RSAConference2022
San Francisco & Digital | February 7 – 10

TRANSFORM

SESSION ID: **OST-T09**

Attacking and Defending Kubernetes Cluster: Kubesploit vs KubiScan

Eviatar Gerzi

Sr. Security Researcher
CyberArk
@g3rzi

Disclaimer

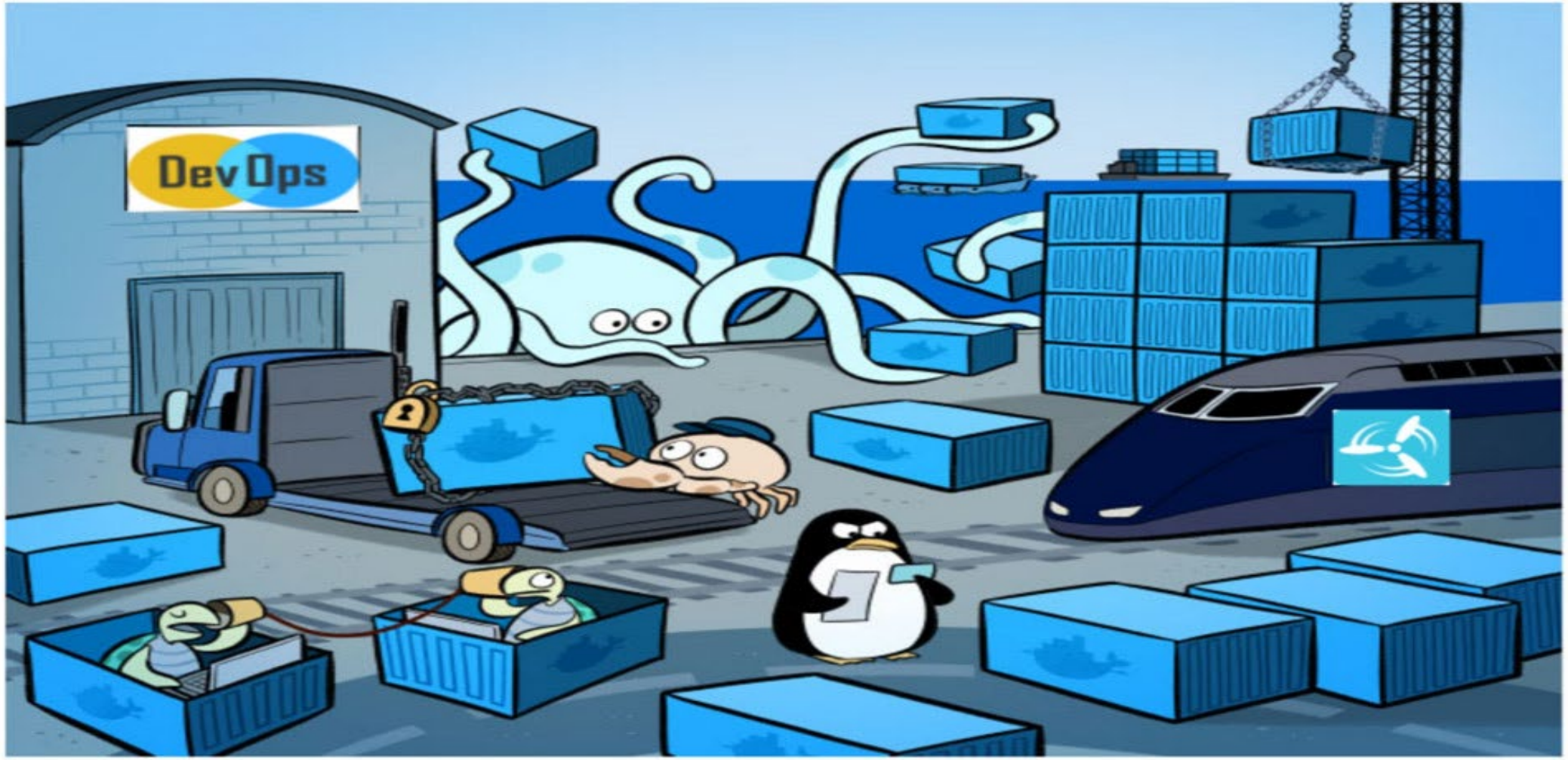
Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

What We Will Talk About?

