



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

When Multitenancy Goes Wrong

A Deep Dive into kcp's First CVE



Your Speakers



Marko Mudrinić

Senior Software Engineer



Marvin Beckers

Cloud Software Engineer



KUBERNETIC



Table of Contents

- What is kcp?
- What is a virtual workspace?
 - A look at the virtual workspace storage layer
- How did we discover the vulnerability?
- CVE-2025-29922 in-depth
- The importance of a security process
- Wrapping up



KubeCon



CloudNativeCon

Europe 2026

What is kcp?

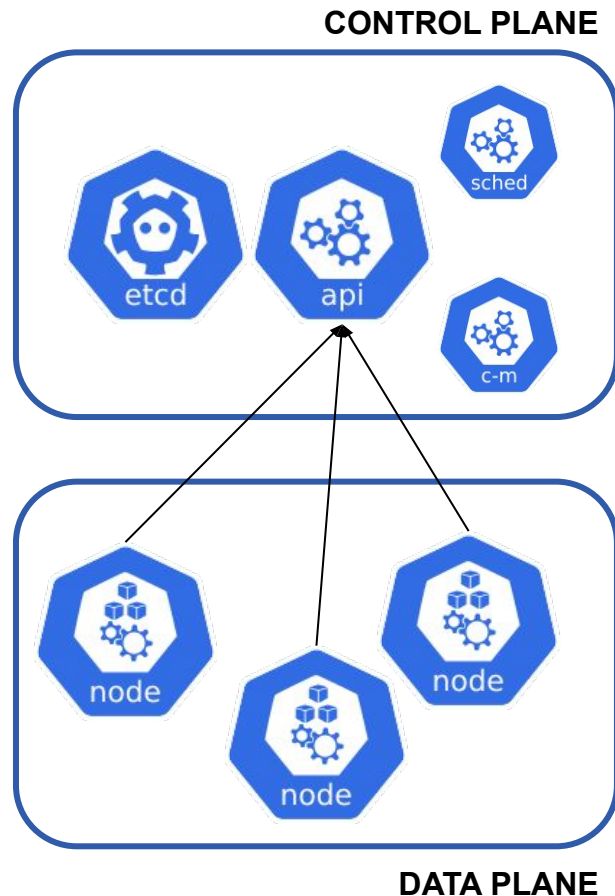


kcp is a multi-tenant horizontally
scalable **control plane** for
Kubernetes-like APIs.

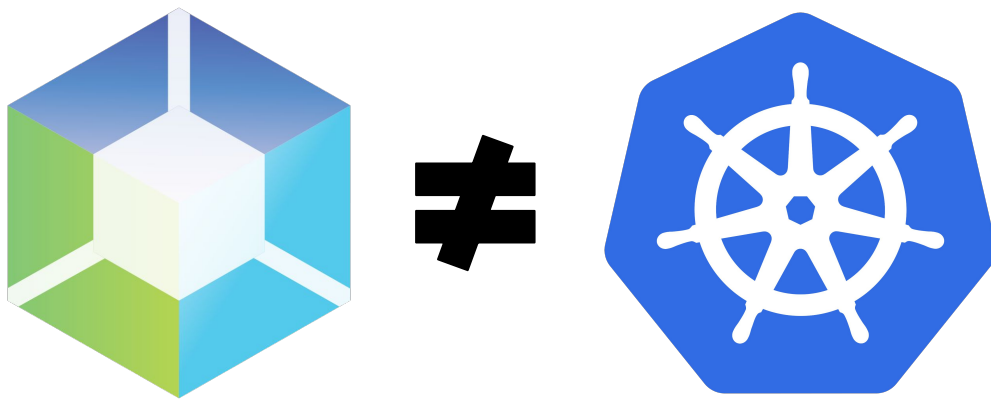


What is a Control Plane?

- Control planes make administrative decisions for a system
- They don't execute the actual workload but instruct what is often called the *data plane* to do something
- They **orchestrate** a system
- Kubernetes is a Control Plane for running containers



What Do You Mean, **Kubernetes-like APIs**?



That Being Said ...

○ ○ ○

```
1 kubectl get configmaps
2 No resources found in default namespace.
3
4 export KUBECONFIG=.kcp/admin.kubeconfig
5
6 kubectl get configmaps
7 No resources found in default namespace.
```

What is **kcp** for?

