



MAY 12-13

BRIEFINGS

kdigger

A Context Discovery Tool for Kubernetes Penetration Testing

About me

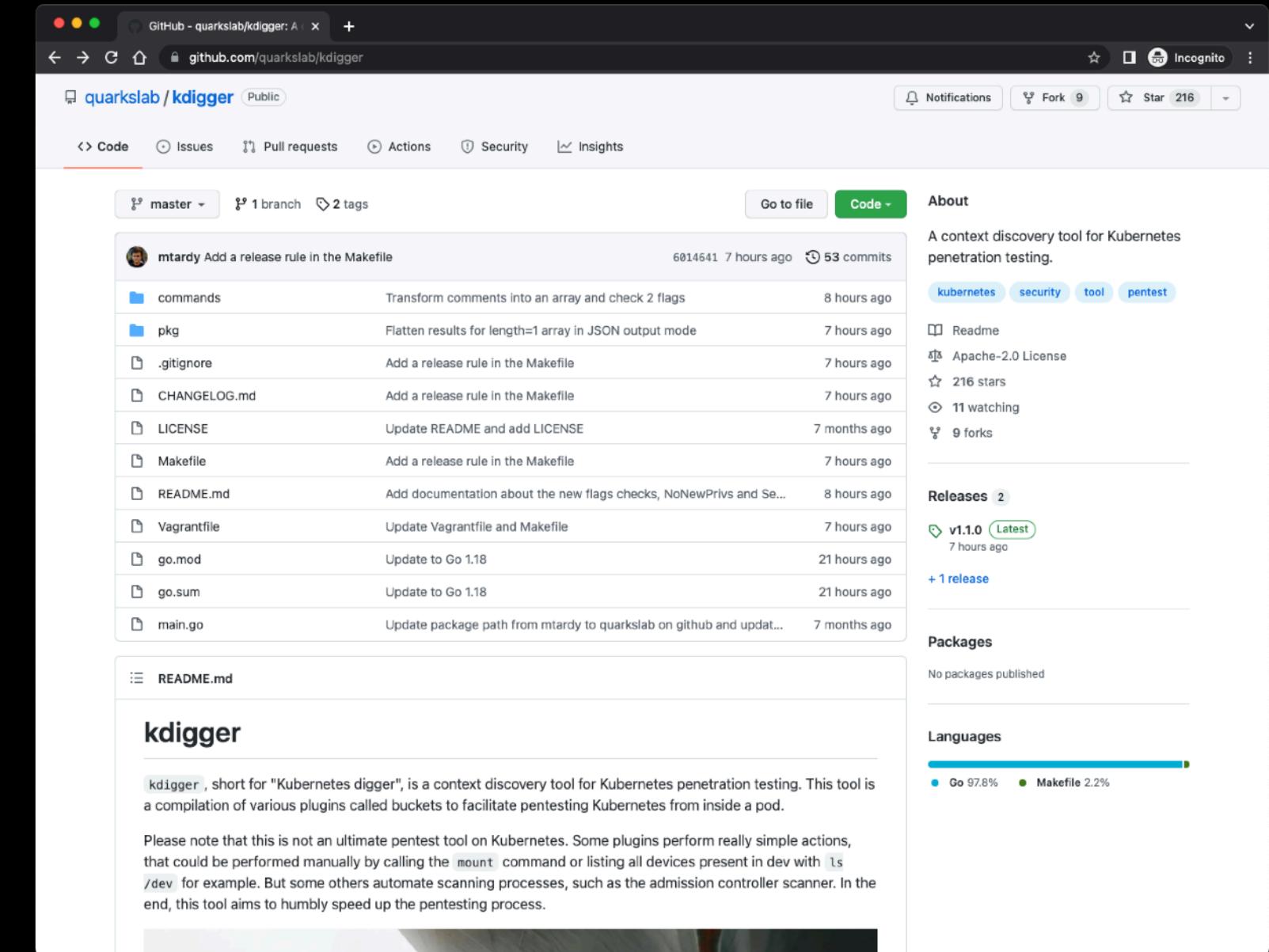


- **Mahé Tardy**
- **Security R&D Engineer @ Quarkslab**
- **Doing research on Kubernetes**
- **@mtardy_ on Twitter**
- **mtardy@quarkslab.com**

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-2022-0812** for overwrite of /proc/
processes/[pid]/fd/0.
- a. DirtyPipe. The kernel should
not allow overwriting of fd 0.

allows

- **CVE-2022-0813** allows for container breakout.
Container handles can be shared
between containers. This handling can
allow for container breakout if no isolation
layers in place.

CRI-O

- **CVE-2022-0814** OpenShift managed by CRI-O
allows for setting sysctls in k8s/
OpenShift manifest files to allow container breakout.

Containers

- **CVE-2022-0815** Container vulnerability
allows for file mounting
while read from the underlying host,
leading to an indirect container breakout.





Kubernetes security
today is more about
configuration than
vulnerabilities.

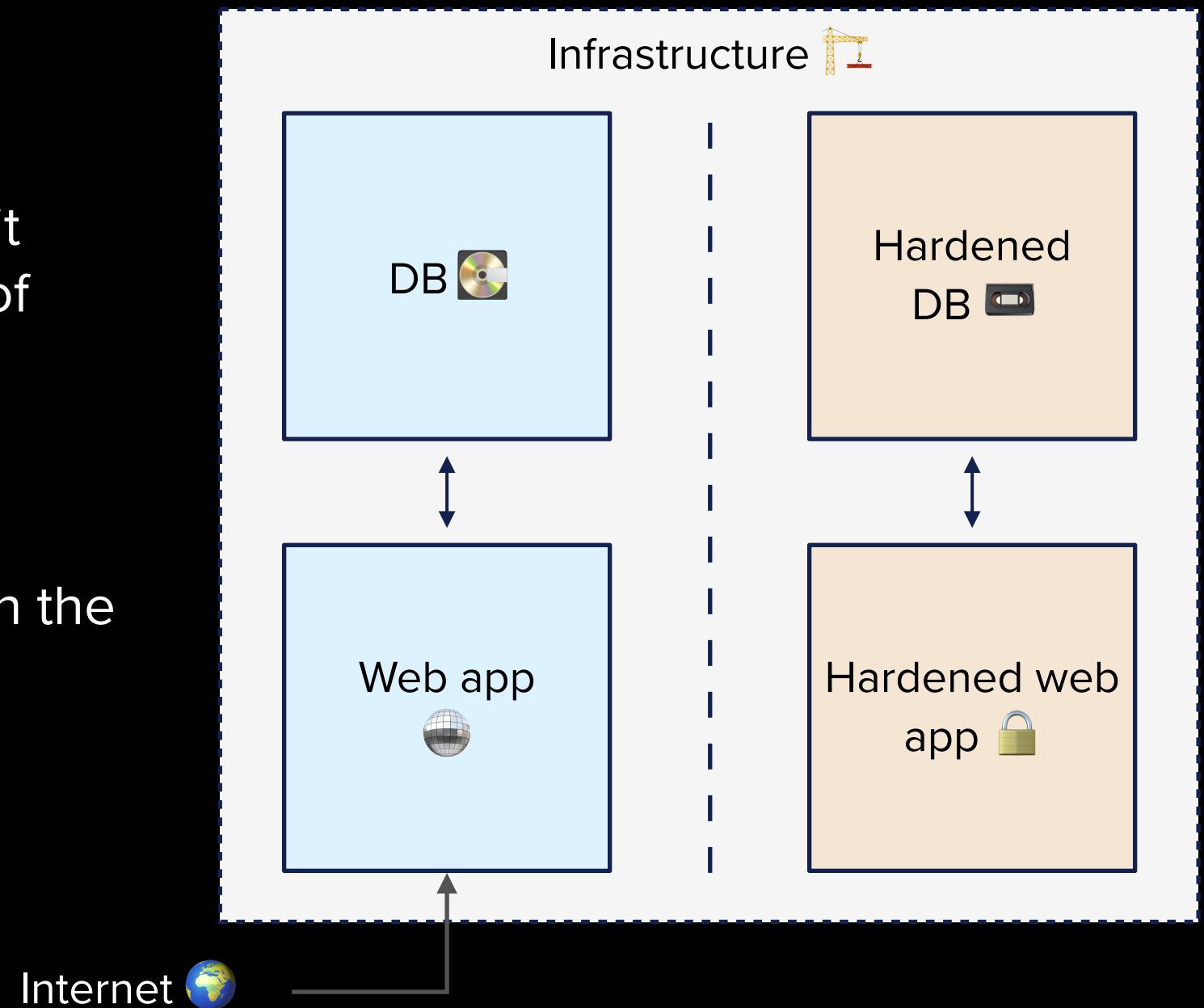
Demonstration!

