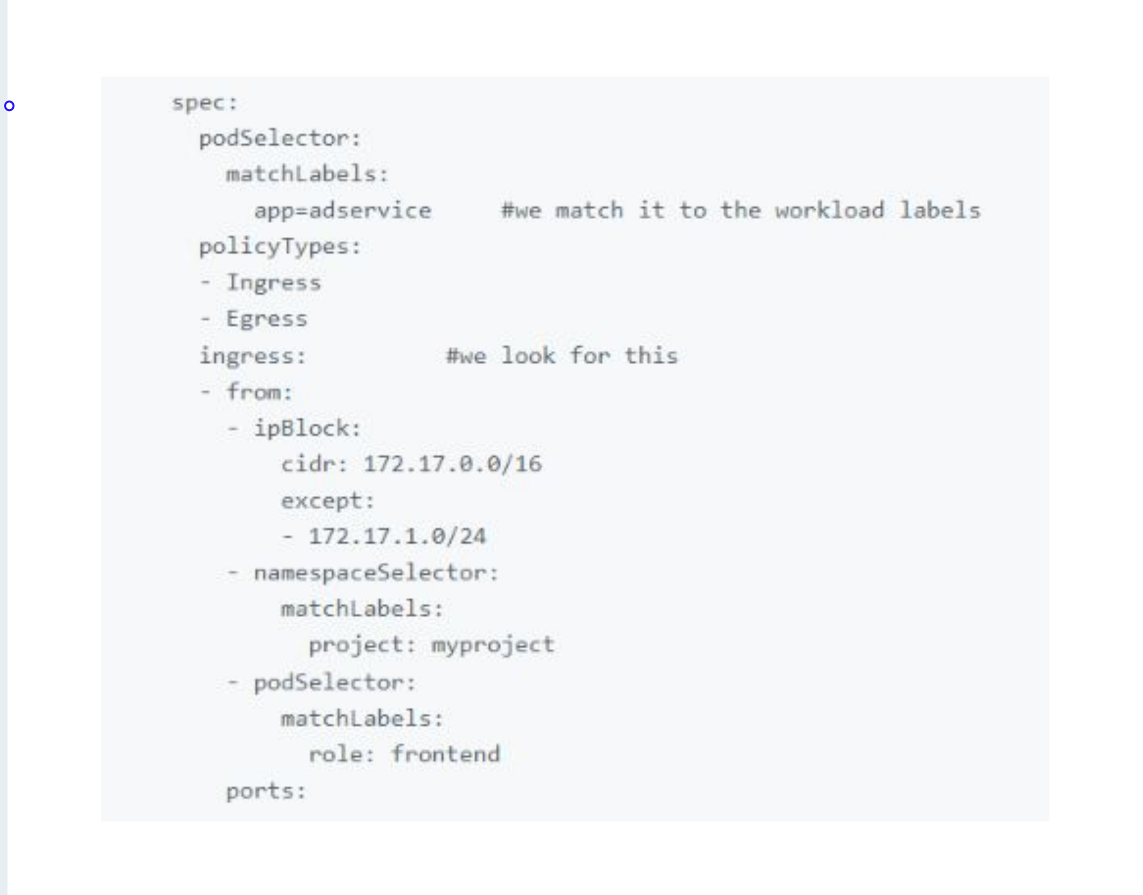
Ben Hirschberg
CTO & Co-founder

What have we learned from scanning over 10,000 clusters?