- Internal (Developer) Platforms (IDPs)
- (Internal) SaaS control plane orchestrating the service across multiple regions
- REST API for customer orders (of e.g. IaaS, PaaS, SaaS)
- Meshing service providers (“Platform Mesh”)



KubeCon



CloudNativeCon

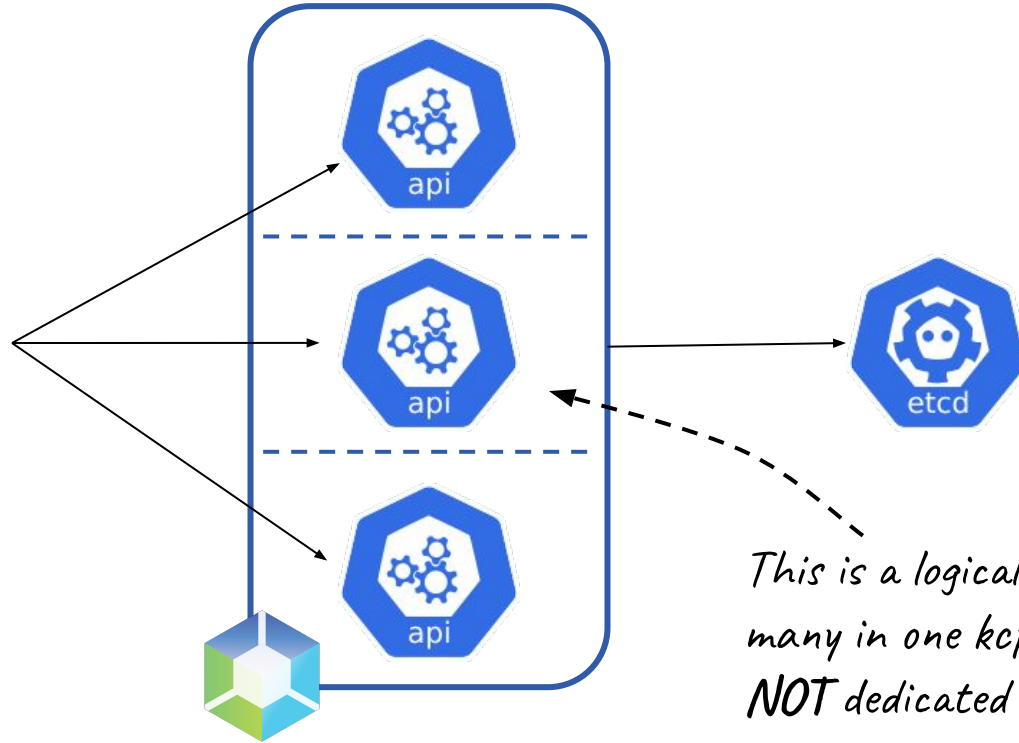
Europe 2026

What is a Virtual Workspace?