Simplified Infrastructure

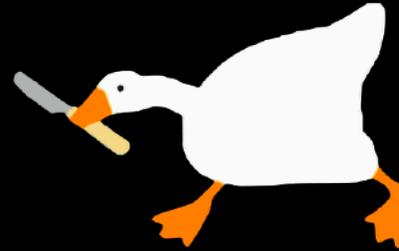
This setup was inspired by a real audit mission and highlights the problems of colocation in clusters.

Context:

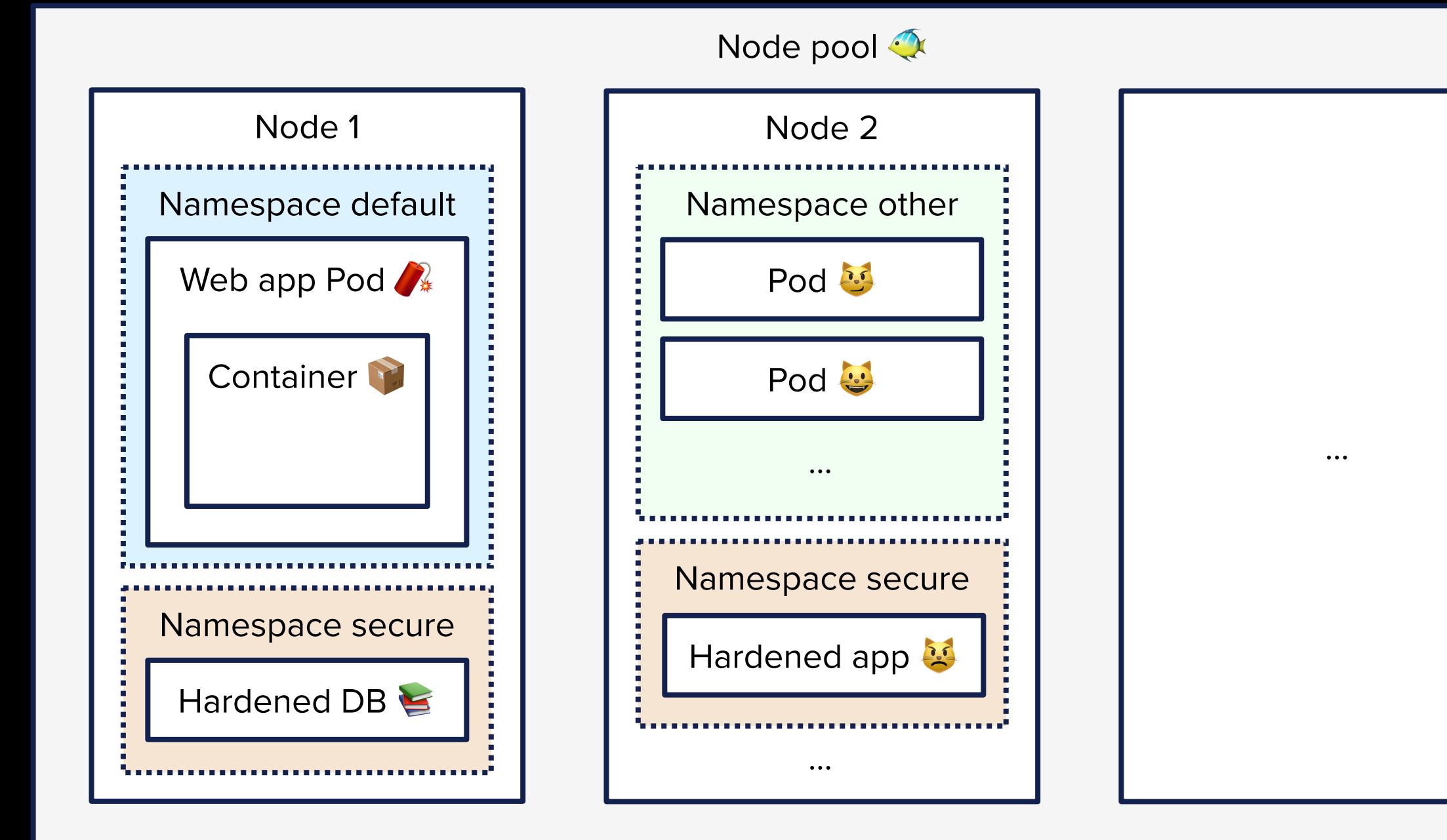
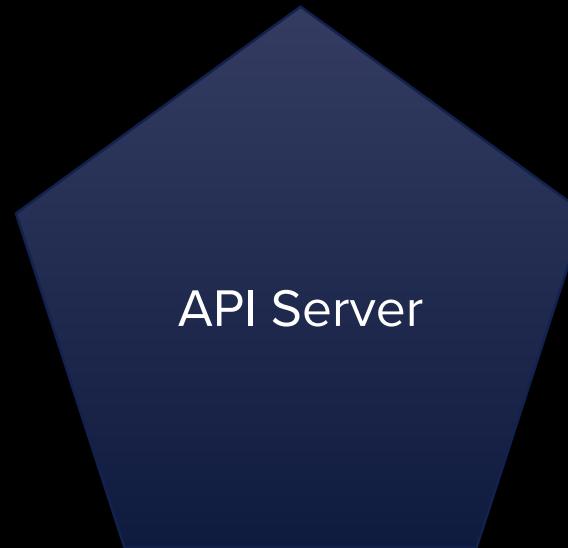
- Web app exposed to internet.
- Hardened web app, only exposed on the internal network.



