KubeCon | CloudNativeCon

North America 2023

RBACdoors: How Cryptominers Are Exploiting RBAC Misconfigs

Greg Castle

linkedin.com/in/greg-castle/
@mrgcastle
@gregcastle.bsky.social
GKE Security, Google Cloud

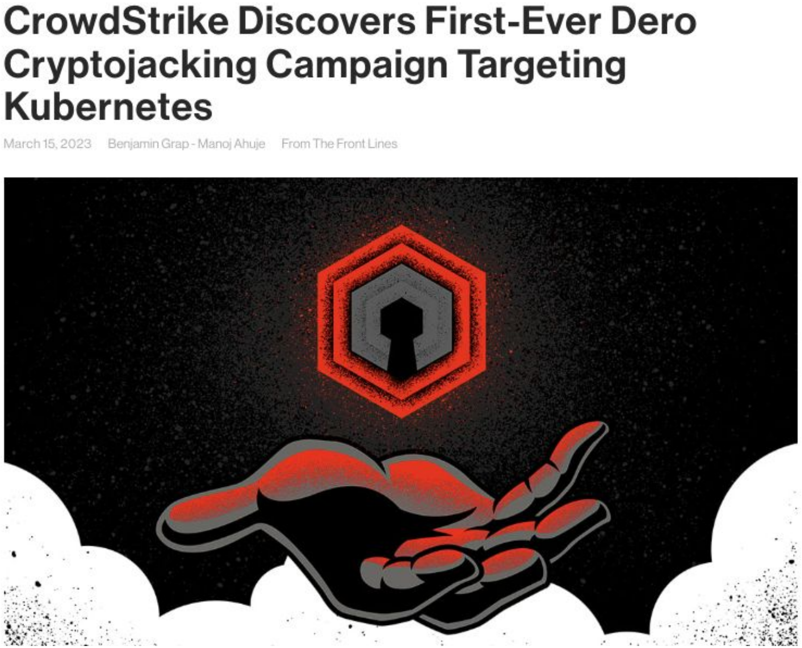
Vinayak Goyal

@vglovespizza
GKE Security, Google Cloud

Slides: https://sched.co/1R2tp

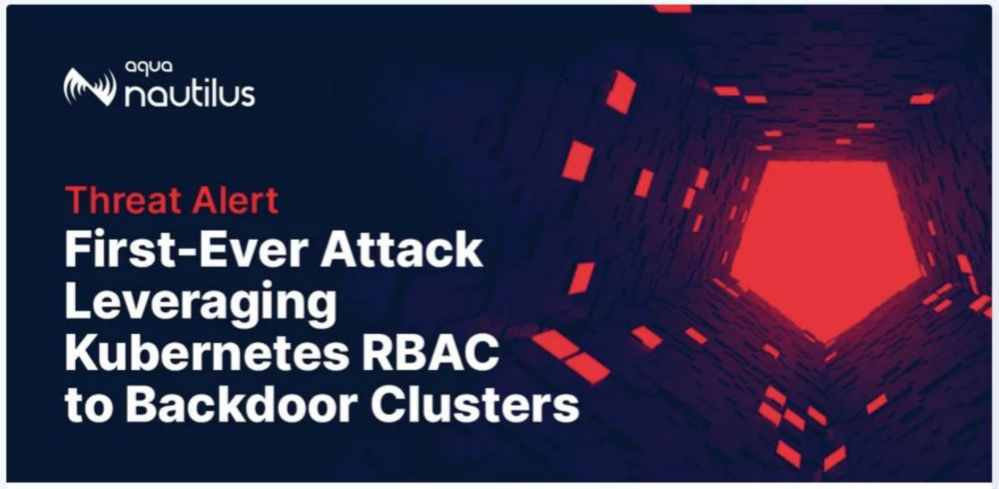
Cryptomining on Kubernetes

March/April 2023



CrowdStrike Discovers First-Ever Dero Cryptojacking Campaign Targeting Kubernetes

March 15, 2023 Benjamin Grap - Manoj Ahuje From The Front Lines



aqua nautilus

Threat Alert

First-Ever Attack Leveraging Kubernetes RBAC to Backdoor Clusters

Today

Attack walkthrough: what we saw on GKE

Root cause: Customer misconfig

Prevention

Detection

What’s new here?