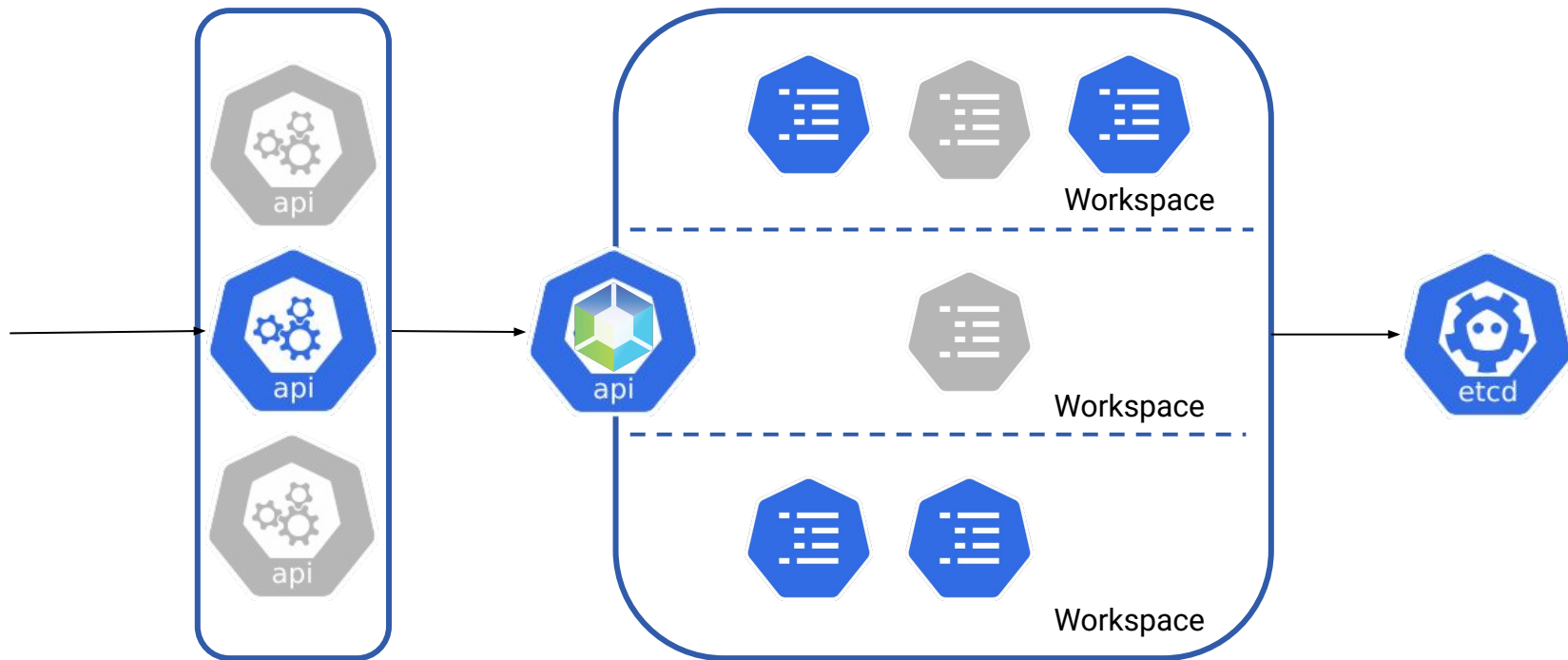
Workspaces

\$ kubectl



*This is a logical “cluster” of many in one kcp process, **NOT** dedicated apiservers*

Sort of Workspaces, but ✨ Virtual ✨



Under the Hood

○ ○ ○

```
1 &virtualdynamic.DynamicVirtualWorkspace{
2     RootPathResolver: framework.RootPathResolverFunc(func(urlPath string,
3         ctx context.Context) (accepted bool, prefixToStrip string,
4         completedContext context.Context) { ... })),
5
6     ReadyChecker: framework.ReadyFunc(func() error { ... })),
7
8     BootstrapAPISetManagement: func(mainConfig
9         genericapiserver.CompletedConfig) (apidefinition.APIDefinitionSetGetter,
10        error) { ... },
11
12     Mutator:      apiExportAdmission,
13     Validator:    apiExportAdmission,
14 }
```

Parses information
from the URL.

Admission (Mutation &
Validation).

Getter for all information
required to serve an API
(schema, storage, etc)



KubeCon



CloudNativeCon

Europe 2026

How Did We Find It?

(Story Time)

Finding Security Vulnerabilities



Expectation



Reality

Read. Test. Report.

✨ and Stay Curious ✨

feature: Extend permission claims with verbs #3255

✓ Closed

Listed in ✓ #8



mjudeikis opened on Jan 18, 2025 · edited by mjudeikis

Edits ▾

Member



Feature Description

Extend permissions claims with verbs:

Description here: https://docs.google.com/document/d/1toEU8_tldfc-klWDgKfROyPVJGhhWDrr2cU0ayRj76I/edit?tab=t.0#heading=h.24igybjpe3n

Proposed Solution

https://docs.google.com/document/d/1toEU8_tldfc-klWDgKfROyPVJGhhWDrr2cU0ayRj76I/edit?tab=t.0#heading=h.24igybjpe3n

Related to [#1937](#)

Alternative Solutions

No response

Want to contribute?

☐ I would like to work on this issue.



Additional Context

No response

Create sub-issue ▾



KubeCon



CloudNativeCon

Europe 2026

@embik & @xmudrii

○ ○ ○

```
1 apiVersion: apis.kcp.io/v1alpha2
2 kind: APIExport
3 metadata:
4   name: example.kcp.io
5 spec:
6   resources:
7     - group: example.kcp.io
8       name: widgets
9       schema: v220801.widgets.example.kcp.io
10      storage:
11        crd: {}
12   permissionClaims:
13     - group: ""
14       resource: configmaps
```

