KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA

KubeCon CloudNativeCon Europe 2022

Trampoline Pods

Node to Admin PrivEsc Built Into Popular K8s Platforms

Yuval Avrahami & Shaul Ben Hai, Palo Alto Networks

whoami

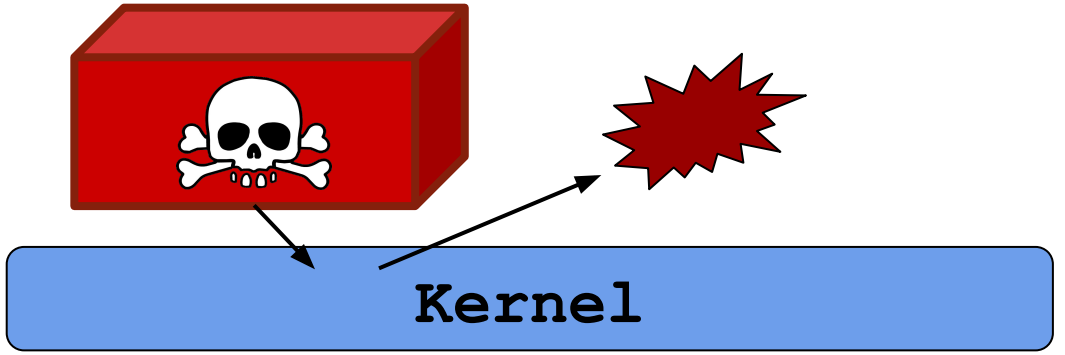
- Security researchers @ Prisma Cloud
- Vulnerability research in the cloud
- Threat hunting in the cloud
- NBA fans
  - May 1st CF prediction: Celtic vs Heat, Suns vs Warriors

Abstract

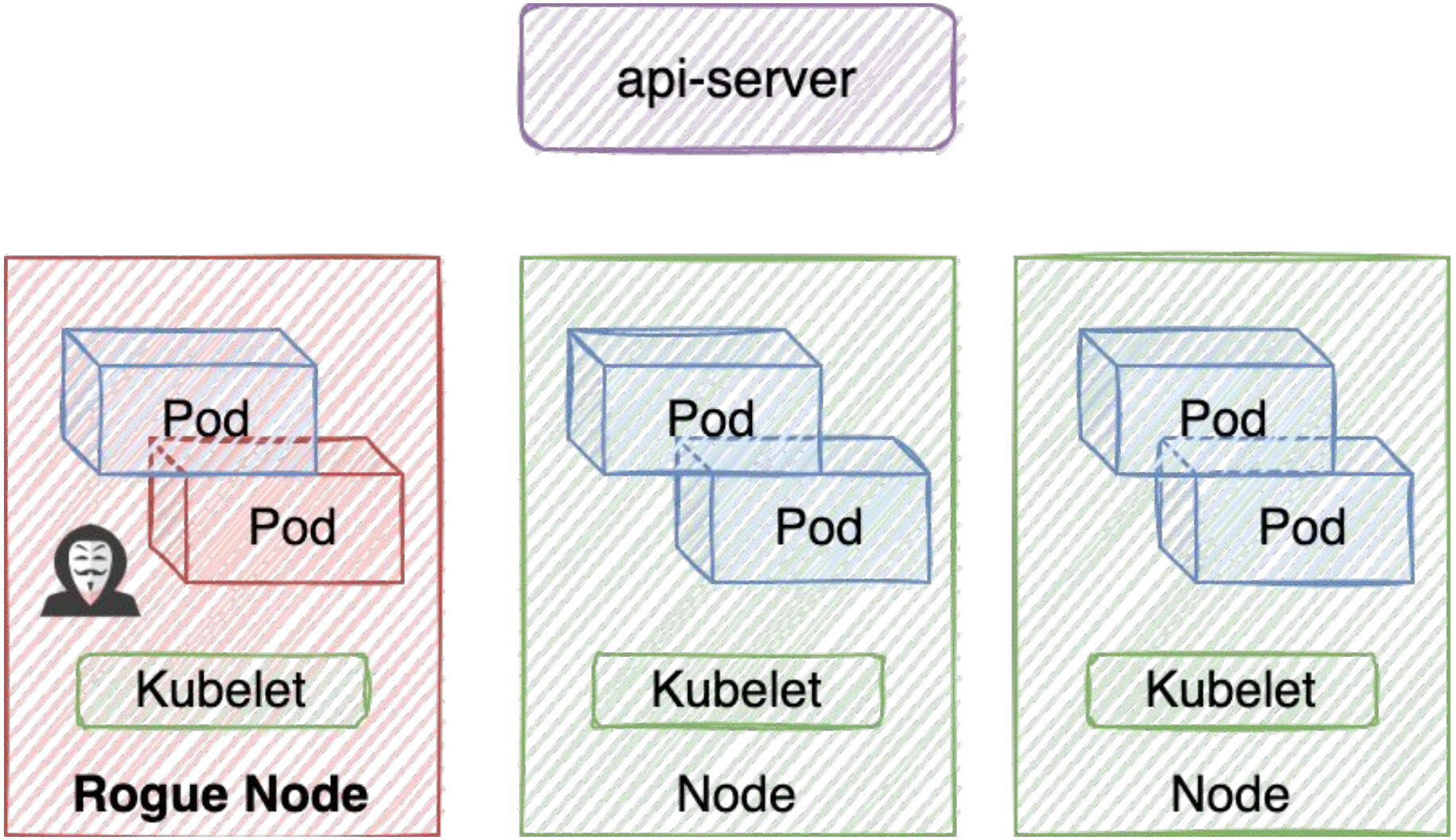
- What happens following a container escape?
- Trampoline pods
- Node→admin privilege escalation in popular K8s platforms
- Recommendations
- rbac-police

Container Escapes - An Inevitable Reality

- Containers are great for packaging & deploying software
- Weak security boundary
- Escapes will inevitably occur
  - **Vulns in 2022 alone**: containerd CVE-2022-23648, cri-o CVE-2022-0811, DirtyPipe, multiple kernel vulns @ Google's kctf
  - **Misconfigurations**: privileged containers, host mounts, etc
  - **In-the-wild malware**: Siloscape, TeamTNT
- **What's the impact?**



Obvious Impact: Compromised Node



Container Escape == Cluster Admin?

Terminology

- Admin

```
ya@demo:~$ kctl auth can-i "*" "*" --all-namespaces
yes
```

- Admin-equivalent

```
ya@demo:~$ kctl auth can-i list secrets -n kube-system
yes
```

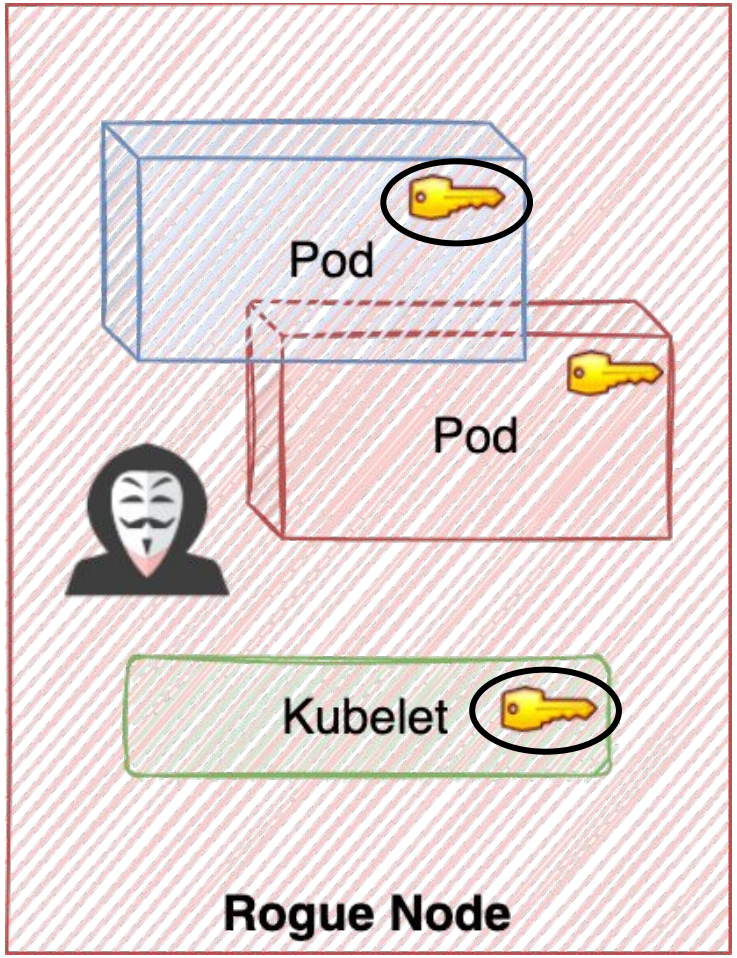
Few trivial steps

```
ya@demo:~$ kctl auth can-i "*" "*" --all-namespaces
yes
```

