**MMLAB Main Seminar**

# Take Over the Whole Cluster: Attacking Kubernetes via Excessive Permissions of Third-party Applications

Nanzi Yang, Wenbo Shen*, Jinku Li, Xunqi Liu, Xin Guo, Jianfeng Ma
*Xidian University(Xi'an, China), Zhejiang University(Hangzhou, China)**

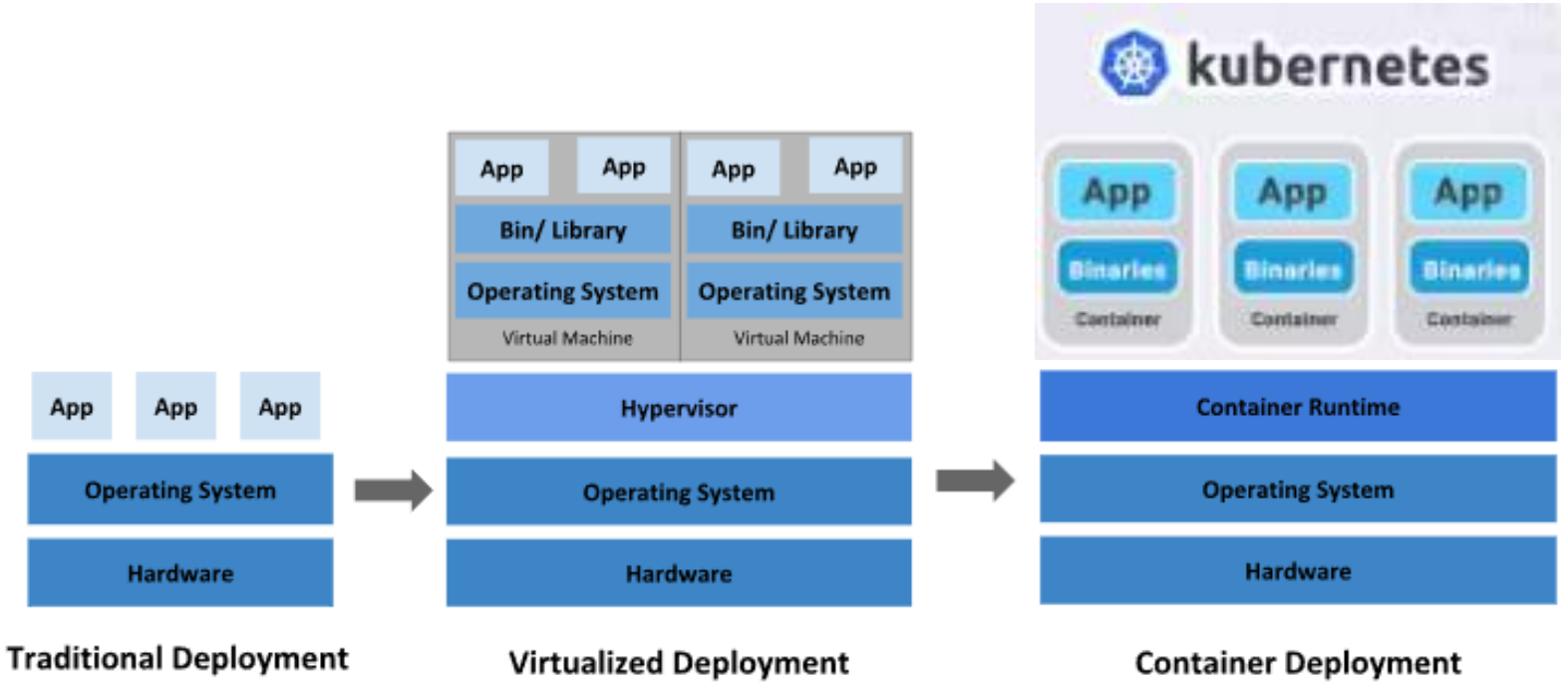
ACM conference on Computer and Communications Security 2023 (CCS 23')

2024.11.19

HyeongUk Ko (huko@mmlab.snu.ac.kr)

