MAY 12-13

BRIEFINGS

kdigger

A Context Discovery Tool for Kubernetes Penetration Testing

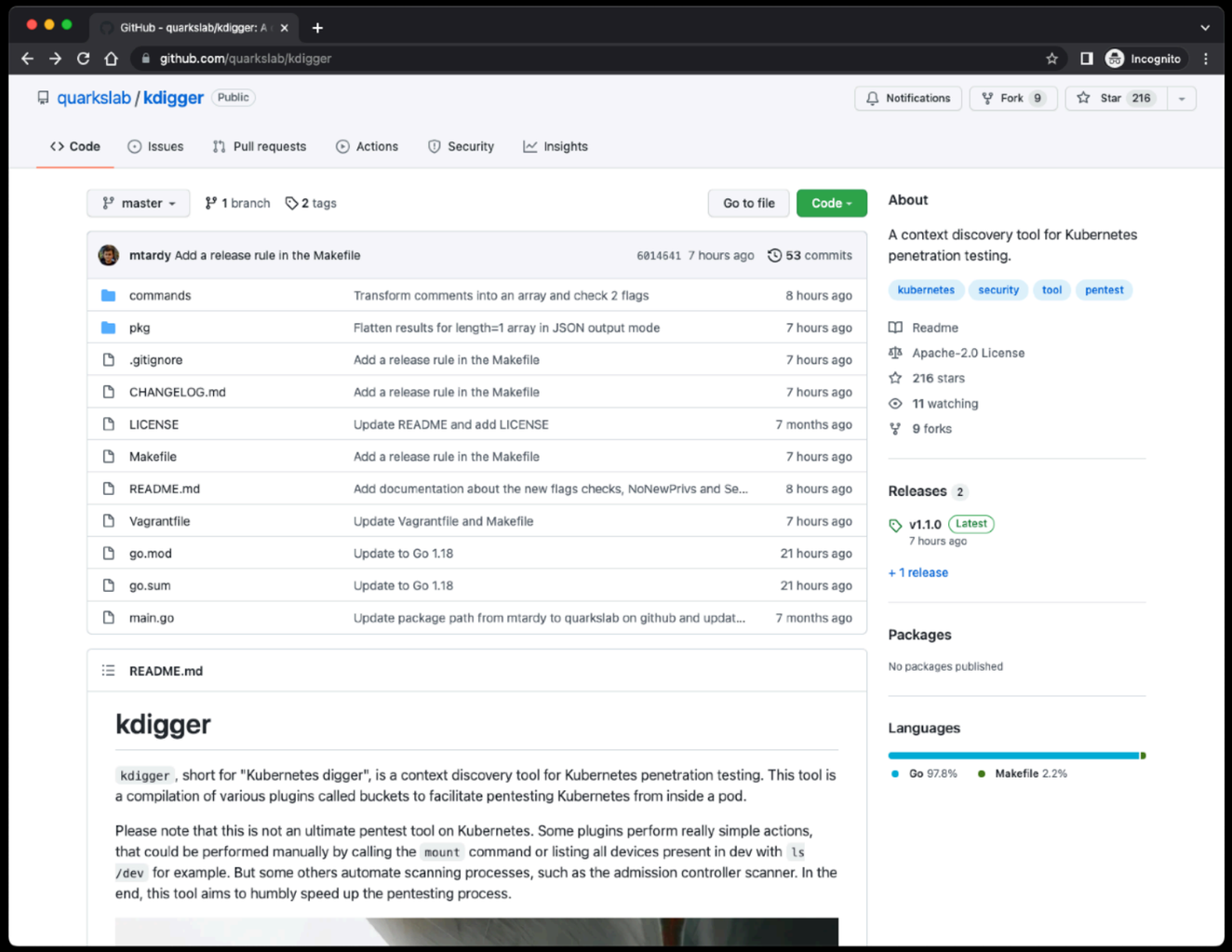
About me

- Mahé Tardy
- Security R&D Engineer @ Quarkslab
- Doing research on Kubernetes
- @mtardy_ on Twitter
- mtardy@quarkslab.com

Introduction on kdigger

A statically linked Go binary CLI

- Written in Go.
- Statically linked binary.
- Easy to install and use CLI.



Why another tool?

- Participated in the last KubeCon Cloud-Native Security Day CTFs.
- Learned the habits of many experts in the fields by watching live solving sessions.
- Loved **amicontained** by Jessie Frazelle.
- Decided to automate a security checklist from inside a Kubernetes Pod!

What can it do for you?

Like **amicontained** you can:

- Try to guess your container runtime.
- See your capabilities.
- Scan for namespace activation and configuration.
- Scan for the allowed syscalls.

But from a **Kubernetes** perspective:

- Retrieve service account token.
- Scan token permissions.
- List interesting environment variables.
- Retrieve all available services in a cluster.
- Retrieve leaked information by cgroups v1.
- Retrieve the specifications of the node.
- Scan the admission controller chain!

And more **basic** stuff: check mounts, uid, processes, devices, status flag, and API versions.

Disclaimer on the usage

Some checks rely on implementation details or fragile features:

- PID namespace or container runtime check from **amicontained**.
- CoreDNS wildcard feature for services listing has been removed in v1.9.0. https://github.com/coredns/coredns/issues/4984

Needs to be updated and extended with new checks

- Pretty straightforward to extend, look at the contributing guide in the README if you have interesting checks to add.

Kubernetes attacks?

CVEs of March for container escapes

Linux

- **CVE-2022-0847** a.k.a. DirtyPipe. Vulnerability allows for overwrite of files that should be read-only.

- **CVE-2022-0492**. Vulnerability in cgroup handling can allow for container breakout depending on isolation layers in place.

CRI-O

- **CVE-2022-0811.** Vulnerability in setting sysctls in k8s/ OpenShift manifests allows for container breakout.