Kubescape by ARMO

63%

of clusters had workloads exposed outside the cluster without proper ingress blocked

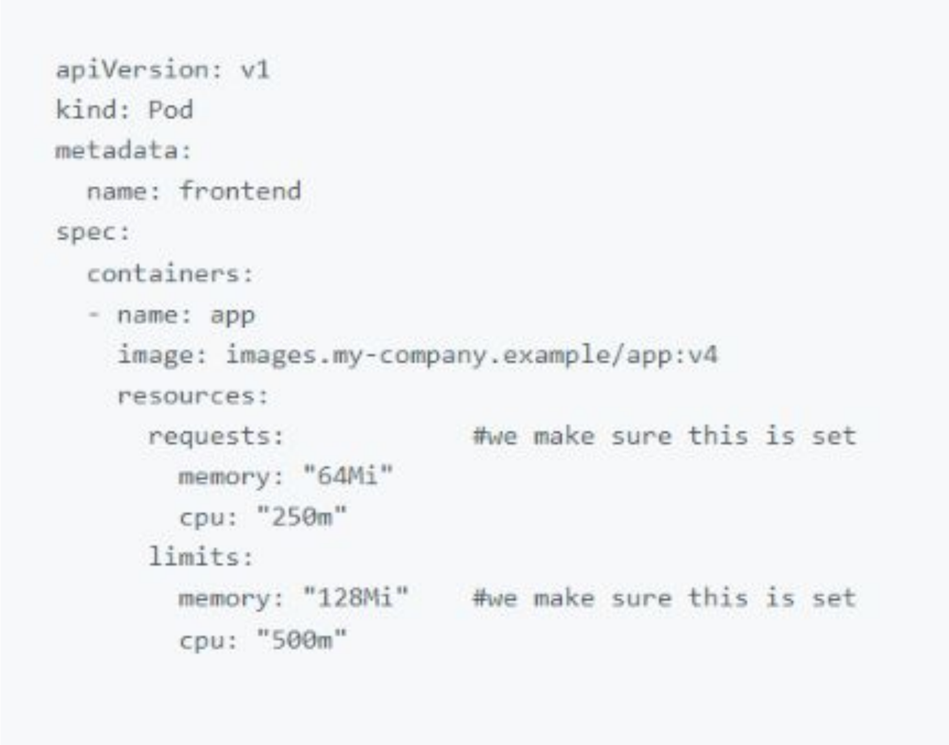


```
spec:
  podSelector:
    matchLabels:
      app=adservice     #we match it to the workload labels
  policyTypes:
  - Ingress
  - Egress
  ingress:          #we look for this
  - from:
    - ipBlock:
        cidr: 172.17.0.0/16
        except:
        - 172.17.1.0/24
    - namespaceSelector:
        matchLabels:
          project: myproject
    - podSelector:
        matchLabels:
          role: frontend
    ports:
```

What have we learned from scanning over 10,000 clusters?