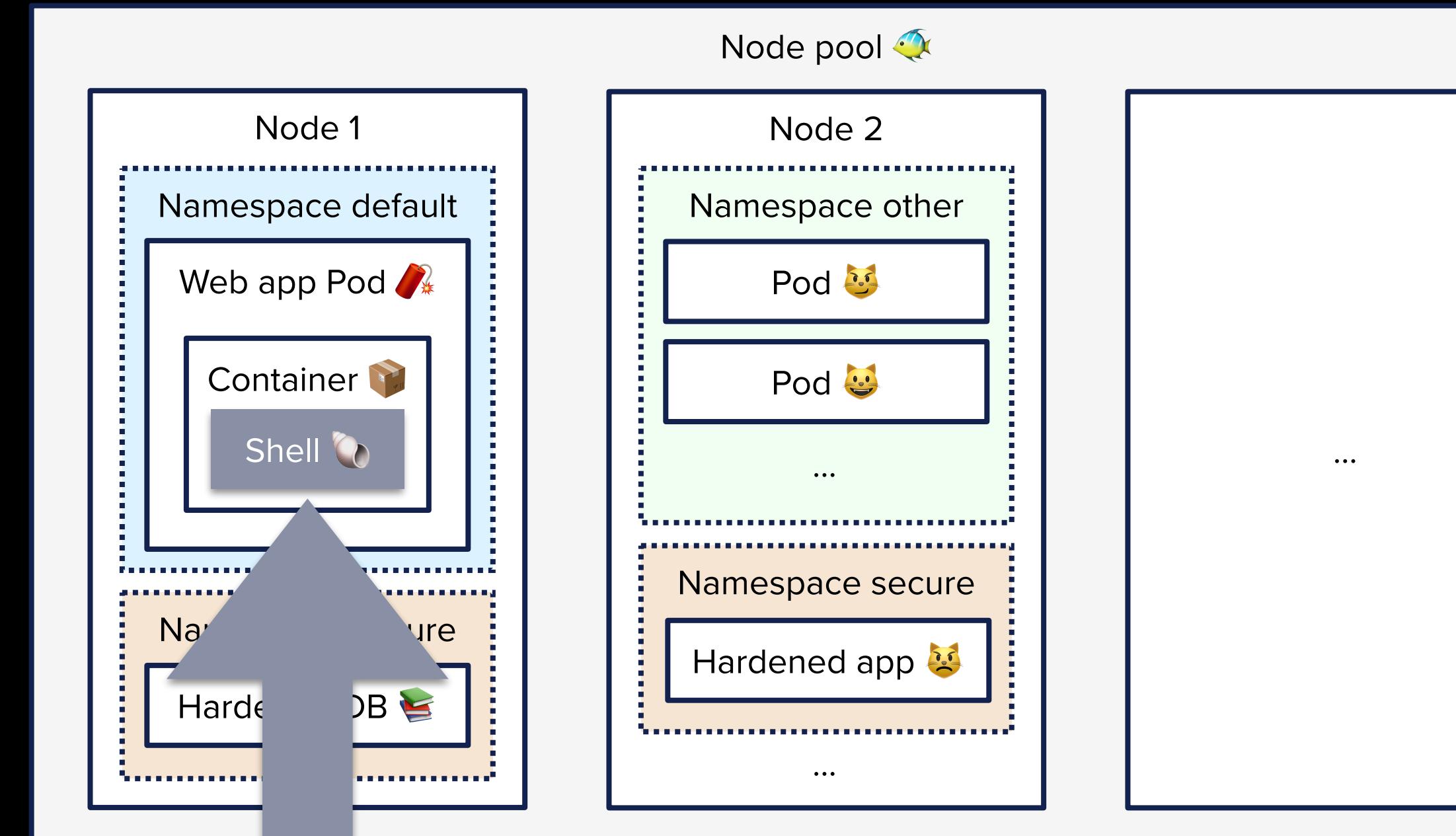
Honk time!