Containerd

- **CVE-2022-23648**. Vulnerability in volume mounting allows for arbitrary file read from the underlying host, leading to likely indirect container breakout.

Source: https://www.container-security.site/ by @raesene

CVEs of March for container escapes

Linux

- **CVE-2022** a. DirtyPipe. allows for overwrite t should

- **CVE-2022-0** abi handling can allow for conta on isolation layers in place.

CRI-O

- **CVE-2022-081** etting sysctls in k8s/ OpenShift mar ner breakout.

Containe

- **CVE-2** ulnera e mounting allows fo ile read fr derlying host, leading t direct container breakout.

Kubernetes security today is more about **configuration** than **vulnerabilities**.

Demonstration!

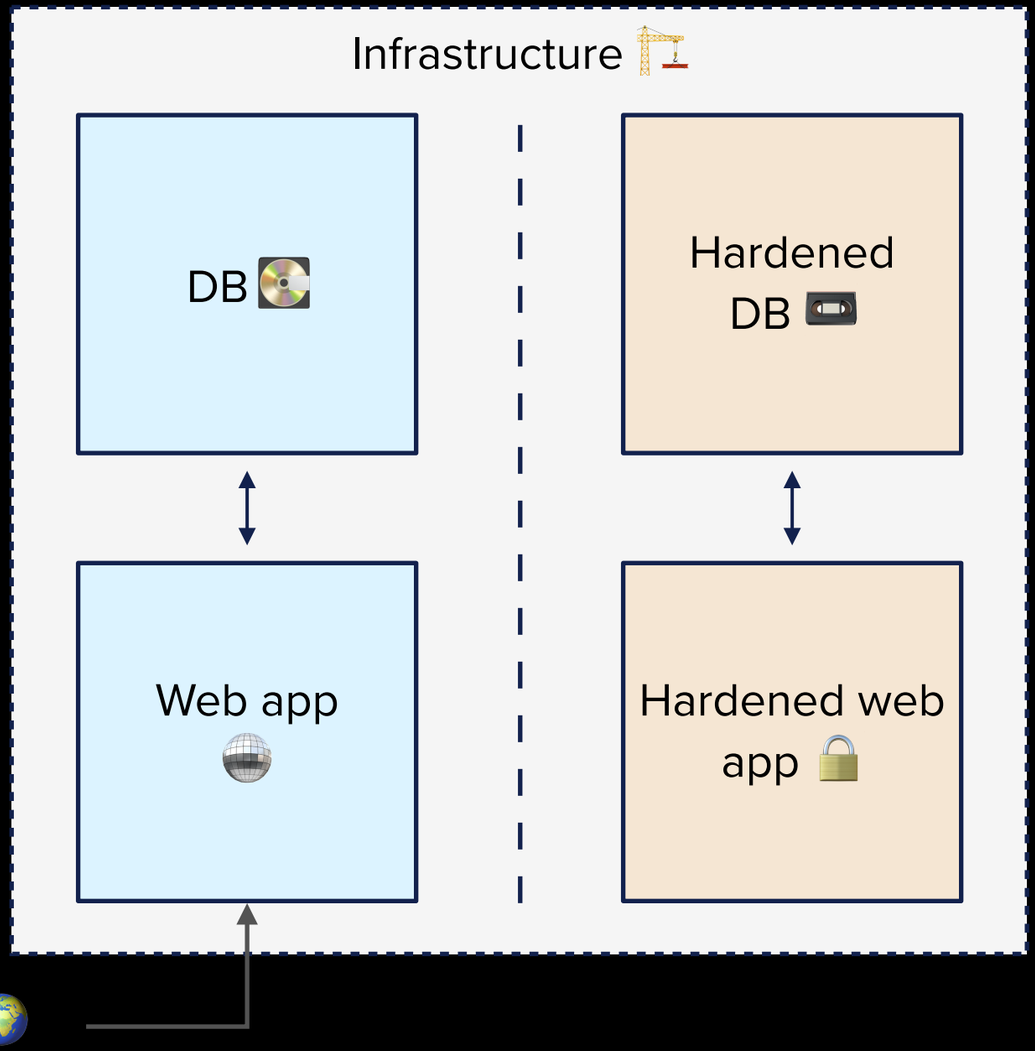
Simplified Infrastructure

This setup was inspired by a real audit mission and highlights the problems of colocation in clusters.

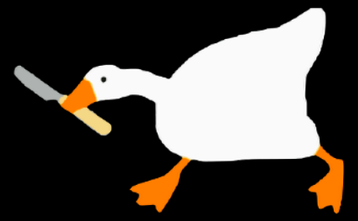
Context:

- Web app exposed to internet.