63%

of clusters had workloads without proper resource limitations

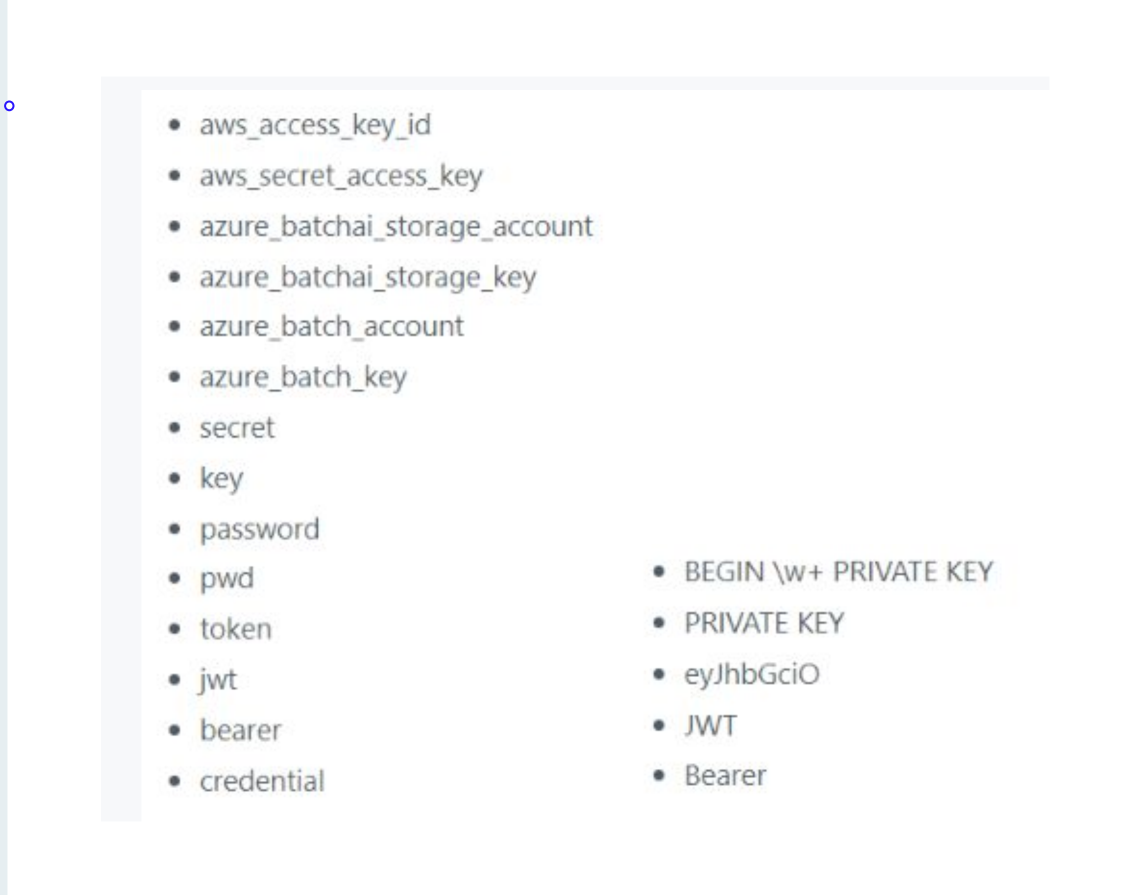


```
apiVersion: v1
kind: Pod
metadata:
  name: frontend
spec:
  containers:
  - name: app
    image: images.my-company.example/app:v4
    resources:
      requests:            #we make sure this is set
        memory: "64Mi"
        cpu: "250m"
      limits:
        memory: "128Mi"    #we make sure this is set
        cpu: "500m"
```

What have we learned from scanning over 10,000 clusters?

37%

of clusters had applications with credentials in configuration files



- aws_access_key_id
- aws_secret_access_key
- azure_batchai_storage_account
- azure_batchai_storage_key
- azure_batch_account
- azure_batch_key
- secret
- key
- password
- pwd
- token
- jwt
- bearer
- credential

- BEGIN \w+ PRIVATE KEY
- PRIVATE KEY
- eyJhbGciO
- JWT
- Bearer

What have we learned from scanning over 10,000 clusters?