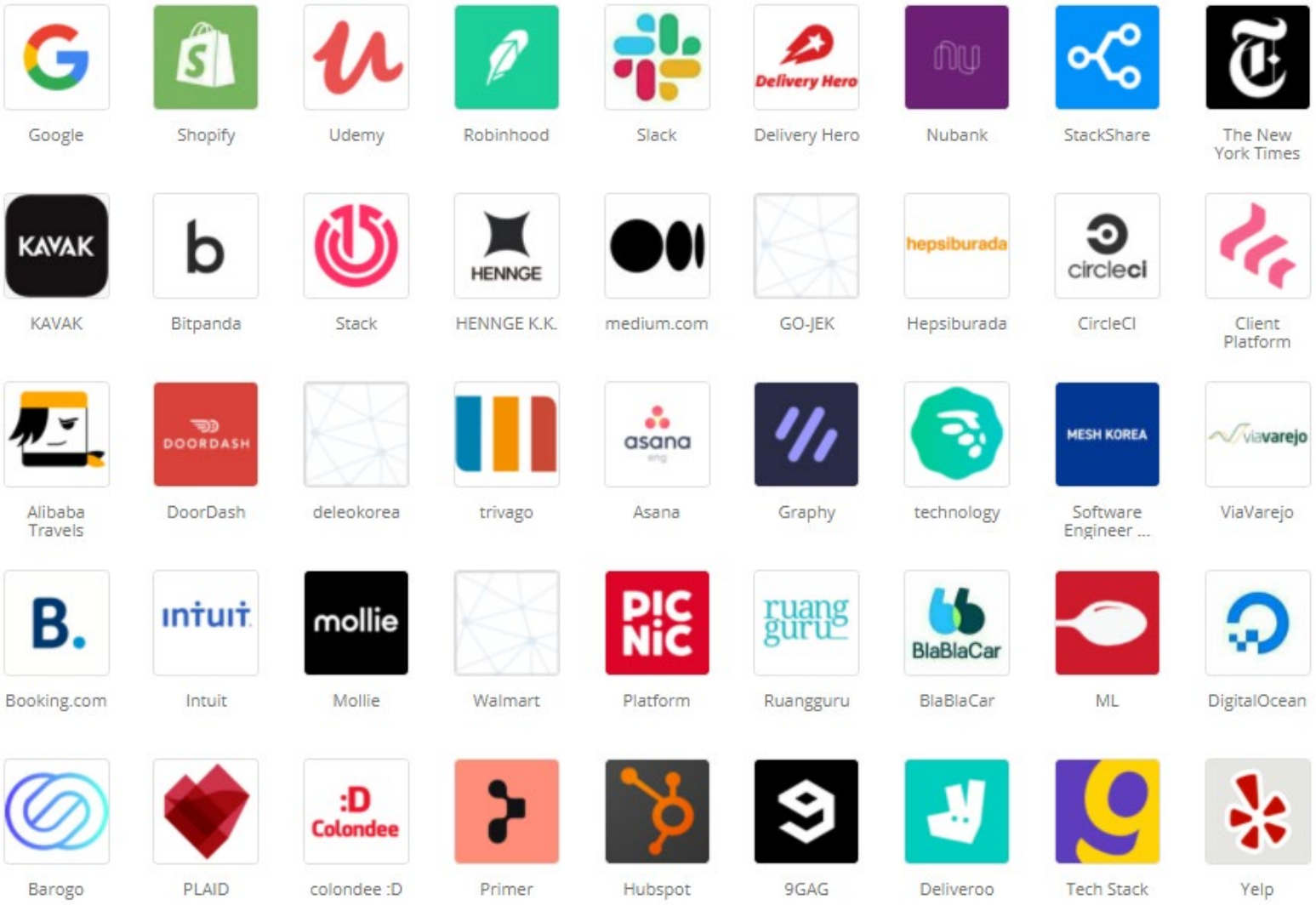
- Quick overview on Kubernetes
- Attacking surface
- Demonstrate attack with **Kubesploit**
- Defending k8s cluster
- Demonstrate defend with **KubiScan**
- 10 Tips to protect your cluster

What Does Kubernetes Do?



https://hackernoon.com/practical-introduction-to-docker-compose-d34e79c4c2b6

Who Uses Kubernetes?