○ ○ ○

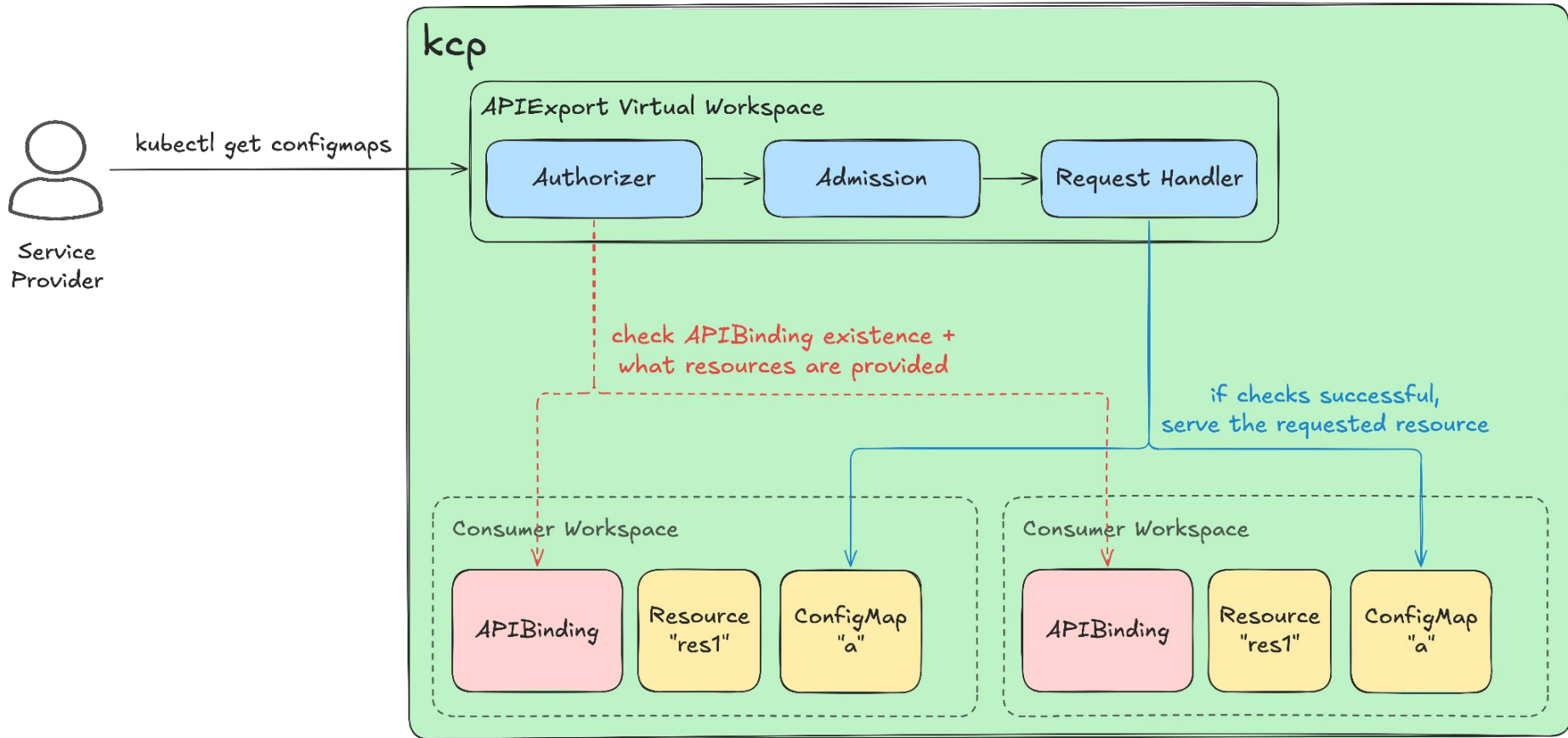
```
1 apiVersion: apis.kcp.io/v1alpha2
2 kind: APIBinding
3 metadata:
4   name: example.kcp.io
5 spec:
6   reference:
7     export:
8       name: example.kcp.io
9       path: "root:api-provider"
10  permissionClaims:
11    - resource: configmaps
12      state: Accepted
```

○ ○ ○

```
1 apiVersion: apis.kcp.io/v1alpha2
2 kind: APIExport
3 metadata:
4   name: example.kcp.io
5 spec:
6   resources:
7     - group: example.kcp.io
8       name: widgets
9       schema: v220801.widgets.example.kcp.io
10      storage:
11        crd: {}
12 permissionClaims:
13   - group: ""
14     resource: configmaps
15     verbs: ["get", "list", "watch"]
```

