Defenders Guide to Kubernetes Security

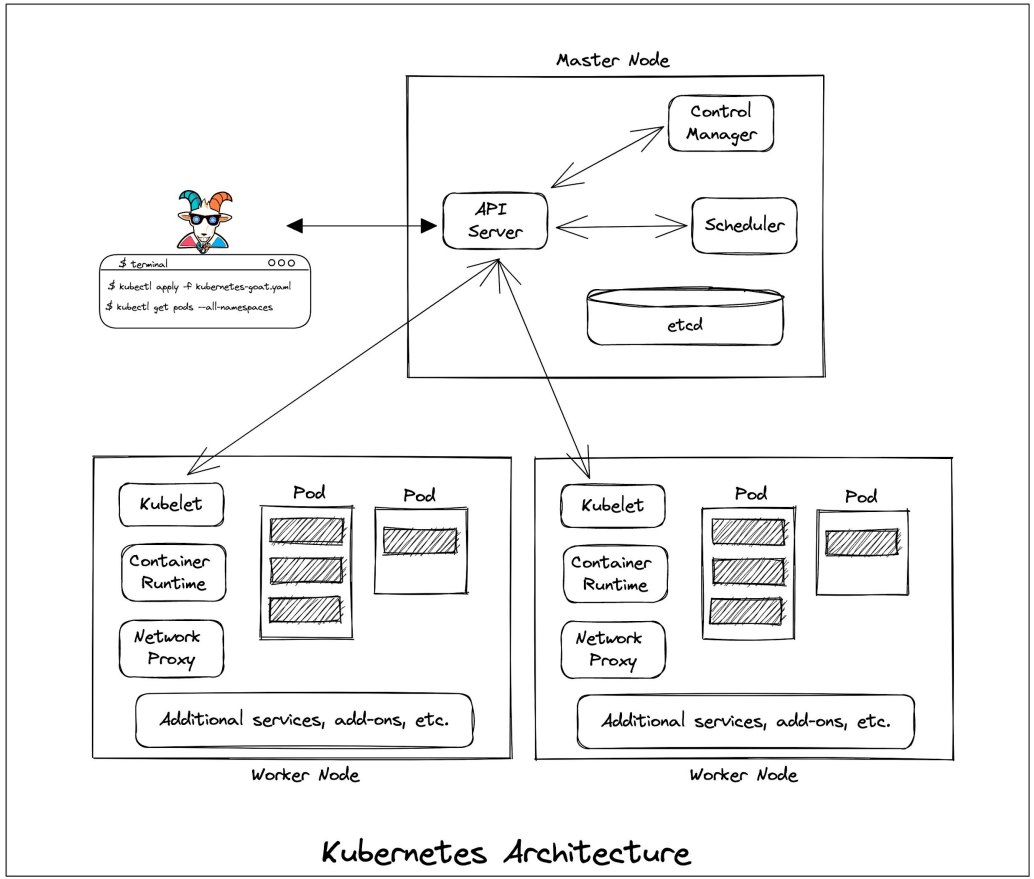
Madhu Akula

@madhuakula

About Me 😊

- Creator of Kubernetes Goat, Hacker Container, tools.tldr.run, many other OSS projects.
- Speaker & Trainer at Blackhat, DEFCON, GitHub, USENIX, OWASP, All Day DevOps, SANS, DevSecCon, CNCF, c0c0n, Nullcon, SACON, null, many others.
- Author of Security Automation with Ansible2, OWASP KSTG, whitepapers, etc.
- Technical reviewer (multiple books) & Review board member of multiple conferences, organizations, communities, etc.
- Found security vulnerabilities in 200+ organizations and products including Google, Microsoft, AT&T, Adobe, WordPress, Ntop, etc.
- Community member of null, ADDO, AWS, CNCF, OWASP, USENIX, Snyk Ambassadors, etc.
- Certified Kubernetes Administrator, Offensive Security Certified Professional, etc.
- Never ending learner!

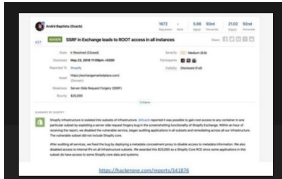
Overview of the Kubernetes



[Kubernetes] is an open source container orchestration engine for automating deployment, scaling, and management of containerized applications. The open source project is hosted by the Cloud Native Computing Foundation ([CNCF]).

@madhuakula

Why do we have to think about Security?



https://hackerone.com/reports/341876

BSidesSF CTF cluster pwn

The challenges for the BsidesSF CTF were run in Docker containers on Kubernetes using Google Container Engine. Because of the two infrastructure issues, it was possible to exploit one of the early challenges, steal service account keys, and then use those keys to directly access flags.

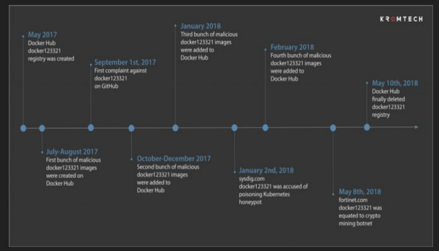
https://hackernoon.com/capturing-all-the-flags-in-bsidessf-ctf-by-owning-our-infrastructure-3570b99b4d0

docker

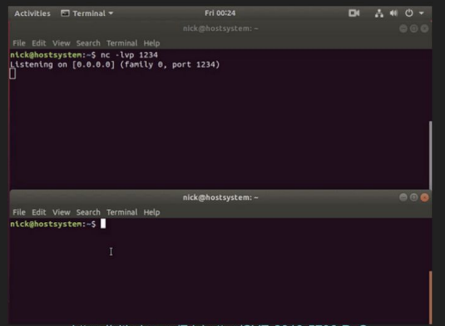
On Thursday, April 25th, 2019, we discovered unauthorized access to a single Hub database storing a subset of non-financial user data. Upon discovery, we acted quickly to intervene and secure the site.

We want to update you on what we've learned from our ongoing investigation, including which Hub accounts are impacted, and what actions users should take.