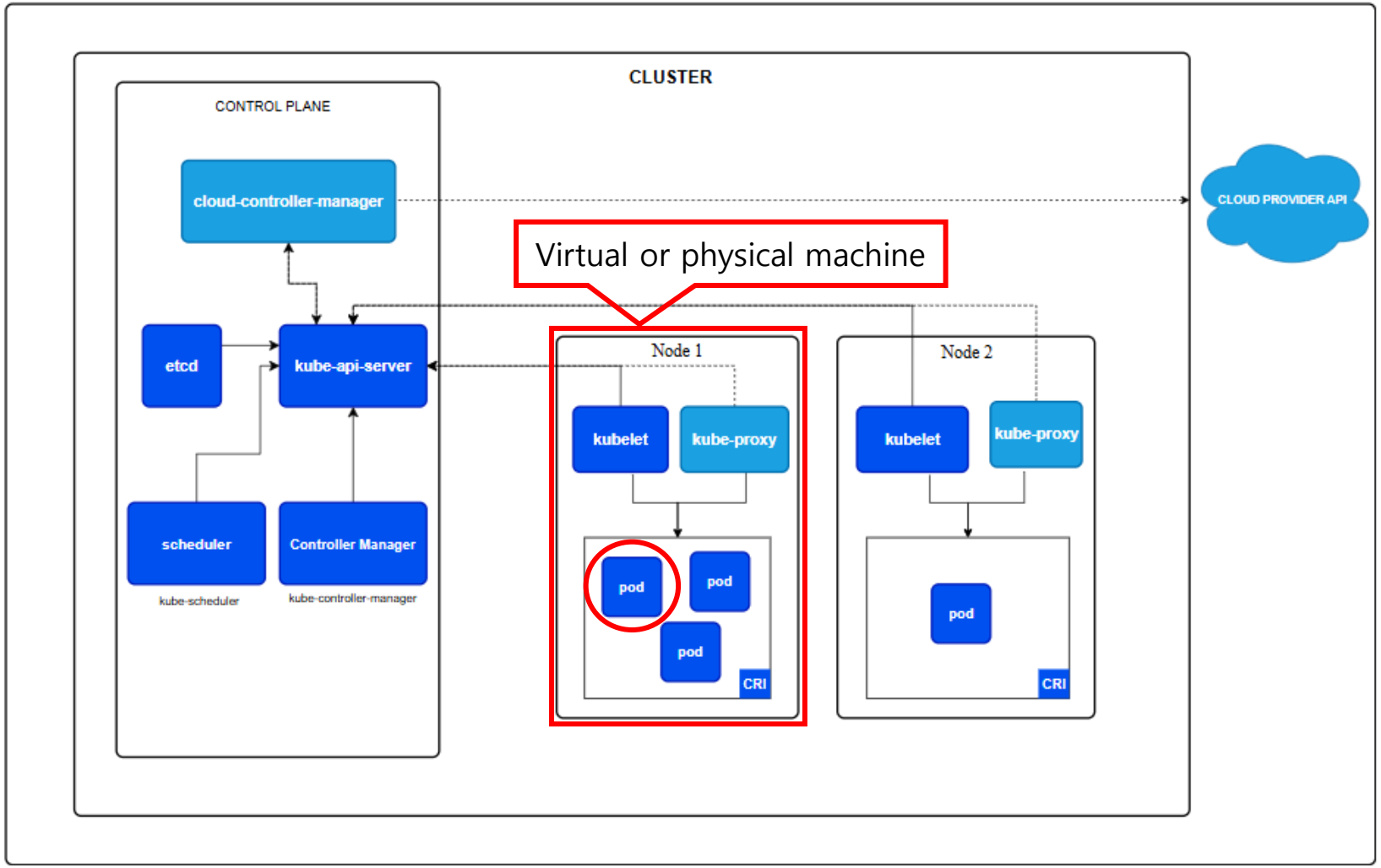
# Contents

# What is Kubernetes?

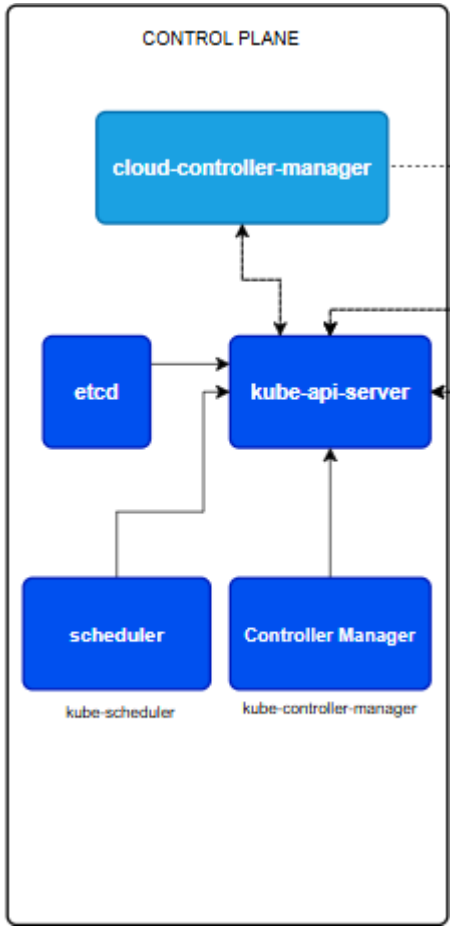



- Container orchestration tool
- Adopted by numerous companies including well-known cloud vendors (e.g. Google GKE, Amazon EKS, …)

# Kubernetes Architecture



# Control Plane

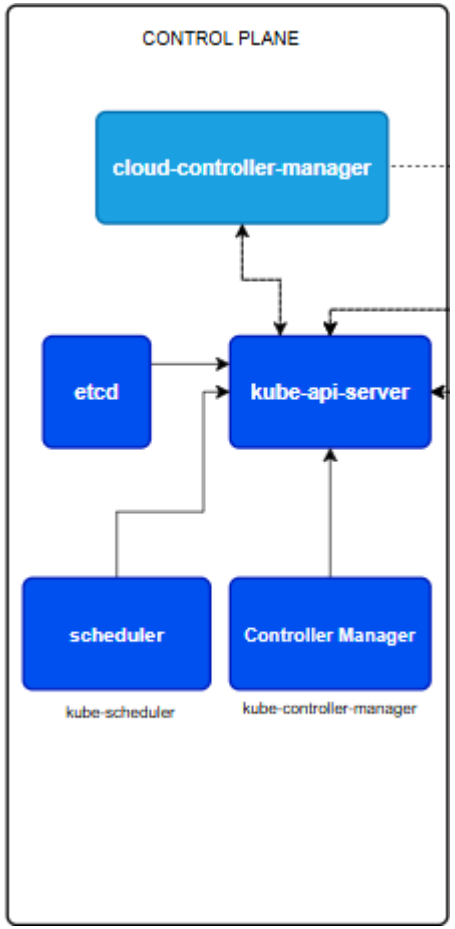



1. kube-api-server
   - Exposes Kubernetes API to user
   - Processes requests from users and components to manage resources like pods, services, nodes, etc. (Communication hub)

2. etcd
   - Stores the entire state of the cluster including configuration, workloads, node status, etc. (Data storage)

3. scheduler
   - Watches for unscheduled pods in the API server and determines which node to run them based on scheduling policies

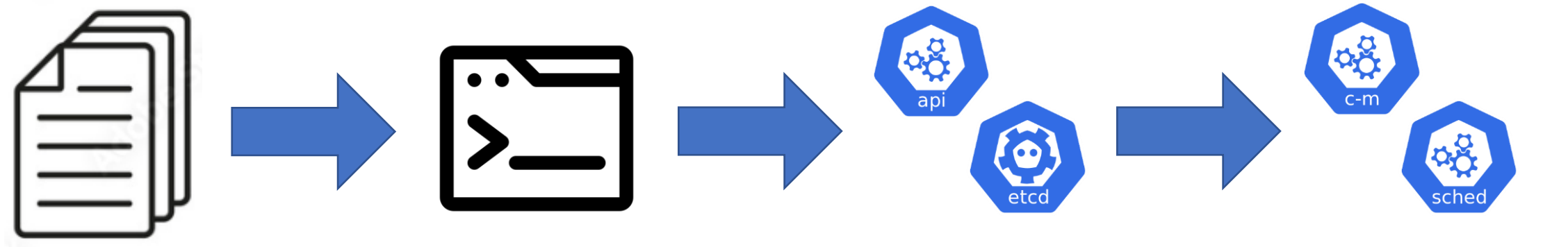
# Control Plane




4. controller manager
   - Manages states of cluster and workload by running controllers, such as Node controller, Deployment controller, etc.
   - Monitors the cluster state and makes adjustments to ensure the cluster matches the desired state specified in resource definitions

5. cloud-controller-manager (optional)
   - Integrates Kubernetes clusters with services provided by cloud providers
   - Handles tasks that require cloud provider APIs, such as managing load balancers, network routes, and persistent storage volumes

# Application Installation




manifest.YAML

kubectl apply

Store spec files

Create Objects&Pods

Persistent entity that represents the state and configuration of resources in the cluster

# Objects

- Deployment
  - Manages a set of Pods to run an application workload
  - Ensures Kubernetes cluster to maintain the desired state
- Daemonset
  - Ensures that all nodes run a copy of a Pod basically
  - Typically used to deploy system-level applications such as monitoring agents, networking helper tool, log collecting daemons, or other add-ons that need to run on every node
- Role
  - Contains rules that represent a set of permissions which work only within specified namespace
  - Can be granted to users by using RoleBinding object
- ClusterRole
  - Contains rules that represent a set of permissions which work within whole cluster
  - Can be granted to users by using RoleBinding, ClusterRoleBinding object