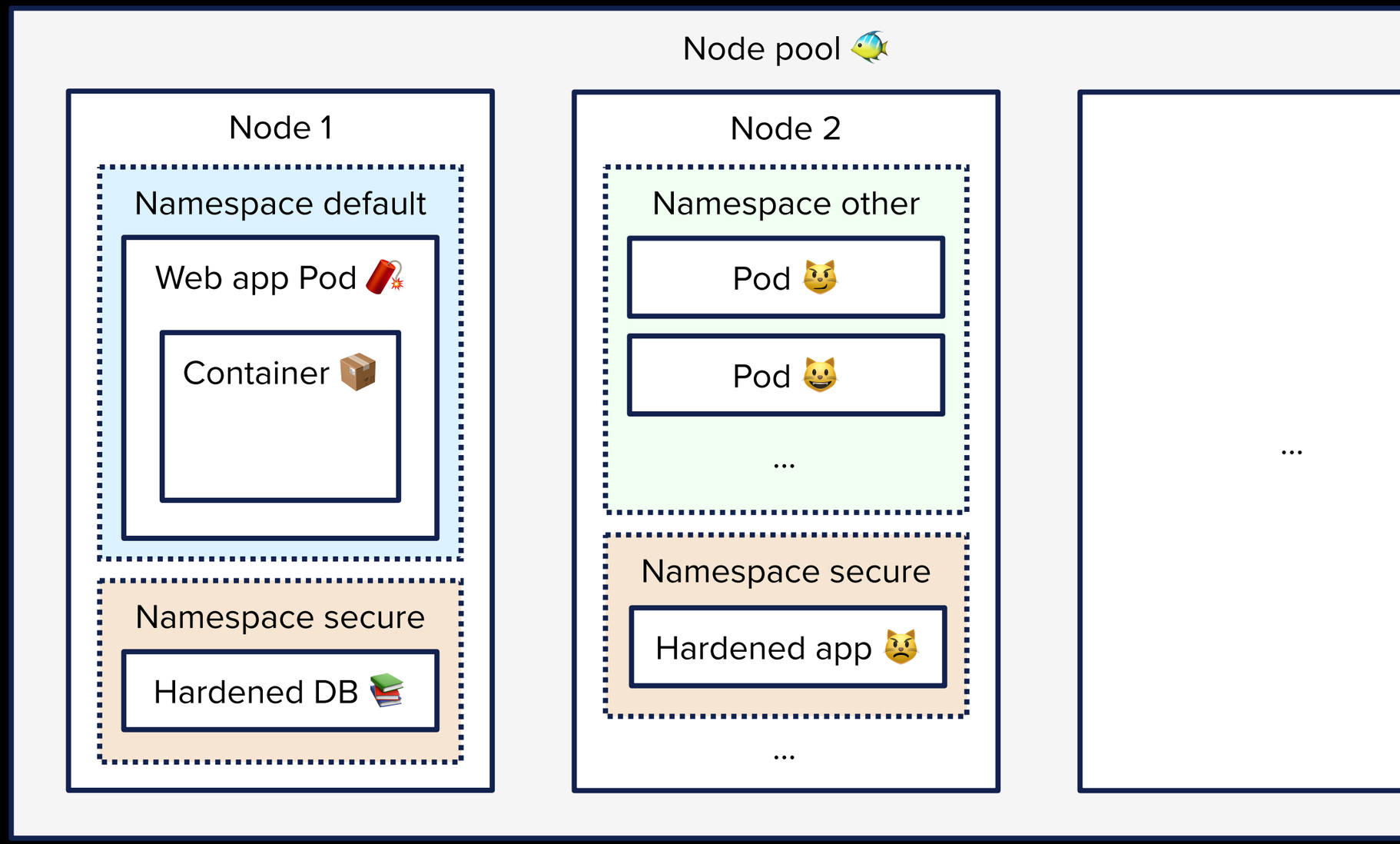
- Hardened web app, only exposed on the internal network.



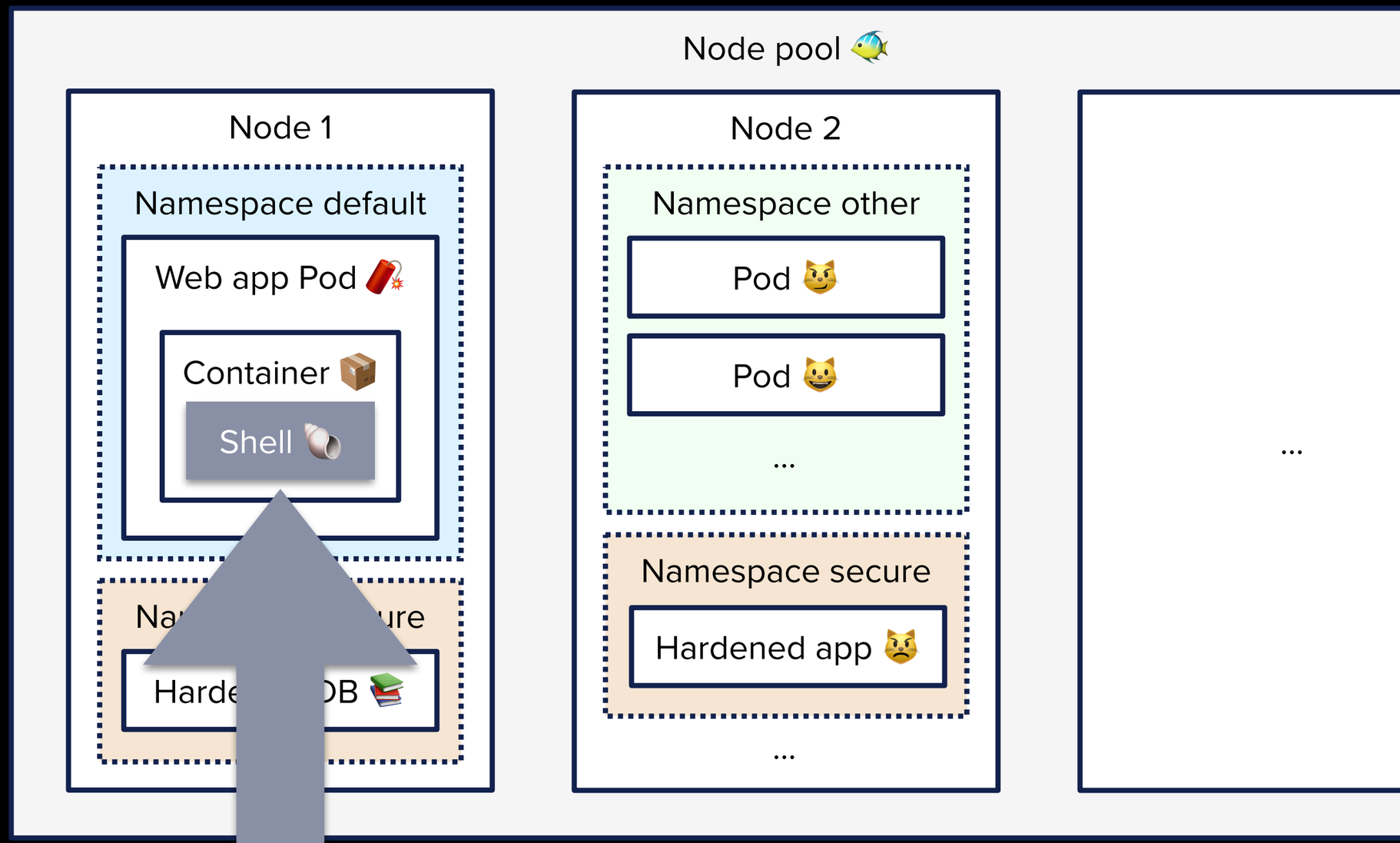
Honk time!