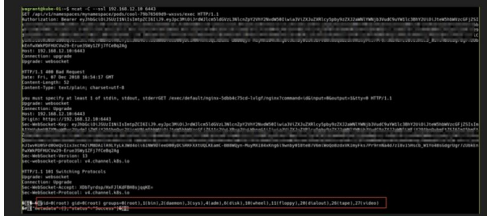
https://blog.madhuakula.com/some-tips-to-review-docker-hub-hack-of-190k-accounts-addcd602aade



https://kromtech.com/blog/security-center/cryptojacking-invades-cloud-how-modern-containerization-trend-is-exploited-by-attackers



https://github.com/Frichetten/CVE-2019-5736-PoC



https://www.youtube.com/watch?v=4CTK2aUXTHo

The bug

I set up a simple project with a web server and deployed it on Kubernetes. The web application had two endpoints `/public/` and `/secret/`. I added an authorization policy which tried to grant access to anything below `/public/`:

```
rules:
- services: ["backend.fishy.svc.cluster.local"]
  methods: ["GET"]
  paths: ["/public/*"]
```

I then used standard **path traversal** from curl:

```
curl -vvvv --path-as-is  "http://backend.fishy.svc.cluster.local:8081/public/../secret/"
```

And was able to reach `/secret/`.

https://github.com/eoftedal/writings/blob/master/published/CVE-2019-9901-path-traversal.md

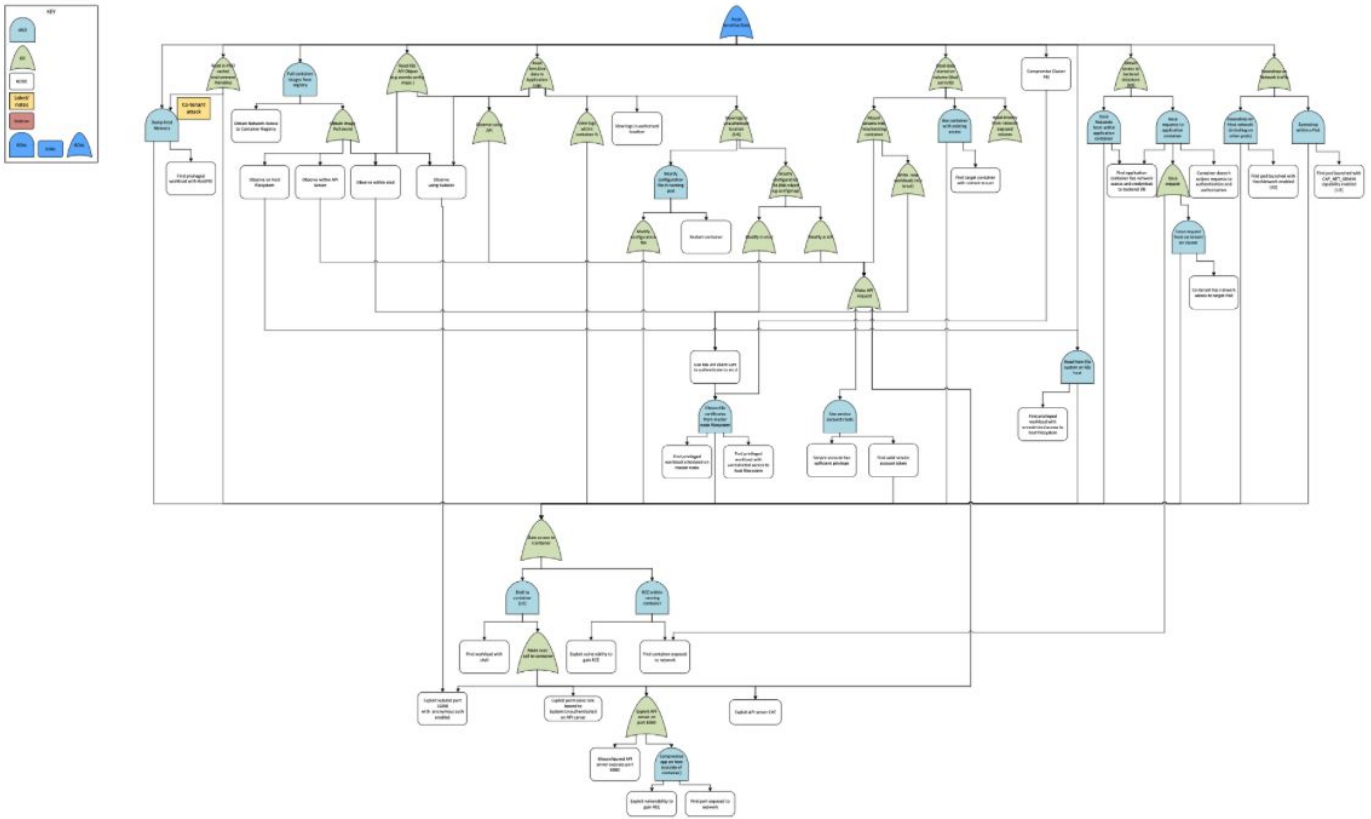
```
root@incluster:/# helm --host tiller-deploy.kube-system:44134 version
Client: &version.Version{SemVer:"v2.12.3", GitCommit:"eecf22f77df5f65c8
Server: &version.Version{SemVer:"v2.12.1", GitCommit:"02a47c7249b1fc6d
root@incluster:/#
```

```
root@incluster:/# helm --host tiller-deploy.kube-system:44134 install /pwnchart
NAME:   maudlin-rabbit
LAST DEPLOYED: Mon Feb  4 09:00:56 2019
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1beta1/ClusterRole
NAME          AGE
all-your-base  0s

==> v1beta1/ClusterRoleBinding
NAME          AGE
belong-to-us  0s
```

https://engineering.bitnami.com/articles/helm-security.html

@madhuakula

Why do we have to think about Security?



https://github.com/cncf/financial-user-group/blob/master/projects/k8s-threat-model/AttackTrees/AccessSensitiveData.md