Credentials on a Rogue Node

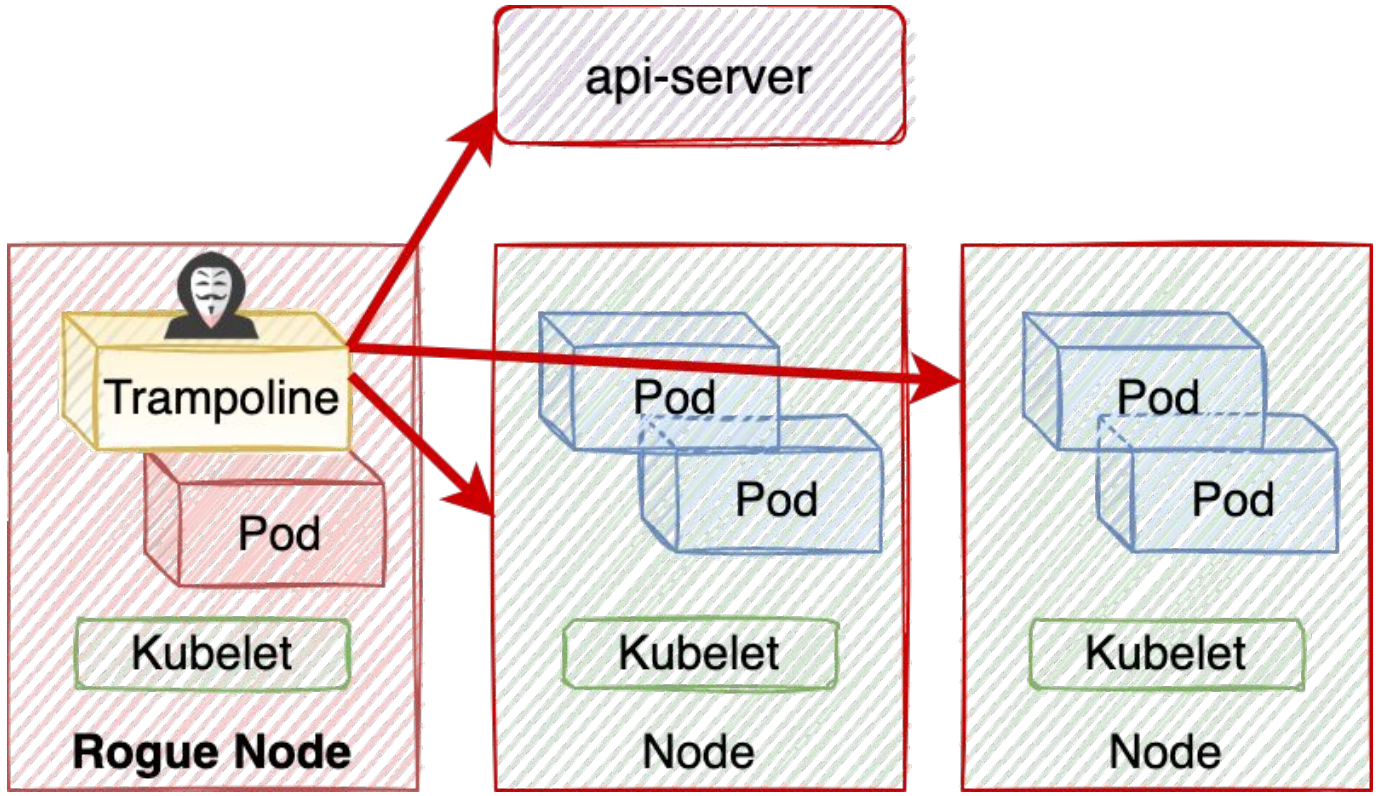
- Kubelet credentials
  - Restricted by NodeAuthorizer & NodeRestriction
    - KubeCon NA 19 talk: Walls Within Walls
  - By default, node != admin
- Neighboring pods' service accounts
  - Permissions vary

Node interesting permissions largely dictated by its pods' permissions



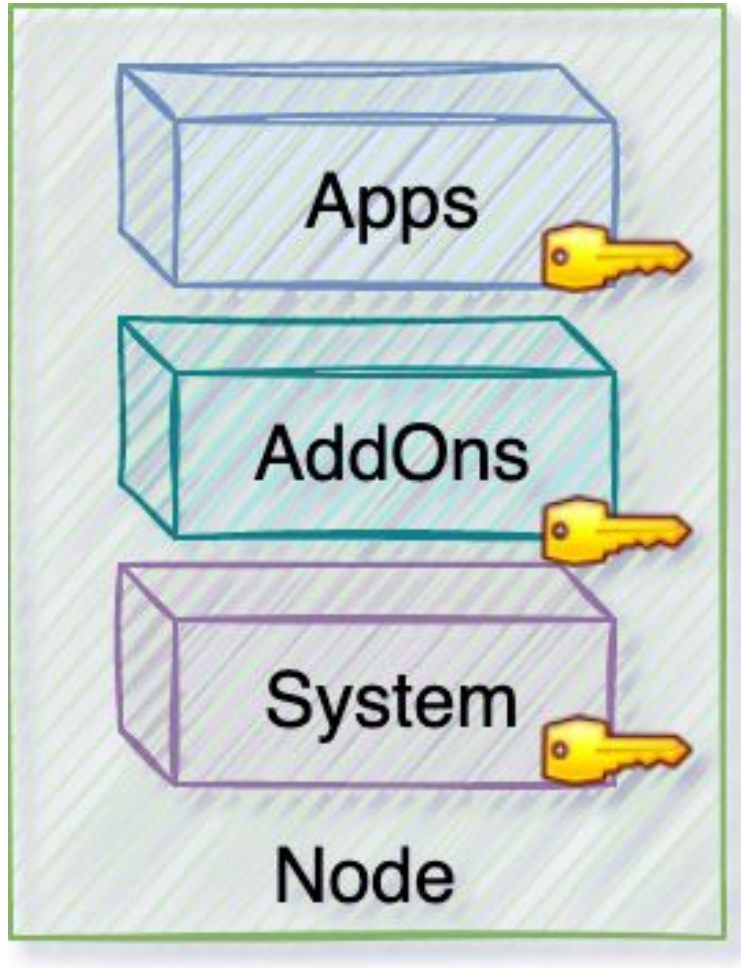
Trampoline Pods

- Powerful pods with enough permissions to bounce you around the cluster
  - Jump to other nodes
  - Reach higher privileges
  - Feel young again



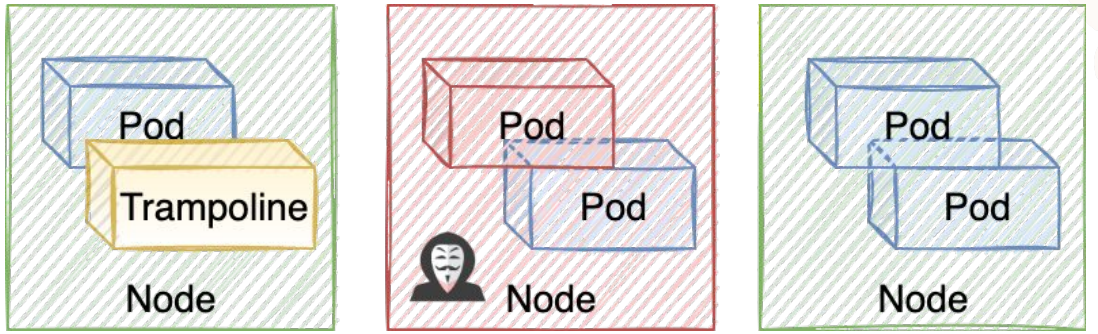
Know Your Nodes

- What pods run on your nodes?
  - Applications
  - Add-on (Prometheus, Istio)
  - System (kube-proxy, coredns)
- Blind spot: system and add-on pods
  - Permissions?
  - Often run as DaemonSets on all nodes