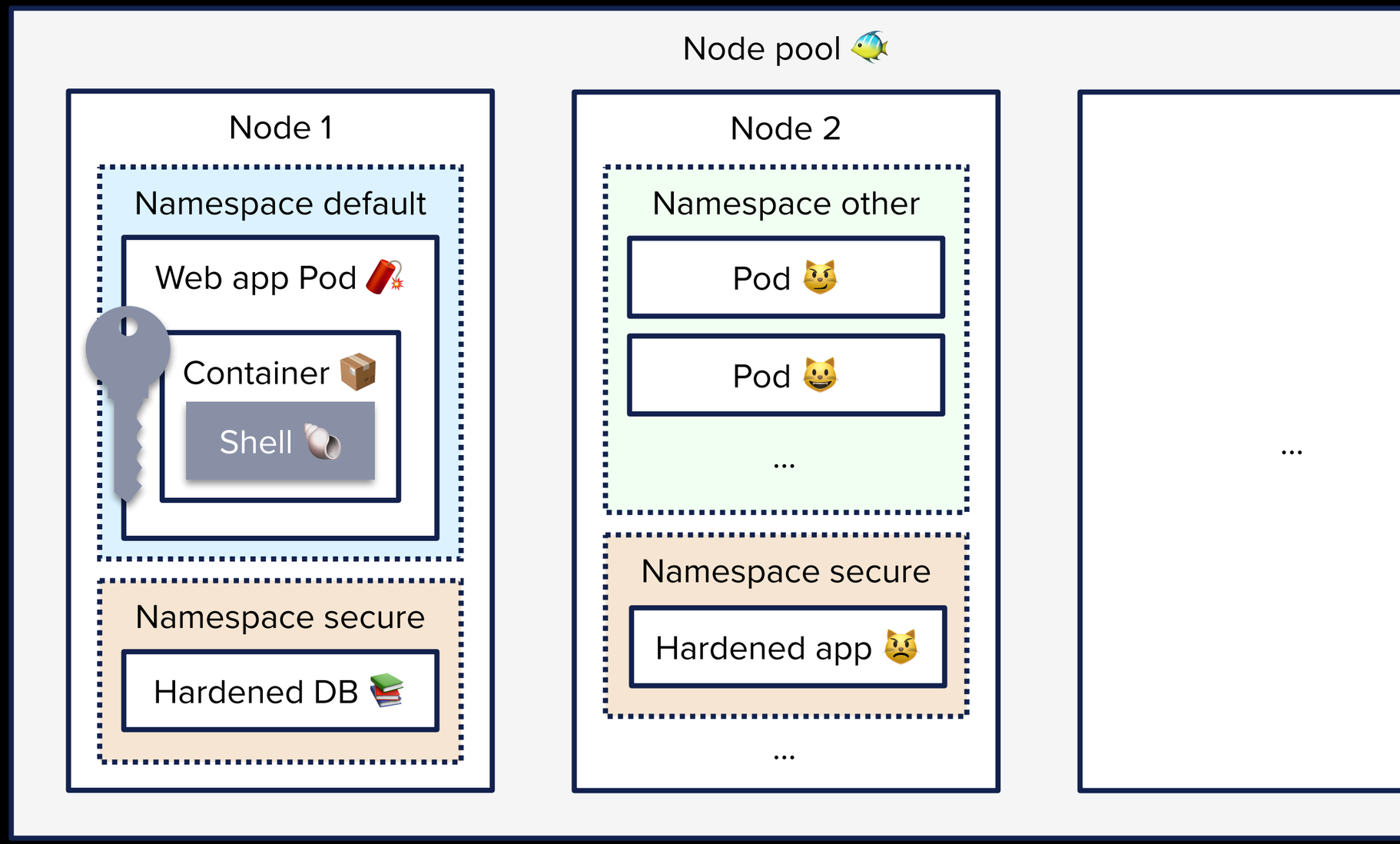
What happened? - Three main issues



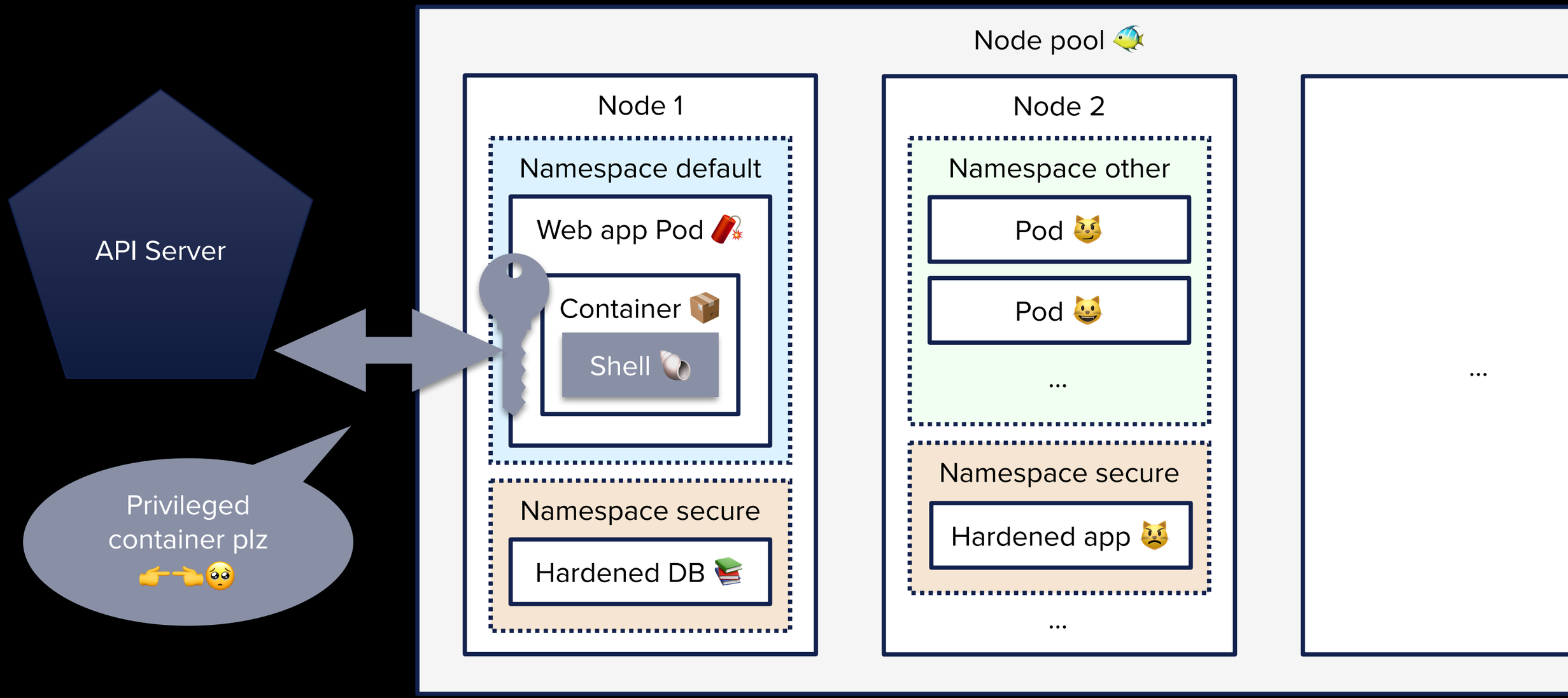