Oops! that isn't good

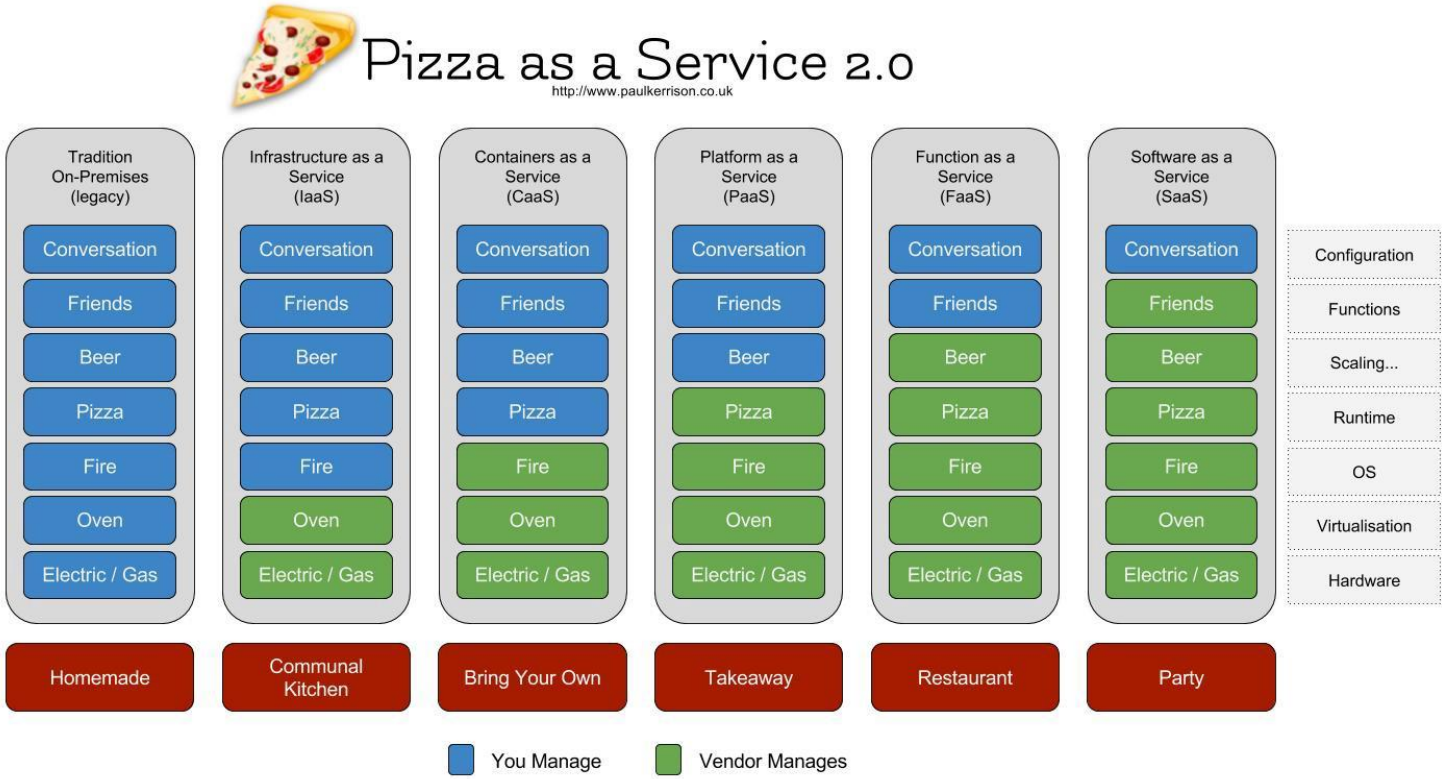
Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Impact
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access the K8S API server	Access cloud resources	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

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

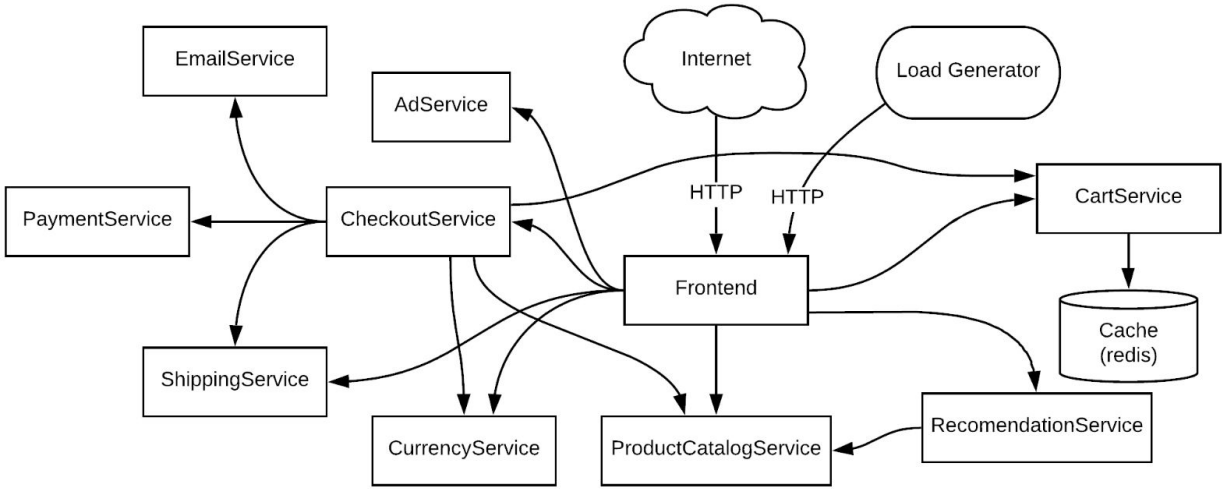
That's Crazy! Isn't our managed providers solving this?