# Problems

- Third-party applications are used widely in Kubernetes
  - Used to extend the control functionality of Kubernetes
  - Granted critical permissions(excessive permission) for cluster management
  - Security of them has not been systemically studied so far
  - Utilizes third-party apps (e.g. CNCF projects) which are granted excessive permissions

The Cloud Native Computing Foundation (CNCF) is a Linux Foundation project that was started in 2015 to help advance container technology

CNCF manages its projects through a structured process, which includes 3 main maturity levels:

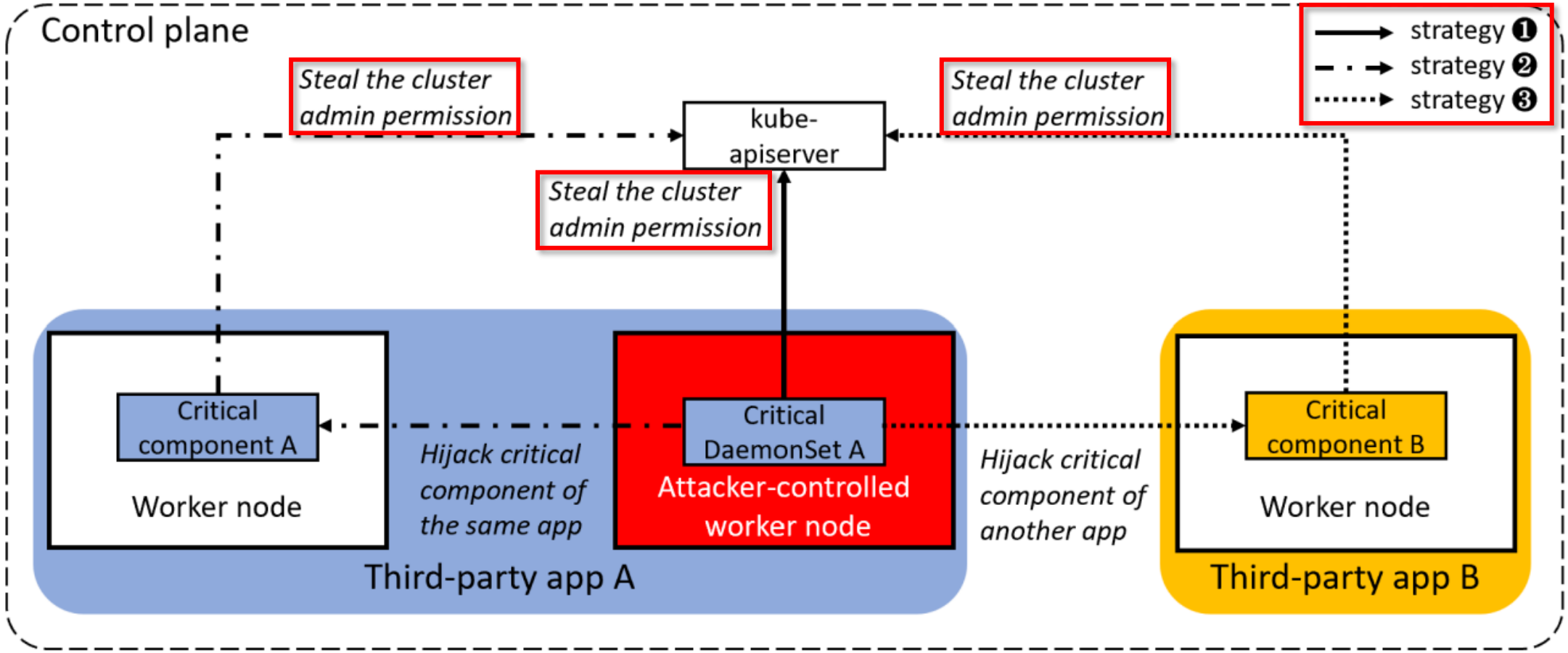
| Maturity level | Explanation |
|---|---|
| Sandbox | **Experimental projects** not yet widely tested in production on the bleeding edge of technology |
| Incubating | Projects **used successfully in production by a small number users** with a healthy pool of contributors |
| Graduated | Projects considered **stable, widely adopted**, and production ready, **attracting thousands of contributors** |

# Excessive Permission Attack

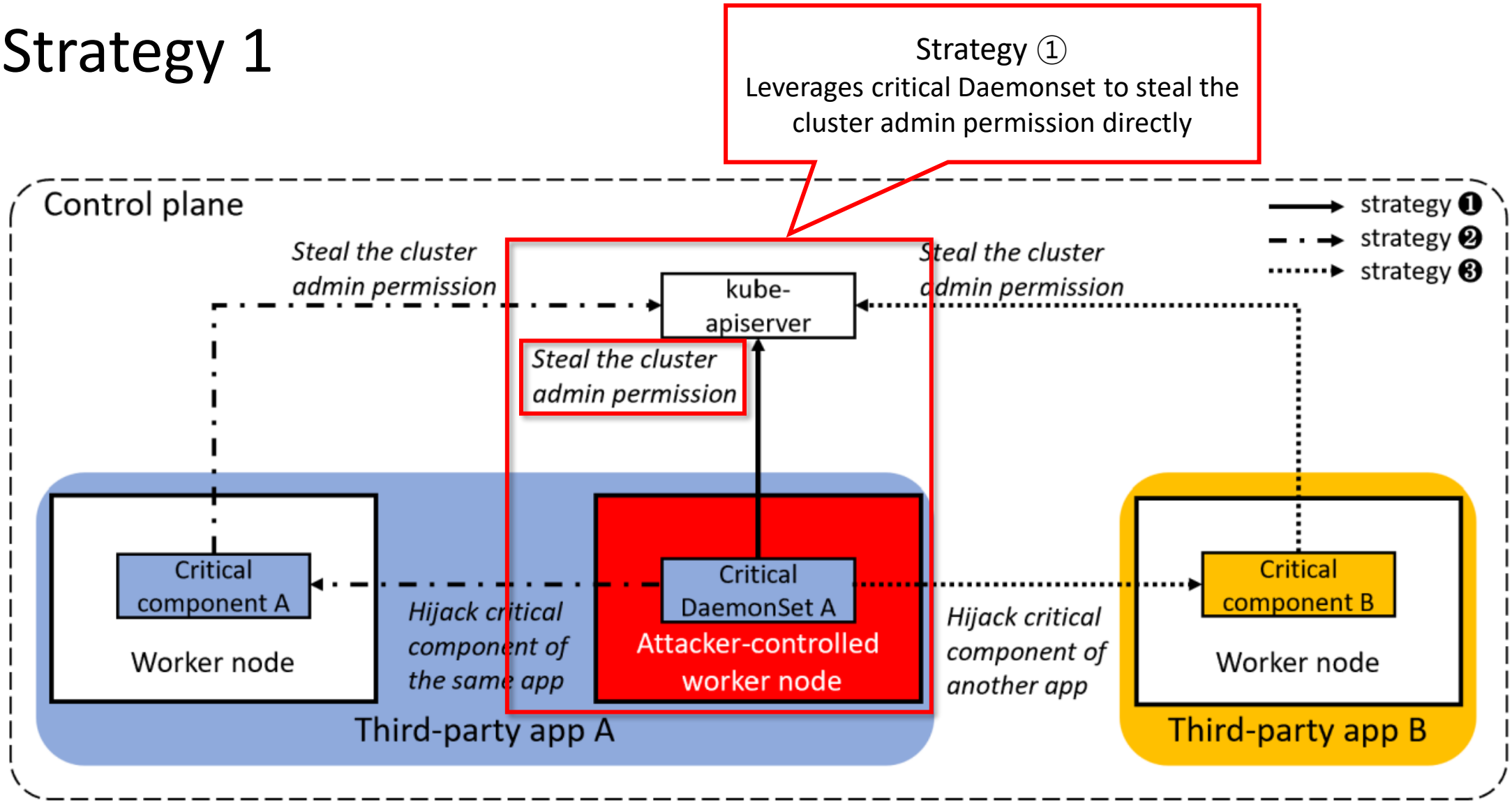
- Features
  - New attack surface which can be exploited to attack the whole Kubernetes cluster
  - Multiple attacks which can be made by abusing excessive permission
  - Utilized third-party apps' critical Daemonset & critical component

