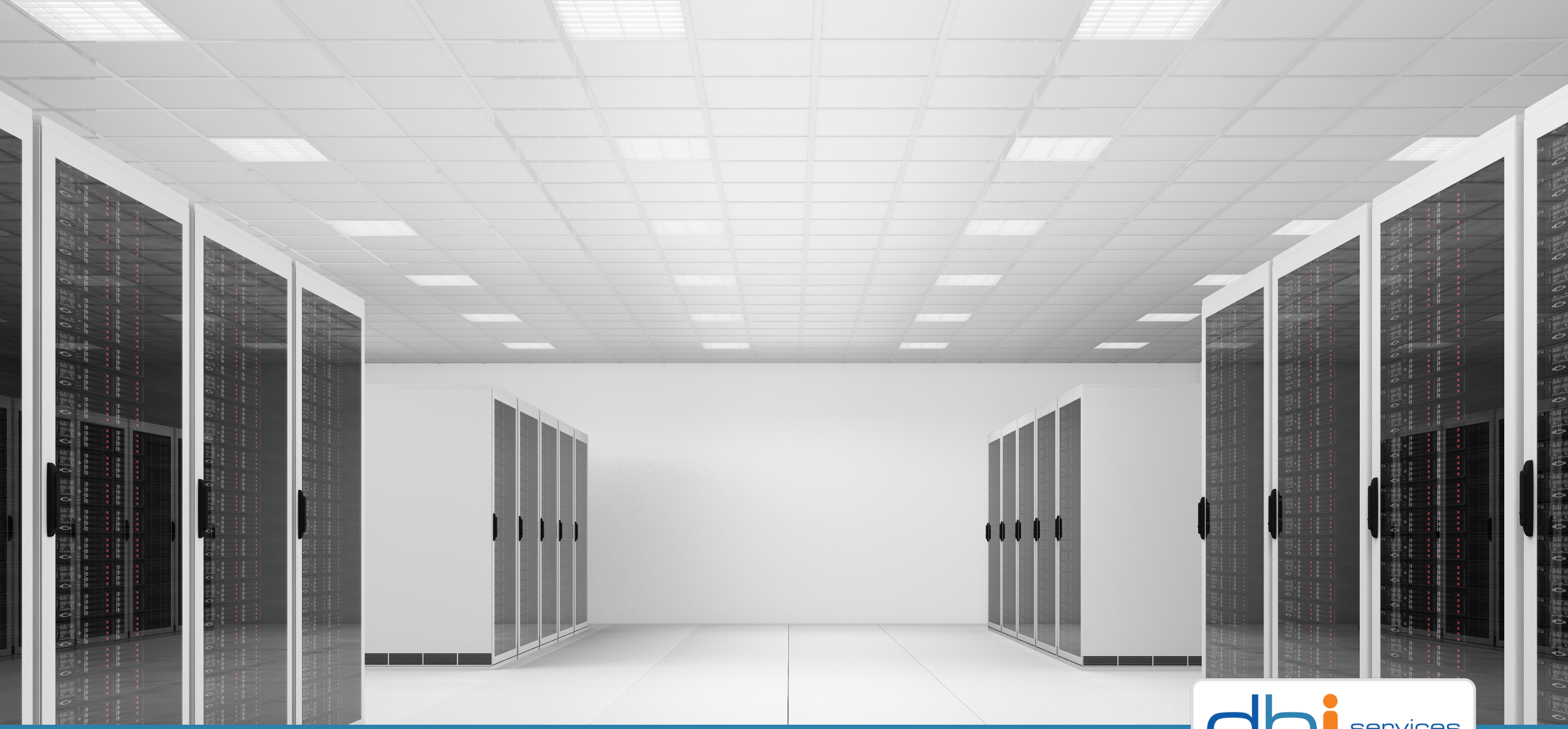


dbi services by Sequotech

Oh No! Our K8s cluster has been compromised

Benoît Entzmann
Consultant

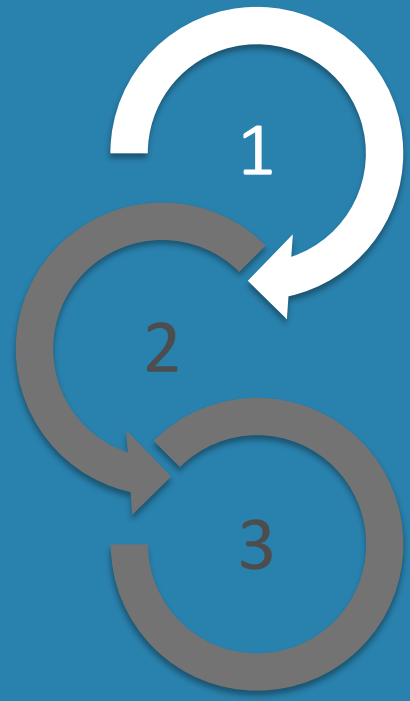
Chay Té
Consultant

Agenda

1. Let’s talk about security
2. Real use cases
3. More about security

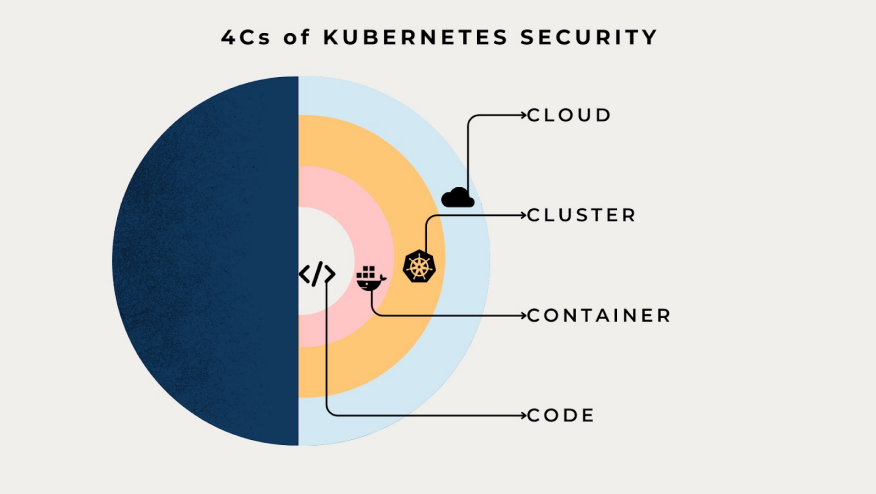
Let’s talk about security

- The 4 C’s in Kubernetes
- Modern threats
- RBAC at the edge
- Runtime under the hood



Let’s talk about security
The 4 C’s in Kubernetes

Security model is designed in 4 layers



Some examples for each layer

> Cloud provider: Network access to API server or nodes, …
> Cluster: RBAC, Secure API server, Authentication, Network policies, …
> Container: Container vulnerability scanner, Image signing, ...
> Code: Access over TLS, Code analysis, Data encryption, ...

Alert 1
Help us to analyze!

- **A** Ask if someone complains about it
- **B** Investigate in Grafana alerts
- **C** Silent the alarm for 2 days
- **D** Take the day off

Alert 1
Help us to analyze!

- **A** Ask if someone complains about it