What happened? - First issue: vulnerability in software



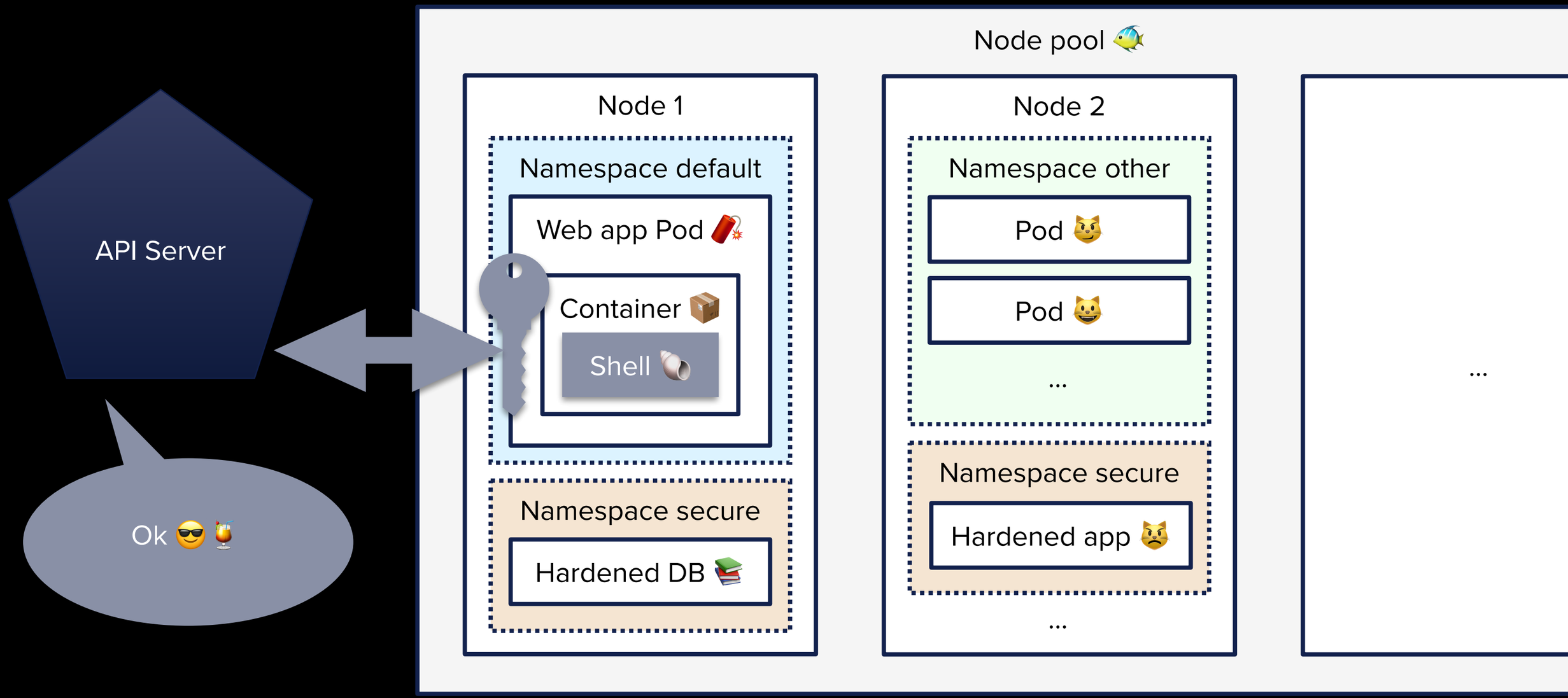
What happened? - Second issue: credentials in container