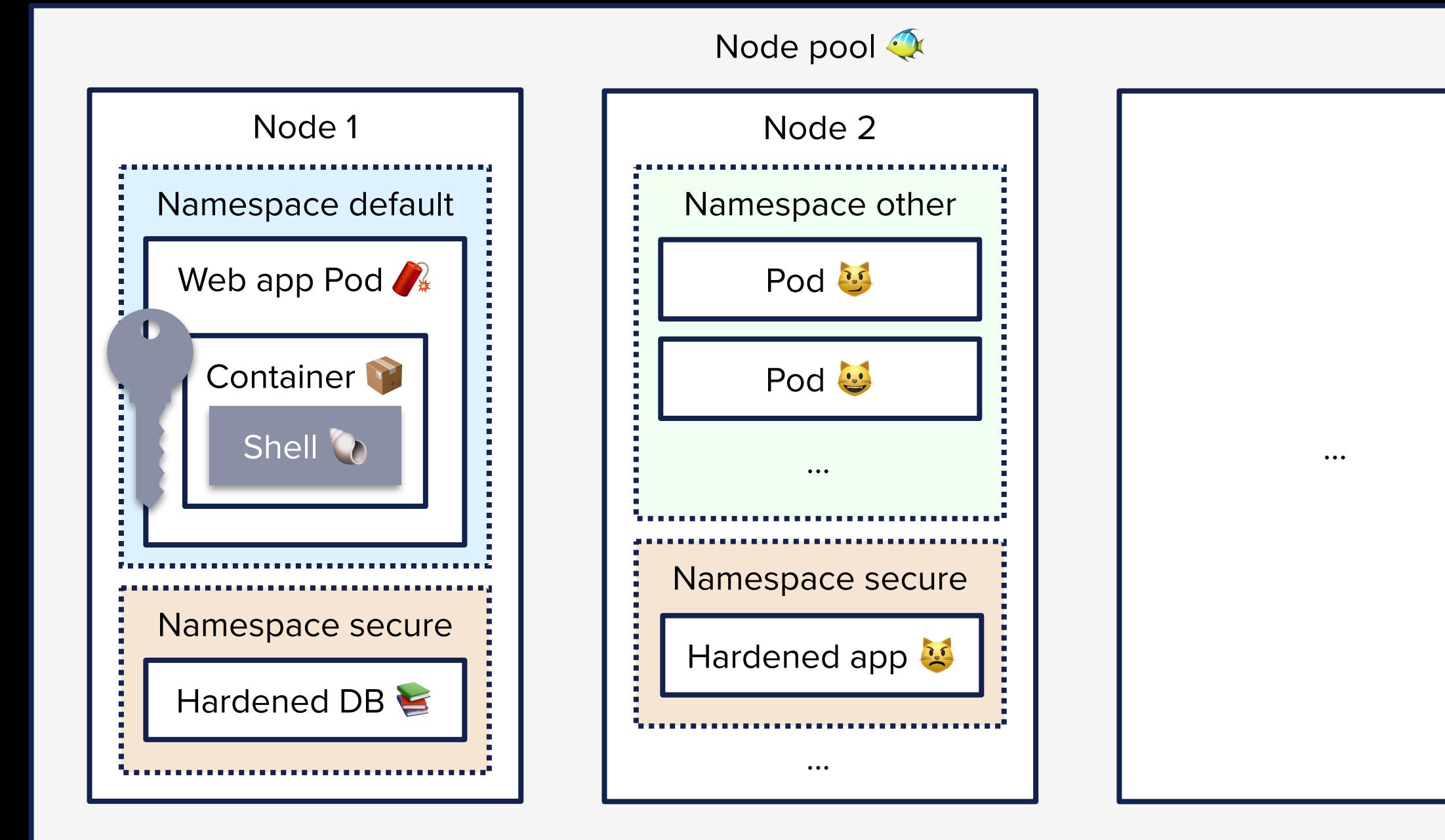
What happened? - Three main issues



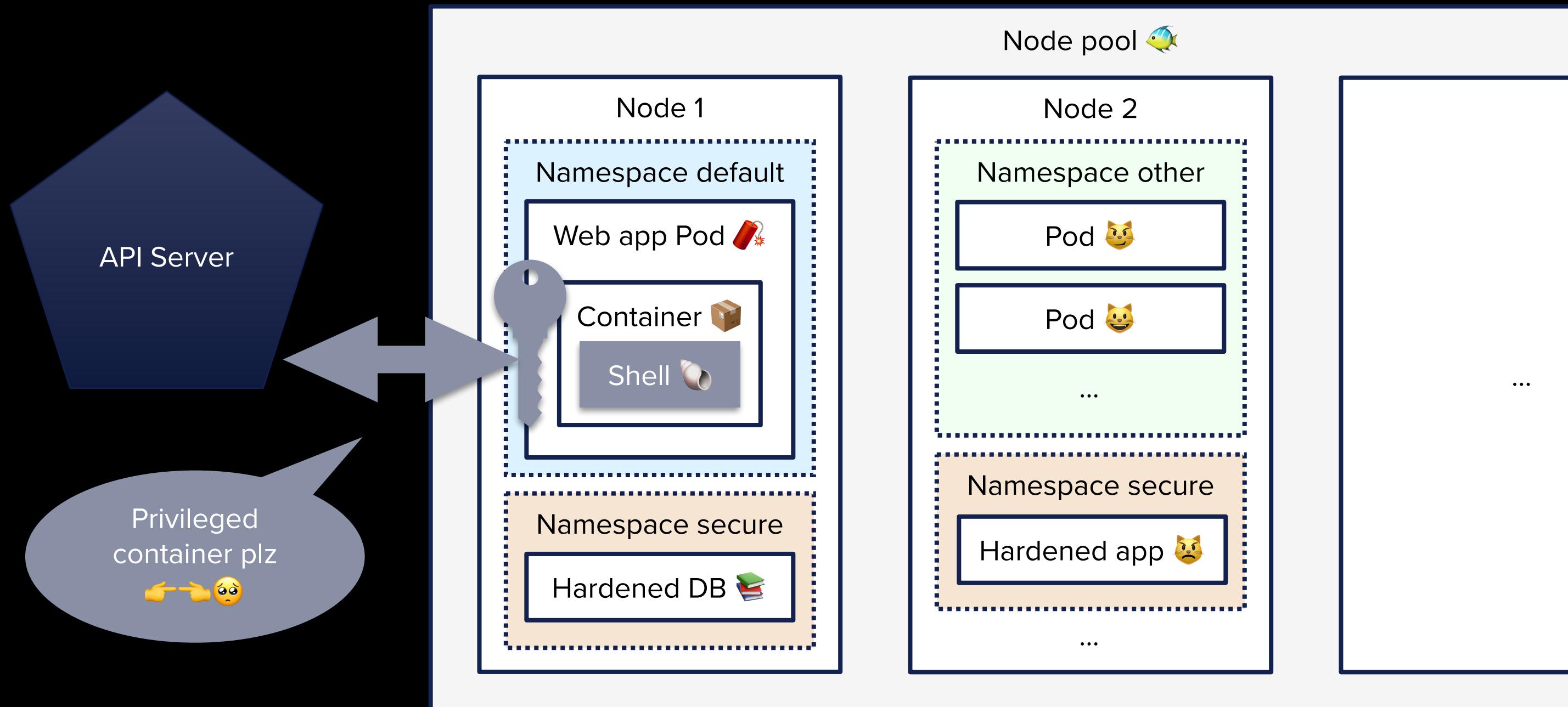
What happened? - First issue: vulnerability in software



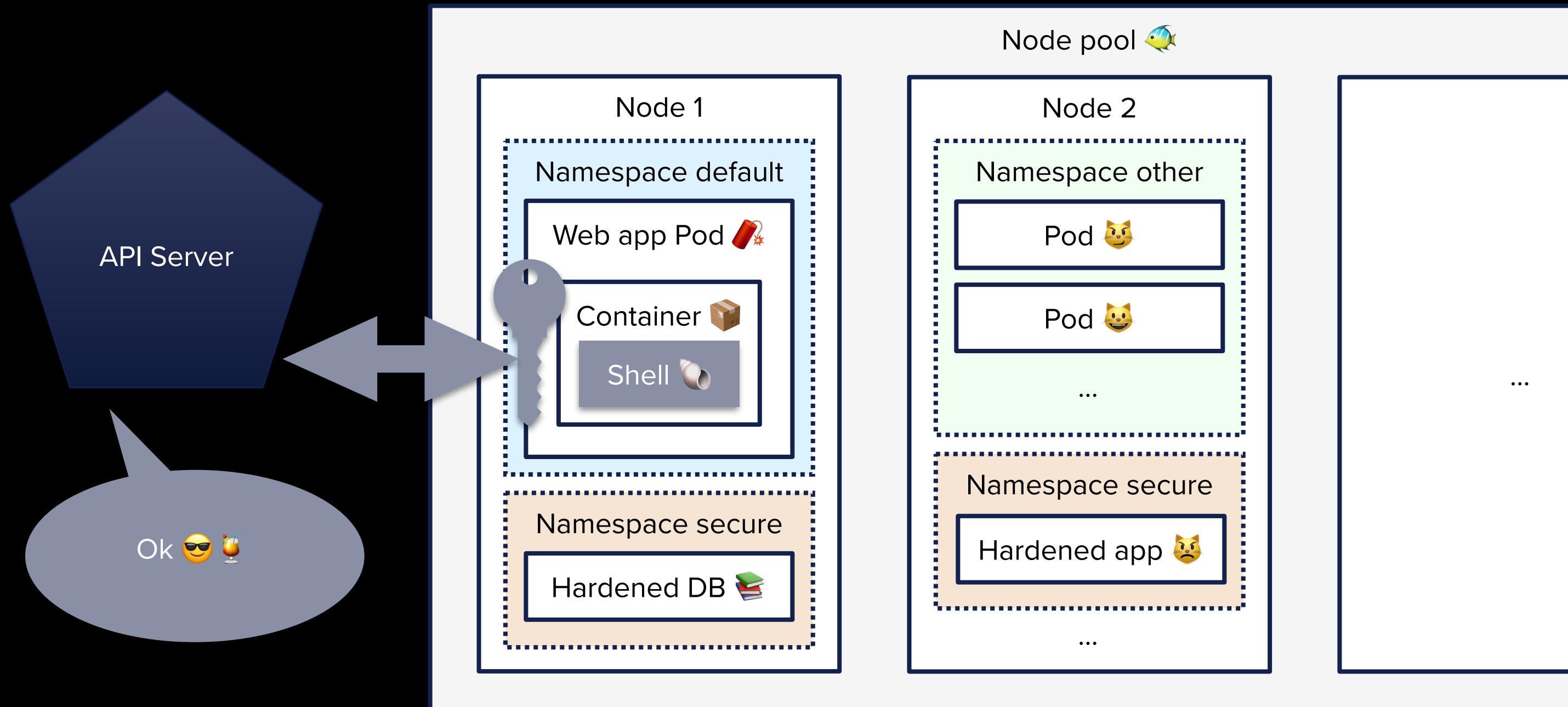
What happened? - Second issue: credentials in container