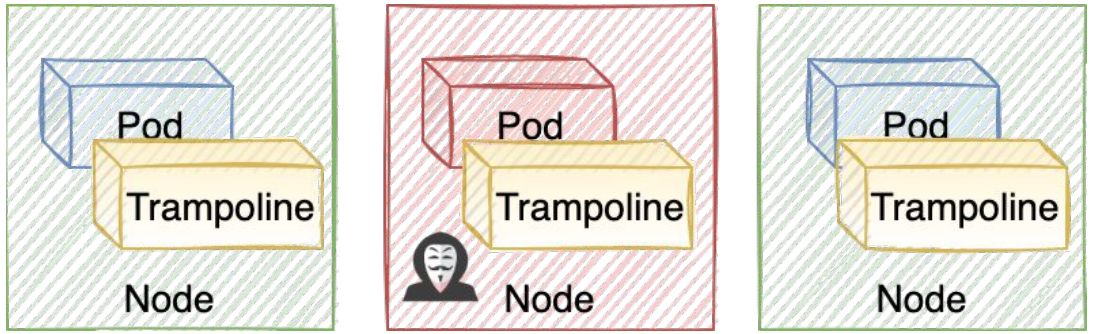


DaemonSets vs Non-DaemonSets

- **Non-DaemonSet Trampoline**
  - Attacker might hit jackpot

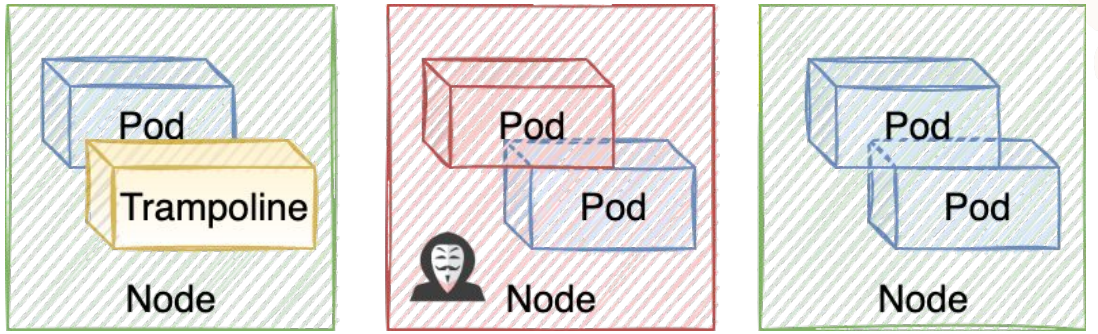


- **Trampolines DaemonSets**
  - Attacker **guaranteed** to hit jackpot

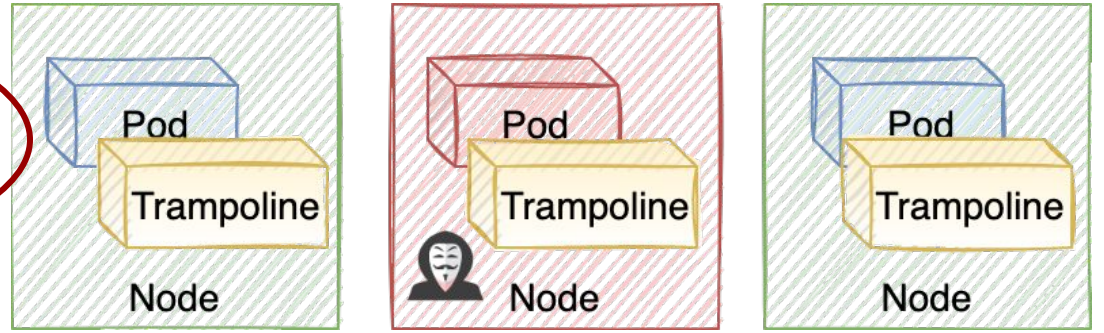


DaemonSets vs Non-DaemonSets

- **Non-DaemonSet Trampoline**
  - Attacker might hit jackpot



- **Trampolines DaemonSets**
  - Attacker **guaranteed** to hit jackpot



Real impact on "container escape == cluster admin"

KubeCon CloudNativeCon Europe 2022

Spotting Trampolines

What Makes a Pod Bouncy?

Which Permissions are Powerful?

- No public list
- Crucial to answer key questions
  - "Can my publicly-exposed pod escalate privileges?"
  - "Is this add-on asking for risky permissions?"
- Seemingly restricted permissions can be surprisingly powerful
- **Approach**: Define interesting attack classes & classify accordingly

Attack Classes

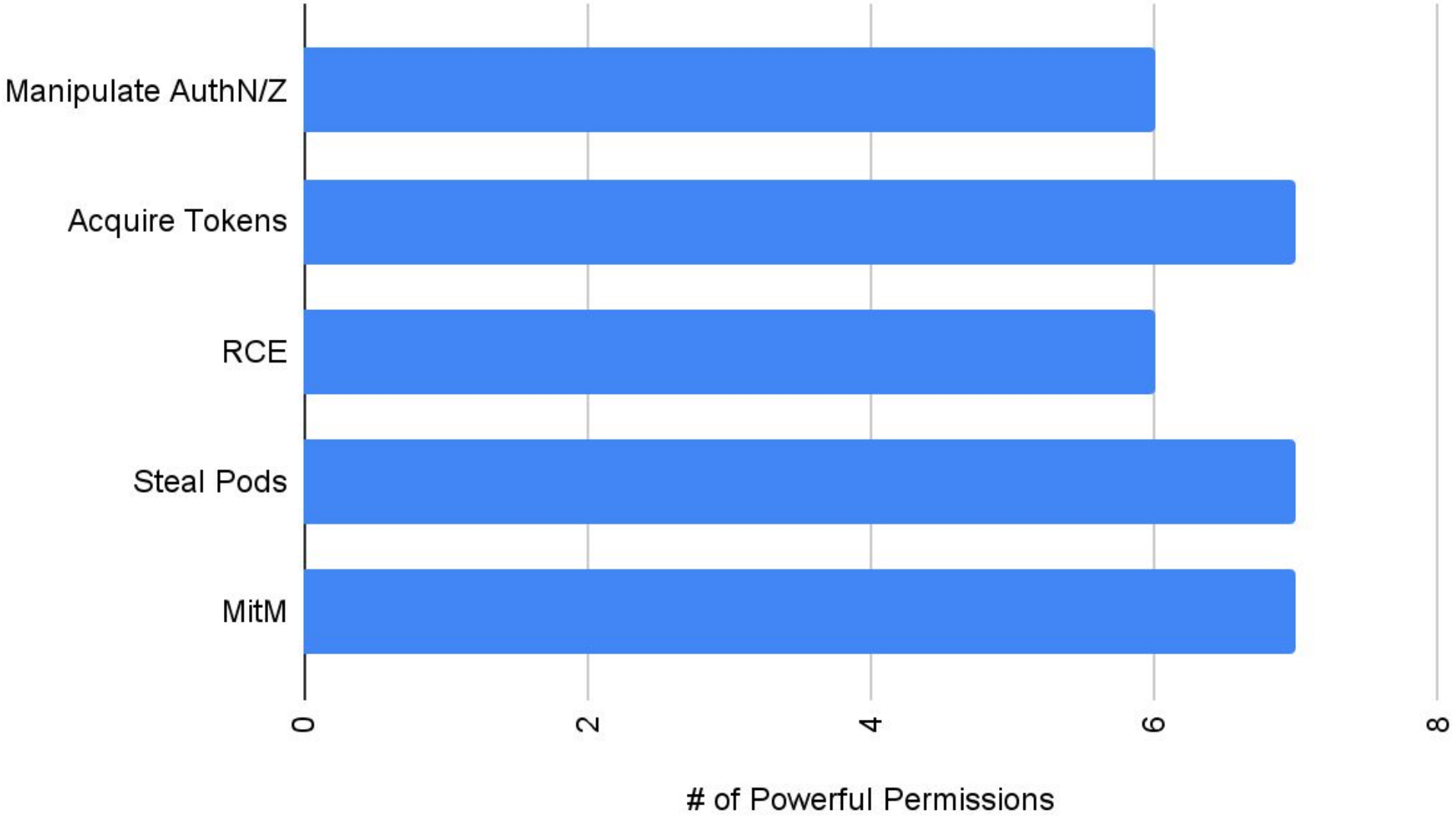
- **Manipulate AuthN / AuthZ**
  - Change identity or permissions (impersonate users)
- **Acquire Tokens**
  - Retrieve or issue a service account token (list secrets)
- **Remote Code Execution**
  - Execute code on pods / nodes (create pods/exec)
- **Steal Pods**
  - Move pods from one node to another (update nodes)
- **Meddler-in-The-Middle**
  - Intercept traffic (create endpointslices)

Classifying Powerful Permissions

Manipulate AuthN/Z	Acquire Tokens	RCE	Steal Pods	MitM
impersonate	list secrets	create pods/exec	modify nodes	control endpointslices
escalate	create secrets	update pods/ephemeralcontainers	modify nodes/status	modify endpoints
bind	create serviceaccounts/token	create nodes/proxy	create pods/eviction	modify services/status
approve signers	create pods	control pods	delete pods	modify pods/status
update certificatesigningrequests/approval	control pod controllers	control pod controllers	delete nodes	modify pods
control mutating webhooks	control validating webhooks	control mutating webhooks	modify pods/status	create services
	control mutating webhooks		modify pods	control mutating webhooks

- Unlikely includes all powerful perms, but it's a good start
- Yellow - permission isn't enough to carry out entire attack

Classifying Powerful Permissions



What Makes a Pod Bouncy?