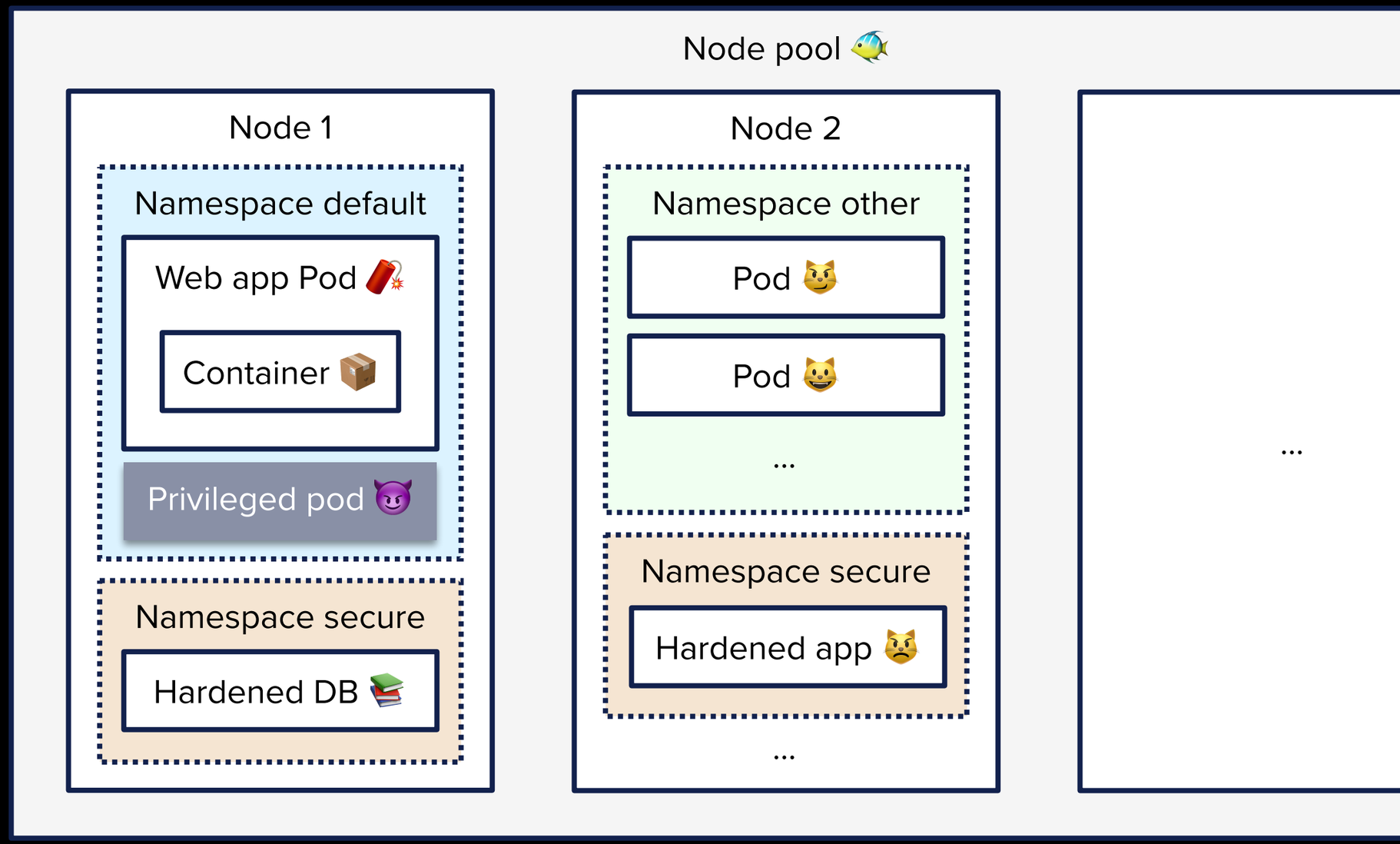
What happened? - Third issue: no admission control



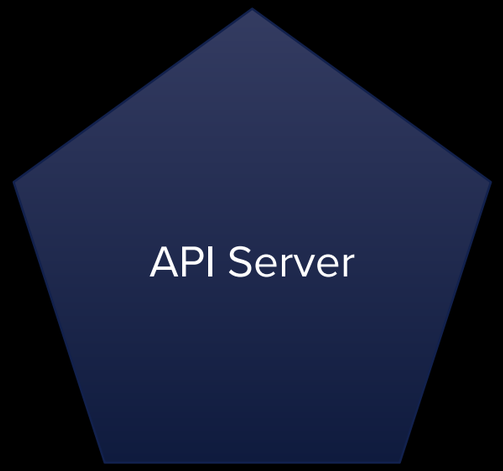
What happened? - Third issue: no admission control



What happened? - Third issue: no admission control



What happened? - Third issue: no admission control



Node pool 🐟

Node 1

Namespace default

Web app Pod 🧨

Container 📦

Escalate to node 😈

Namespace secure

Hardened DB 📚

Node 2

Namespace other

Pod 😺

Pod 😺

...

Namespace secure

Hardened app 😾

...

...