- **B** Investigate in Grafana alerts
- **C** Silent the alarm for 2 days
- **D** Take the day off

> The issue is real, we must do something

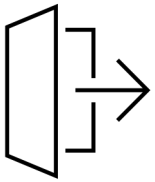
> A threat may not have visible impact to the customer

Alert 1
Help us to analyze!

- **A** Ask if someone complains about it
- **B** Investigate in Grafana alerts
- **C** Silent the alarm for 2 days
- **D** Take the day off

> This is the first step, get more information from Prometheus and Grafana

> Then continue the investigation to identify where the exploit comes from



Alert 1
Help us to analyze!

A	Ask if someone complains about it
B	Investigate in Grafana alerts
C	Silent the alarm for 2 days
D	Take the day off

> This can be done if the privilege escalation is legit (in this case, the rule should be adapted) or a known ongoing test

> What about the pod that allowed this privilege escalation?

> Is this alert only the visible part of the iceberg?

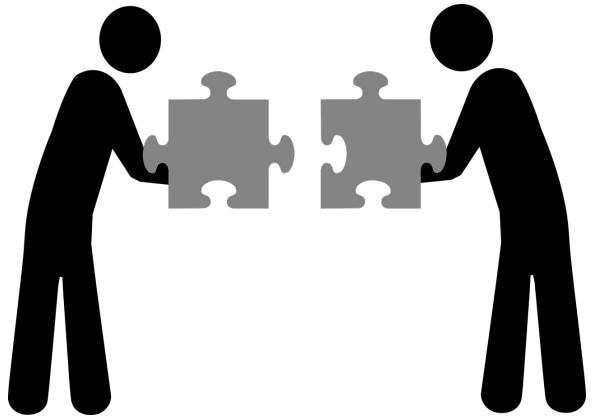
Alert 1
Help us to analyze!