- 🤷‍♀️ K8s misconfigs: docker, kubelet, dashboard
- 🤷‍♀️ Container-delivered Cryptominers
- 👍 Kubernetes-specific hiding and persistence

High Level Attack Overview

Root Cause	Customer RBAC misconfiguration: give the whole Internet cluster-admin
Discovery	Scan for API servers for access
Execution	Create role bindings to privilege kube-system service account Create cryptomining daemonset “kube-controller” in kube-system
Persistence	Create “cluster-admin” certificate with cluster-admin permissions
Defense Evasion	Remove evidence of cluster-admin cert creation

Google cloud signals alert customer, GKE assists with investigation, GKE notifies handful of other customers

What is cluster-admin?

- **All** permissions
- for **all** resources

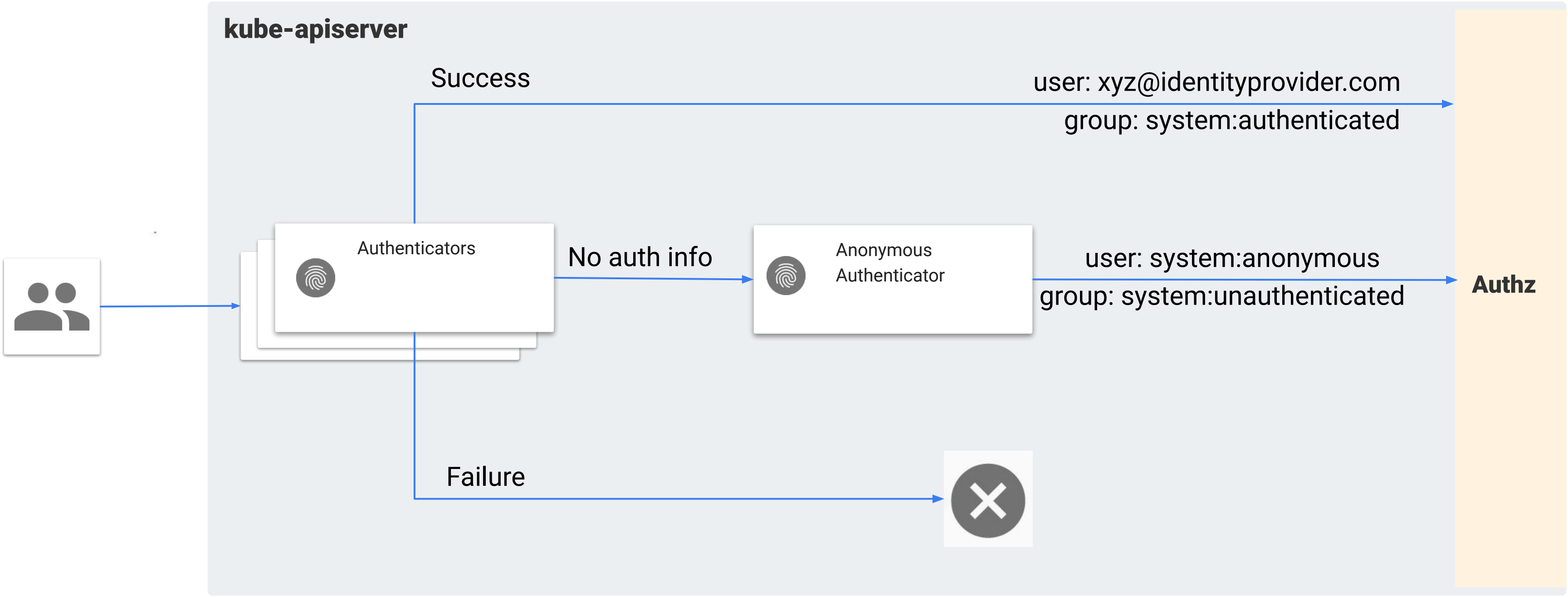
Root in cluster if used in a ClusterRoleBinding

Used many places, even when it shouldn’t be

```
$ kubectl get clusterrole cluster-admin -o yaml

apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  annotations:
  labels:
    kubernetes.io/bootstrapping: rbac-defaults
  name: cluster-admin
rules:
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - '*'
- nonResourceURLs:
  - '*'
  verbs:
  - '*'
```

What are system Users/Groups?