- Threat model
  1. Attacker already compromised the applications running inside the container
  2. Attacker already performed a container escape to compromise a worker node
  3. Attacker uses 3 strategies to perform excessive permission attack

# Three Strategies



# Strategy 1

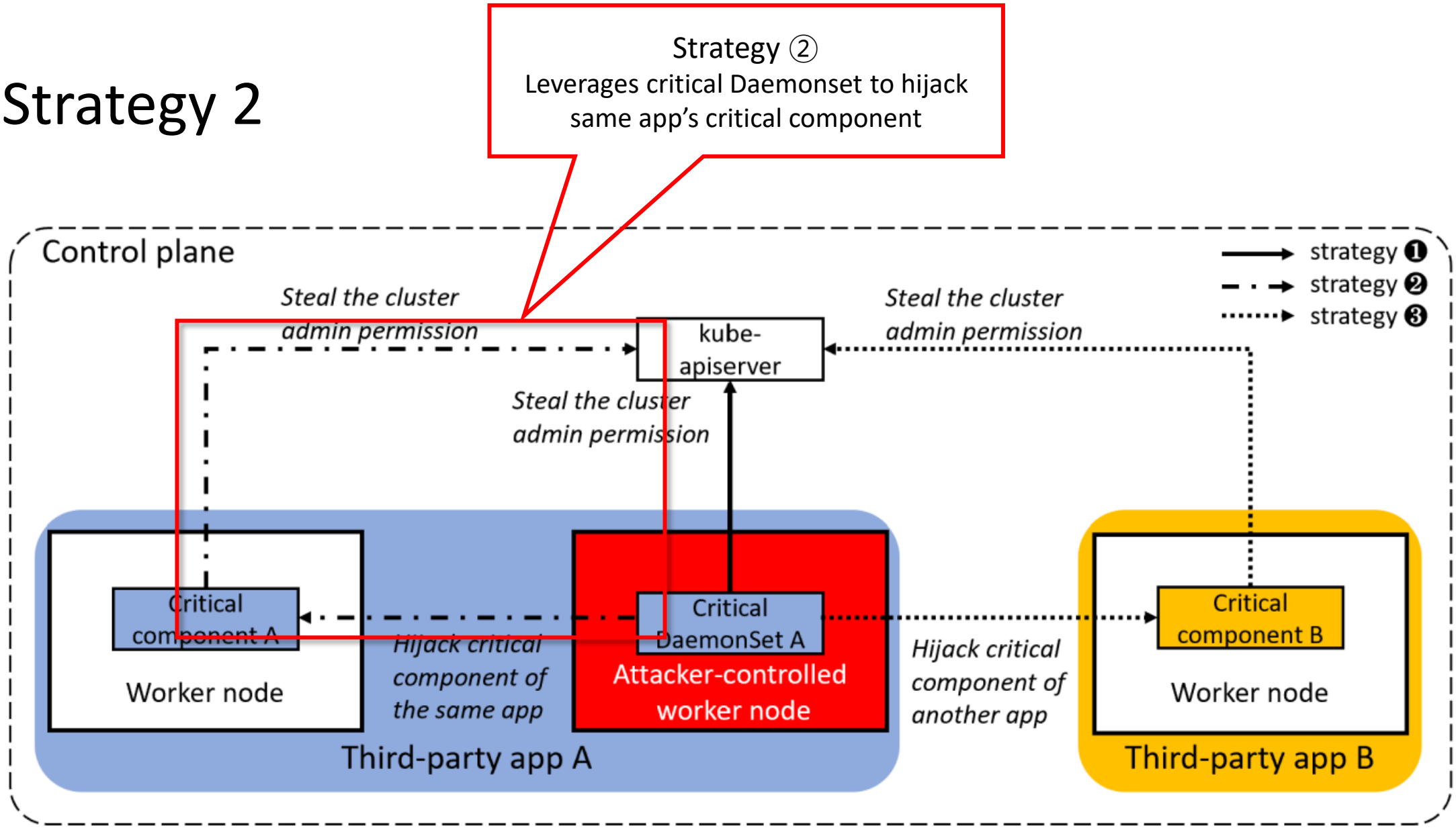

Strategy ①
Leverages critical Daemonset to steal the cluster admin permission directly
Control plane
strategy ❶
strategy ❷
strategy ❸
Steal the cluster admin permission
kube-apiserver
Steal the cluster admin permission
Steal the cluster admin permission
Critical component A
Worker node
Hijack critical component of the same app
Critical DaemonSet A
Attacker-controlled worker node
Third-party app A
Hijack critical component of another app
Critical component B
Worker node
Third-party app B