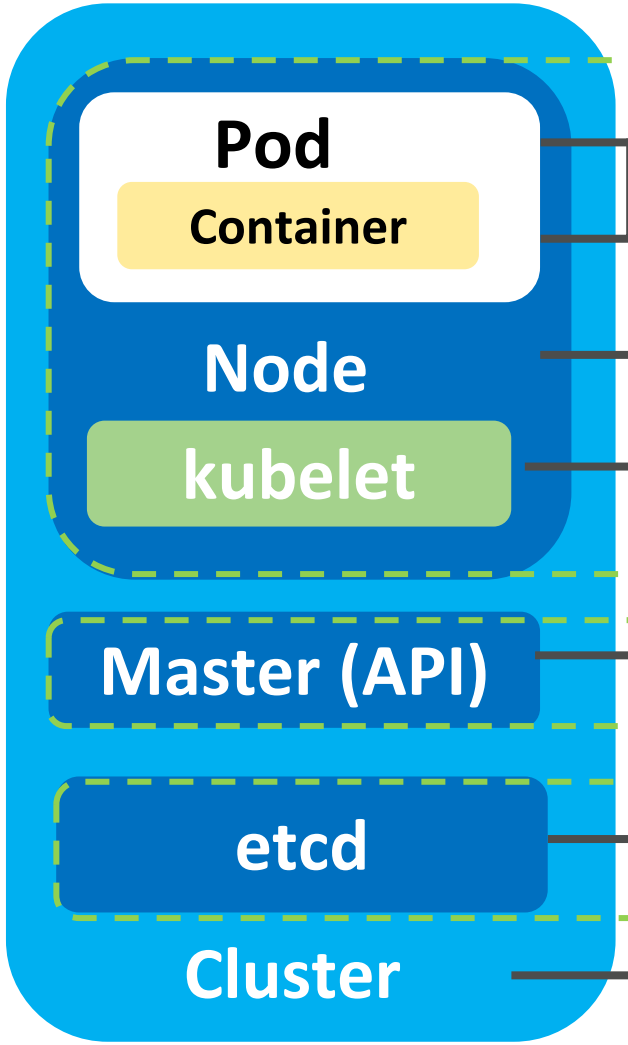
2863

https://stackshare.io/kubernetes

RSAConference2022

Attacking Kubernetes Cluster

Vital Target Components



If attacker controls a **container**, they may be able to escape and attack the node

If attacker controls **nodes**, they also control the pods running on them and can potentially attack masters or abuse compute

If attacker controls the **kubelet**, they control the pods

If attacker controls the **master**, they control your cluster

If attacker controls the **etcd**, they can access, modify, or destroy the cluster

A **cluster** is set of nodes on which containers are scheduled

Attack Vectors

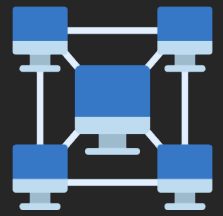
Kernel Exploit

CVE-2019-5736

Container
Runtime Exploit

Kubernetes
misconfiguration

Application



Network

Threat Matrix for Kubernetes

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Impact
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access the K8S API server	Access cloud resources	Images from a private registry	Data Destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account		Resource Hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access container service account	Network mapping	Cluster internal networking		Denial of service
Application vulnerability	Application exploit (RCE)	Malicious admission controller	Access cloud resources	Connect from Proxy server	Applications credentials in configuration files	Access Kubernetes dashboard	Applications credentials in configuration files		
Exposed Dashboard	SSH server running inside container				Access managed identity credential	Instance Metadata API	Writable volume mounts on the host		
Exposed sensitive interfaces	Sidecar injection				Malicious admission controller		Access Kubernetes dashboard		
							Access tiller endpoint		
							CoreDNS poisoning		
							ARP poisoning and IP spoofing		

= New technique

= Deprecated technique

CYBERARK

https://www.microsoft.com/security/blog/2021/03/23/secure-containerized-environments-with-updated-threat-matrix-for-kubernetes/