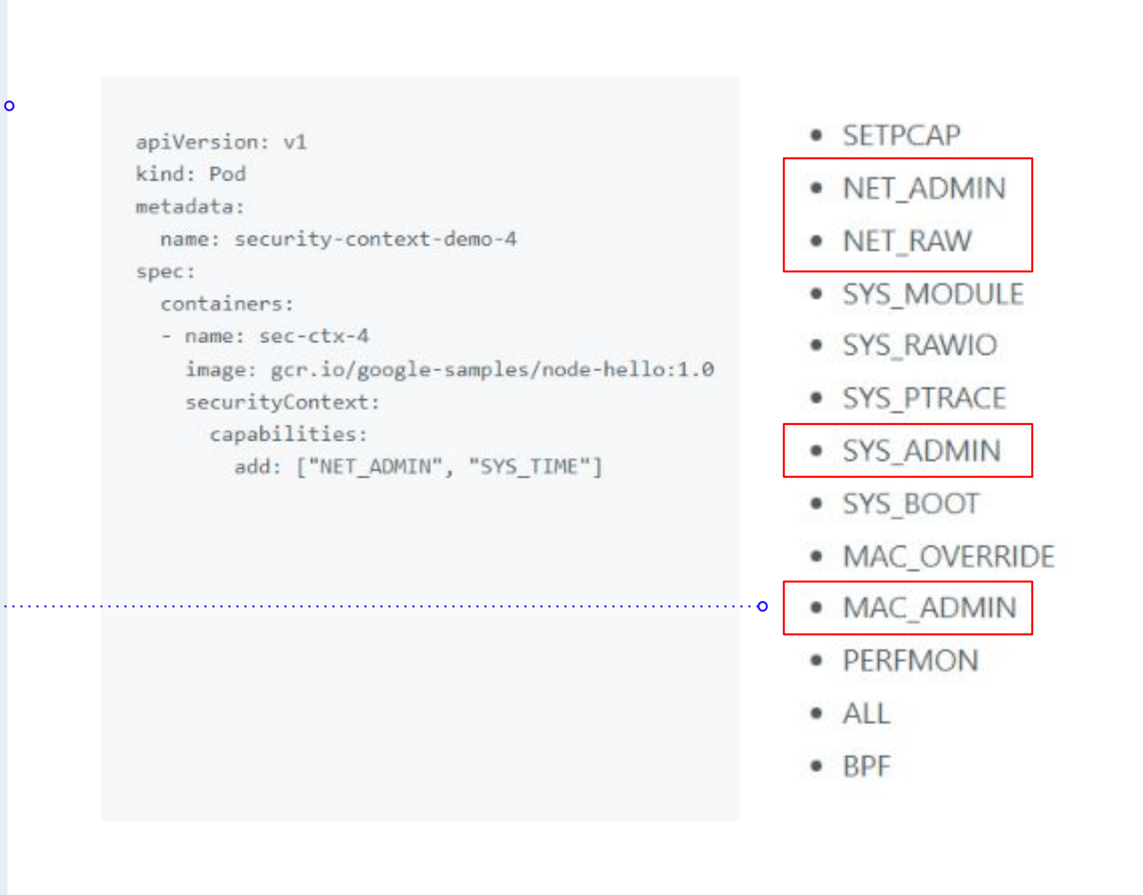
Kubescape by ARMO

23%

of clusters had applications runing with dangerous Linux capabilities

35%

of clusters had workloads running with insecure capabilities

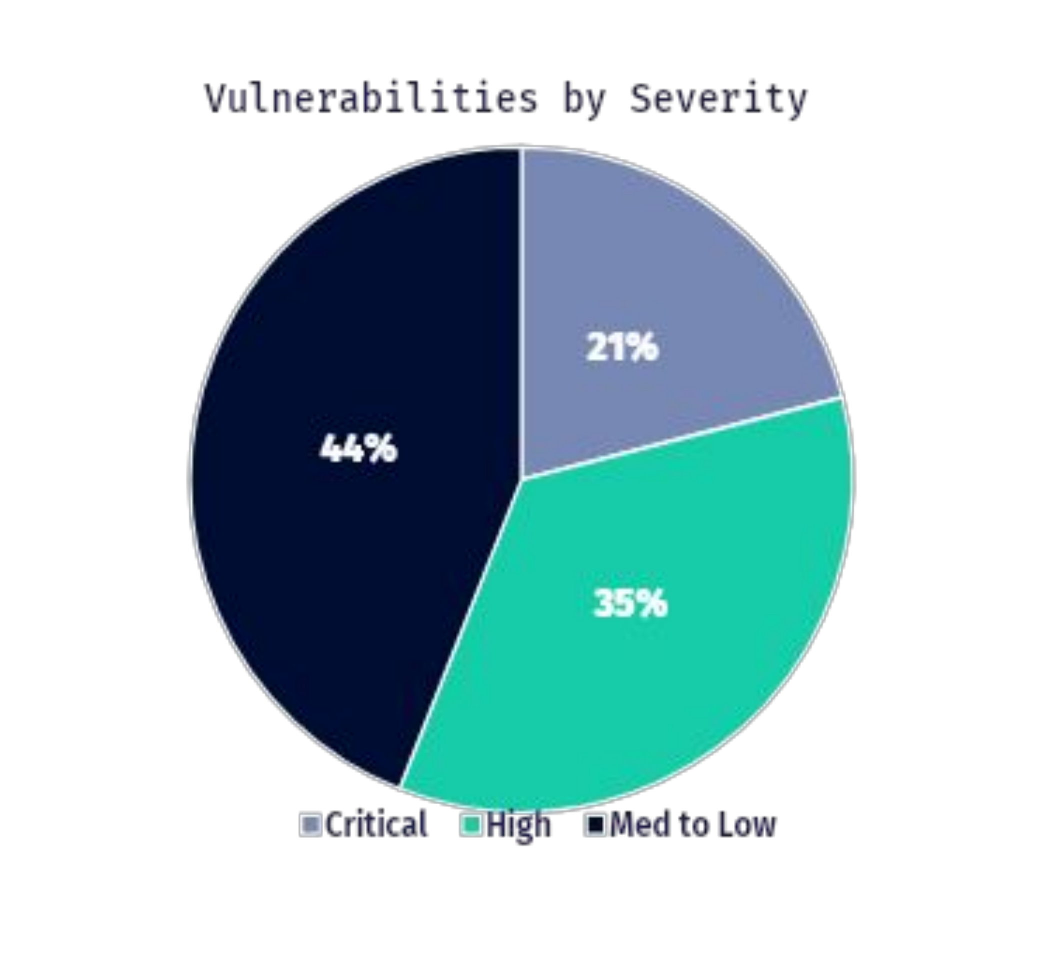