- Trampolines are pods with permissions to:
  - Manipulate AuthN/AuthZ
  - Acquire Tokens
  - Remote Code Execution
  - Steal Pods
- Real shot at getting cluster admin

Recap

Recap

- Container escapes likely to continue happening
- Impact largely depends on presence of powerful pods
- Trampoline DaemonSets install powerful tokens onto every node
- **Escape == Admin? → Are There Trampoline DaemonSets?**

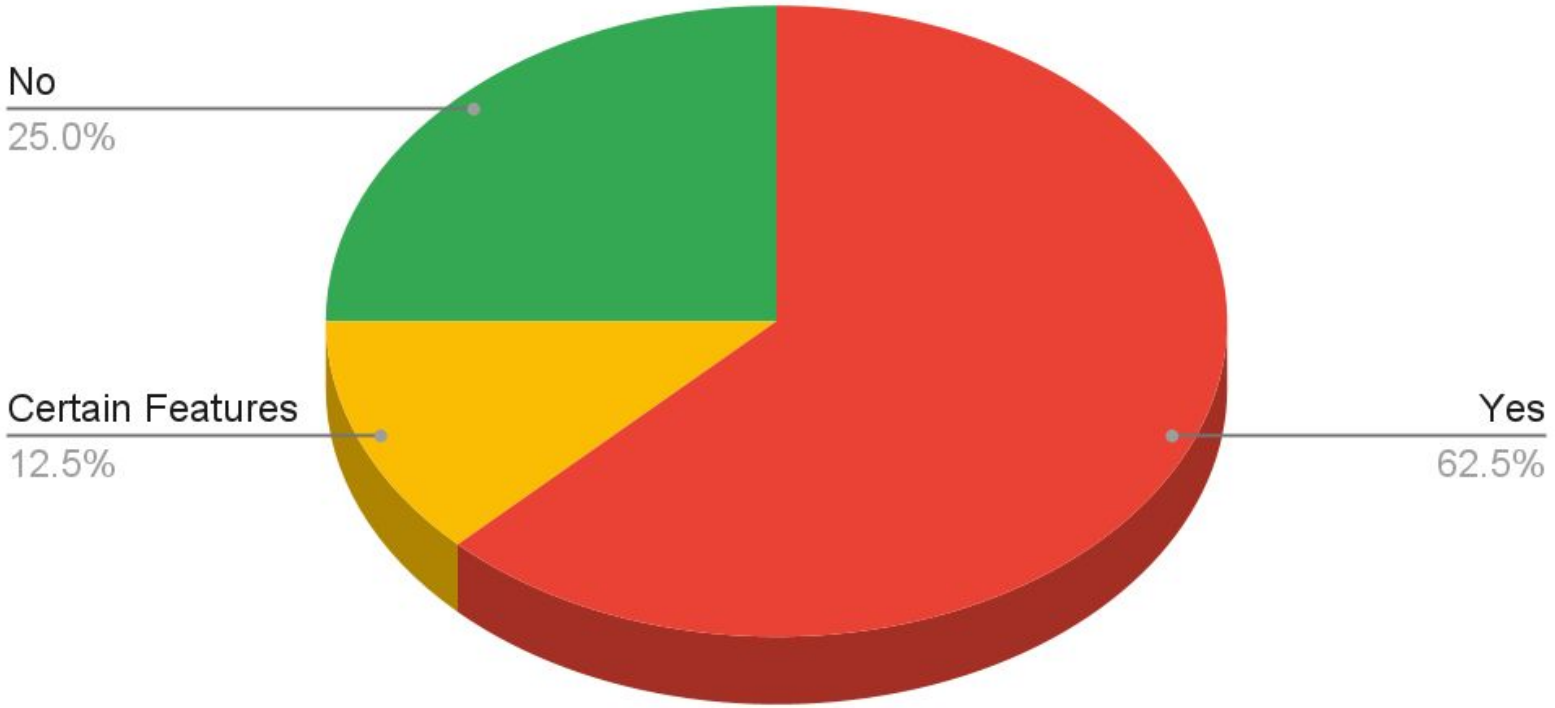
Trampoline DaemonSets In Popular K8s Platforms

Analyzed Platforms

- Focused on common infra components
- Managed K8s Services & K8s Distributions
  - AKS, EKS, GKE, OpenShift
- CNIs
  - Antrea, Calico, Cilium, WeaveNet

Trampoline DaemonSets (Feb 22)

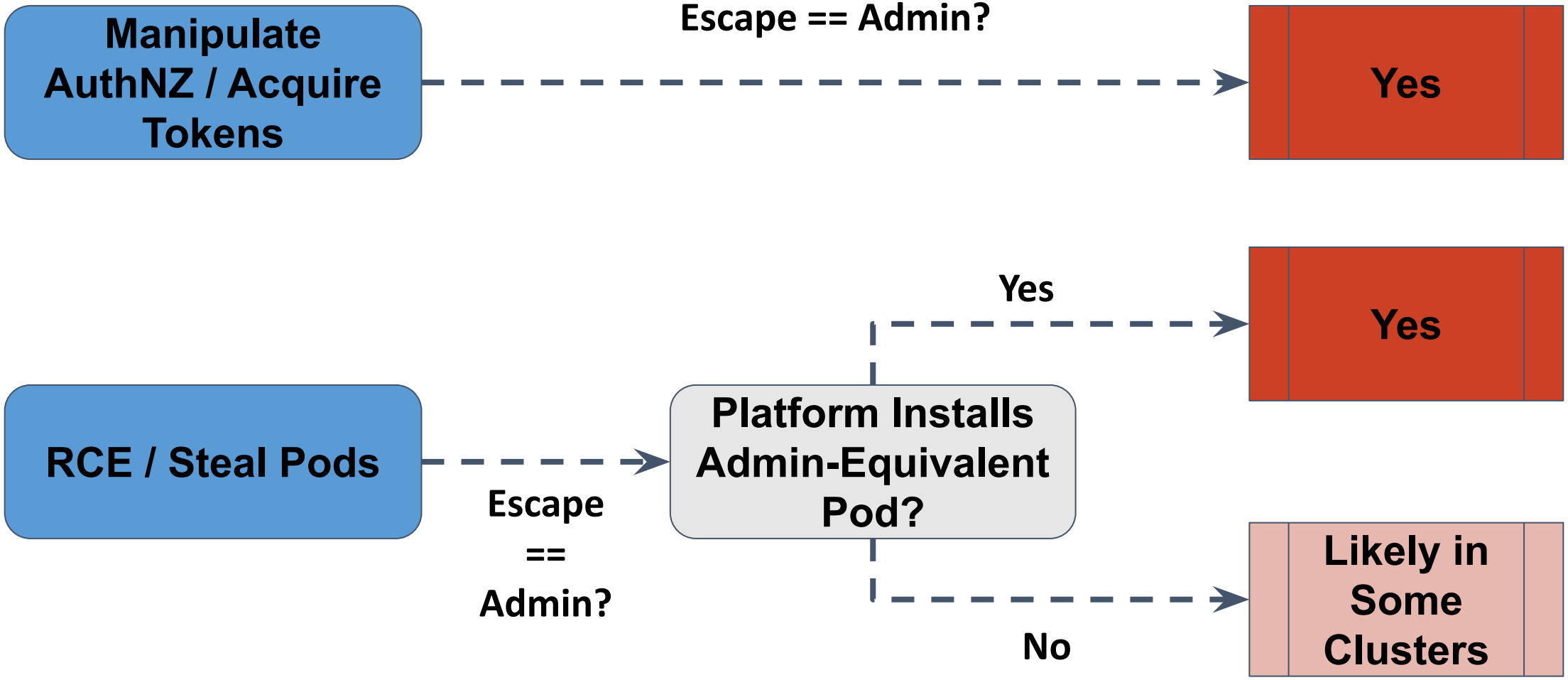
- Most (62.5%) install powerful DaemonSets by default



Trampoline DaemonSets (Feb 22)