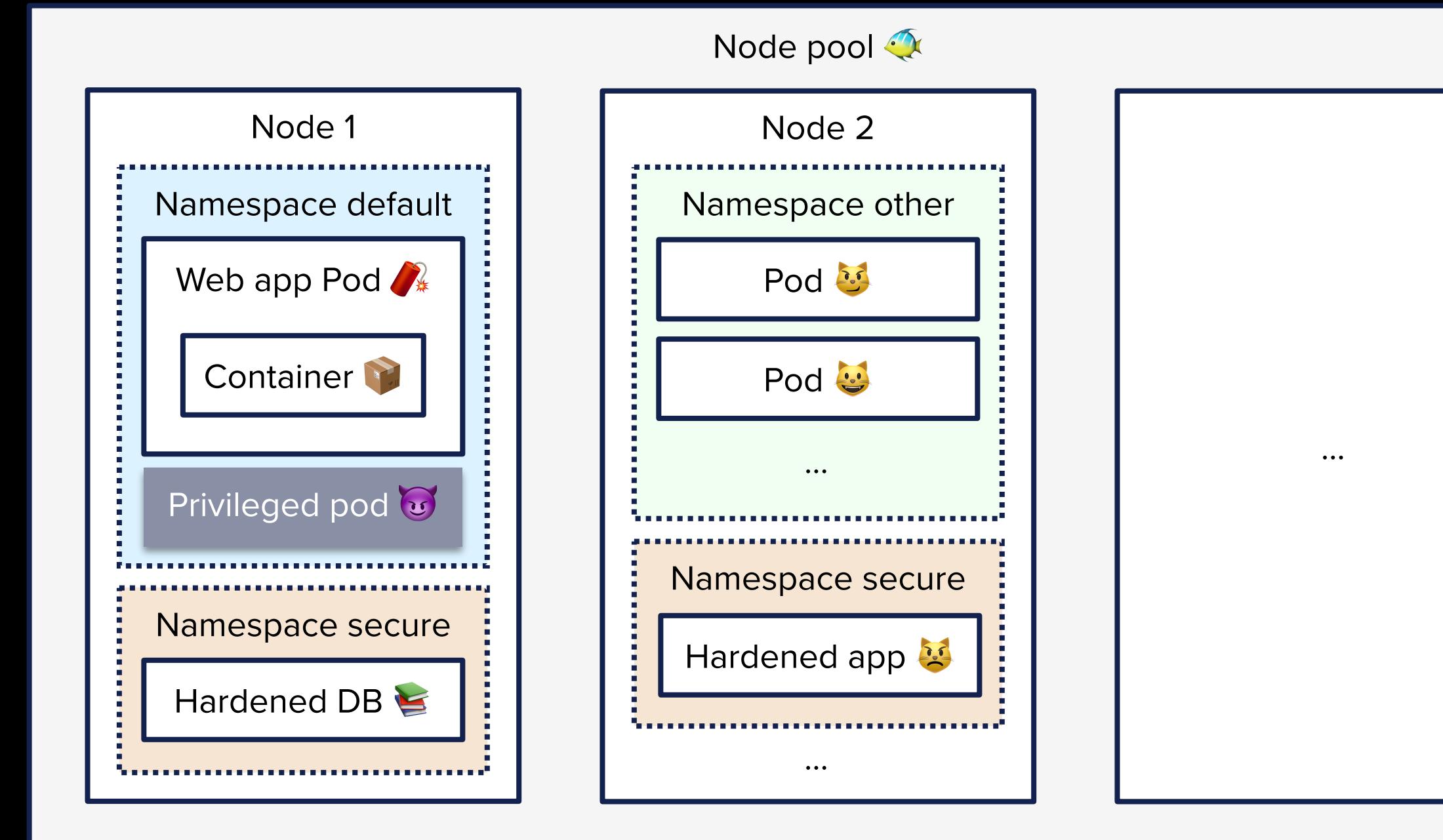
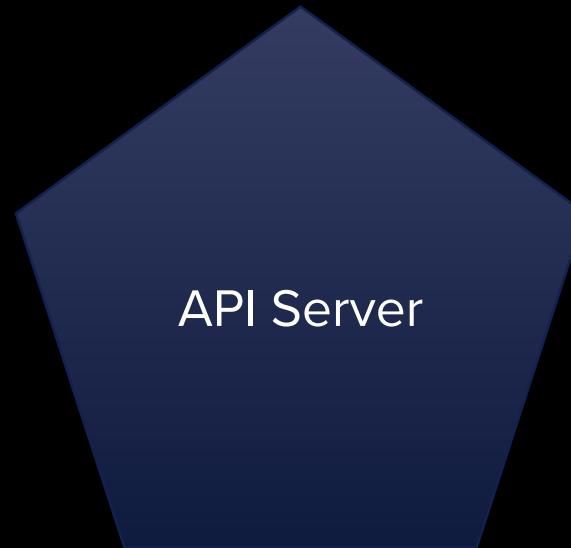
What happened? - Third issue: no admission control



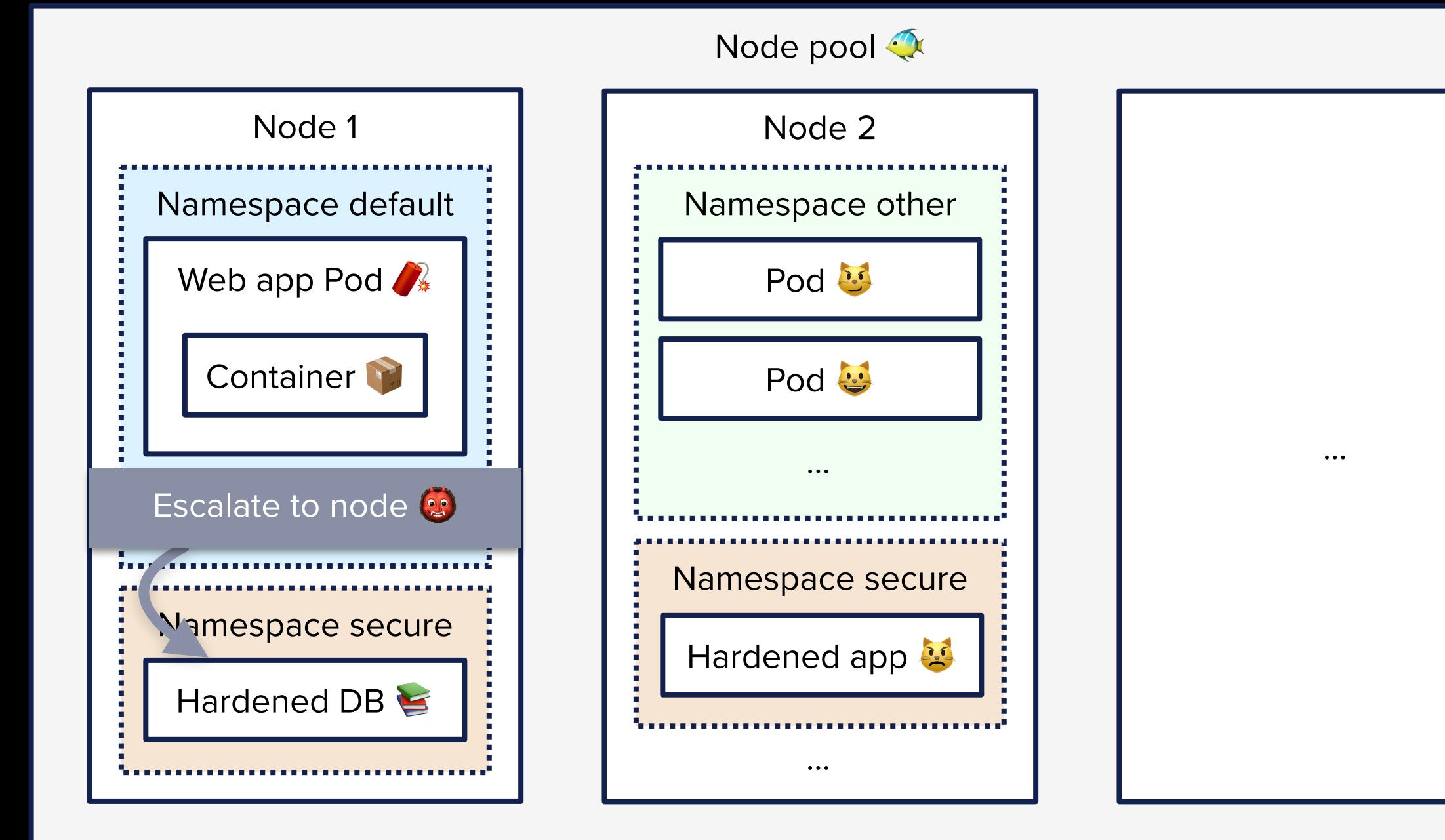
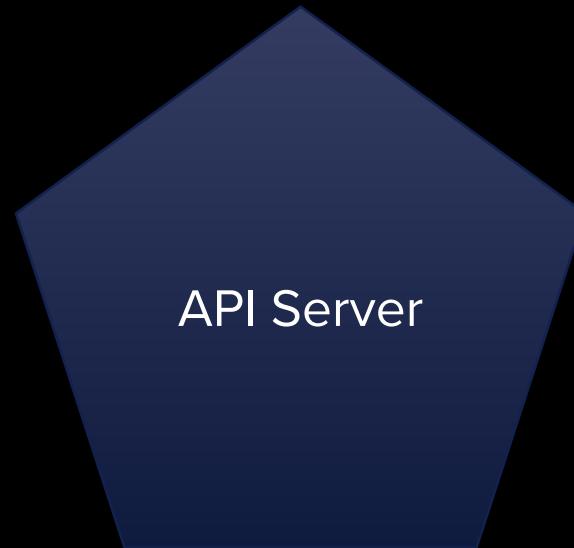
What happened? - Third issue: no admission control