https://medium.com/@pkerrison/pizza-as-a-service-2-0-5085cd4c365e

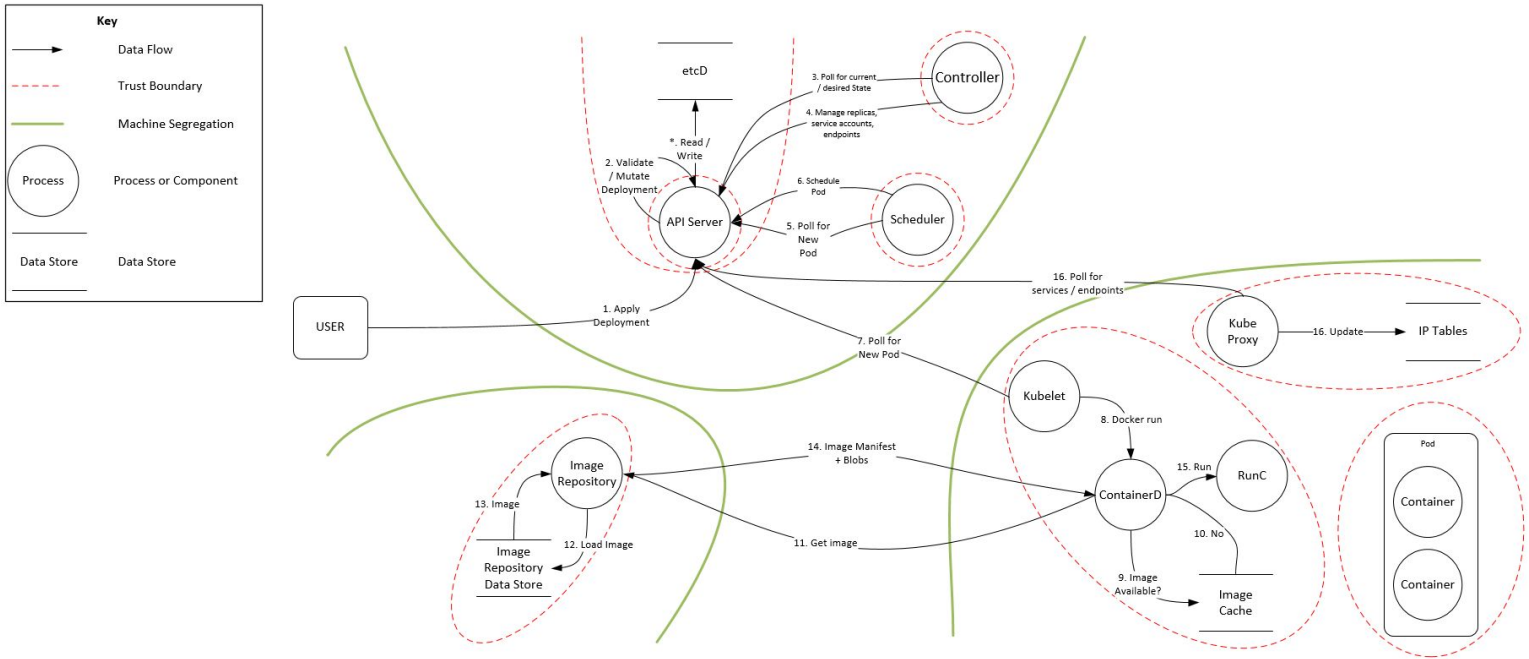
Okay, Let's start by writing a simple Microservice?

Online Boutique is a cloud-native demo application with 10 microservices showcasing Kubernetes, Istio, gRPC and OpenCensus.



https://github.com/GoogleCloudPlatform/microservices-demo/

Okay, Let's start by writing a simple Microservice?



https://github.com/cncf/financial-user-group/tree/master/projects/k8s-threat-model

Write the application code

- Code quality analysis (Ex: SonarQube)
- Security linters (Ex: Findsecbugs)
- Sensitive Info/Secrets Analysis
- Dependency security Analysis Checks
- Supply chain security analysis
- Static Code Security Analysis
- Dynamic Security Analysis
- Semantic/Variant Analysis (Ex: Semgrep, CodeQL)
- Many more...

pip install ‘pyyaml==5.4’

```
import flask
import yaml

app = flask.Flask(__name__)
app.config["DEBUG"] = True

@app.route('/', methods=['GET'])
def home():
    return "Welcome to Kubernetes world!"

app.run()
```

Package the application into a container aka Dockerfile

- Dockerfile best practices
- Linters, tools, techniques
- BuildKit for the safety
- Hadolint, Dockle, Checkov, KICS, etc.
- docker-slim for looking deeper layers
- dive: explore layers!
- IDE integrations (VSCode, k8slens.dev, IntelliJ, etc.)
- OPA & Conftest with custom policies & Rego
- Always context matters 😎

```
FROM randomuser/python:latest

ENV SECRET AKIGG23244GN2344GHG

USER root

WORKDIR /app

COPY requirements.txt requirements.txt
RUN pip3 install -r requirements.txt

COPY . .

CMD [ "flask", "run", "--host=0.0.0.0" ]
```

https://medium.com/miro-engineering/a-practical-guide-to-writing-secure-dockerfiles-bf561224dd80

Push these changes to Version Control System

- Pre/Post commit hooks
- Secrets scanning (cool project: OWASP WrongSecrets) - Trufflehog, Gitleaks, etc.
- Scanning for the container vulnerabilities (System, SBOM, Dependencies, Packages, etc.)
- Supply chain security risks (signing, verification, packages, artefacts, etc.)
- Permissions, privileges and changes
- Risk analysis of the code, packages, permissions, build
- All the amazing automation comes here 😊

So, what happens now?

It's time for the CI/CD stuff!

- Build systems, configuration and the context
- Runners, segmentation, privileges, socket mounts, volumes, many other…
- All your pipelines comes handy here
    - SCA, SAST, DAST, Secrets, Container, IaC, Code, Supply Chain, RBAC, etc.
- Having policies, processes for registries, artefacts
- Podman, Distroless, Docker-Slim, Custom stuff
- Short-lived, Least privileged access for the infrastructure
- Many others…

I'm ready now, where do I go?