Vulnerabilities from HackerOne

100	Exposed Kubernetes API - RCE/Exposed Creds	By txtsrob to Snapchat	Resolved	Critical	$25,000.00	disclosed 3 months ago
10	Unauthorized Kubernetes to RCE (root) and found TEAMTNT Crypto Miner on it	By un_kn0wn to IBM	Resolved	Critical		disclosed 9 days ago
16	SSRF for kube-apiserver cloudprovider scene	By lazydog to Kubernetes	Resolved	Medium		disclosed 20 days ago
42	Loading YAML in Java client can lead to command execution	By j0v to Kubernetes	Resolved	Medium	$1,000.00	disclosed 3 months ago
4	Man in the middle leading to root privilege escalation using hostNetwork=true (CAP_NET_RAW considered harmful)	By champtar to Kubernetes	Informative	Medium		disclosed 19 days ago
11	Node Validation Admission does not observe all oldObject fields	By ariellima to Kubernetes	Resolved	Medium	$1,000.00	disclosed 2 months ago
13	Index Out Of Bounds in protobuf unmarshalling	By pulpkk to Kubernetes	Resolved	None	$250.00	disclosed 2 months ago
5	Holes in EndpointSlice Validation Enable Host Network Hijack	By howardjohn to Kubernetes	Resolved	Low	$200.00	disclosed 2 months ago
100	Node disk DOS by writing to container /etc/hosts	By kebe to Kubernetes	Resolved	Medium	$1,000.00	disclosed about 1 year ago

https://hackerone.com/hacktivity?querystring=kubernetes

RSAConference2022

Kubesploit

Kubesploit – what is it?

A cross-platform post-exploitation HTTP/2 Command & Control server and agent dedicated for containerized environments written in Golang and built on top of Merlin project by Russel Van Tuyl (@Ne0nd0g).

Kubesploit

- **Container breakout**
  - Mounting
  - Docker.sock
  - CVE-2019-5736 (runC)
  - Pod escape through /var/log
  - cGroup
  - Kernel Module
- **Kubelet attack**
  - Scan for containers with RCE
  - Execute arbitrary commands with multiple options
  - Scan for Pods and containers
  - Scan for tokens form all available containers
- **Scan for Kubernetes cluster known CVEs**
- **Kubernetes service scan**
- **Port scanning**
- **Module scanner**
- **Deepce**

https://github.com/cyberark/kubesploit

Demo #1 - Kubesploit

```
root@priv-pod:/tmp#
```

3. Agent in privileged container

```
[kubesploit]$
```

2. Kubesploit

Katacoda playground

https://www.katacoda.com/cyberarkcommons/scenarios/kubesploit

Welcome!

Kubernetes with Kubesploit

Difficulty: Beginner Estimated Time: 60

In this scenario, you will learn how to work with Kubesploit, a cross-platform post-exploitation HTTP/2 Command & Control server and agent written in Golang, focused on containerized environments. We created a Kubernetes environment for you to play with Kubesploit. Let's start!

Mitigation

- YARA rules
- Agent recording
- Module mitigation table

RSAConference2022

Defending Kubernetes Cluster

Best practices to harden your cluster

Maturity

Set up a cluster

- Restrict access to kubectl
- Use RBAC
- Use a Network Policy
- Use namespaces
- Bootstrap TLS

Follow security hygiene

- Keep Kubernetes updated
- Use a minimal OS
- Use minimal IAM roles
- Use private IPs on your nodes
- Monitor access with audit logging
- Verify binaries that are deployed

Prevent known attacks

- Disable dashboard
- Disable default service account token
- Protect node metadata
- Scan images for known vulnerabilities

Prevent/limit impact of microservice compromise

- Set a Pod Security Policy
- Protect secrets
- Consider sandboxing
- Limit the identity used by pods
- Use a service mesh for authentication & encryption

Google Cloud

https://static.sched.com/hosted_files/kccna18/1c/KubeCon%20NA%20-%20This%20year%2C%20it%27s%20about%20security%20-%2020181211.pdf

RSA Conference2022

KubiScan

KubiScan

- Risky (Cluster)Roles
- Risky (Cluster)RoleBindings
- Risky Subject (Users, Groups and ServiceAccounts)
- Risky Pods\Containers
- All mounted volumes to Pods
- All mounted environment variables to Pods
- Privileged Pods
- Other cool stuff ☺

https://github.com/cyberark/KubiScan

Demo #2 - KubiScan

```
[kubiscan]#
```

RSAConference2022

10 Tips to Protect Your Kubernetes Cluster

1. Access to Kubelet

/var/lib/kubelet/config.yaml

```yaml
authentication:
  anonymous:
    enabled: false
  ...
authorization:
  mode: Webhook
  ...
readOnlyPort: 0
```

Not set to "AlwaysAllow"