A	Ask if someone complains about it
B	Investigate in Grafana alerts
C	Silent the alarm for 2 days
D	Take the day off

> A pod allowing the usage of Linux capabilities should be handled with care

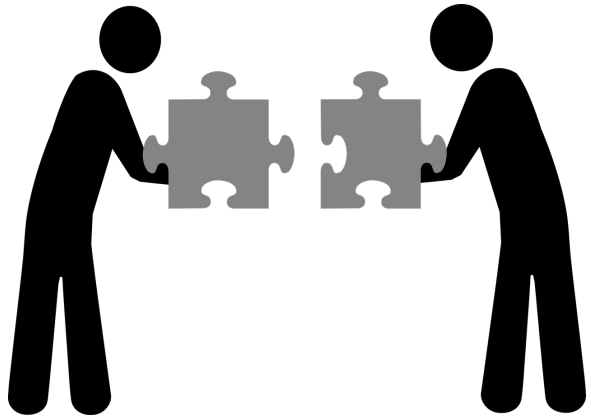
> You must identify what's wrong

Alert 1
Help us to remediate!



- **A** Delete the pod
- **B** Add a taint to not schedule this pod
- **C** Install Norton 360 for protection
- **D** Investigate the audit logs

Alert 1
Help us to remediate!



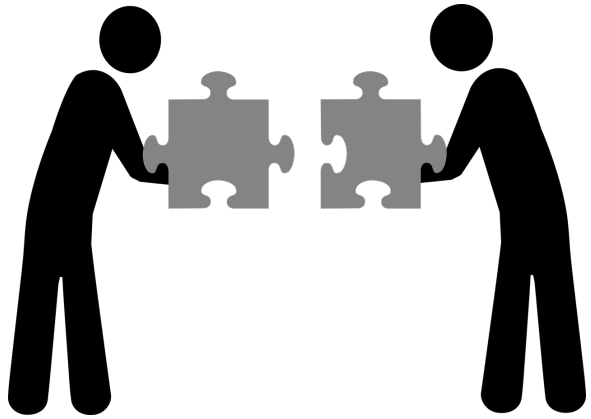
- **A** — **Delete the pod**
- **B** — **Add a taint to not schedule this pod**
- **C** — **Install Norton 360 for protection**
- **D** — **Investigate the audit logs**

> We didn’t took the time to analyze the pod

> Who created the pod?

> The issue may reoccur

Alert 1
Help us to remediate!

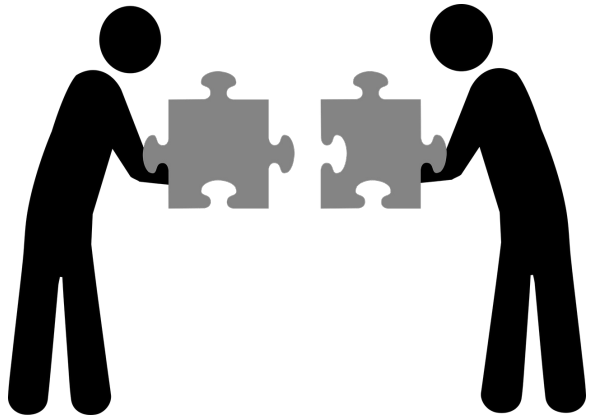


- **A** Delete the pod
- **B** Add a taint to not schedule this pod
- **C** Install Norton 360 for protection
- **D** Investigate the audit logs

> A pod can be scheduled when specifically mentioned in the manifest

> We have to investigate further the pod manifest

Alert 1
Help us to remediate!

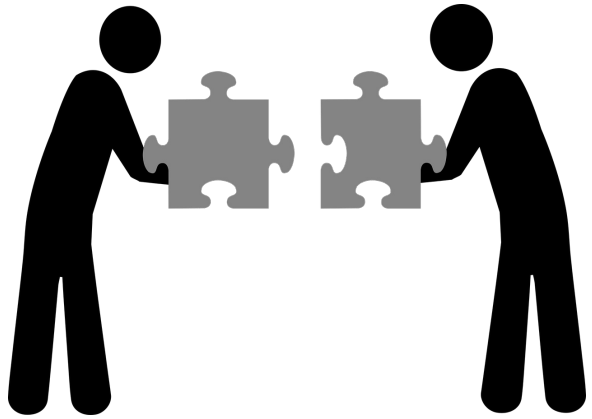


- **A** Delete the pod
- **B** Add a taint to not schedule this pod
- **C** Install Norton 360 for protection
- **D** Investigate the audit logs

> There is nothing to do with Norton 360, it's not related

> Our issue is container related

Alert 1
Help us to remediate!