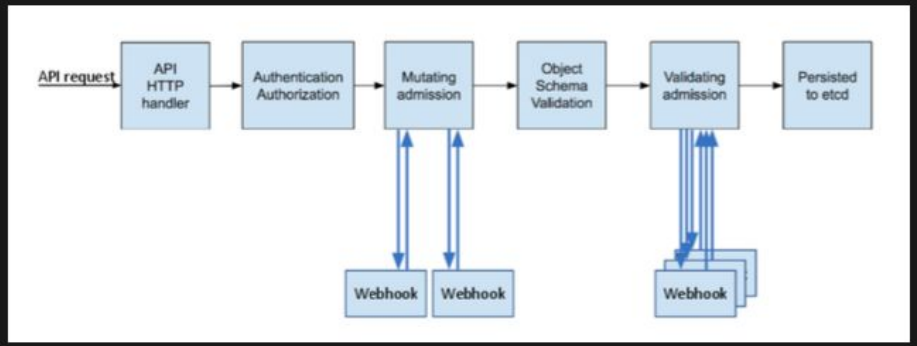
Here comes the Infrastructure aka our K8S cluster ⎈

- Infrastructure Code (Terraform, Ansible, AMIs, Configurations, etc.)
  - KICS, Kubescape, Checkov, Kubesec.io, etc. for performing scanning for these IaC
- Hardening using standards and benchmarks like CIS, NSA, etc.
- Applying sane secure defaults (AppArmor, gVisor, NSP, PSS, RBAC, OPA, many others.)
- Handling the operations well (Secrets Management, TLS, mTLS, Ingress, LB, Storage, etc.)
- Cloud providers security configurations and best practices (Metadata, IAM, NSG, etc.)
- Preventive & Detective controls (OPA, Kyverno, SecurityContext, PSS, Webhooks, etc.)
- Continuous security visibility, monitoring, detection and alerting in place
  - Audits, Risk analysis, Runtime Sandboxing, External Connections, Add-ons, etc.

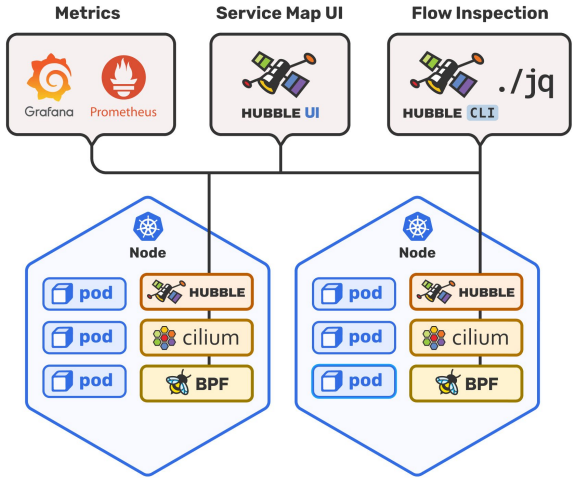
Oh! This is pretty cool, how can I be more awesome?

Go beyond normal paranoia and threat actors🕵

I think something went wrong!!!

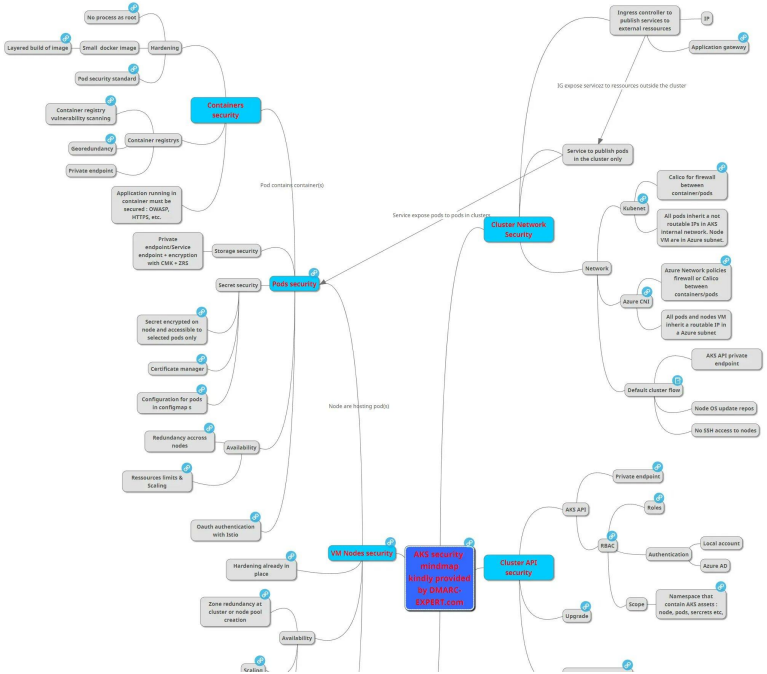



```
20:45:03.307315701: Warning Sensitive file opened for reading by non-trusted program (user=root user_loginuid=-1 program=cat command=cat /etc/shadow file=/etc/shadow parent=sh gparent=<NA> ggparent=<NA> gggparent=<NA> container_id=ad1146dd2f84 image=madhuakula/hacker-container) k8s.ns=default k8s.pod=bash container=ad1146dd2f84 k8s.ns=default k8s.pod=bash container=ad1146dd2f84
20:45:05.548736395: Error File below / or /root opened for writing (user=root user_loginuid=-1 command=cilium-cni parent=kubelet file=/root/.config/gops/23078 program=cilium-cni container_id=host image=<NA>) k8s.ns=<NA> k8s.pod=<NA> container=host k8s.ns=<NA> k8s.pod=<NA> container=host
20:45:05.961919096: Error File below / or /root opened for writing (user=root user_loginuid=-1 command=cilium-cni parent=kubelet file=/root/.config/gops/25548 program=cilium-cni container_id=host image=<NA>) k8s.ns=<NA> k8s.pod=<NA> container=host k8s.ns=<NA> k8s.pod=<NA> container=host
20:45:10.597825630: Error File below / or /root opened for writing (user=root user_loginuid=-1 command=cilium-cni parent=kubelet file=/root/.config/gops/23112 program=cilium-cni container_id=host image=<NA>) k8s.ns=<NA> k8s.pod=<NA> container=host k8s.ns=<NA> k8s.pod=<NA> container=host
20:45:11.009677355: Error File below / or /root opened for writing (user=root user_loginuid=-1 command=cilium-cni parent=kubele

~ #
~ # cat /etc/shadow
root:!::0:::::
bin:!::0:::::
daemon:!::0:::::
adm:!::0:::::
lp:!::0:::::
```



https://github.com/cilium/hubble

Cool, anything else?