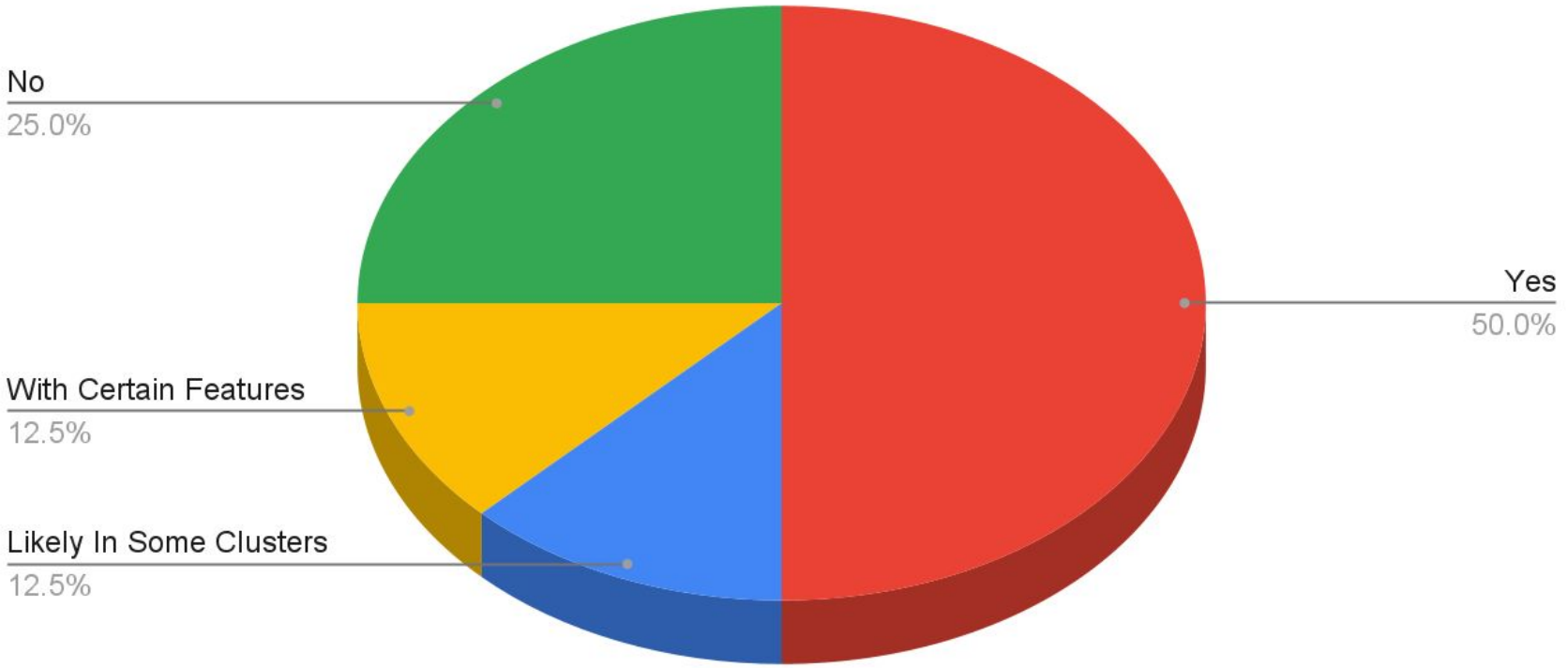
Platform	Powerful DaemonSets	DaemonSets	Most Powerful Permissions
AKS	Yes	cloud-node-manager, csi-azurefile-*, Kubelet	list secrets, update nodes
EKS	Yes	aws-node	update nodes
GKE	Only with Dataplane v2	anetd	update nodes, update pods
OCP	Yes	machine-config, sdn, multus-*	create pods, update validatingwebhookconfigurations
Antrea	Yes	antrea-agent	patch nodes, patch pods, update services, update services/status
Calico	No	-	
Cilium	Yes	cilium	update nodes, update pods, delete pods
Weave Net	No	-	

Escape == Admin by DaemonSet Perms



Container Escape == Cluster Admin? (Feb)

- In half the platforms, escape == admin by default



Escape == Admin (Feb 22)

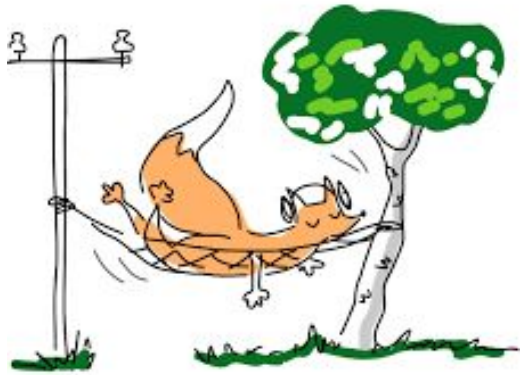
Platform	Escape == Admin	Attack	Prerequisite
AKS	Yes	Acquire Token → Manipulate AuthN/Z	-
EKS	Likely in some clusters	Steal Pods	Stealable admin-eq pod
GKE	With Dataplane v2	Steal Pod / RCE → Acquire token → Manipulate AuthN/Z	Dataplane v2 enabled
OCP	Yes	Acquire Token	-
Antrea	Yes	Steal Pods / RCE → Acquire token → Manipulate AuthN/Z	-
Calico	No	-	-
Cilium	Yes*	Steal Pod / RCE -→ Acquire token → Manipulate AuthN/Z	-
Weave Net	No	-	-

* Cilium via Helm: Likely in some clusters

No Panic Please :)

- To exploit powerful DaemonSets, attackers must compromise a node
  - Take over a pod
  - Escape it
- Several platforms already removed powerful DS 🎉

* If you run multi-tenant clusters, you're at greater risk



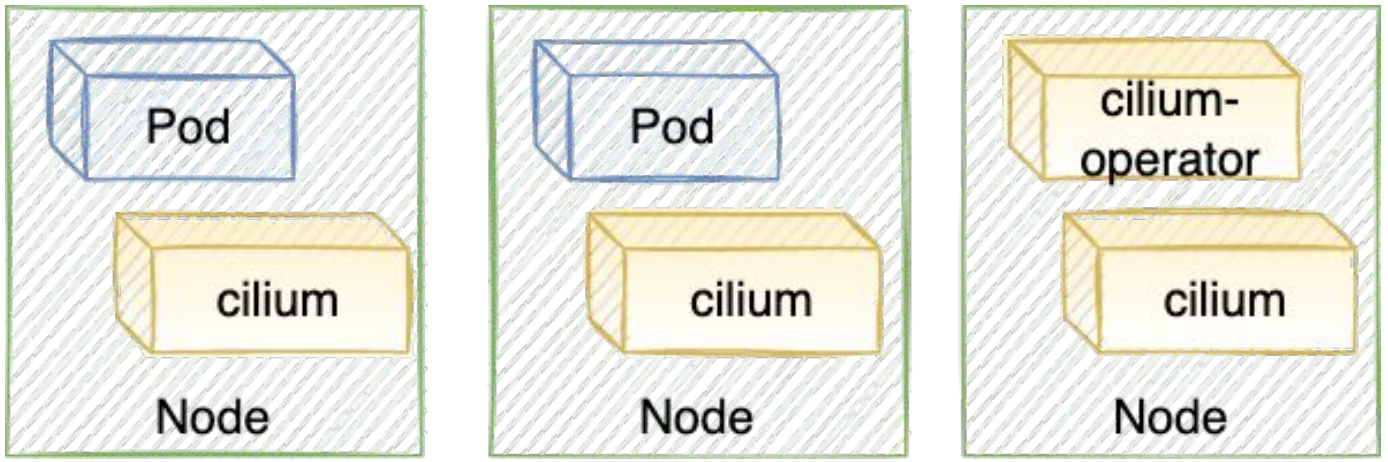
Attack Walkthrough

Attack Walkthrough

- No time to go over every attack
- Why Cilium?
  - Showcases a number of attack classes
  - Released fixes

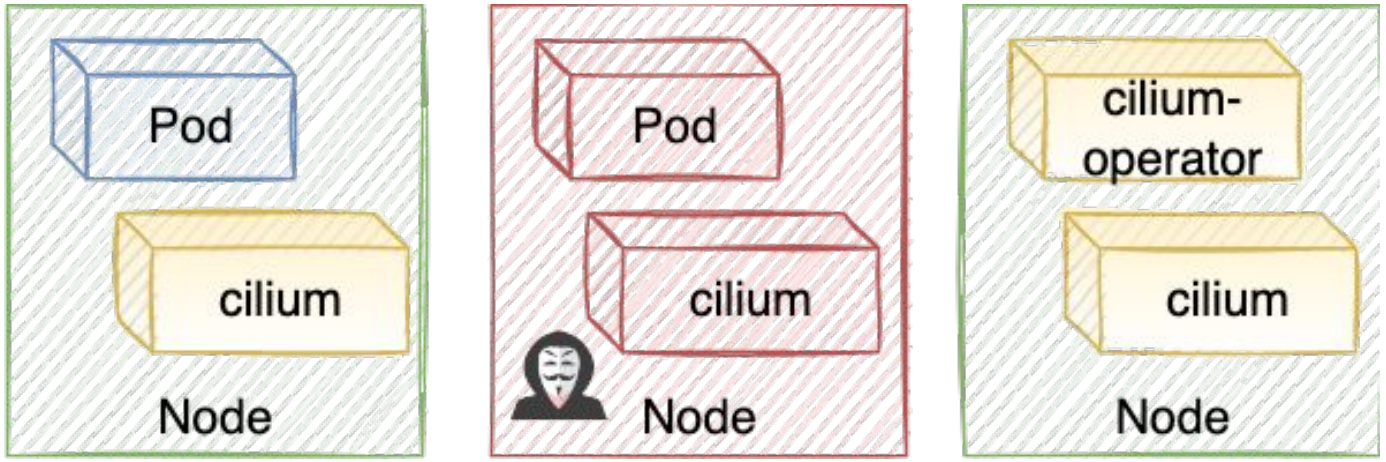
Cilium: Trampolines

- cilium DaemonSet
  - Steal Pods: can delete pods & update nodes/status
- cilium-operator Deployment
  - Acquire Tokens: can list secrets