# Strategy 1 Example

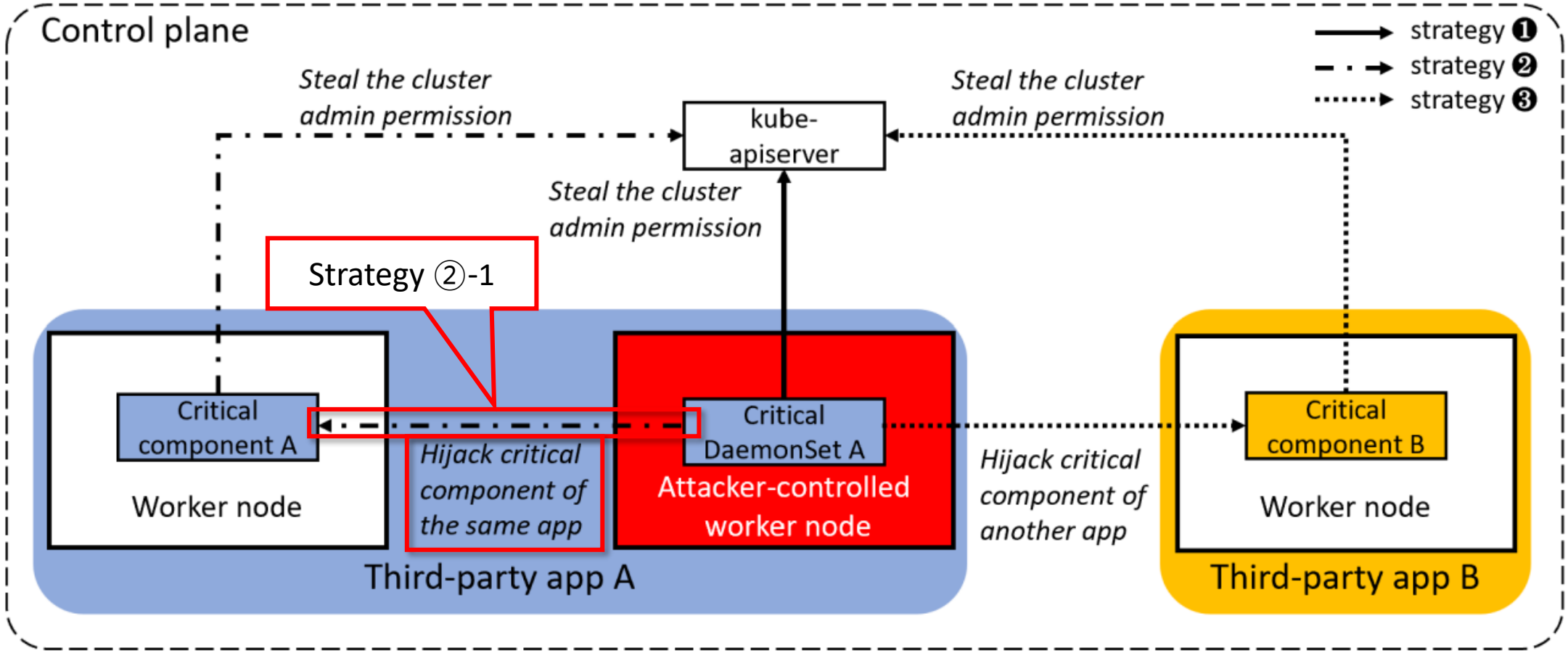
- CubeFS
  - Open-source cloud-native file storage system
  - CNCF incubating project

- Attack scenario
  1. It has a critical Daemonset **cfs-csi-node** which uses a service account **cfs-csi-service-account**, which is assigned ClusterRole **cfs-csi-cluster-role** that has the "get/list" verbs of the "secrets" resource
  2. Attacker can see admin's secrets utilizing that permission

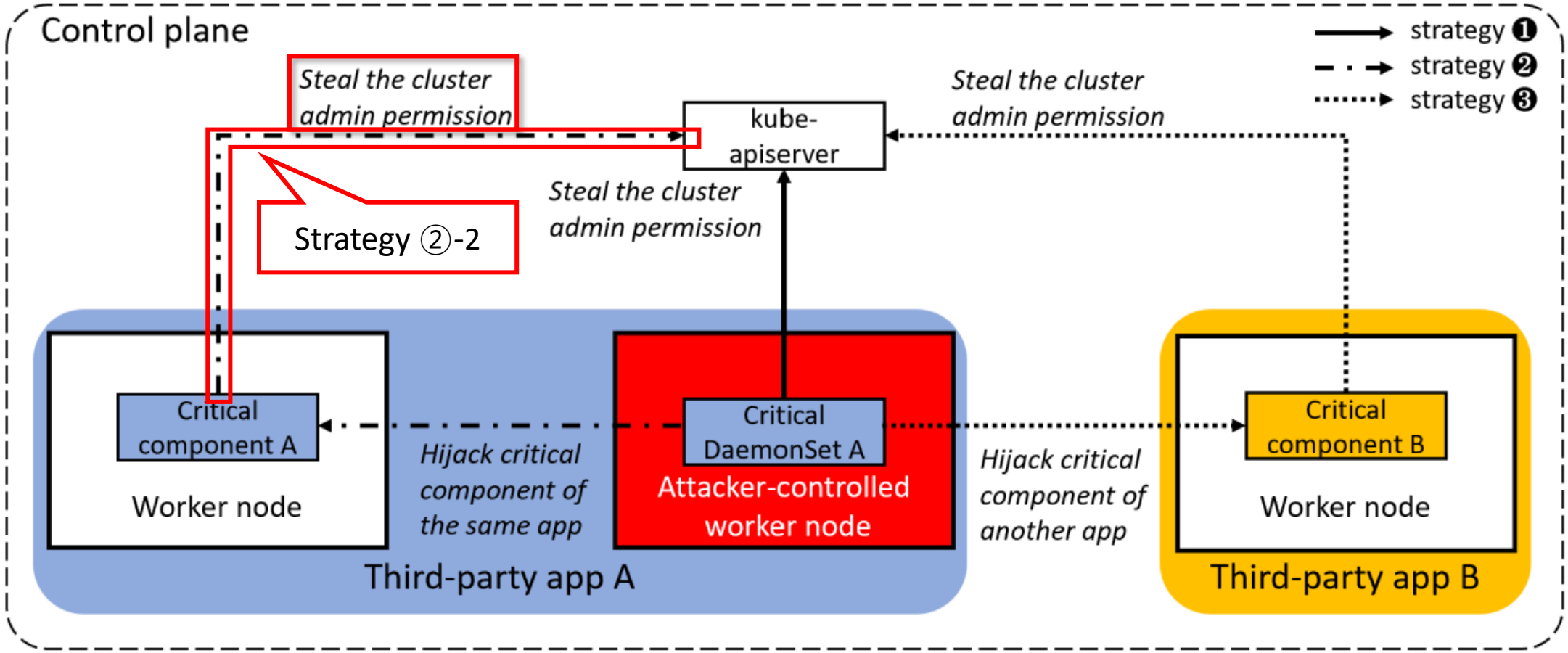
# Strategy 2


Strategy ②
Leverages critical Daemonset to hijack same app's critical component
Control plane
strategy ❶
strategy ❷
strategy ❸
Steal the cluster admin permission
kube-apiserver
Steal the cluster admin permission
Steal the cluster admin permission
Critical component A
Worker node
Hijack critical component of the same app
Critical DaemonSet A
Attacker-controlled worker node
Third-party app A
Hijack critical component of another app
Critical component B
Worker node
Third-party app B