Uses for system:anonymous

- On by default in K8s
- Allows access to health status and version via `system:public-info-viewer` clusterRoleBinding
- Load balancers check liveness before routing
- kubeadm trust bootstrap

Note on system:authenticated

All authenticated users! That could mean:

- No-one, or something custom
- everyone@yourcompany.com
- everyone@identityprovider.com
- allcustomers@cloudprovider.com

Avoid binding large groups: least privilege

On GKE: system:authenticated same as IAM allAuthenticatedUsers

Attack Demo

Attack Observations

- Time to exploitation: 8 days from misconfig
- Blend into system noise
- XMRig bitcoin miner payload
- kubectl user-agent
- K8s updates: new image versions rather than in-payload

Prevention

Misconfig surface

- **User** system:anonymous ← Misconfig attacked
- **Group** system:unauthenticated
- **Group** system:authenticated (config dependent)

3 principals x [cluster|namespace]

= 6 different misconfig possibilities

YAML examples

Prevention Options

- Limit API server network access
- Disable anonymous auth completely
- Block *cluster-admin* bindings to system users/groups
- Block *any* bindings to system users/groups
- Block *any* bindings to cluster-admin

Limit Network Access to API Server

- Protects against: DoS, Authn/z misconfig, API server vulns
- Run in private address space
- Put a firewall in front

On GKE: private endpoint, authorized networks

Disable Anonymous Auth

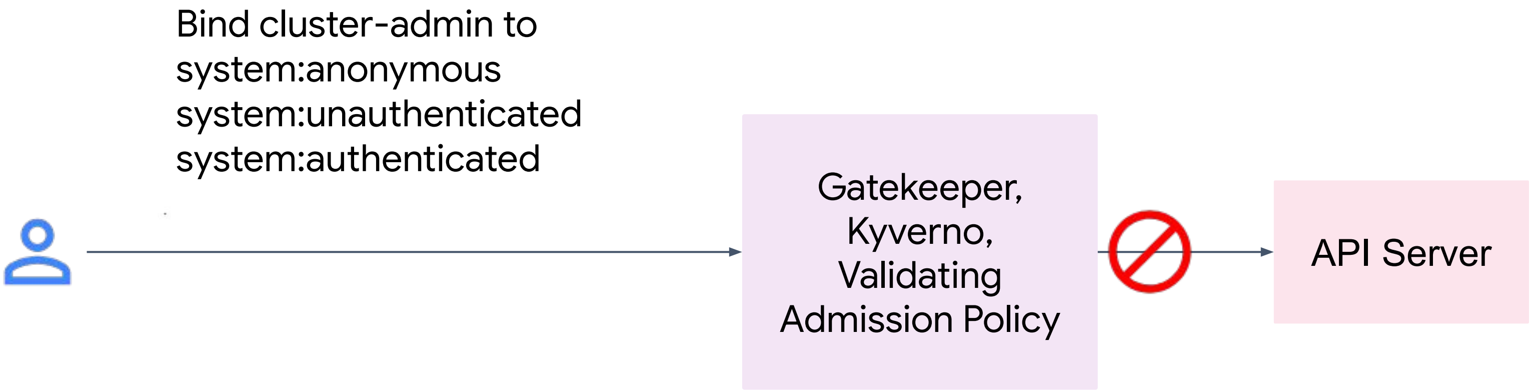
> “If you are using RBAC authorization, it is generally considered reasonable to allow anonymous access to the API Server for health checks and discovery purposes.”
>
> — K8s CIS Security Benchmark

Disable Anonymous Auth

- On by default, API server flag `anonymous-auth=false` to disable
- Can’t be disabled on many managed K8s platforms

Let’s make improvements here! Discuss at sig-auth Nov 22

Block cluster-admin Bindings



Bind cluster-admin to
system:anonymous
system:unauthenticated
system:authenticated

Gatekeeper,
Kyverno,
Validating
Admission Policy

API Server

Prevents misconfig used in the attack

On GKE: As of 1.28 blocked by default

Block *Any* Bindings to system

From our research, this may affect:

- Limited pre-auth APIs: kubeadm, rancher, bitnami sealed secrets
- CI/CD metrics
- Legacy Pod Security Policy bindings (EOL anyway)

Can be done with admission, or as detection

Block Any Bindings To cluster-admin

> *"Ensure that the cluster-admin role is only used where required"*
>
> —K8s CIS Benchmark

Careful! Widely used by privileged components

On GKE: Audit/enforce with CIS benchmark policy bundle in Policy Controller (K8sRestrictRoleBindings)

Prevent Demo

Detection

Detection Opportunities

- Misconfiguration: root cause
- Exploitation: attacker actions on K8s API
- Mining: the crypto miner itself