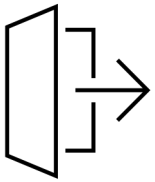


- **A** Delete the pod
- **B** Add a taint to not schedule this pod
- **C** Install Norton 360 for protection
- **D** Investigate the audit logs

> Let's have a look at the audit logs

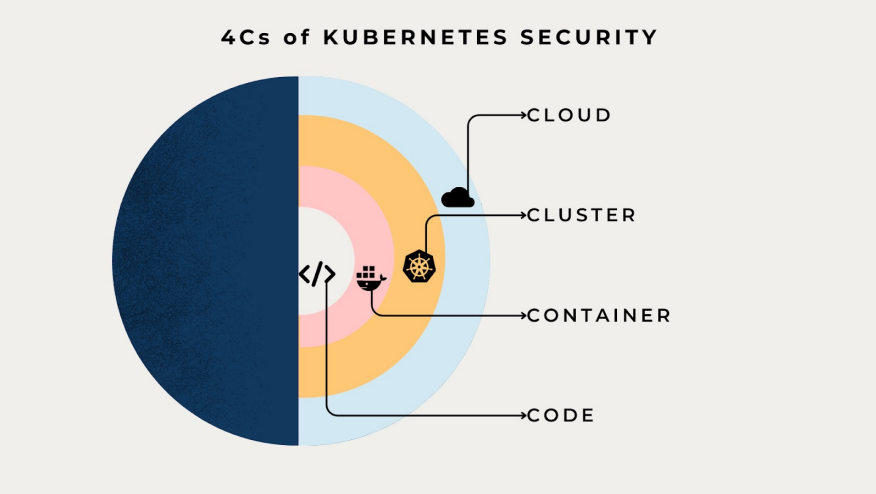
> Find which account has been compromised and renew its token (kubeadm)

> Install a tool to protect against privilege escalation



Let’s talk about security
The 4 C’s in Kubernetes

Security model is designed in 4 layers



Some examples for each layer

> Cloud provider: Network access to API server or nodes, …
> Cluster: RBAC, Secure API server, Authentication, Network policies, …
> Container: Container vulnerability scanner, Image signing, ...
> Code: Access over TLS, Code analysis, Data encryption, ...

Alert 2
Help us to analyze!

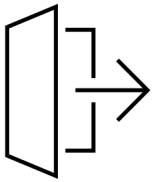
- **A** — etcd may have been compromised, check how
- **B** — Kyverno is not working, look for another tool
- **C** — Hackers are magicians, we can't fight them
- **D** — etcd-controller is a legit pod, add an exception in Kyverno

Alert 2
Help us to analyze!

A	**etcd may have been compromised, check it**
B	**Kyverno is not working, look for another tool**
C	**Hackers are magicians, we can't fight them**
D	**etcd-controller is a legit pod, add an exception in Kyverno**

> Remember we deployed Kyverno in our cluster with a specific cluster rule that avoid privilege escalation

> The only possibility is for someone to write directly into the etcd



Alert 2
Help us to analyze!

A	**etcd may have been compromised, check it**
B	**Kyverno is not working, look for another tool**
C	**Hackers are magicians, we can't fight them**
D	**etcd-controller is a legit pod, add an exception in Kyverno**

> Kyverno is a proven product, our rule is correct

> It is based on admission controllers and are the last gate before the schedule

> Attackers didn’t schedule a suspicious pod through the API Server

Alert 2
Help us to analyze!