# Strategy 2


Control plane
strategy ❶
strategy ❷
strategy ❸
Steal the cluster admin permission
kube-apiserver
Steal the cluster admin permission
Steal the cluster admin permission
Strategy ②-1
Critical component A
Worker node
Hijack critical component of the same app
Critical DaemonSet A
Attacker-controlled worker node
Third-party app A
Hijack critical component of another app
Critical component B
Worker node
Third-party app B

# Strategy 2

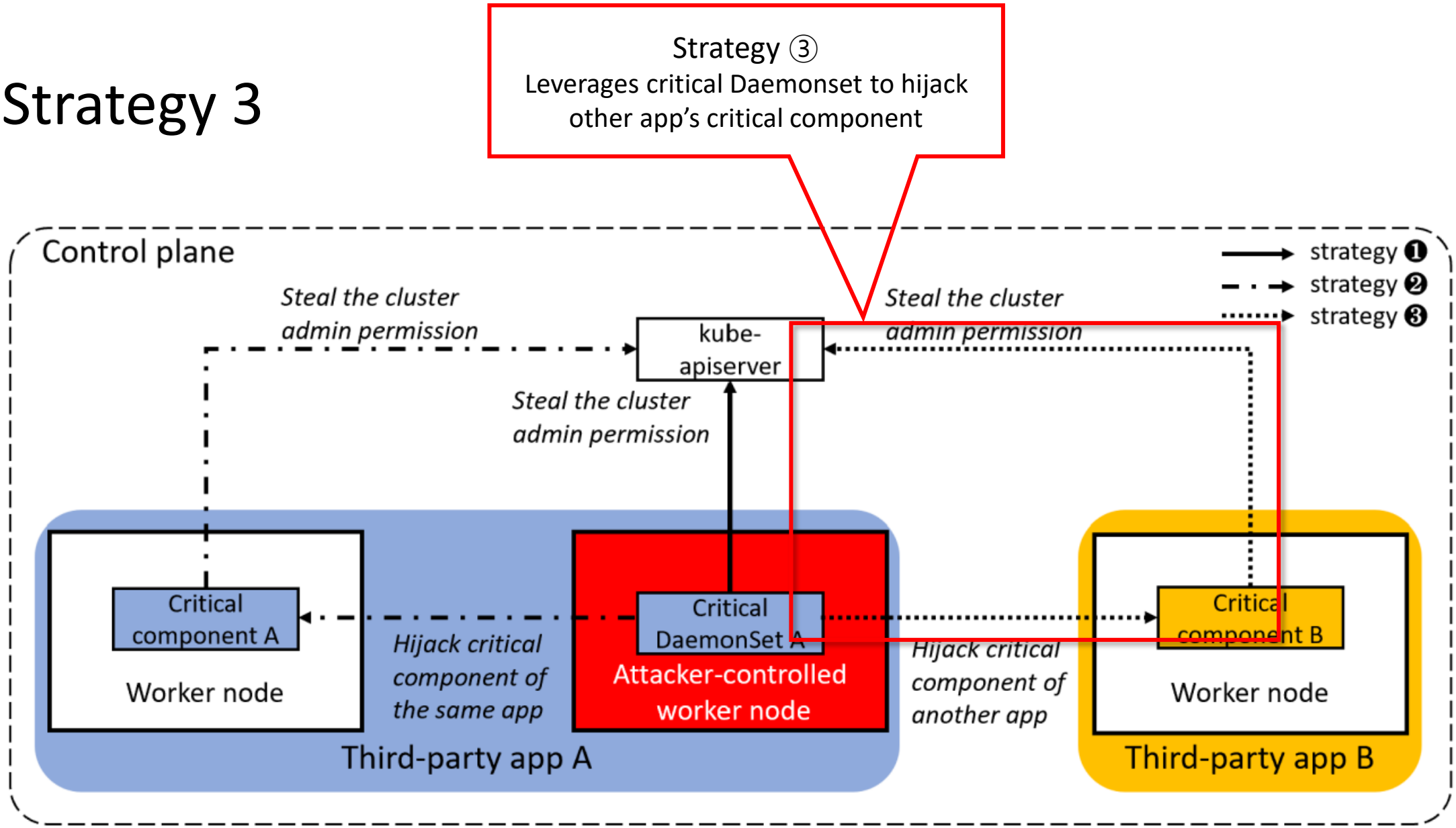

Control plane
strategy ❶
strategy ❷
strategy ❸
Steal the cluster admin permission
kube-apiserver
Steal the cluster admin permission
Steal the cluster admin permission
Strategy ②-2
Critical component A
Worker node
Hijack critical component of the same app
Critical DaemonSet A
Attacker-controlled worker node
Third-party app A
Hijack critical component of another app
Critical component B
Worker node
Third-party app B

# Strategy 2 Example

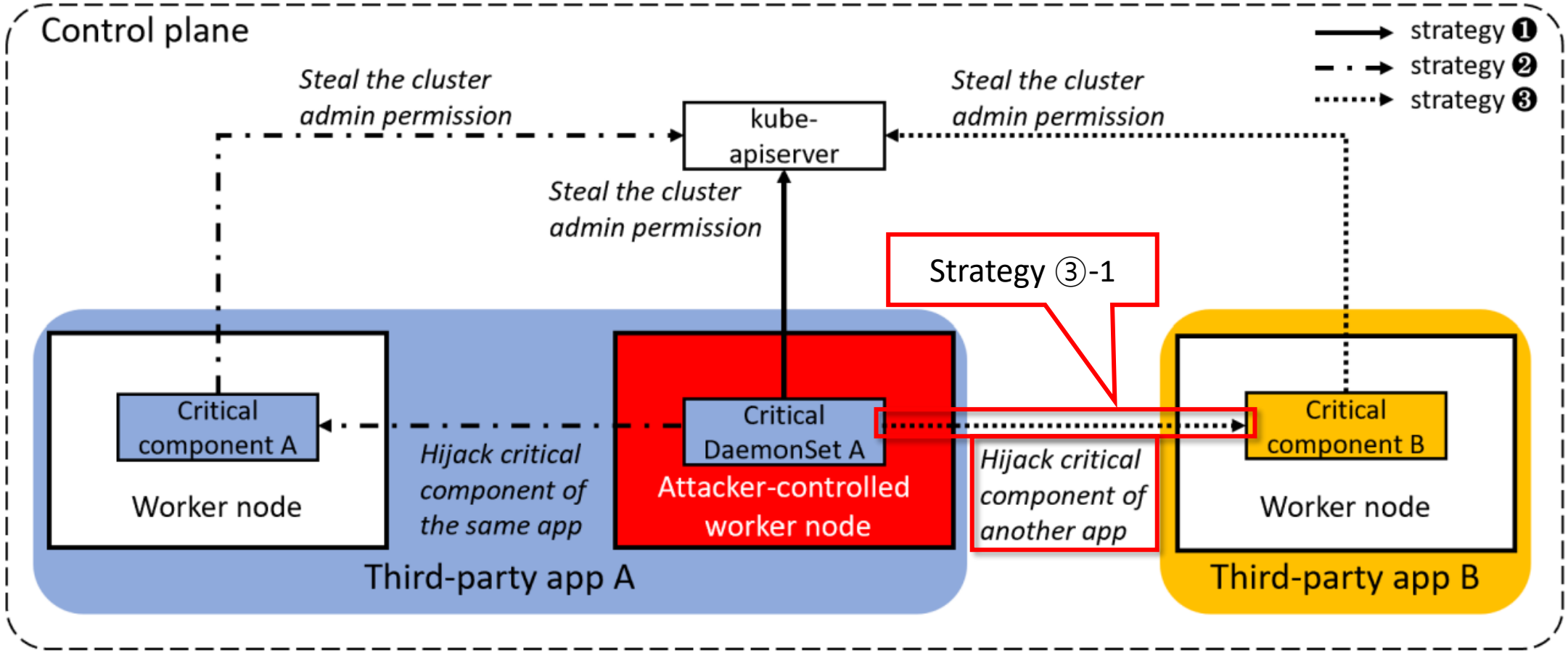
- Kubevirt
  - Virtualization workload manager which operates inside a Kubernetes cluster
  - CNCF incubating project