[AKS - MindMap]

https://twitter.com/david_das_neves/status/1544367262093791232

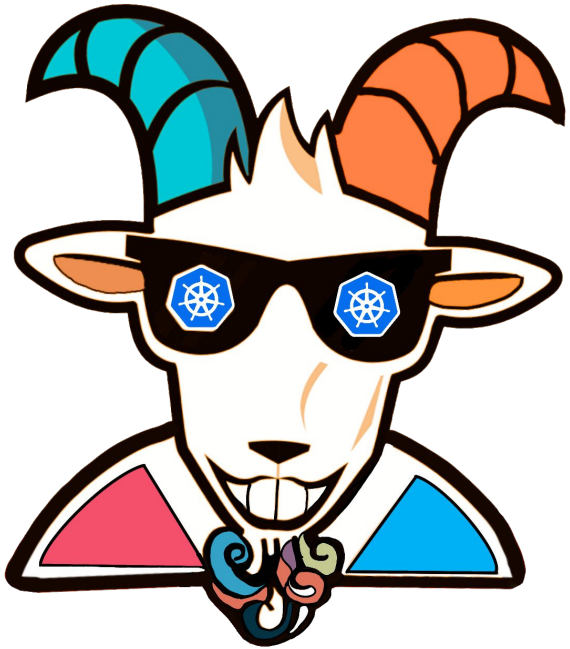
@madhuakula

It’s enough! I love this stuff ❤️

How can I learn, practice, and implement?

Welcome to **Kubernetes Goat** 🎉

What is Kubernetes Goat 🐐



Kubernetes Goat is an interactive Kubernetes security learning playground.

Intentionally vulnerable by design scenarios to showcase the common misconfigurations, real-world vulnerabilities, and security issues in Kubernetes clusters, containers, and cloud native environments.

🚨 Disclaimer

Kubernetes Goat has intentionally created vulnerabilities, applications, and configurations to attack and gain access to your cluster and workloads. Please **DO NOT** run alongside your production environments and infrastructure. So we highly recommend running this in a safe and isolated environment.

Kubernetes Goat is used for educational purposes only, do not test or apply these attacks on any systems without permission. Kubernetes Goat comes with absolutely no warranties, by using it you take full responsibility for all the outcomes.

Can I use Kubernetes Goat for ___? 🤔

Kubernetes Goat is intended for a variety of audiences and end-users.

Which includes hackers, attackers, defenders, developers, architects, DevOps teams, engineers, researchers, products, vendors, and anyone interested in learning about Kubernetes Security.

Below are some of the very high-level categories of audience

💡 **Interested in Kubernetes Security**

💥 **Attackers & Red Teams**

🛡️ **Defenders & Blue Teams**

🔐 **Developers & DevOps Teams**

🏢 **Products & Vendors**

Kubernetes Goat Audience

Attackers & Red Teams

Learn to attack or find security issues, misconfigurations, and real-world hacks within containers, Kubernetes, and cloud native environments. Enumerate, exploit and gain access to the workloads right from your browser.

Defenders & Blue Teams

Understand how attackers think, work and exploit security issues, and apply these learnings to detect and defend them. Also, learn best practices, defenses, and tools to mitigate, and detect in the real world.

Developers & DevOps Teams

Learn the hacks, defenses, and tools. So that you can think like an attacker, and secure your Kubernetes, cloud, and container workloads right from the design, code, and architecture itself to prevent them.

Products & Vendors

Use Kubernetes Goat to showcase the effectiveness of the tools, product, and solution. Also, educate the customers and share your product or tool knowledge in an interactive hands-on way.

Interested in Kubernetes Security

Check out the awesome Kubernetes security resources like popular misconfigurations, hacks, defenses, and tools to gain real-world knowledge. Provide your valuable feedback and suggestions.

Scenarios in Kubernetes Goat 🚀

1. Sensitive keys in codebases
2. DIND (docker-in-docker) exploitation
3. SSRF in the Kubernetes (K8S) world
4. Container escape to the host system
5. Docker CIS benchmarks analysis
6. Kubernetes CIS benchmarks analysis
7. Attacking private registry
8. NodePort exposed services
9. Helm v2 tiller to PwN the cluster - [Deprecated]
10. Analyzing crypto miner container
12. Gaining environment information
13. DoS the memory/cpu resources
14. Hacker Container preview
15. Hidden in layers
16. RBAC Least Privileges Misconfiguration
17. KubeAudit - Audit Kubernetes Clusters
18. Sysdig Falco - Runtime Security Monitoring & Detection
19. Popeye - A Kubernetes Cluster Sanitizer
20. Secure network boundaries using NSP

Scenarios going to be updated with defenders, developers, tools & vendor sections for reach scenario 🥳