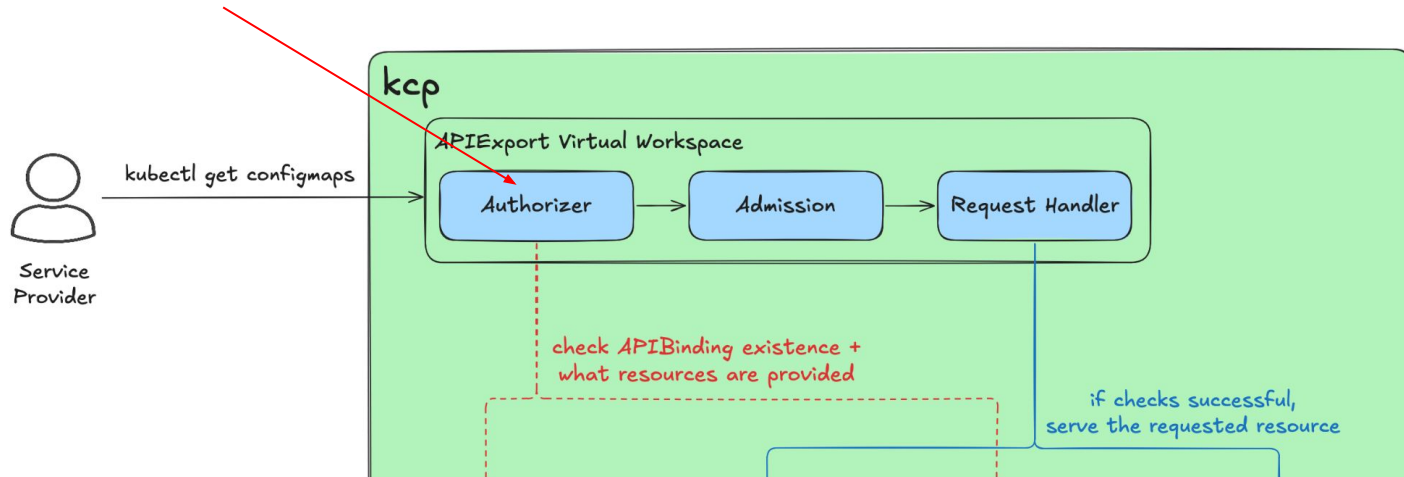
○ ○ ○

```
1 apiVersion: apis.kcp.io/v1alpha2
2 kind: APIBinding
3 metadata:
4   name: example.kcp.io
5 spec:
6   reference:
7     export:
8       name: example.kcp.io
9       path: "root:api-provider"
10 permissionClaims:
11   - resource: configmaps
12     state: Accepted
13     verbs: ["get", "list", "watch"]
```



How to solve the ticket? Simple!

- This all makes solving the mentioned ticket very easy!
- Just tap into the existing APIBinding checks (Authorizer) to verify if the requested verb/operation is permitted!



KubeCon



CloudNativeCon

Europe 2026

@embik & @xmudrii



How to solve the ticket? Simple!

- The issue is... **those checks do not actually exist*** in the code
 - Checks are **very well documented**, but they didn't actually exist
- There are APIBindings references in the APIExport Virtual Workspace code, but none of those references are relevant

```
~/Projects/src/github.com/kcp-dev/kcp/pkg/virtual/apiexport git #v0.26.1  
grep -Ri apibinding | wc -l # Before the fix  
18
```

- And that's how we got to CVE-2025-29922 ✨

* The APIBinding existence check was partially implemented at the storage layer, more about that later