A	etcd may have been compromised, check it
B	Kyverno is not working, look for another tool
C	Hackers are magicians, we can't fight them
D	etcd-controller is a legit pod, add an exception in Kyverno

> They are not magicians, however they know how to take advantage of security loopholes

> As an administrator, you are responsible to reduce the attack surface of the platform

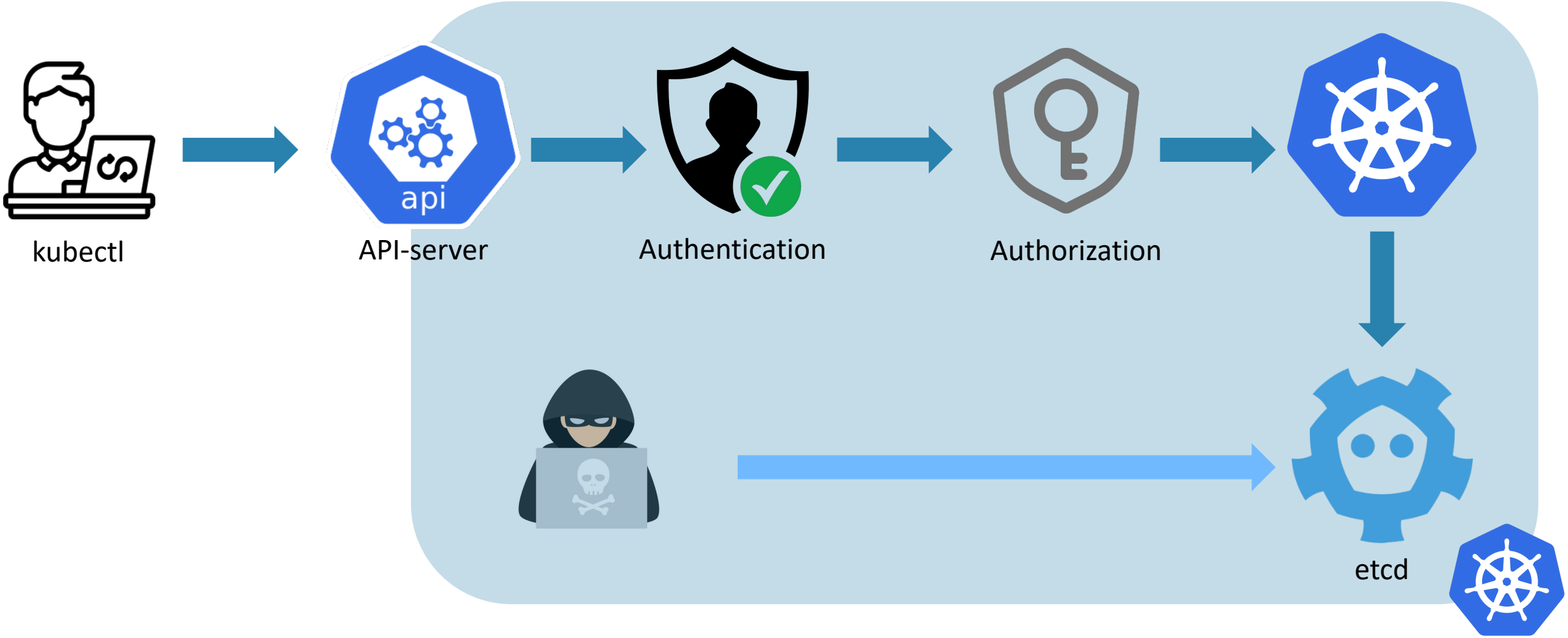
Alert 2
Help us to analyze!

A	**etcd may have been compromised, check it**
B	**Kyverno is not working, look for another tool**
C	**Hackers are magicians, we can't fight them**
D	**etcd-controller is a legit pod, add an exception in Kyverno**

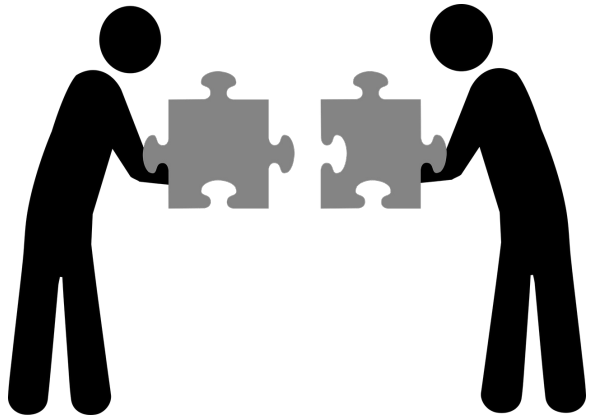
> The etcd-controller pod could be a legit pod, but not in the default namespace

> In case you have a doubt, describe the pod to know the image that is running

Alert 2
Help us to analyze!