15+ more scenarios releasing soon... ❤️

@madhuakula

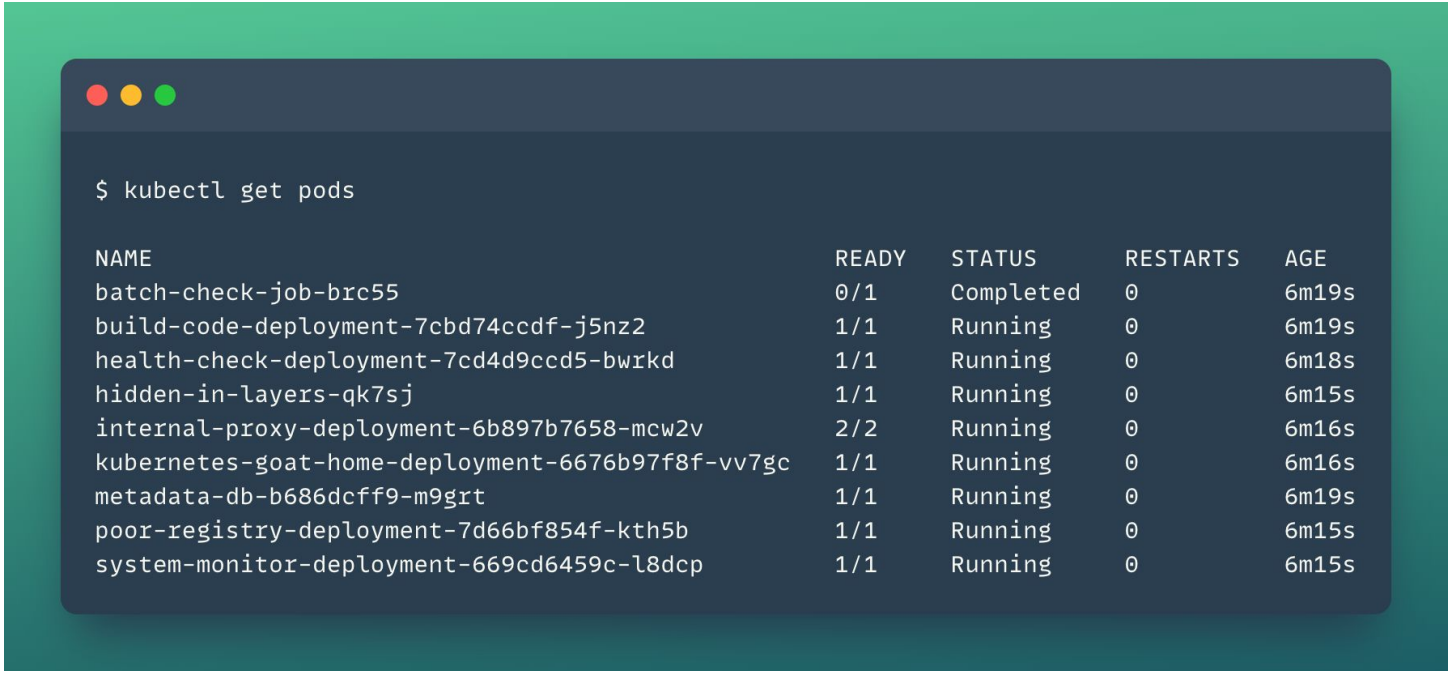
⎈ Setting up in your Kubernetes Cluster

- Make sure you have Kubernetes cluster with cluster-admin privileges. Also **kubectl** and **helm** installed in your system before running the following commands to setup the Kubernetes Goat

```
$ git clone https://github.com/madhuakula/kubernetes-goat.git
$ cd kubernetes-goat
$ bash setup-kubernetes-goat.sh
$ bash access-kubernetes-goat.sh
```

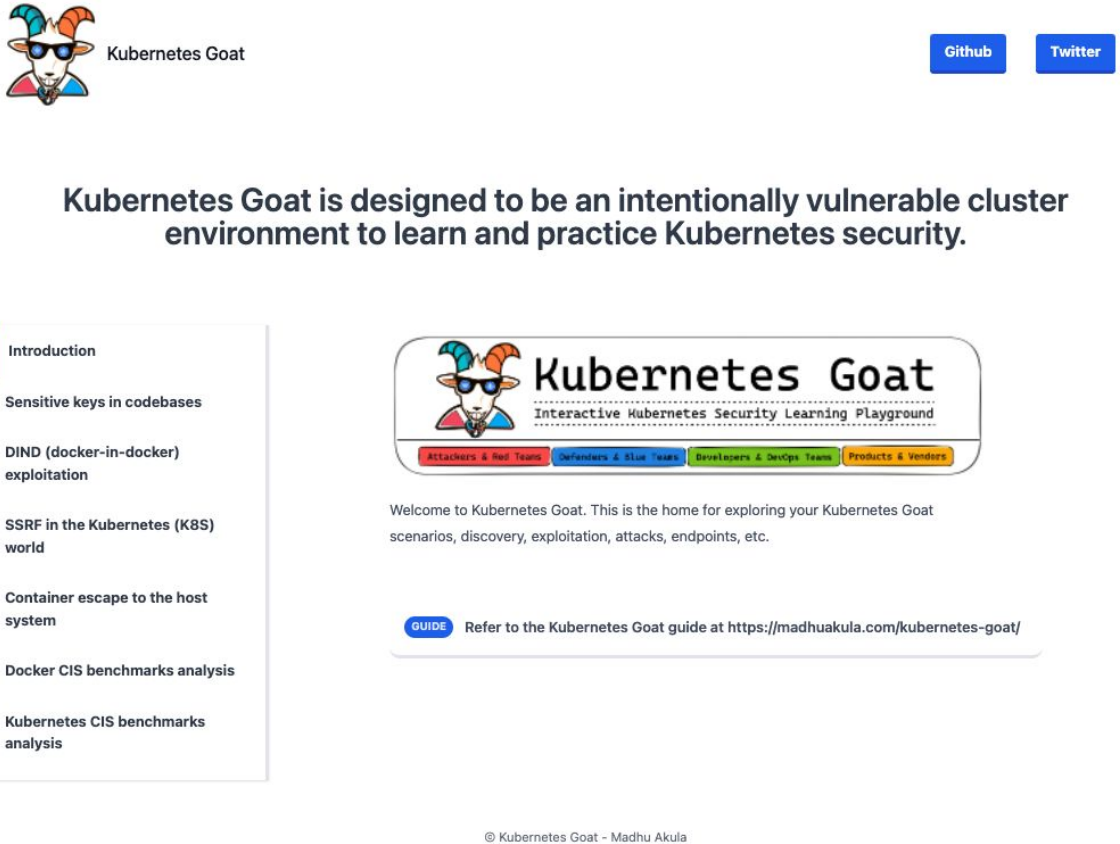
- Now you can access the Kubernetes Goat by navigating to **http://127.0.0.1:1234**

⚡ Get Started with Kubernetes Goat 🐐



```
$ kubectl get pods

NAME                                          READY   STATUS      RESTARTS   AGE
batch-check-job-brc55                         0/1     Completed   0          6m19s
build-code-deployment-7cbd74ccdf-j5nz2        1/1     Running     0          6m19s
health-check-deployment-7cd4d9ccd5-bwrkd      1/1     Running     0          6m18s
hidden-in-layers-qk7sj                        1/1     Running     0          6m15s
internal-proxy-deployment-6b897b7658-mcw2v    2/2     Running     0          6m16s
kubernetes-goat-home-deployment-6676b97f8f-vv7gc  1/1  Running     0          6m16s
metadata-db-b686dcff9-m9grt                   1/1     Running     0          6m19s
poor-registry-deployment-7d66bf854f-kth5b     1/1     Running     0          6m15s
system-monitor-deployment-669cd6459c-l8dcp    1/1     Running     0          6m15s
```

⚡ Get Started with Kubernetes Goat



@madhuakula

⚡ Get Started with Kubernetes Goat

https://madhuakula.com/kubernetes-goat

  Demo Time  

Key Takeaways!