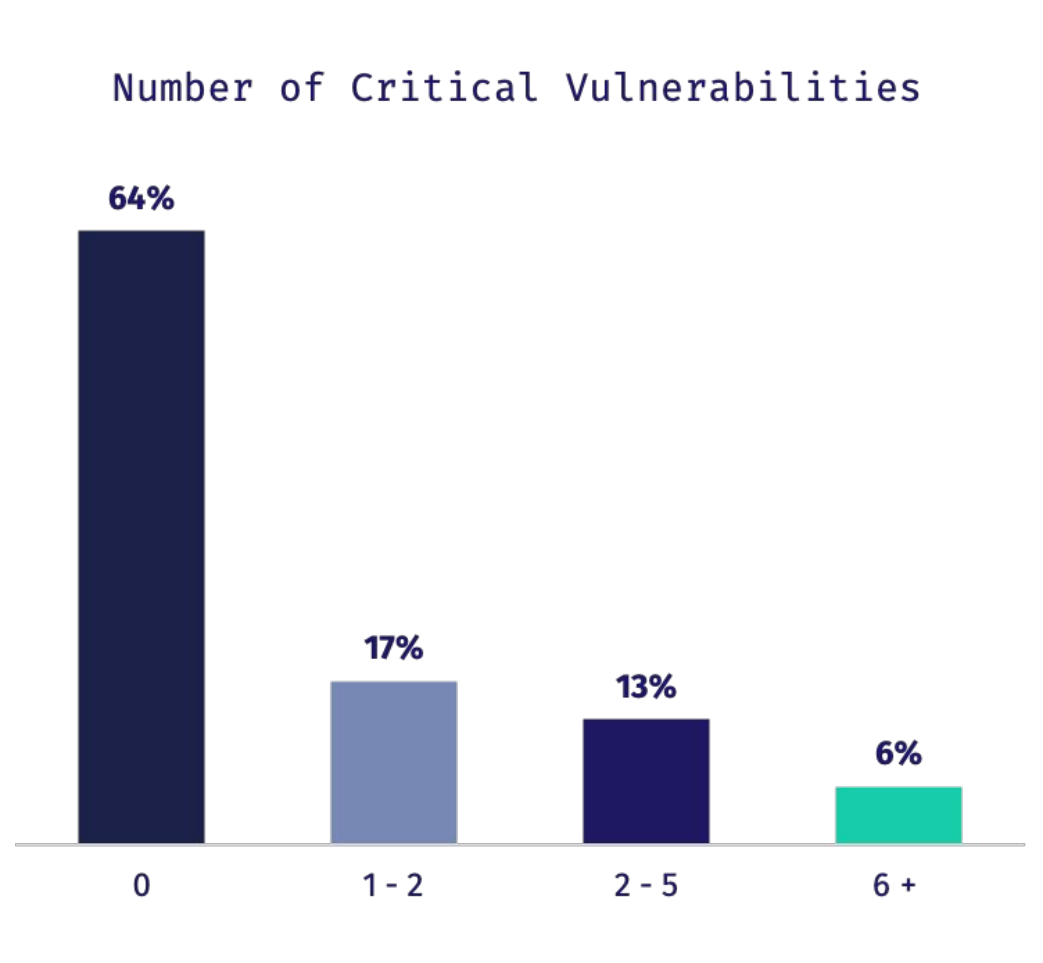
```
apiVersion: v1
kind: Pod
metadata:
  name: security-context-demo-4
spec:
  containers:
  - name: sec-ctx-4
    image: gcr.io/google-samples/node-hello:1.0
    securityContext:
      capabilities:
        add: ["NET_ADMIN", "SYS_TIME"]
```