- Attack scenario
  1. It has a critical Daemonset **virt-handler** which uses a service account **kubevirt-handler** which is assigned ClusterRole **kubevirt-handler** that has the "patch" verb of the "nodes" resource
  2. Attacker can hijack critical component **virt-operator** by setting taint "node.kubernetes.io/unschedulable: NoExecute" to all nodes except attacker-controlled worker node
  3. That component's service account is assigned ClusterRole which has the "get/list" verbs of the "secrets" resource therefore attacker can see admin's secrets

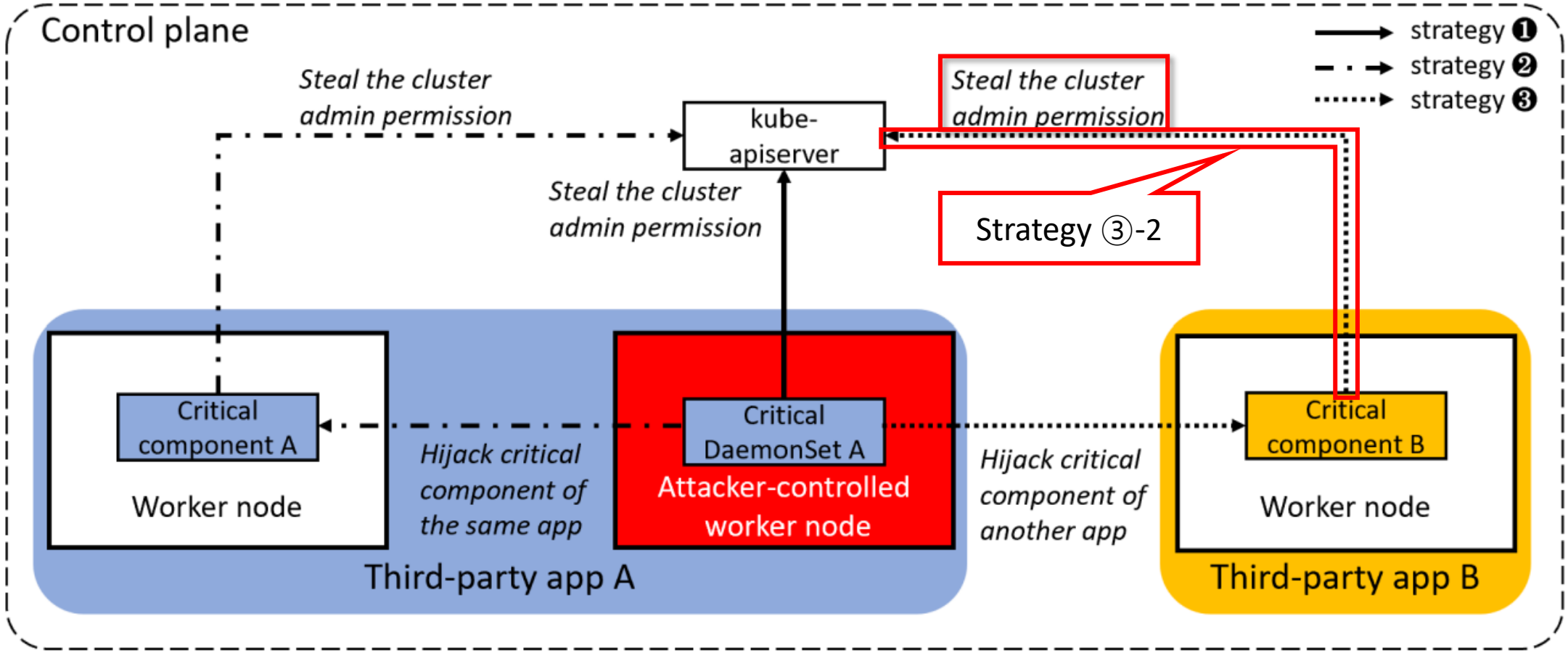
# Strategy 3


Strategy ③
Leverages critical Daemonset to hijack other app's critical component
Control plane
strategy ❶
strategy ❷
strategy ❸
Steal the cluster admin permission
kube-apiserver
Steal the cluster admin permission
Steal the cluster admin permission
Critical component A
Worker node
Hijack critical component of the same app
Critical DaemonSet A
Attacker-controlled worker node
Third-party app A
Hijack critical component of another app
Critical component B
Worker node
Third-party app B

# Strategy 3


Control plane
strategy ❶
strategy ❷
strategy ❸
Steal the cluster admin permission
kube-apiserver
Steal the cluster admin permission
Steal the cluster admin permission
Strategy ③-1
Critical component A
Worker node
Hijack critical component of the same app
Critical DaemonSet A
Attacker-controlled worker node
Third-party app A
Hijack critical component of another app
Critical component B
Worker node
Third-party app B

# Strategy 3


Control plane
strategy ❶
strategy ❷
strategy ❸
Steal the cluster admin permission
kube-apiserver
Steal the cluster admin permission
Steal the cluster admin permission
Strategy ③-2
Critical component A
Worker node
Hijack critical component of the same app
Critical DaemonSet A
Attacker-controlled worker node
Third-party app A
Hijack critical component of another app
Critical component B
Worker node
Third-party app B

# Strategy 3 Example

- Open Cluster Management(OCM)
  - CNCF sandbox project
  - 2 critical components

- Attack scenario
  1. Use another app's critical Daemonset to hijack critical component **cluster-manager-registration-controller-sa** or **cluster-manager** by setting taint to all nodes except attacker-controlled worker node
  2. One is assigned ClusterRole which has the "escalate/bind" verbs of the "clusterroles" resource and another one is additionally assigned "get/list" verbs of the "secrets" resource therefore attacker can take over whole cluster