BBQ en borrel 🥳

Key Takeaways!

✅ Security is everyone's responsibility (Dev, Ops, Security, Management, etc.)

⚠️ Threat model your architecture and identify risks/threats

🙌 Follow and apply secure defaults

📚Know what you have (Inventory of assets)

🧱Adopt zero trust model (Zoning, Containment & Segmentation)

🎯Apply security at each layer (Defense in depth strategy)

🚨Follow least privilege principle

👮AuthN & AuthZ

🔐Encryption at REST & TRANSIT

🛡️Proactive monitoring & Active defense

🔁Continuously analyse and apply feedback loops

👉 Crawl 🐢, Walk 🚶, Run 🏃, Fly 🛫

🔖 Resources & References

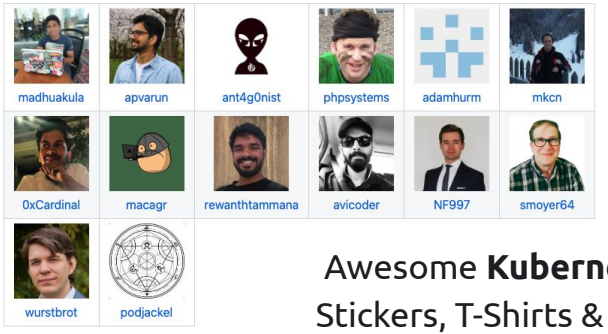
- 👉 https://madhuakula.com/content
- 👉 https://kubernetes.io
- 👉 https://github.com/madhuakula/hacker-container
- 👉 https://kubernetes-security.info
- 👉 https://github.com/kelseyhightower/kubernetes-the-hard-way
- 👉 https://container.training
- 👉 https://github.com/freach/kubernetes-security-best-practice
- 👉 https://kubernetes.io/docs/tasks/administer-cluster/securing-a-cluster
- 👉 https://github.com/docker/labs
- 👉 https://labs.play-with-docker.com
- 👉 https://labs.play-with-k8s.com
- 👉 https://landscape.cncf.io
- 👉 https://github.com/cncf/sig-security/tree/master/security-whitepaper
- 👉 https://tools.tldr.run
- 👉 https://github.com/magnologan/awesome-k8s-security
- 👉 https://github.com/ramitsurana/awesome-kubernetes
- 👉 https://github.com/tomhuang12/awesome-k8s-resources
- 👉 CNCF Slack
- 👉 Kubernetes Slack
- 👉 https://k8s.af
- 👉 https://contained.af
- 👉 https://github.com/genuinetools/img
- 👉 https://github.com/genuinetools/bane
- 👉 https://github.com/genuinetools/amicontained
- 👉 CNCF YouTube Playlists for the KubeCon

Spread the ❤️ Kubernetes Goat

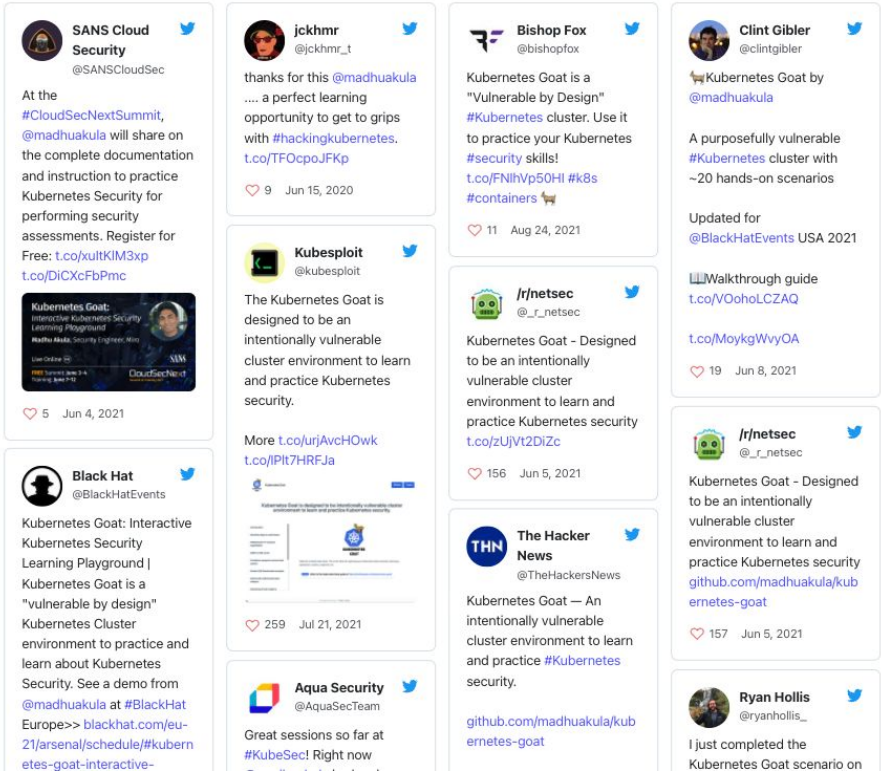
- 🙌 Give it a try
- 🚀 Contribute ideas & suggestions
- 🤝 Work with the project & improve
- 🙏 Share your valuable feedback
- 🌟 Star in our GitHub
- 🎉 Spread the word in social media

✨ **Acknowledgements**

Thanks go to these wonderful people 🎉



Awesome **Kubernetes Goat** Stickers, T-Shirts & Some cool goodies on the way 🥳



https://madhuakula.com/kubernetes-goat/docs/wall-of-love

Dank je wel 🙏

Want to learn more, have some idea, or just wanted to say 👋

@madhuakula

https://madhuakula.com