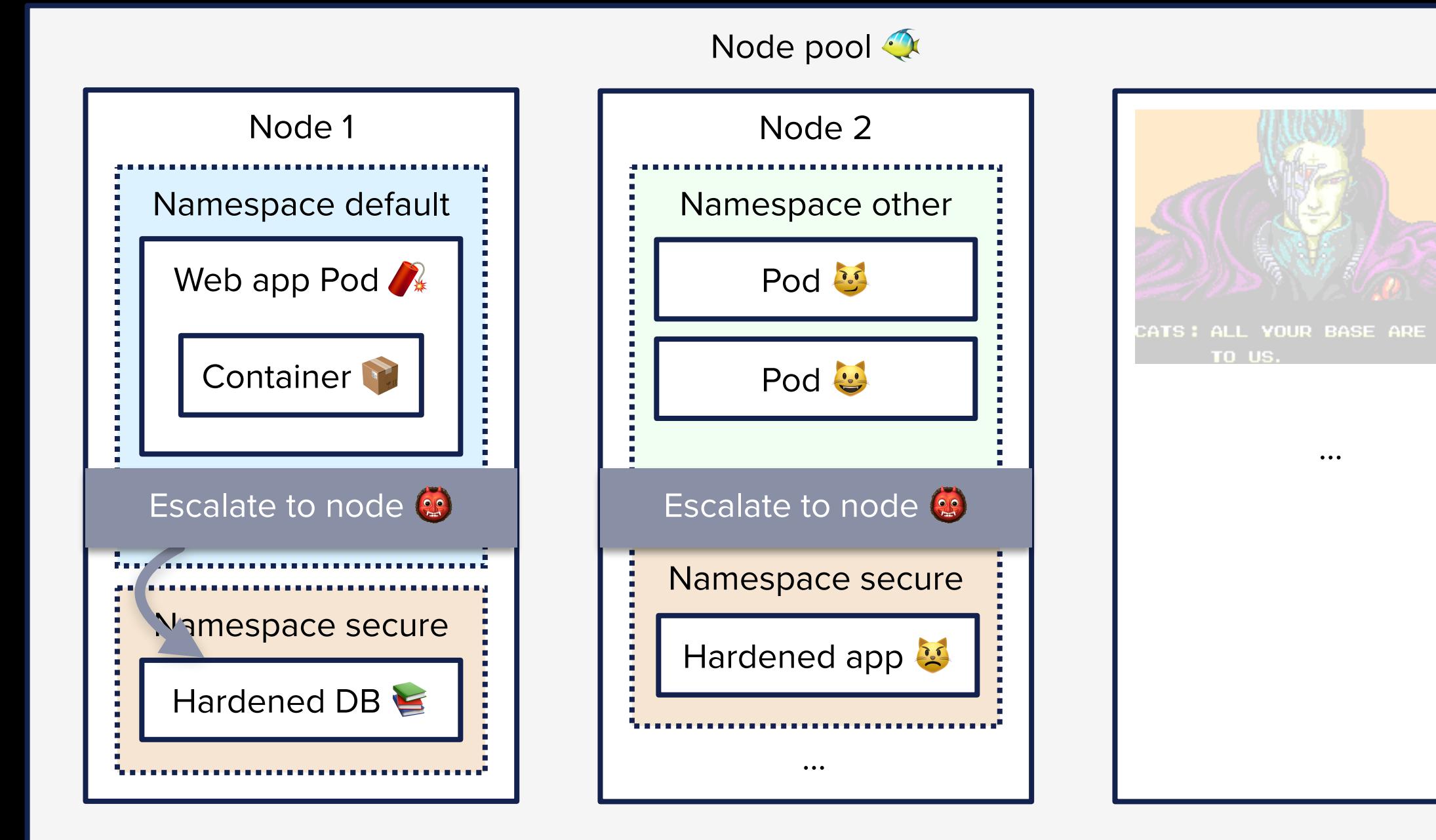
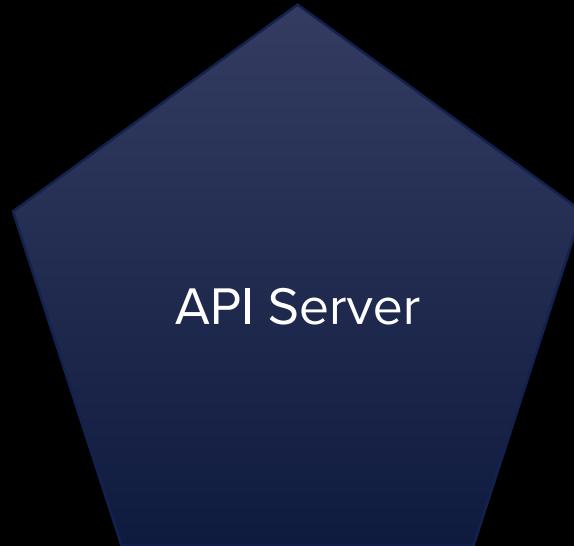
What happened? - Third issue: no admission control