KubeCon

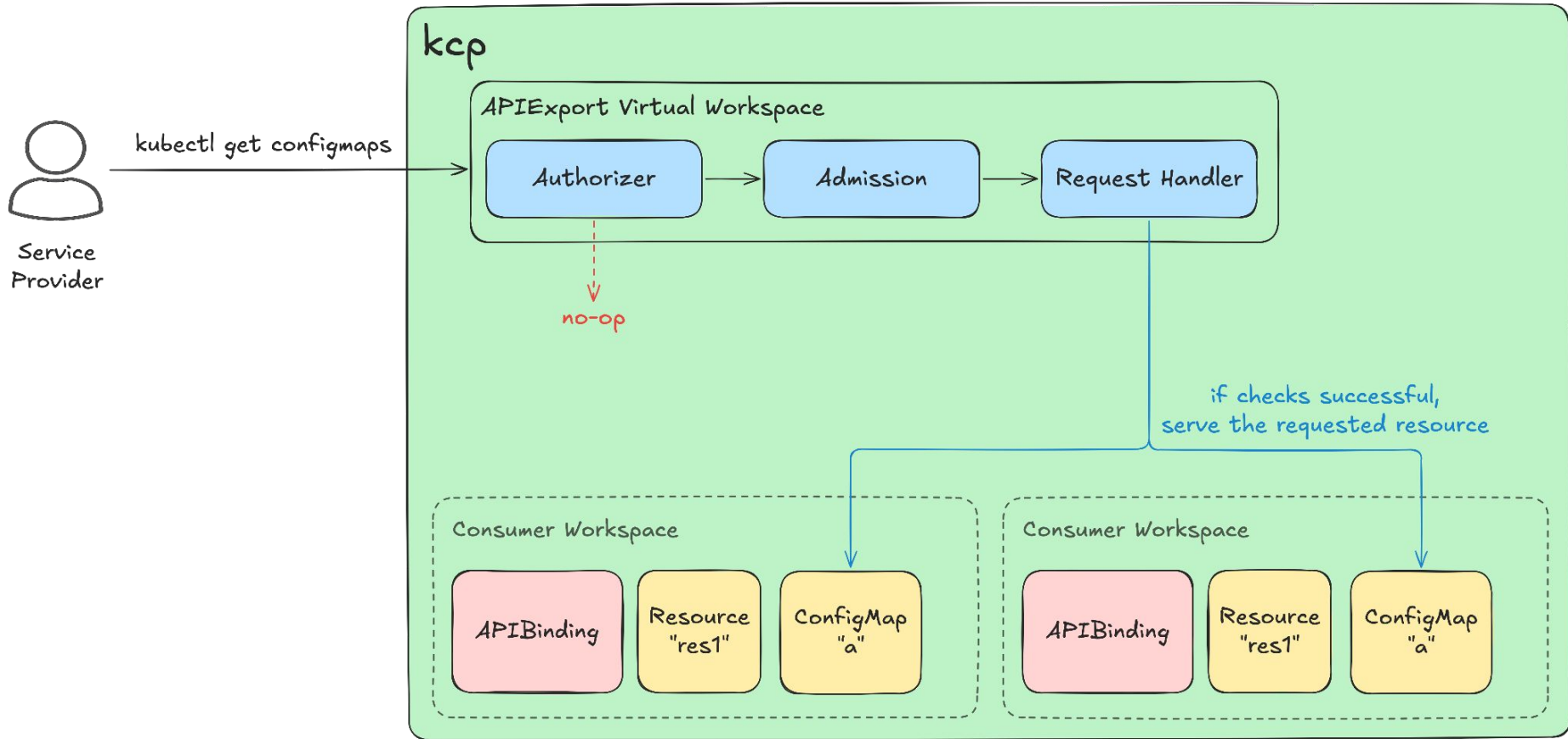


CloudNativeCon

Europe 2026

CVE-2025-29922





kcp



step 1: reconcile APIBindings

Consumer Workspace

APIBinding

Resource
"res1"

ConfigMap
"a"

Consumer Workspace

APIBinding

Resource
"res1"

ConfigMap
"a"



step 2: label resources provided by APIBinding

Consumer Workspace

APIBinding

Resource
"res1"

ConfigMap
"a"

Consumer Workspace

APIBinding

Resource
"res1"

ConfigMap
"a"



KubeCon



CloudNativeCon

Europe 2026

@embik & @xmudrii

Implications

- Missing **APIBinding checks*** means that, by default:
 - User can create a random APIExport in their own workspace
 - Via that APIExport, they can access the APIExport Virtual Workspace
 - Because there are no APIBinding checks, the user can access resources in any workspace
 - Resources => the resource that is exported by the APIExport and resources specified in PermissionClaims

Limitations

- Luckily, there are some limitations:
 - You need to know the target workspace (logical cluster) ID, which is random
 - Only **CREATE** and **DELETE** operations are supported
 - Other operations require objects to have the appropriate label applied by kcp
 - You can't claim any resource via PermissionClaims
 - You can only built-in APIs and resources provided by other APIExports (but you need to know an identity hash of those APIExports)

Demo Time

The Initial Analysis

- Initially, we thought that this is bad, but not as bad as it can be (how wrong we were!)
- You can CREATE and DELETE resources, but you cannot access any resource
- => You can still affect integrity of an environment(!), but you can't read or change data
- => Gives us a solid **5.4** base score for CVE

CVSS 3.1 Vector

AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*

Network (AV:N) Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*

Low (AC:L) High (AC:H)