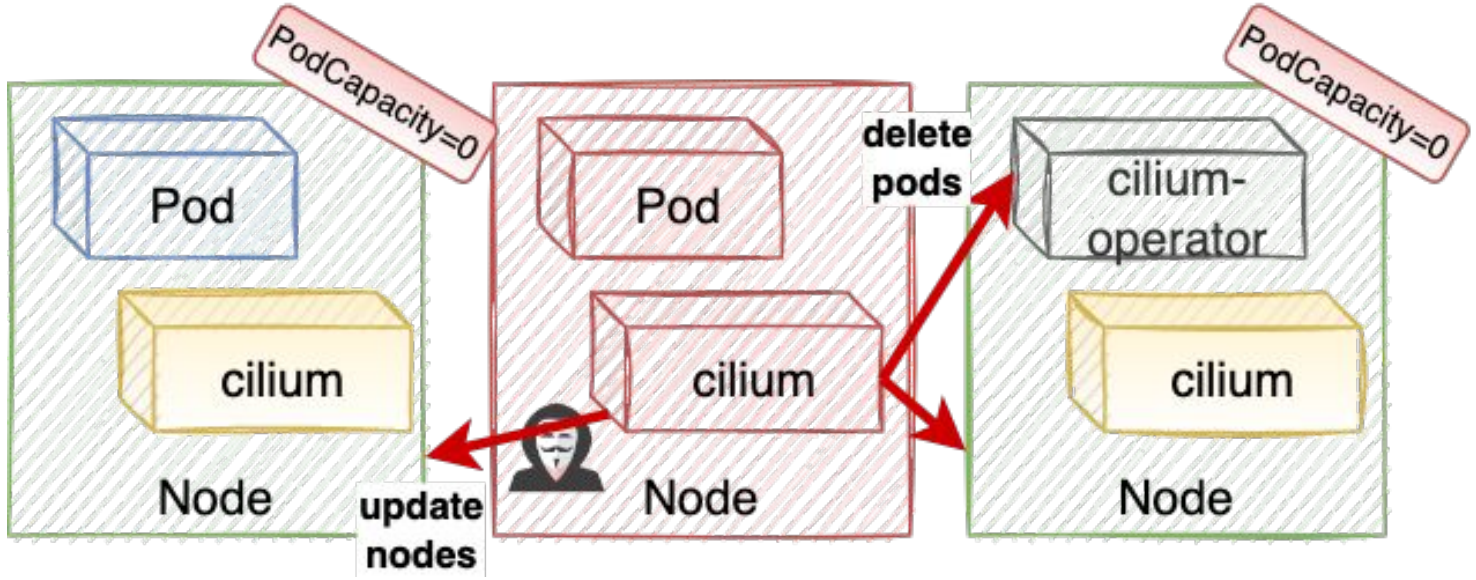
Attack Walkthrough

- **Starting assumption**: compromised pod and escaped to node
- **Goal**: cluster admin