What happened? - Third issue: no admission control

API Server

Node pool 🐟

Node 1

Namespace default

Web app Pod 🧨

Container 📦

Escalate to node 😈

Namespace secure

Hardened DB 📚

Node 2

Namespace other

Pod 😺

Pod 😺

Escalate to node 😈

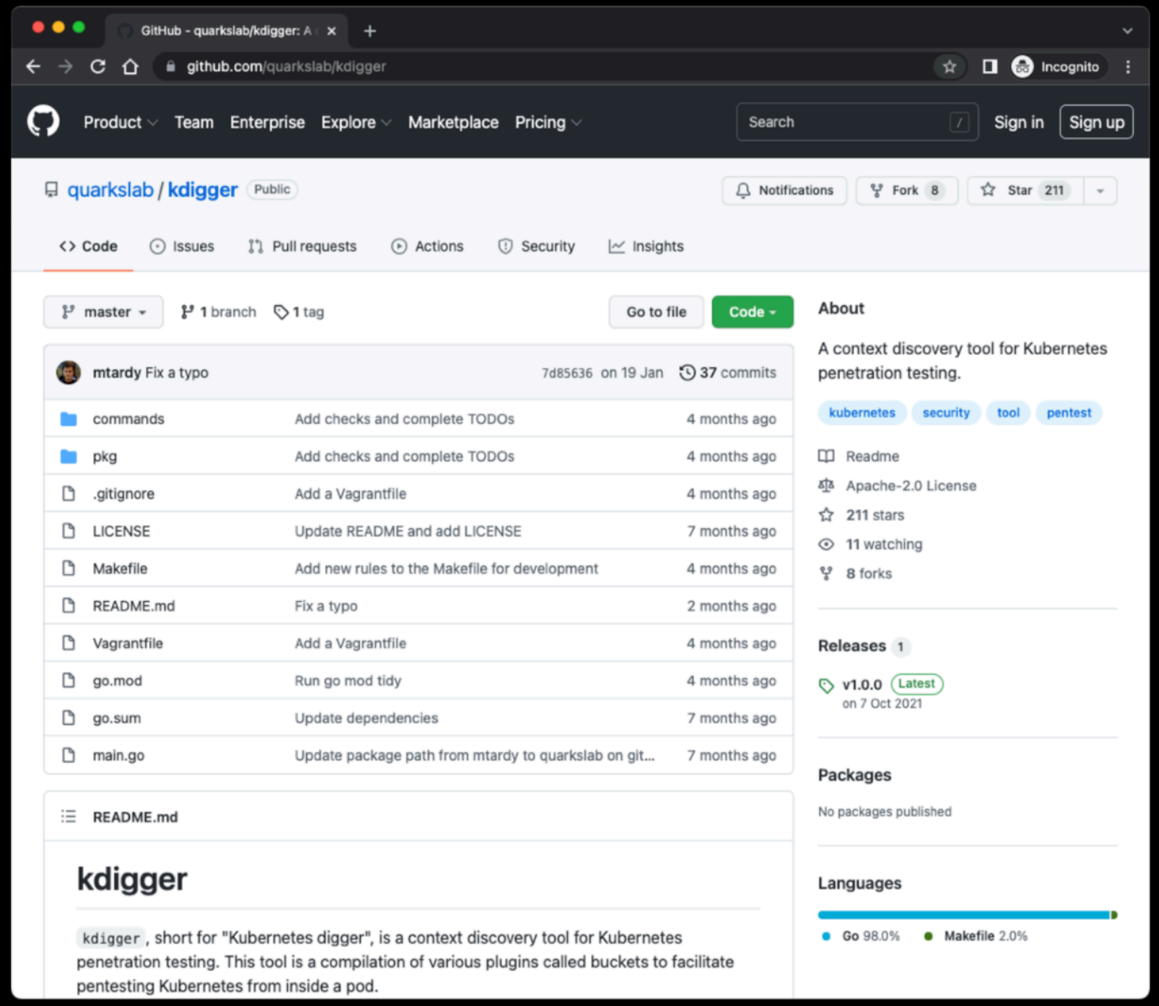
Namespace secure

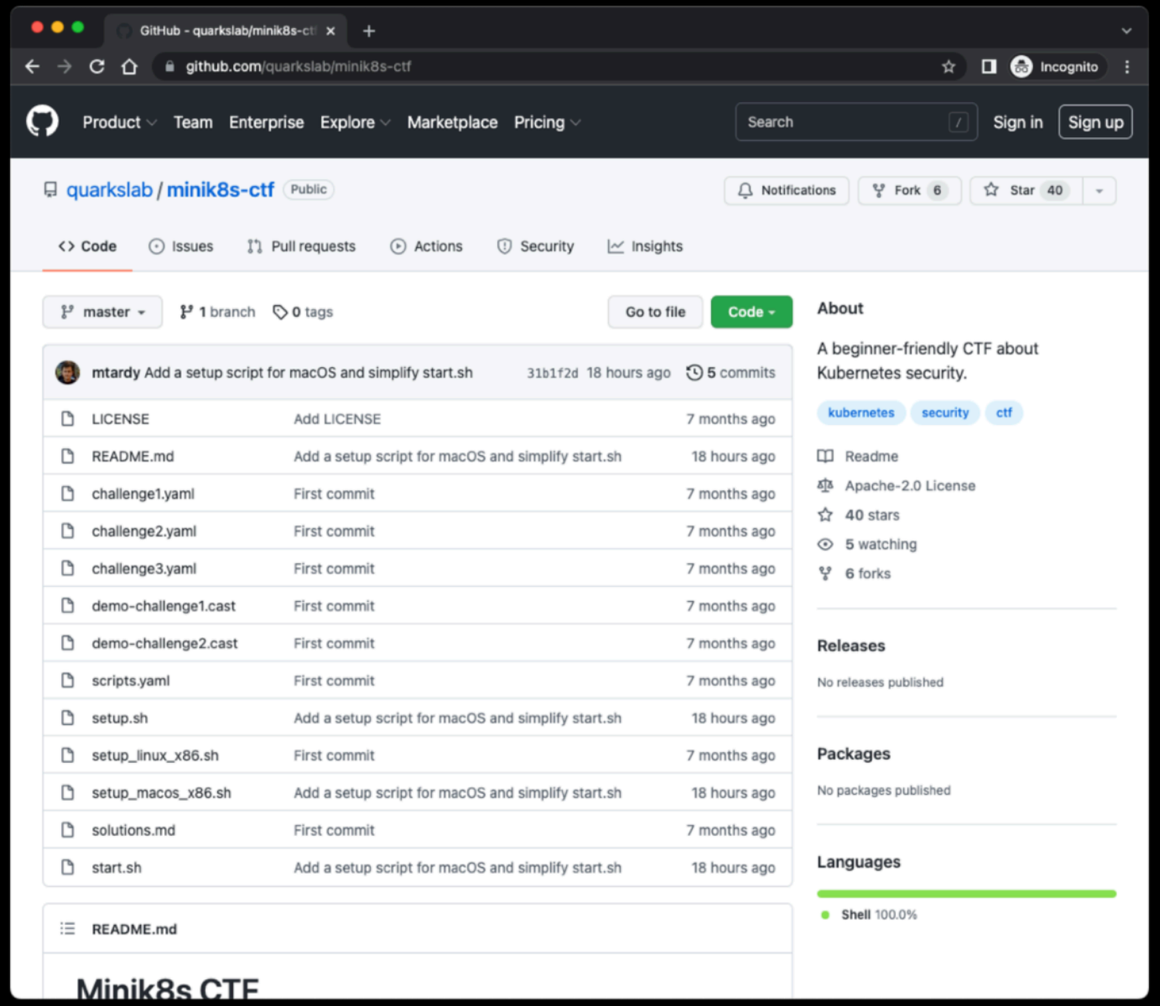
Hardened app 😾

...

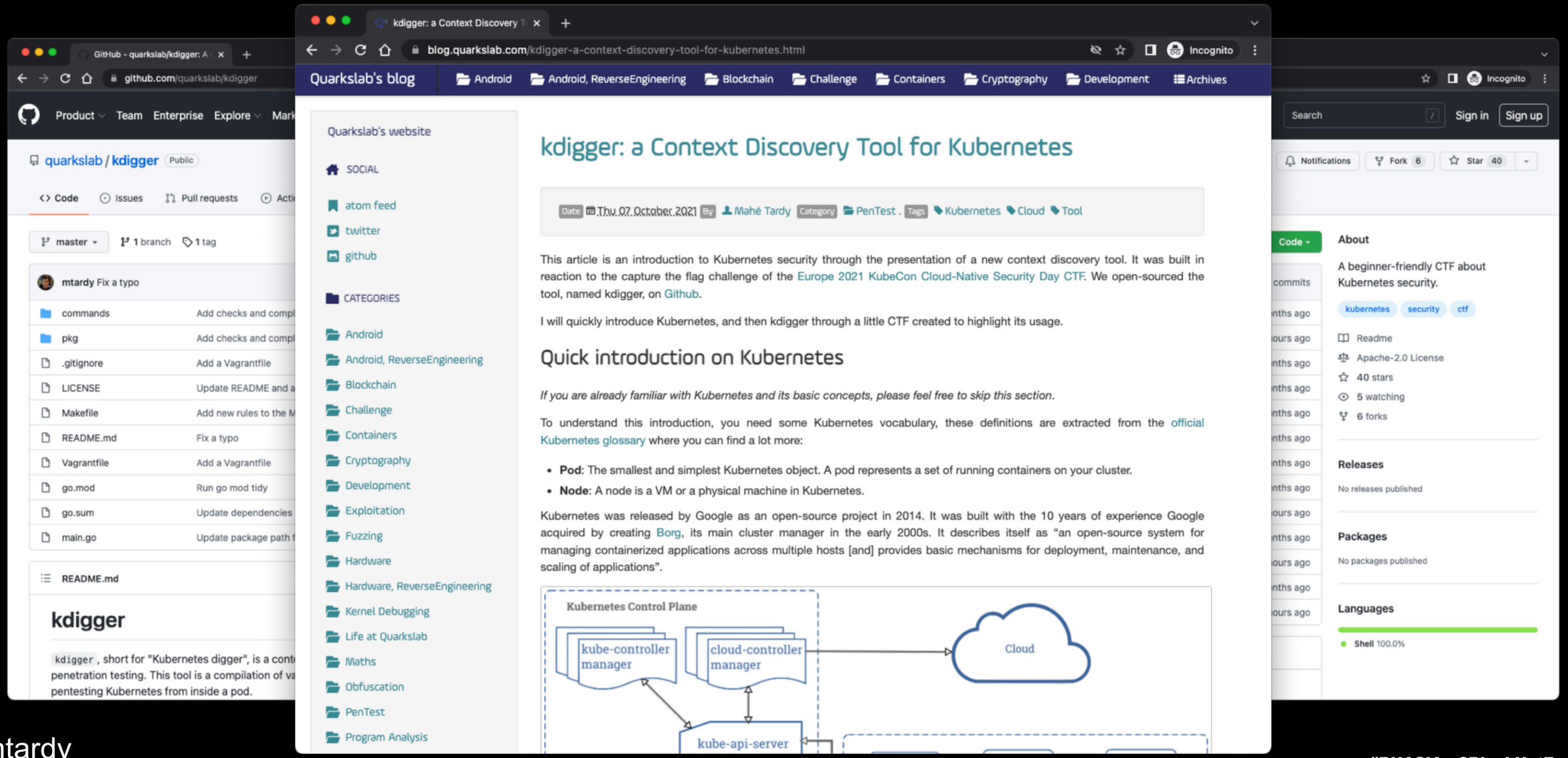
...

If you want to experiment by yourself





If you want to experiment by yourself



Conclusion

Key takeaways

- **kdigger** is open-source and available at https://github.com/quarkslab/kdigger.
- **minik8s-ctf** is another open-source project to directly experiment with **kdigger** that you can find at https://github.com/quarkslab/minik8s-ctf.
- Easier context discovery during a Kubernetes pentest from inside the pods.

- **Mahé Tardy**
- **@mtardy_ on Twitter**
- **mtardy@quarkslab.com**