Privileges Required (PR)*

None (PR:N) Low (PR:L) High (PR:H)

User Interaction (UI)*

None (UI:N) Required (UI:R)

Scope (S)*

Unchanged (S:U) Changed (S:C)

Impact Metrics

Confidentiality Impact (C)*

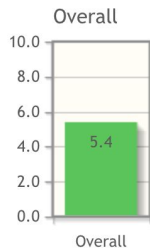
None (C:N) Low (C:L) High (C:H)

Integrity Impact (I)*

None (I:N) Low (I:L) High (I:H)

Availability Impact (A)*

None (A:N) Low (A:L) High (A:H)



CVSS Base Score: 5.4

Impact Subscore: 2.5

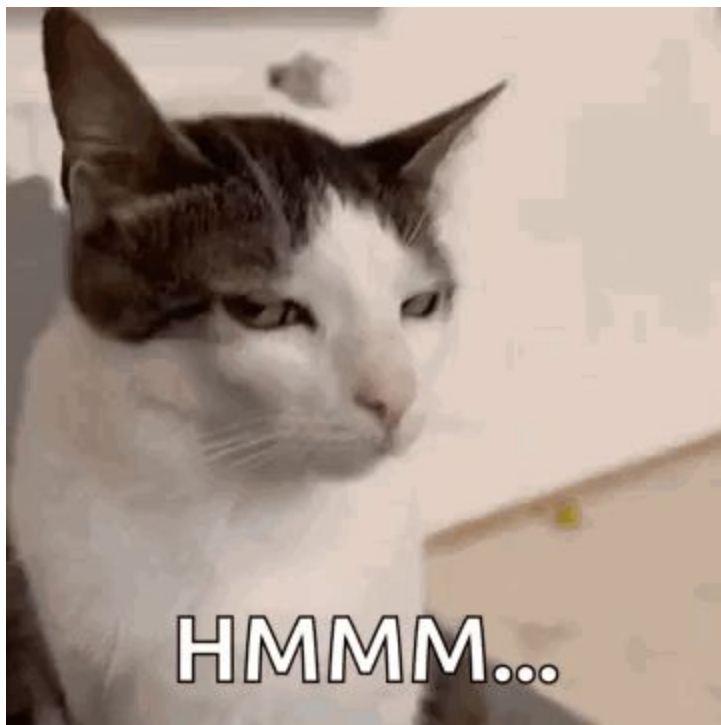
Exploitability Subscore: 2.8

CVSS Temporal Score: NA

CVSS Environmental Score: NA

Modified Impact Subscore: NA

Overall CVSS Score: 5.4



A Few Questions ...

- What if I create an APIExport that exports some resource, and has **ClusterRoles** and **ClusterRoleBindings** claimed via PermissionClaims?
- Can I create a cluster-admin role in every workspace?
- Can I just create a cluster-admin role in the **root** workspace?

Demo Time

(again)

So ...

- What if I create an APIExport that exports some resource, and has **ClusterRoles** and **ClusterRoleBindings** claimed via PermissionClaims?
- Can I create a cluster-admin role in every workspace across the kcp setup?
- Or can I just create a cluster-admin role in the root workspace and get admin access to every single workspace?

The answer is... unfortunately, yes!

Updated Analysis

- Now you can access any workspace, any resource, and do any operation!
- => Gives us a very bad 9.6 base score for CVE

CVSS 3.1 Vector

[AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N](#)

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*

Network (AV:N) Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*

Low (AC:L) High (AC:H)

Privileges Required (PR)*

None (PR:N) Low (PR:L) High (PR:H)

User Interaction (UI)*

None (UI:N) Required (UI:R)

Scope (S)*

Unchanged (S:U) Changed (S:C)

Impact Metrics

Confidentiality Impact (C)*

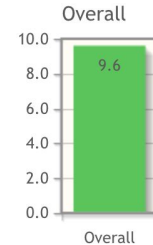
None (C:N) Low (C:L) High (C:H)

Integrity Impact (I)*

None (I:N) Low (I:L) High (I:H)

Availability Impact (A)*

None (A:N) Low (A:L) High (A:H)