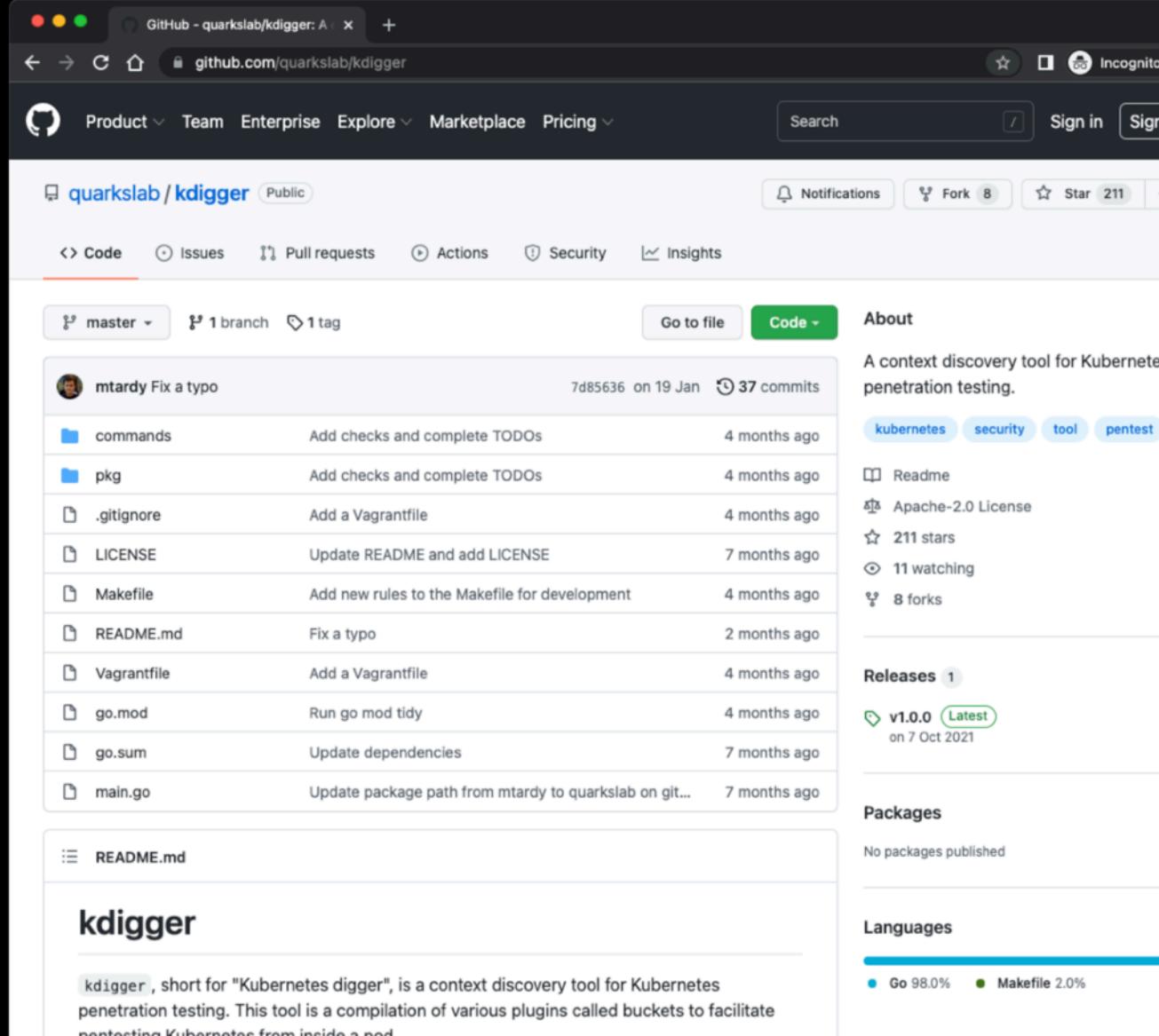
What happened? - Third issue: no admission control



What happened? - Third issue: no admission control



If you want to experiment by yourself



quarkslab / kdigger Public

About
A context discovery tool for Kubernetes penetration testing.

Code

- master · 1 branch · 1 tag
- Go to file · Code ·
- mtardy Fix a typo · 7d85636 on 19 Jan · 37 commits
- commands · Add checks and complete TODOs · 4 months ago
- pkg · Add checks and complete TODOs · 4 months ago
- .gitignore · Add a Vagrantfile · 4 months ago
- LICENSE · Update README and add LICENSE · 7 months ago
- Makefile · Add new rules to the Makefile for development · 4 months ago
- README.md · Fix a typo · 2 months ago
- Vagrantfile · Add a Vagrantfile · 4 months ago
- go.mod · Run go mod tidy · 4 months ago
- go.sum · Update dependencies · 7 months ago
- main.go · Update package path from mtardy to quarkslab on git... · 7 months ago

Releases 1

- v1.0.0 · Latest · on 7 Oct 2021

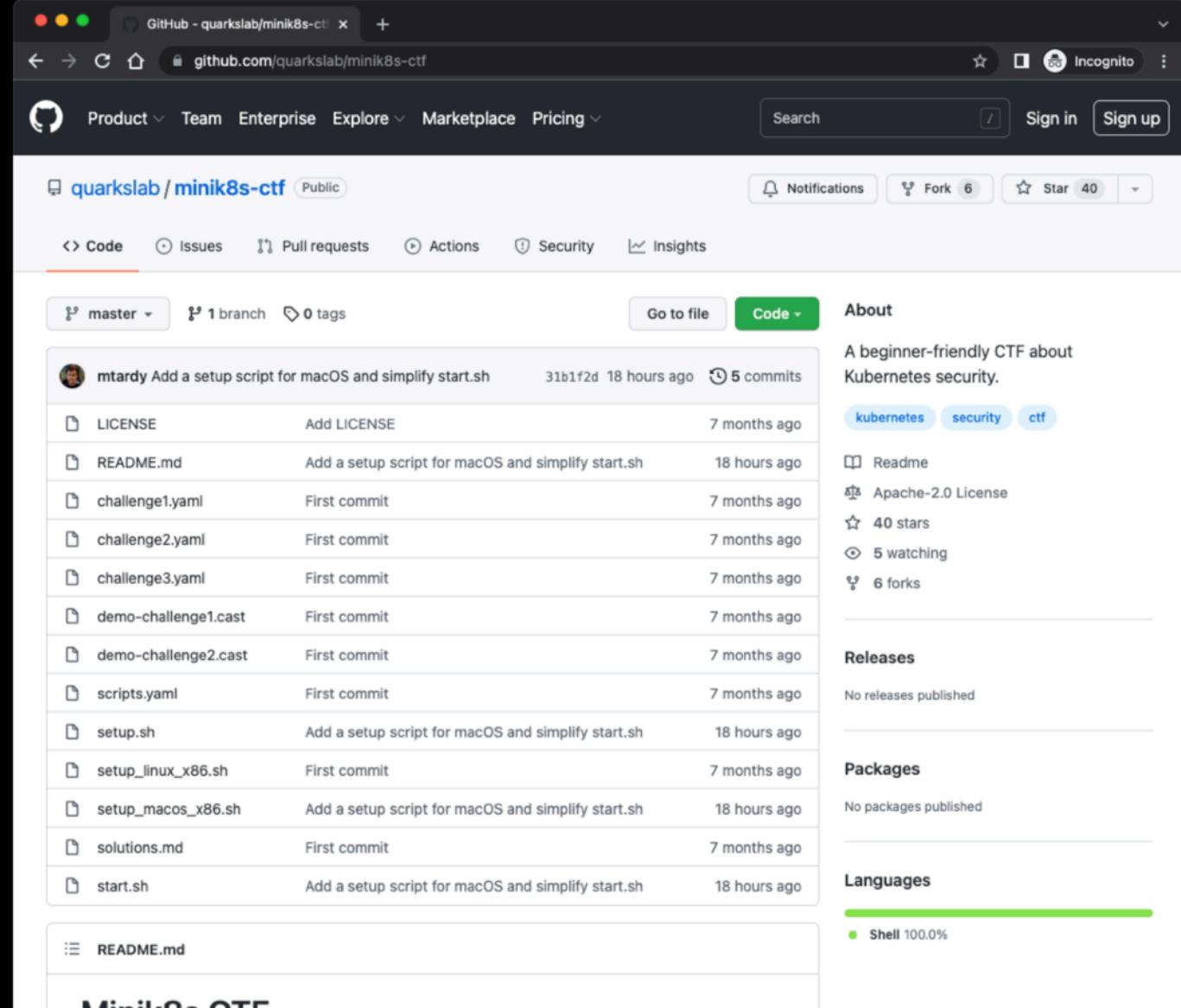
Packages
No packages published

Languages

Go 98.0% · Makefile 2.0%

kdigger

kdigger, short for "Kubernetes digger", is a context discovery tool for Kubernetes penetration testing. This tool is a compilation of various plugins called buckets to facilitate pentesting Kubernetes from inside a pod.



quarkslab / minik8s-ctf Public

About
A beginner-friendly CTF about Kubernetes security.