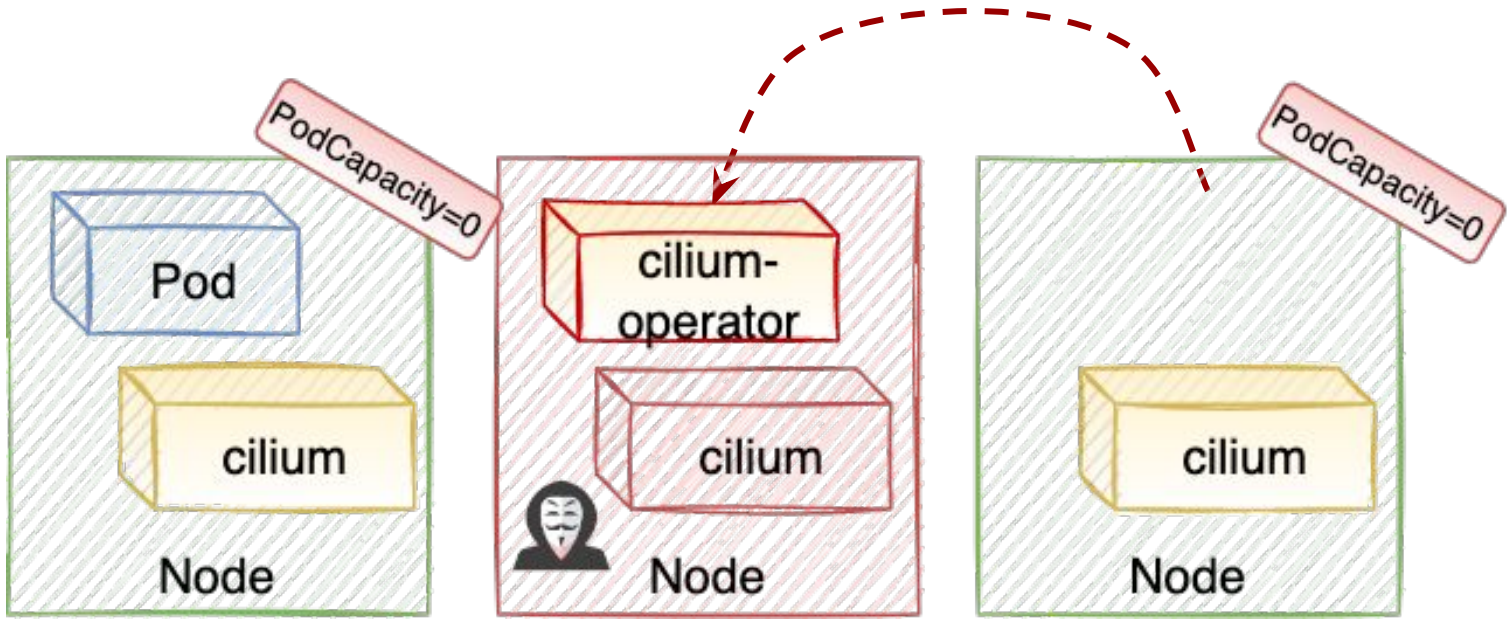
Attack Walkthrough

1. Zero other nodes' pod capacity & delete cilium-operator



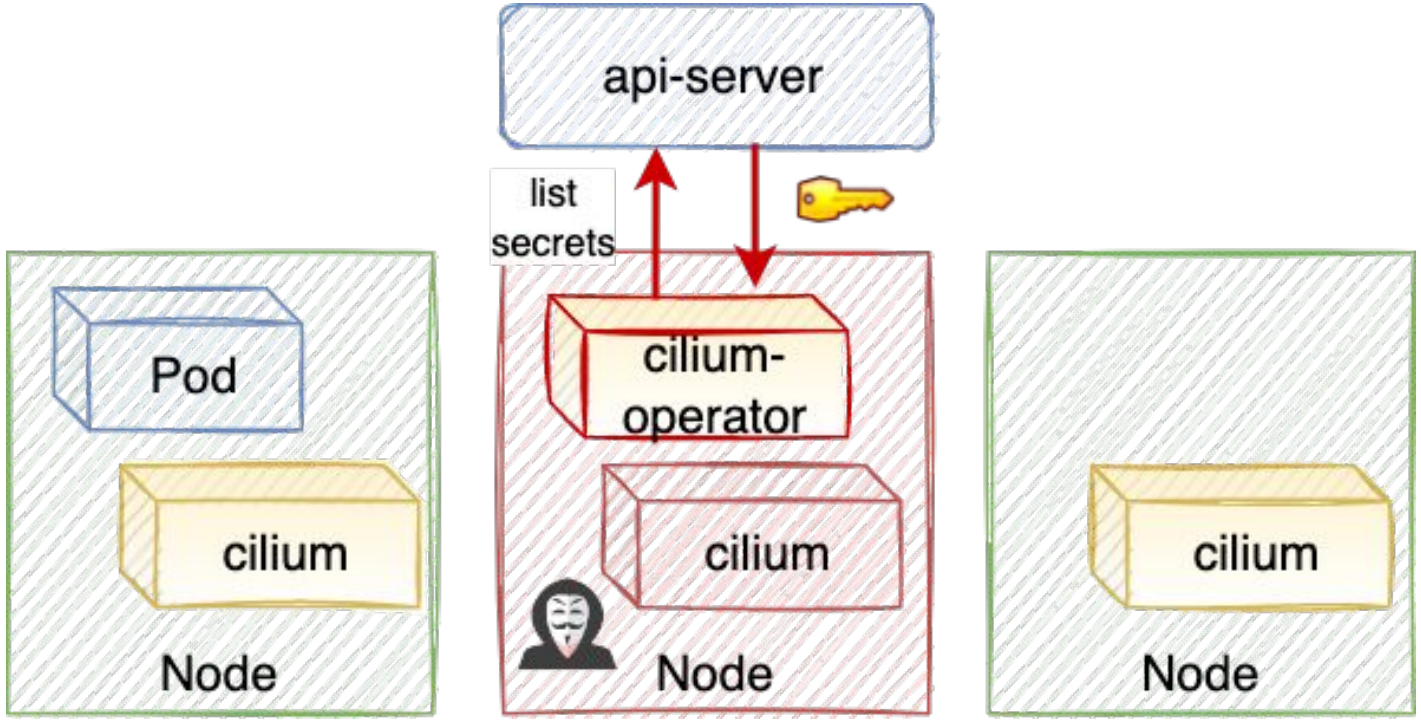
Attack Walkthrough

1. Zero other nodes' pod capacity & delete cilium-operator



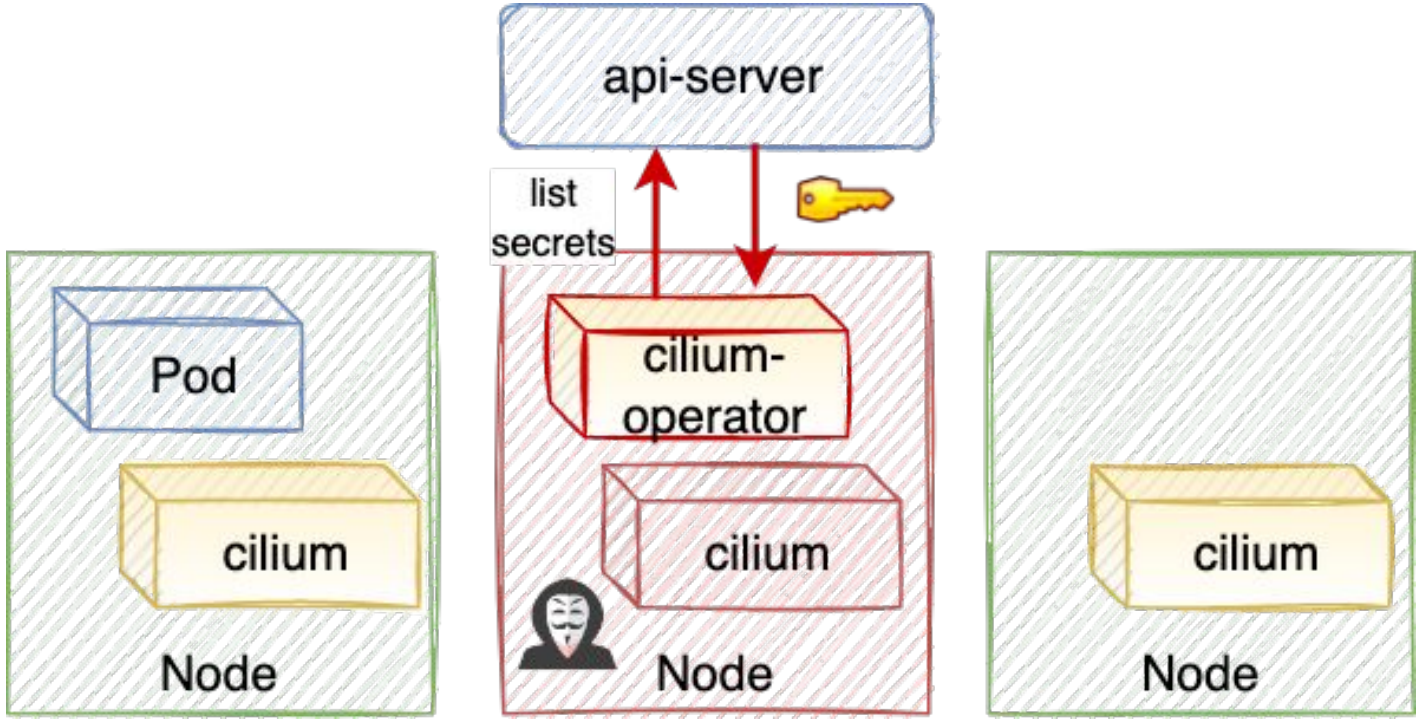
Attack Walkthrough

1. Zero other nodes' pod capacity & delete cilium-operator
2. **Abuse cilium-operator to retrieve powerful built-in token**



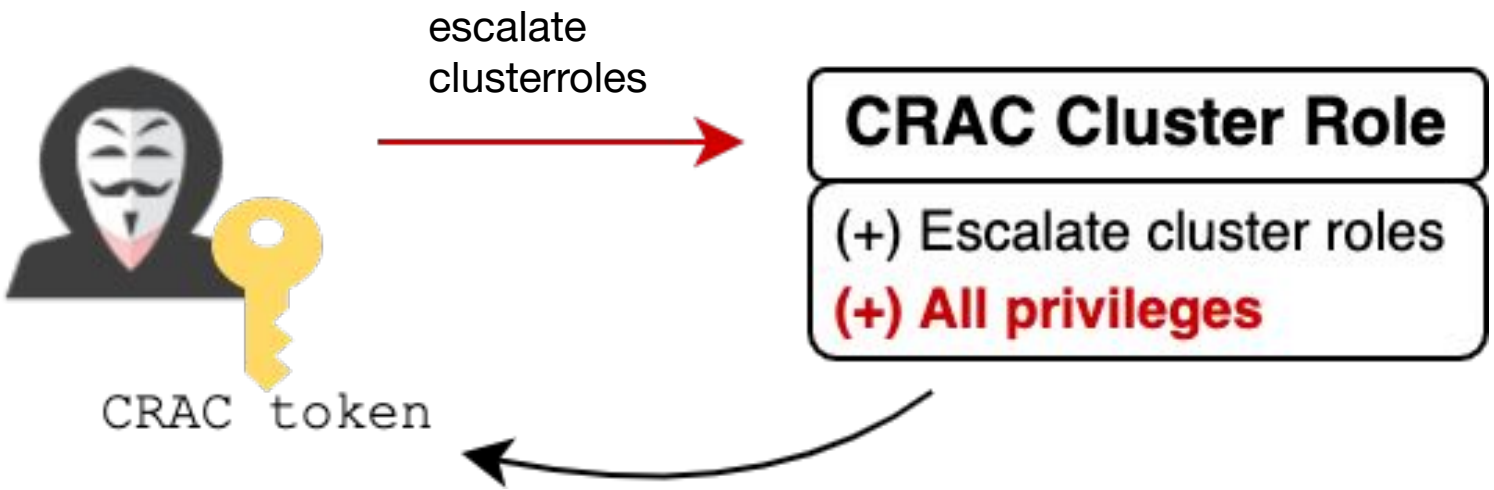
Attack Walkthrough

1. Zero other nodes' pod capacity & delete cilium-operator
2. **Abuse cilium-operator to retrieve powerful built-in token**
   a. clusterrole-aggregation-controller (CRAC) can manipulate ClusterRoles



Attack Walkthrough

1. Zero other nodes' pod capacity & delete cilium-operator
2. Abuse cilium-operator to retrieve powerful built-in token
3. **Add admin perms to the ClusterRole binded to our token**



Demo

Attack Classes in Demo

1. Zero other nodes' pod capacity & delete cilium-operator

   Steal Pods

2. Abuse cilium-operator to retrieve powerful built-in token

   Acquire Tokens

3. Add admin perms to the ClusterRole binded to our token

   Manipulate AuthN/Authz

Final Remarks on Cilium

- Fixed in v1.12.0-rc2 with several fixes backported 🙌
  - None of the powerful permissions remain
- Other platforms allowed similar attacks
- Demo targeted cilium-cil installation (default method)
  - Via Helm: still Trampoline DaemonSets but lower impact :) see report for more info

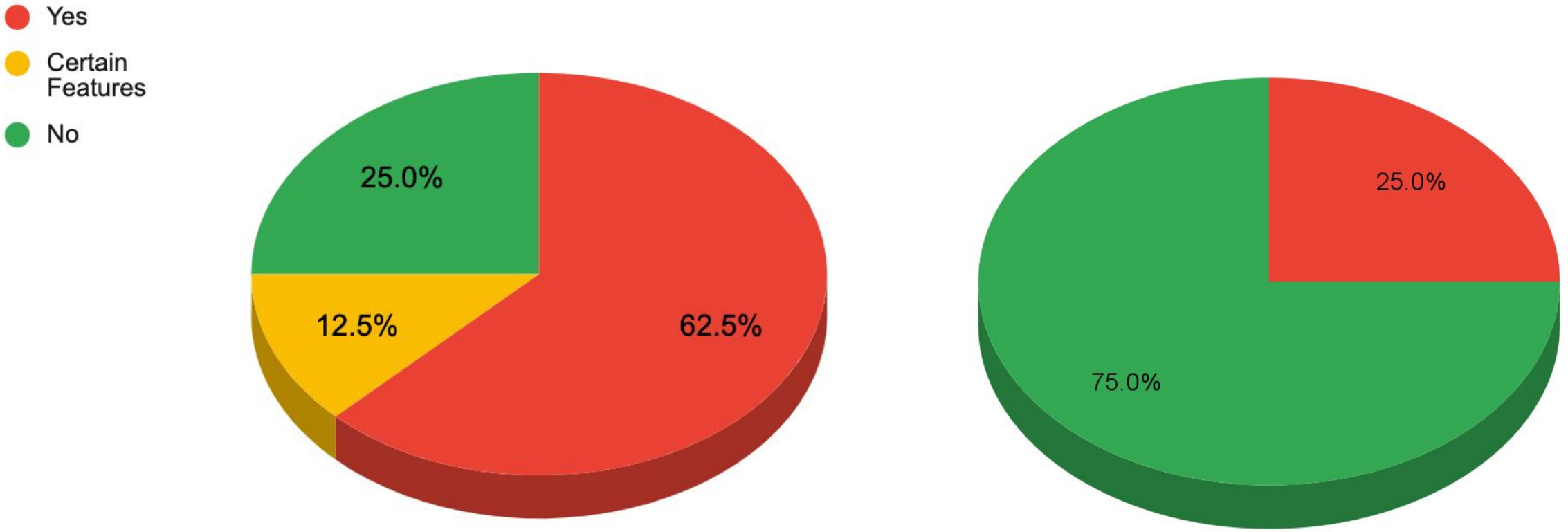
Disclosure, Fixes & Mitigations

Disclosure & Fixes

- Disclosed findings between Dec 21 to Feb 22
- Great disclosure experience all around!
- Approaches to fix
  - Strip excessive / unneeded privileges
  - Move functionally from DS to Deployments / control plane
  - Release admission policies that prevent misuse of powerful DS