- SETPCAP
- NET_ADMIN
- NET_RAW
- SYS_MODULE
- SYS_RAWIO
- SYS_PTRACE
- SYS_ADMIN
- SYS_BOOT
- MAC_OVERRIDE
- MAC_ADMIN
- PERFMON
- ALL
- BPF

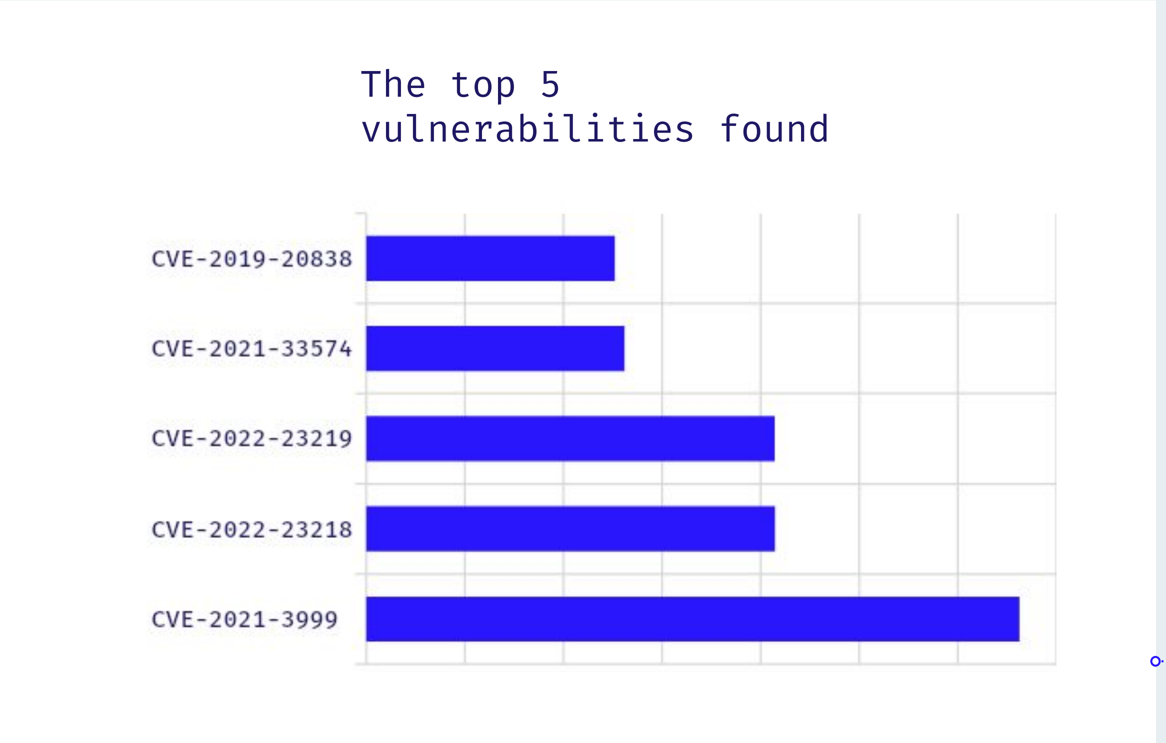
What have we learned from scanning over 10,000 clusters?