Code

- master · 1 branch · 0 tags
- Go to file · Code ·
- mtardy Add a setup script for macOS and simplify start.sh · 31b1f2d 18 hours ago · 5 commits
- LICENSE · Add LICENSE · 7 months ago
- README.md · Add a setup script for macOS and simplify start.sh · 18 hours ago
- challenge1.yaml · First commit · 7 months ago
- challenge2.yaml · First commit · 7 months ago
- challenge3.yaml · First commit · 7 months ago
- demo-challenge1.cast · First commit · 7 months ago
- demo-challenge2.cast · First commit · 7 months ago
- scripts.yaml · First commit · 7 months ago
- setup.sh · Add a setup script for macOS and simplify start.sh · 18 hours ago
- setup_linux_x86.sh · First commit · 7 months ago
- setup_macos_x86.sh · Add a setup script for macOS and simplify start.sh · 18 hours ago
- solutions.md · First commit · 7 months ago
- start.sh · Add a setup script for macOS and simplify start.sh · 18 hours ago

Releases
No releases published

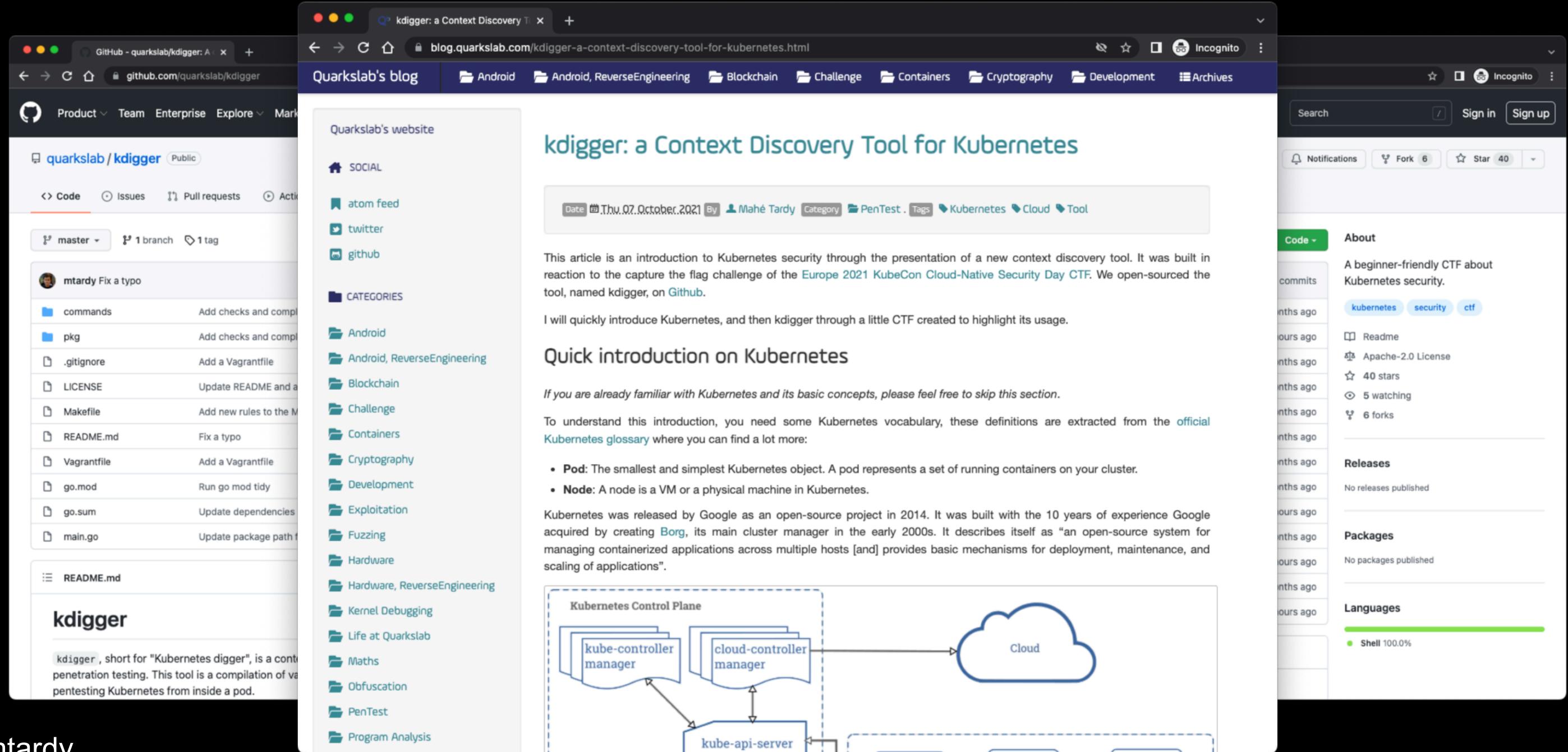
Packages
No packages published

Languages

Shell 100.0%

Minik8s CTF

If you want to experiment by yourself



The screenshot shows a web browser with three tabs open:

- Github - quarkslab/kdigger:** A GitHub repository page for the kdigger tool. It contains code files like `commands`, `pkg`, and `.gitignore`, and a `README.md` file.
- QuarksLab's blog:** A blog post titled "kdigger: a Context Discovery Tool for Kubernetes". The post discusses the tool's purpose, its creation in response to a CTF challenge, and provides a quick introduction to Kubernetes. It includes a diagram of the Kubernetes Control Plane.
- blog.quarkslab.com/kdigger-a-context-discovery-tool-for-kubernetes.html:** The same blog post from the QuarksLab website.

kdigger: a Context Discovery Tool for Kubernetes

Date: Thu.07.October.2021 By: Mahé Tardy Category: PenTest Tags: Kubernetes, Cloud, Tool

This article is an introduction to Kubernetes security through the presentation of a new context discovery tool. It was built in reaction to the capture the flag challenge of the Europe 2021 KubeCon Cloud-Native Security Day CTF. We open-sourced the tool, named kdigger, on Github.

I will quickly introduce Kubernetes, and then kdigger through a little CTF created to highlight its usage.

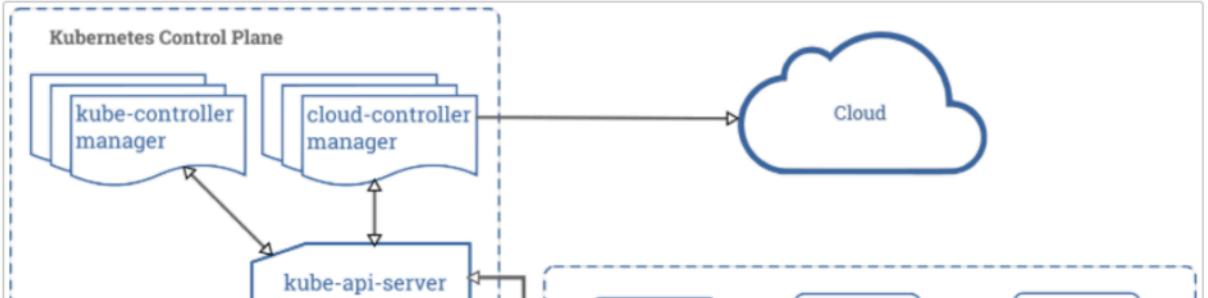
Quick introduction on Kubernetes

If you are already familiar with Kubernetes and its basic concepts, please feel free to skip this section.

To understand this introduction, you need some Kubernetes vocabulary, these definitions are extracted from the official [Kubernetes glossary](#) where you can find a lot more:

- **Pod:** The smallest and simplest Kubernetes object. A pod represents a set of running containers on your cluster.
- **Node:** A node is a VM or a physical machine in Kubernetes.

Kubernetes was released by Google as an open-source project in 2014. It was built with the 10 years of experience Google acquired by creating [Borg](#), its main cluster manager in the early 2000s. It describes itself as “an open-source system for managing containerized applications across multiple hosts [and] provides basic mechanisms for deployment, maintenance, and scaling of applications”.



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](https://twitter.com/mtardy_)
- mtardy@quarkslab.com