Misconfig Detection

Find in logs:

- Critical: cluster-admin binding to system user/groups
- Medium: any system user/groups binding
- Low: any cluster-admin binding

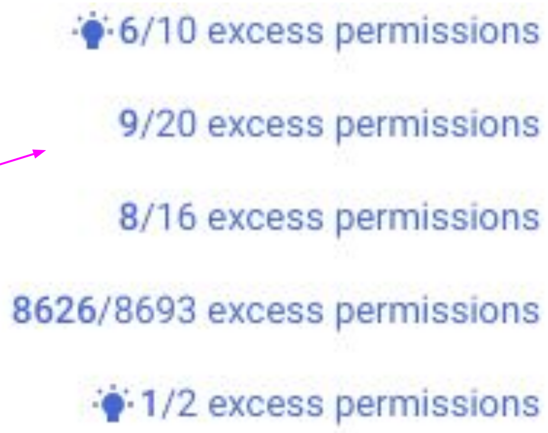
Audit existing rolebindings

On GKE: Event Threat Detection

Detect Demo

Detect Unused Permissions

- cluster-admin isn’t the only privileged role
- You can make your own!
- RBAC Police can tell you about privileged roles, but not if they are unused
- Third party tooling for unused permissions


6/10 excess permissions

9/20 excess permissions

8/16 excess permissions

8626/8693 excess permissions

1/2 excess permissions

On GKE: IAM recommender does this for bindings through IAM, but not for RBAC, yet.

Exploitation Detection

system:anonymous activity

```
protoPayload.authenticationInfo.principalEmail="system:anonymous"
protoPayload.authorizationInfo.granted="true"
```

CSR creation/approval

```
protoPayload.request.@type="certificates.k8s.io/v1.CertificateSigningRequest"
-protoPayload.authenticationInfo.principalEmail="kubelet-bootstrap"
-protoPayload.authenticationInfo.principalEmail="system:gcp-controller-manager"
-protoPayload.authenticationInfo.principalEmail=~"^system:node:"
```

Miner Detection

- Communication with known-bad IPs/domains
- Known-bad containers
- Known-bad binaries

On GKE (opt-in): Event Threat Detection and VM Threat Detection

Summary

- Interesting K8s-specific attack
- Prevent:
    - Limit network access
    - Block RBAC bindings to system users/groups
- Detect:
    - Audit existing rolebindings
    - system:anonymous and CSRs in logs
    - Unused permission detection
    - Cryptominer detection

Links

- Demo Code
- Attack Blogs: Crowdstrike, Aqua, Raesene
- K8s Anonymous Auth
- System Groups on GKE
- RBAC best practices: K8s, GKE
- Limit network access to API server hardening advice
- Gatekeeper DisallowAnonymous
- Validating admission policy: beta in K8s 1.28
- Auditing and cleaning up anonymous bindings

Talk Feedback

Detection Log Queries

cluster-admin bindings to system:*

```
resource.type="k8s_cluster"
resource.labels.cluster_name="vulnerable-cluster"
resource.labels.location="us-central1-c"
protoPayload.request.roleRef.name="cluster-admin"
protoPayload.request.subjects.name="system:anonymous" OR
protoPayload.request.subjects.name="system:authenticated" OR
protoPayload.request.subjects.name="system:unauthenticated"
protoPayload.methodName:"io.k8s.authorization.rbac.v1.rolebindings" OR
protoPayload.methodName:"io.k8s.authorization.rbac.v1.clusterrolebindings"
-protoPayload.response.code="403"
```

System anonymous actions:

```
resource.type="k8s_cluster"
resource.labels.cluster_name="vulnerable-cluster"
resource.labels.location="us-central1-c"
protoPayload.authenticationInfo.principalEmail="system:anonymous"
protoPayload.authorizationInfo.granted="true"
```

Detection Log Queries

CSR Creation and Approvals

```
resource.type="k8s_cluster"
resource.labels.cluster_name="vulnerable-cluster"
resource.labels.location="us-central1-c"
protoPayload.request.@type="certificates.k8s.io/v1.CertificateSigningRequest"
-protoPayload.authenticationInfo.principalEmail="kubelet-bootstrap"
-protoPayload.authenticationInfo.principalEmail="system:gcp-controller-manager"
-protoPayload.authenticationInfo.principalEmail=~"^system:node:"
```