# Summary of CNCF Projects

| Project type | Strategy❶ No. | Strategy❷ No. | Strategy ❸ No. | Identified/Total |
|---|---|---|---|---|
| Graduated | 1 | - | 1 | 2/20 |
| Incubating | 2 | 1 | 13 | 16/38 |
| Sandbox | 2 | 5 | 26 | 33/95 |
| Summary | 5 | 6 | 40 | 51/153 |

| Identified projects | 51 | | |
|---|---|---|---|
| No Response | 19 | | |
| Responded | 32 | Fixed (CVE) | 8 |
| | | Fixed (no CVE) | 2 |
| | | Pending | 22 |

# Summary of Excessive Permissions in Cloud Environments

| Vendor | App | Component | Type | Excessive permission | Report channel | Report status |
|---|---|---|---|---|---|---|
| Google GKE | Calico Network Policy | calico-node | DaemonSet | patch nodes/status | Google Bug Hunters | Confirmed |
| | Config Connector | configconnector -operator | StatefulSet | get/list secrets | | |
| | Anthos | istio-cni-node | DaemonSet | delete pods | | |
| | | activator | Deployment | get/list secrets | | |
| | | autoscaler | | | | |
| | | controller | | | | |
| | | metrics | | | | |
| | | webhook | | | | |
| Amazon EKS | Amazon VPC CNI | aws-node | DaemonSet | update nodes | AWS Security | Confirmed |
| | Tetrate Istio Distro | istiod | Deployment | get/list secrets | | |
| | Upbound Universal Crossplane | crossplane | Deployment | get/list secrets | | |
| Azure AKS | Secret store CSI driver | aks-secrets-store -csi-driver | DaemonSet | get/list secrets | Microsoft Security Response Center (MSRC) | Confirmed |
| | Azure Policy | gatekeeper-audit | Deployment | get/list *.* resources | | |
| | | gatekeeper-controller | | | | |
| Alibaba Cloud ACK | Prometheus Monitoring | node-exporter | DaemonSet | get/list secrets | Alibaba Security Response Center (ASRC) | Confirmed |
| | | arms-prometheus-ack -arms-prometheus | Deployment | get/list secrets | | |
| | | kube-state-metrics | Deployment | get/list secrets | | |
| | CSI Volume Plugin | csi-plugin | DaemonSet | get/list secrets | | |
| | Node-problem -detector | ack-node-problem -detector-daemonset | DaemonSet | * verbs of nodes | | |
| | Flannel | kube-flannel-ds | DaemonSet | patch nodes/status | | |
| | Terway | terway-eniip | DaemonSet | update/patch nodes | | |
| | Nginx Ingress | nginx-ingress -controller | Deployment | list secrets | | |
| | ALB Ingress | alb-ingress -controller | Deployment | get/list secrets | | |
| | MSE Ingress | ack-mse-ingress -controller | Deployment | get/list secrets | | |
| | CloudMonitor Agent | alicloud-monitor -controller | Deployment | get/list * resources | | |

# Mitigations

- Removing unnecessary permissions
  - Remove excessive permissions which are not required by the app’s functionalities
- Using more complex designs for service accounts
  - Use multiple service accounts with varying permission levels than a single service account with excessive permissions
- Using RoleBinding to remove cluster efforts
  - Restrict the scope of permission from whole cluster to certain namespace
  - Difficult since the namespaces requiring resource consumption may not be known in advance
- Using accurate resource names
  - Not use resource as “secrets” but use as “aaa-bb-secret”

# Conclusion

- Reveals that multiple third-party apps in the Kubernetes cluster are granted excessive permissions

- Evaluates the impact by analyzing the CNCF projects in the local cluster and third-party apps in four cloud vendors using three strategies
  - Reported to related communities and got 8 CVEs and a bounty

- Provides several actionable suggestions to mitigate the risks

# Thank you!

# Appendix

## Experiment environments

| Environments | Distributions | Topology | VM | OS | Containerd | Datset selection |
|---|---|---|---|---|---|---|
| Local | v1.25.4 | 1 control plane, 2 worker nodes | 4 cores, 8GB | Ubuntu 20.04 | 1.6.8 | CNCF project lists |
| Google GKE | v1.24.9-gke.3200 | 3 worker nodes | 2 cores, 4GB | Container-Optimized | 1.6.20 | GKE startup UI |
| Amazon EKS | v1.24.10-eks-48e63af<br>v1.23.16-eks-48e63af | 3 worker nodes | 2 cores, 4GB | Amazon Linux 2 (AL2_x86_64) | 1.6.19 | EKS startup UI |
| Azure AKS | v1.24.9 | 2 worker nodes | 4 cores, 16GB | AKSUbuntu 18.04 | 1.6.18 | AKS startup UI |
| Alibaba Cloud ACK | v1.24.6-aliyun.1 | 3 control planes, 3 worker nodes | 4 cores, 8GB | Alibaba Cloud Linux 2.1903 | 1.5.13 | ACK startup UI |

# Appendix

| App name | CVE-ID | Component name | Type | Service account | ClusterRole | Excessive permission |
|---|---|---|---|---|---|---|
| CubeFS | 2023-30512 | cfs-csi-node | DaemonSet | cfs-csi-service-account | cfs-csi-cluster-role | get/list secrets |
| OpenKruise | 2023-30617 | kruise-daemon | DaemonSet | kruise-daemon | kruise-daemon-role | get/list secrets |
| Kubevirt | 2023-26484 | virt-handler | DaemonSet | kubevirt-handler | kubevirt-handler | list/patch nodes |
| | | virt-operator | Deployment | kubevirt-operator | kubevirt-operator | get/list secrets |
| Fluid | 2023-30840 | csi-nodeplugin-fluid | DaemonSet | fluid-csi | fluid-csi-plugin | patch nodes |
| | | fluid-webhook | Deployment | fluid-webhook | fluid-webhook | get/list secrets |
| Kubewarden | 2023-22645 | kubewarden-controller | Deployment | kubewarden-controller | kubewarden-controller-manager-cluster-role | get/list secrets |
| Open Cluster Management | 2023-2250 | cluster-manager-registration-controller | Deployment | cluster-manager-registration-controller-sa | open-cluster-management:cluster-manager-registration:controller | escalate/bind verbs of clusterroles |
| | | cluster-manager | Deployment | cluster-manager | cluster-manager | escalate/bind verbs of clusterroles, get/list secrets |
| Clusternet | 2023-30622 | clusternet-hub | Deployment | clusternet-hub | clusternet:hub | * verbs of *.* resources |
| OpenFeature | 2023-29018 | open-feature-operator-controller-manager | Deployment | open-feature-operator-controller-manager | open-feature-operator-manager-role | list/update verbs of clusterrolebindings |