2. Access to ETCD

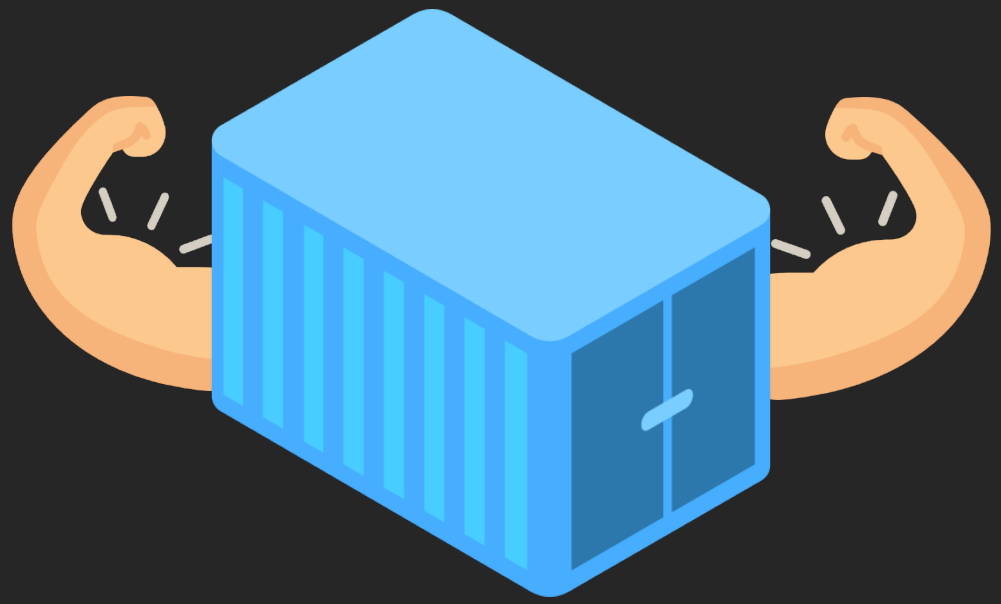
/etc/kubernetes/manifests/etcd.yaml

```
- --client-cert-auth=true
- --auto-tls=false
- --peer-client-cert-auth=true
- --peer-auto-tls=false
```

3. Restrict ServiceAccountToken

```
automountServiceAccountToken: false
```

4. Restrict privileged containers



5. Use PodSecurityPolicy

```yaml
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: restricted
spec:
  privileged: false
  allowPrivilegeEscalation: false
  requiredDropCapabilities:
    - ALL
  # Allow core volume types.
  volumes:
    - 'secret'
  hostNetwork: false
  hostIPC: false
  hostPID: false
  runAsUser:
    # Require the container to run without root privileges.
    rule: 'MustRunAsNonRoot'
```

Will be **deprecated** from Kubernetes 1.25.

It will be replaced by **Pod Security Standards (PSS).**

Pod Security Admission, is the mechanism that implements the **PSS**.

6. Use Network Policy

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: multi-port-egress
  namespace: default
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Egress
  egress:
  - to:
    - ipBlock:
        cidr: 10.0.0.0/24
    ports:
    - protocol: TCP
      port: 32000
      endPort: 32768
```

7. API Server AuthN & AuthZ

/etc/kubernetes/manifests/kube-apiserver.yaml