Kubescape by ARMO





What have we learned from scanning over 10,000 clusters?



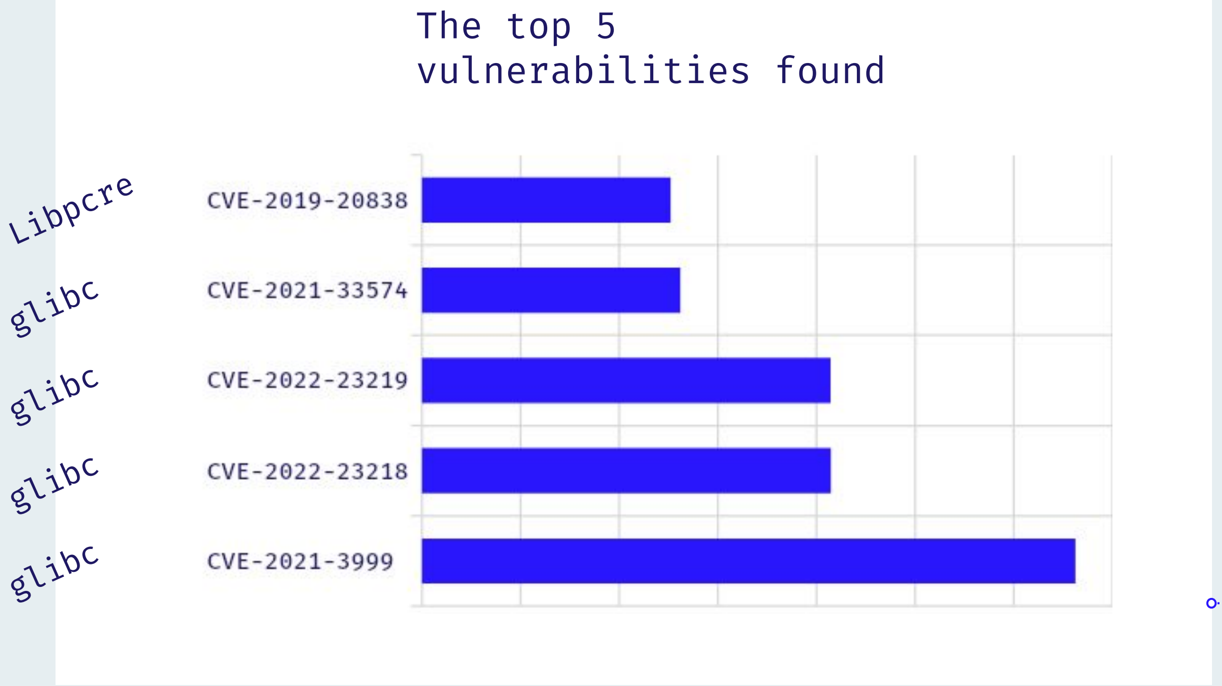
63% of the containers had one or more vulnerabilities

46% of containers had one or more critical vulnerabilities

53% of containers had one or more RCE vulnerabilities

What have we learned from scanning over 10,000 clusters?