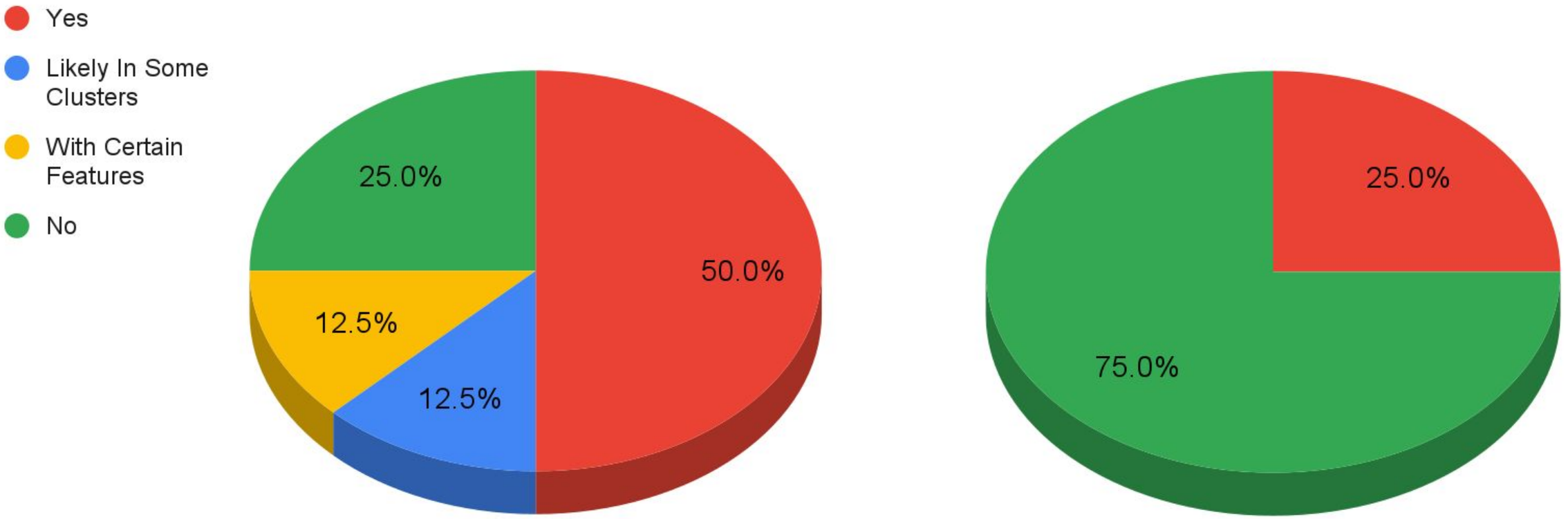
Trampolines DaemonSets

- Feb 22nd → May 18th



Container Escape == Cluster Admin

- Feb 22nd → May 18th



* Due to powerful DaemonSets / K8s-native privEsc, doesn't cover possible platform-specific attacks

Fixes

Platform	Had Powerful DaemonSets	Fixed	Had Powerful Kubelets	Fixed
AKS	Yes	No	Yes	**WIP**
EKS	Yes	**Yes, from Kubernetes v1.18**	No	-
GKE	With Dataplane v2	**Yes, v1.23.4-gke.900, 13022$ Bounty**	No	-
OCP	Yes	**WIP set for v4.11, possible backports**	No	-
Antrea	Yes	**Yes, v1.6.1 alongside an admission policy**	No	-
Calico	No	-	No	-
Cilium	Yes	**Yes, v1.12.0-rc2**	No	-
Weave Net	No	-	No	-

Recommendations

Better RBAC Posture and Stronger Node Isolation

Recommendations for Users & Projects

- **Least-Privileged**
  - Scope permissions to namespace/resourceNames
  - Review your K8s manifests / Helm charts!
- **Track the powerful permissions & pods in your cluster / project**
  - Document ones you're asking for
- **Isolate powerful pods from untrusted / publicly-exposed ones**
  - Scheduling constraints: Taints & Tolerations, Node Affinity, PodAntiAffin
- **Remove powerful DaemonSets**
  - Move privileged functionality to non-DaemonSet / control plane,
  - Minimize write perms over core objs, store state in CRDs / ConfigMaps
  - It's not drop everything or drop nothing
- **Restrain powerful permissions**
  - Prevent / detect misuse via policy controllers (OPA Gatekeeper)

rbac-police

Identifying Powerful Permissions and RBAC Risks

rbac-police

- New open-source tool
- Evaluate the RBAC perms of pods, SA & nodes
  - Find powerful pods & understand the attacks they enable
- ~20 policies out-of-the-box
  - Each targeting a different powerful perms / privEsc technique
- Customizable: policies written in Rego (OPA), add your own!
  - CRDs? Platform specific attacks? PrivEsvs we missed?

github.com/PaloAltoNetworks/rbac-police

```
yavrahami@M-C02YT7FTLVDQ:~/rbac-police$ ./rbac-police eval lib
{
    "policyResults": [
        {
            "policy": "lib/modify_pods.rego",
            "severity": "High",
            "description": "SAs and nodes that can update and patch pods in privileged namespaces
(kube-system) can gain code execution on pods that are likey to be privileged",
            "violations": {
                "serviceAccounts": [
                    {
                        "name": "cilium",
                        "namespace": "kube-system",
                        "nodes": [
                            {
                                "ip-172-31-20-29.ec2.internal": [
                                    "cilium-66ssg",
                                    "cilium-eks-node-init-82f9f"
                                ]
                            },
                            {
                                "ip-172-31-33-112.ec2.internal": [
                                    "cilium-bb9s6",
```

Policy & Severity

Powerful pods violating policy

Final Thoughts

- Issues are getting subtler and milder
  - Only downside - false confidence
- Tackling vague areas in K8s security
  - Which permissions can be abused for what attacks?
  - What's the real impact of container escape?
- Ambiguity leads to blind spots, let's clear the fog

Report

Questions?

rbac-police

Extra Slides

Checkov

- Open source Infrastructure as Code (IaC) security scanner
- 4 new policies for K8s RBAC
  - Targeting perms that can Manipulate AuthN/AuthZ
  - More incoming
- Alerts on RBAC perms **before they're installed to the cluster**
  - Great for inspecting add-ons prior to deployment

github.com/bridgecrewio/checkov

checkov by bridgecrew

Recommendations - K8s Projects

- We focused on infra platforms, countless other K8s projects
- **Review your K8s manifests / Helm charts!**
  - Least-Privileged
- **Remove powerful DaemonSets**
  - Move privileged functionality to non-DaemonSet / control plane
  - Minimize write perms over core objs, store state in CRDs / ConfigMaps
- **It's not drop everything or drop nothing**
  - Some powerful permissions will be harder to remove
  - Partially privileged better than admin-equivalent!
- **Document the powerful permissions you're asking for**

Recommendations - K8s Users

- **Least-Privileged**
  - Scope permissions to namespaces / resourceNames
- **Track powerful permissions & pods in your cluster**
- **Restrain powerful permissions**
  - Prevent / detect misuse via policy controllers (cluster-wide or SA scoped)
    - Patch pods X change image
    - Update deployment X change service account
    - Machine creds (SAs / nodes) querying for their permissions

Recommendations - K8s Itself

- Kubernetes could make unprivileged DaemonSets easier
- Fine-grained permissions - "annotation" subresource
- AuthZ based on pod's node (like Node Authorizer)
  - TokenRequest BoundObjRef

Escape == Admin

- **Not all trampolines DS offer the same bounce**
- Some are admin-equivalent
  - Manipulate AuthN/AuthZ, Acquire Tokens
- Others allow taking over other pods
  - RCE, Steal Pods
  - Cluster hosts admin-equivalent pod?

Previous Work

- "Hardening Kubernetes Identities in 5 Simple Steps" - BH 2021