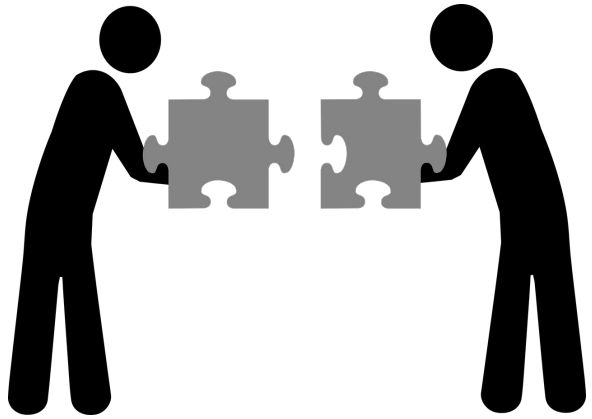


Alert 2
Help us to remediate!



- **A** Delete the etcd certificates to close this door to the attacker
- **B** Activate encryption at rest for the etcd
- **C** Run a CIS benchmark to fix critical vulnerabilities
- **D** Rename etcd certificates

Alert 2
Help us to remediate!

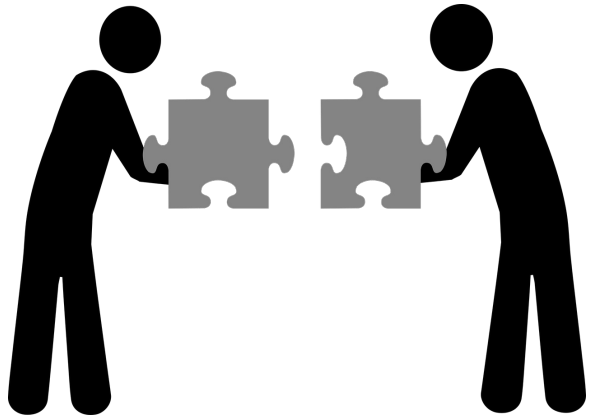


- **A** — **Delete the etcd certificates to close this door to the attacker**
- **B** — **Activate encryption at rest for the etcd**
- **C** — **Run a CIS benchmark to fix critical vulnerabilities**
- **D** — **Rename etcd certificates**

> It sounds like a panic mode, don't do that or you'll break your cluster at the same time

> etcd certificates are vital to your cluster

Alert 2
Help us to remediate!

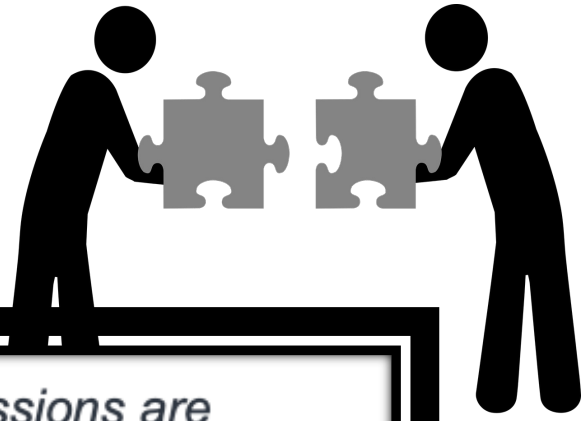


- **A** Delete the etcd certificates to close this door to the attacker
- **B** Activate encryption at rest for the etcd
- **C** Run a CIS benchmark to fix critical vulnerabilities
- **D** Rename etcd certificates

> The intention here is not to steal secrets

> But of course activating encryption at rest is a good practice in case an attacker extracts data from the etcd

Alert 2
Help us to remediate!



- **A** Delete the etcd certificates to close this door to the attacker
- **B** Activate encryptio… etcd
- **C** Run a CIS benchma… vulnerabilities
- **D** Rename etcd certif…

1.1.21 Ensure that the Kubernetes PKI key file permissions are set to 600 (Manual)

Profile Applicability:

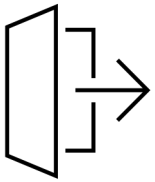
- Level 1 - Master Node

Description:

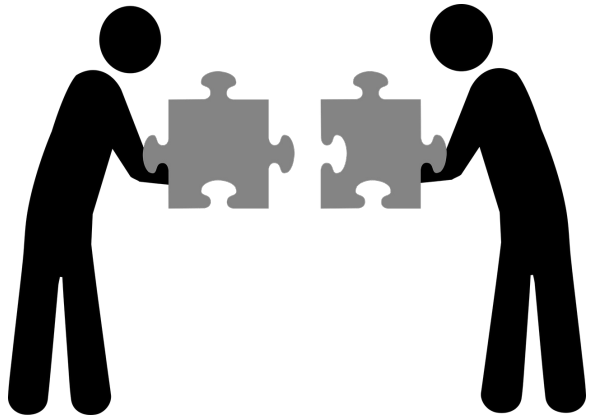
Ensure that Kubernetes PKI key files have permissions of `600`.