CVSS Base Score: 9.6

Impact Subscore: 5.8

Exploitability Subscore: 3.1

CVSS Temporal Score: NA

CVSS Environmental Score: NA

Modified Impact Subscore: NA

Overall CVSS Score: 9.6



KubeCon



CloudNativeCon

Europe 2026

@embik & @xmudrii

Addressing The Vulnerability

- We used GitHub Security Advisory to **manage** it
- It allowed us to **fix** the vulnerability in a private fork
- We **pre-announced** the vulnerability (without details) and release date for patches about three days in advance
- We **released patch releases** with the fix and made the advisory public
- GitHub as CNA took care of assigning the CVE number and publishing it





KubeCon



CloudNativeCon

Europe 2026

The Importance of a Security Process

(duh)

**All projects have security vulnerabilities.
We just don't know about them yet.**

**We want our fellow maintainers to be
prepared.**

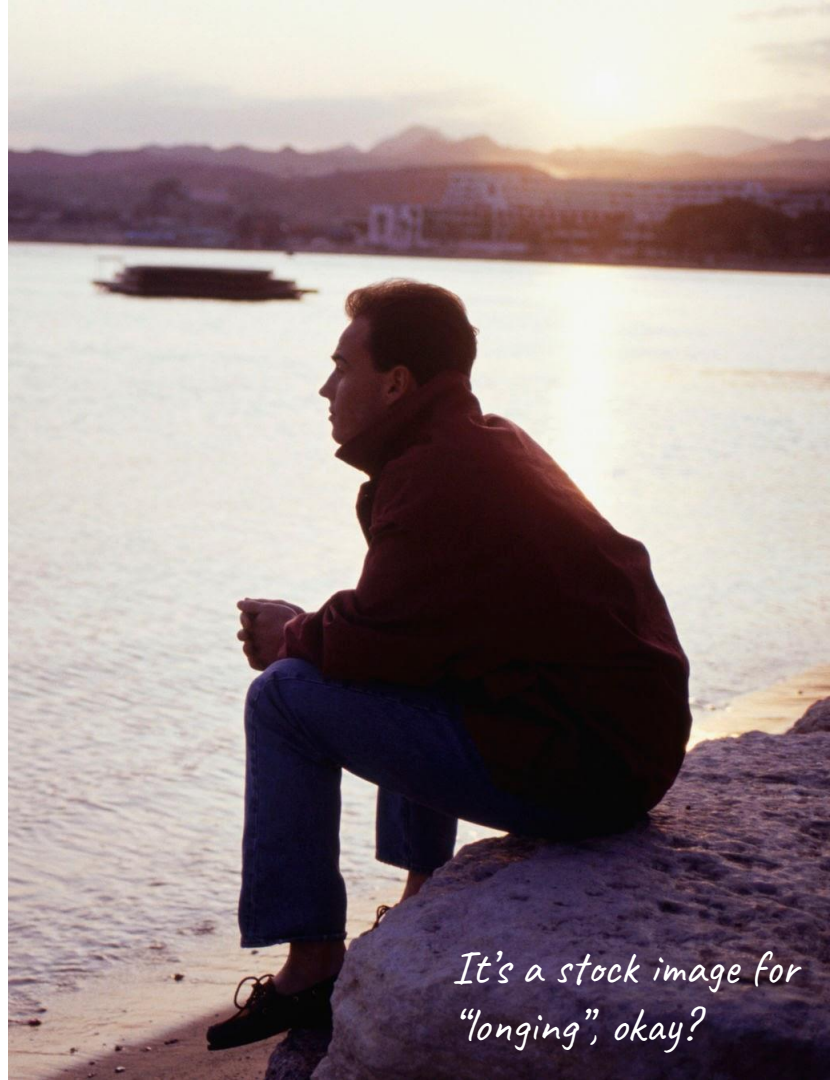
GitHub's Security Advisories

- GitHub is already a trusted platform
- The security advisory feature guided us through the process
- We could focus on fixing it
- The advisory on the repository and automatic dependabot alerts are amazing

 **Use GitHub for this!**

Things We're Still Missing

- We don't really know who uses kcp beyond our [ADOPTERS.md](#) file.
 - OSS working as designed!
- We don't have a list of **trusted vendors**. Who ships kcp as part of their product?
 - We need to build a vendor program like Kubernetes has.
- How do we automatically check for this kind of vulnerability?



*It's a stock image for
"longing", okay?*



KubeCon



CloudNativeCon

Europe 2026

Wrapping Up



Thanks for Listening!

Resources



<https://kcp.io>



[CVE-2025-29922](#)

This Talk



Kubernetes icons are from [here](#), licensed under CC-BY-4.0 by the Kubernetes community.



KubeCon



CloudNativeCon

Europe 2026