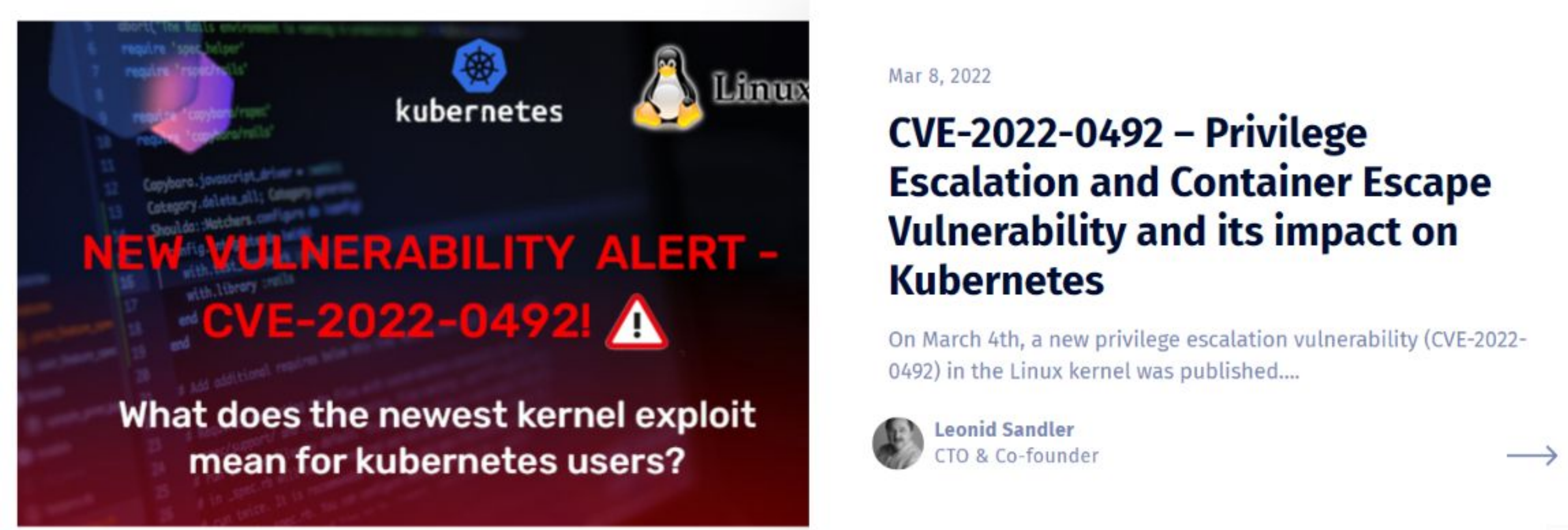
Kubescape by ARMO



- 63% of the containers had one or more vulnerabilities
- 46% of containers had one or more critical vulnerabilities
- 53% of containers had one or more RCE vulnerabilities

What have we learned from scanning over 10,000 clusters?

Kubescape
by ARMO



kubernetes

Linux

NEW VULNERABILITY ALERT - CVE-2022-0492! ⚠

What does the newest kernel exploit mean for kubernetes users?

Mar 8, 2022

CVE-2022-0492 – Privilege Escalation and Container Escape Vulnerability and its impact on Kubernetes

On March 4th, a new privilege escalation vulnerability (CVE-2022-0492) in the Linux kernel was published....

Leonid Sandler
CTO & Co-founder

→

Control – Allow Privileged escalation - 0016
Control – Deny run as root – 0013
Control – Insecure capabilities - (CAP_DAC_OVERRIDE)
But . . We also added a specific control . . .

Closing thoughts

- Most of companies already running on k8s clusters
- Security perspective seems like a big “pain” that no-one wants/can handle
- DevOps are the new security personas in k8s based organizations
- we are overwhelmed with vulnerabilities (to be continued…)

Thank you_

ARMO