Rationale:

Kubernetes makes use of a number of key files as part of the operation of its components. The permissions on these files should be set to `600` to protect their integrity and confidentiality.



Alert 2
Help us to remediate!



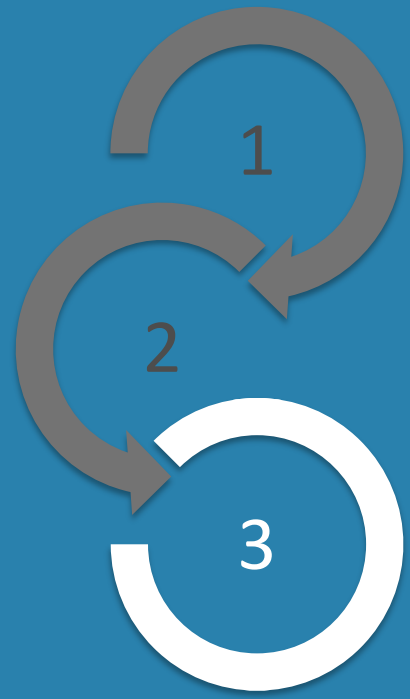
- **A** Delete the etcd certificates to close this door to the attacker
- **B** Activate encryption at rest for the etcd
- **C** Run a CIS benchmark to fix critical vulnerabilities
- **D** Rename etcd certificates

> This move may break your etcd because the etcd manifest points to them

> An attacker will know anyway which certificates are used by the etcd thanks to its manifest