```
--anonymous-auth=false
--basic-auth-file
--token-auth-file
```

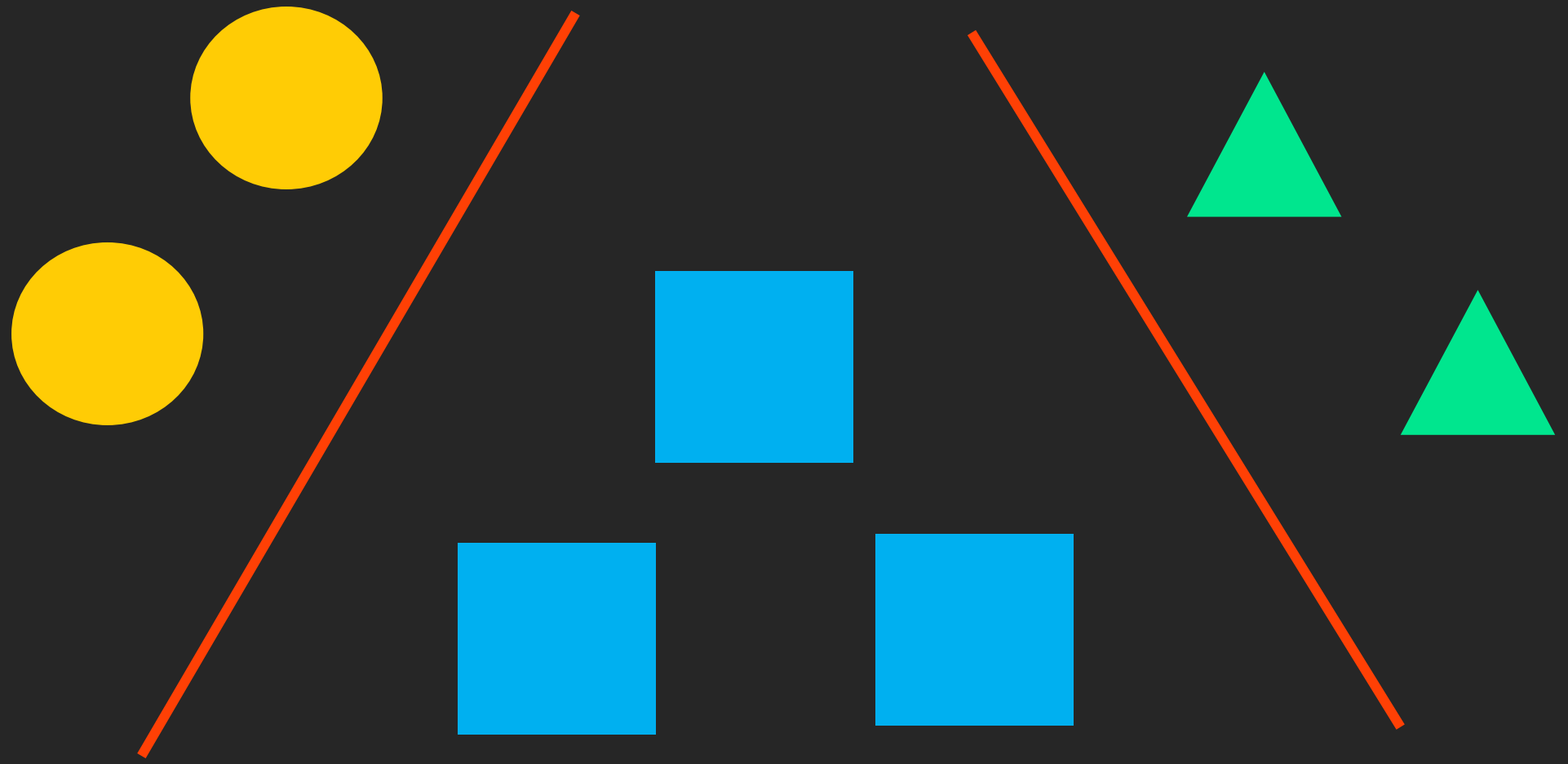
Not set

8. Use RBAC

/etc/kubernetes/manifests/kube-apiserver.yaml

```
--authorization-mode=RBAC
```

9. Namespace separation



10. Use Kubiscan

```
kubiscan -rp
kubiscan -pp
kubiscan -pse
kubiscan -psv
```

- `kubiscan -rp` → Containers with privileged token
- `kubiscan -pp` → Privileged containers
- `kubiscan -pse` → Mounts via Environment Variables
- `kubiscan -psv` → Mounts via Volumes

Recommended Tools and Links

- KubiScan
- Kubesploit
- Kube-bench (CIS Benchmarks)
- Kubestriker
- Kube-hunter
- kubescape
- https://github.com/magnologan/awesome-k8s-security

Summary

- We reviewed common attack vectors on Kubernetes
- We demonstrate attacks with **Kubesploit** and show how it can be defended with **KubiScan**
- Follow the best practices to protect your cluster

Apply What You Have Learned Today

Next week

- Use KubiScan to protect your cluster
- Follow the protections tips and implement them

3 months

- Search for cluster misconfiguration or insecure pods
- Use Kubesploit to test your protections

+6 months

- Have a secured cluster with:
  - Namespace separation
  - No privileged containers
- Create an awareness in your organization

THANK YOU!

@g3rzi

https://github.com/cyberark/kubesploit

https://github.com/cyberark/KubiScan