More about security

- Best practices
- Modern Threats
- Conclusion: The known and the unknown



More about security
Best practices

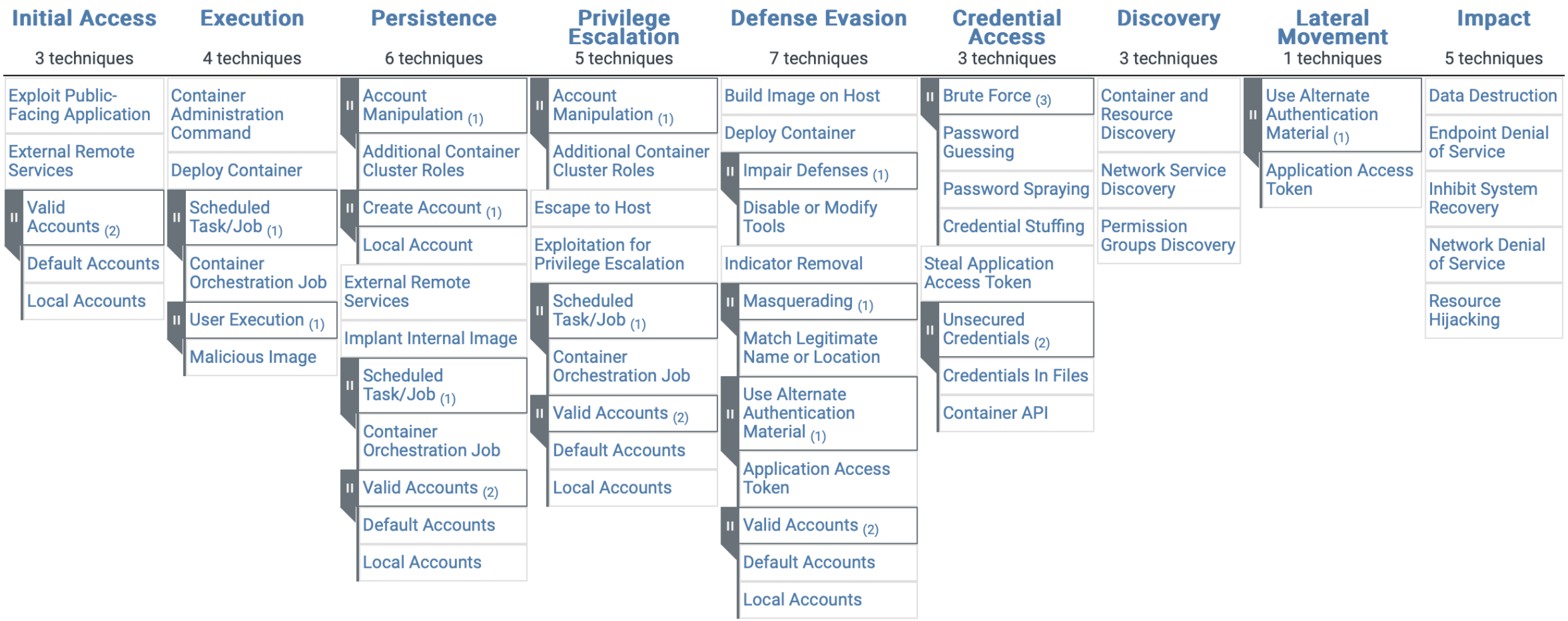
General considerations

More about security
Best practices

General considerations

- Security is a daily business
- Keep your platform and components up-to-date
- Monitor your infrastructure
- Scan your running containers
- Secure communication whenever possible with
  - HTTPS
  - mTLS
  - TLSv1.2
- Use RBAC with minimum privilege principle
- Use a tool to manage all critical/sensitive informations
  - Passwords
  - Certificates
  - Tokens

More about security
Modern threats

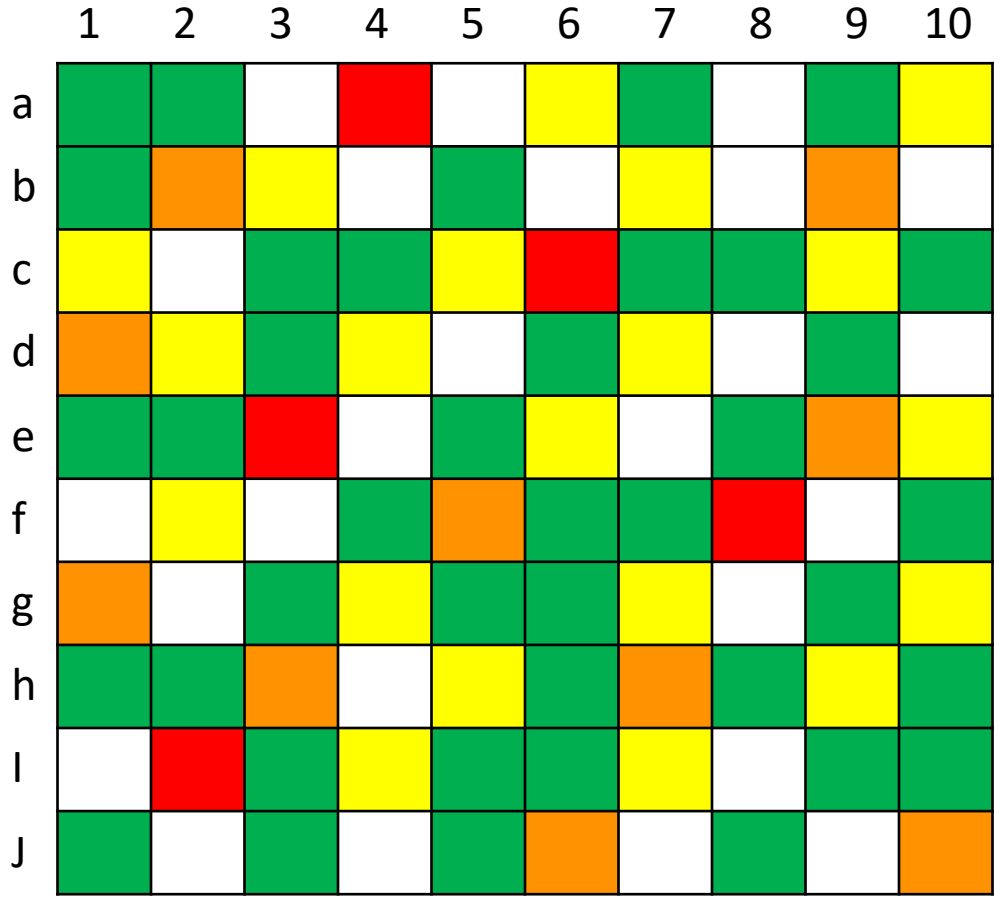


Reference: https://attack.mitre.org/matrices/enterprise/containers/

More about security

Conclusion: The known and the unknown

Rating	CVSS Score
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0



Disclaimer

Special thanks

No Kubernetes clusters were harmed during the sentation.

All pods and scenarios are entirely fictional

Any similarity to actual Kubernetes cluster is unintentional

Any questions?

Please do ask!

We would love to boost your IT-Infrastructure

How about you?