Kubernetes Attack Simulation:
The Definitive Guide

Adversary Village, DEF CON 32

W/TH secure

whoami

Leo Tsaousis

Senior Security Consultant

Attack Path Mapping Lead @ WithSecure

Purple Teams / Threat Simulation

Presented at ROOTCON, BSides

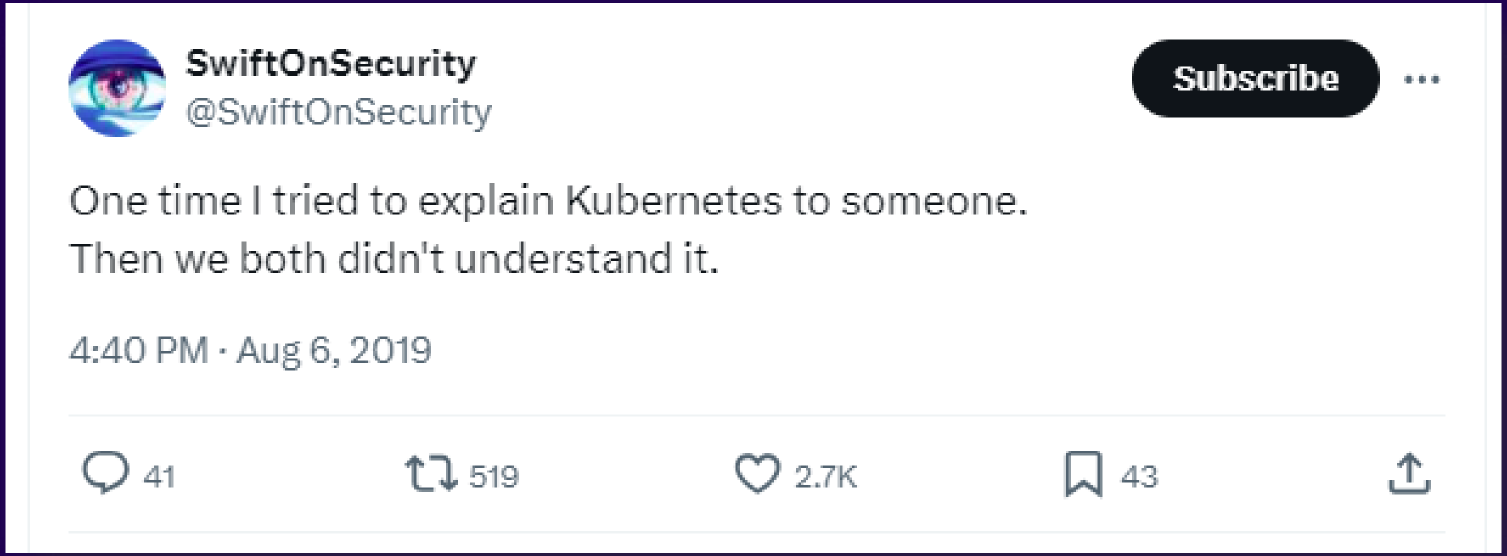
> We need to measure our Attack Detection capability for this {Windows, Linux, On-prem, Cloud, Kubernetes} environment

> We need to measure our Attack Detection capability for this {Windows, Linux, On-prem, Cloud, Kubernetes} environment

Agenda

1. Kubernetes Introduction
2. Threat Modelling
3. K8S Attack Simulation
4. K8S Attack Detection
5. Demo

Intro to Kubernetes

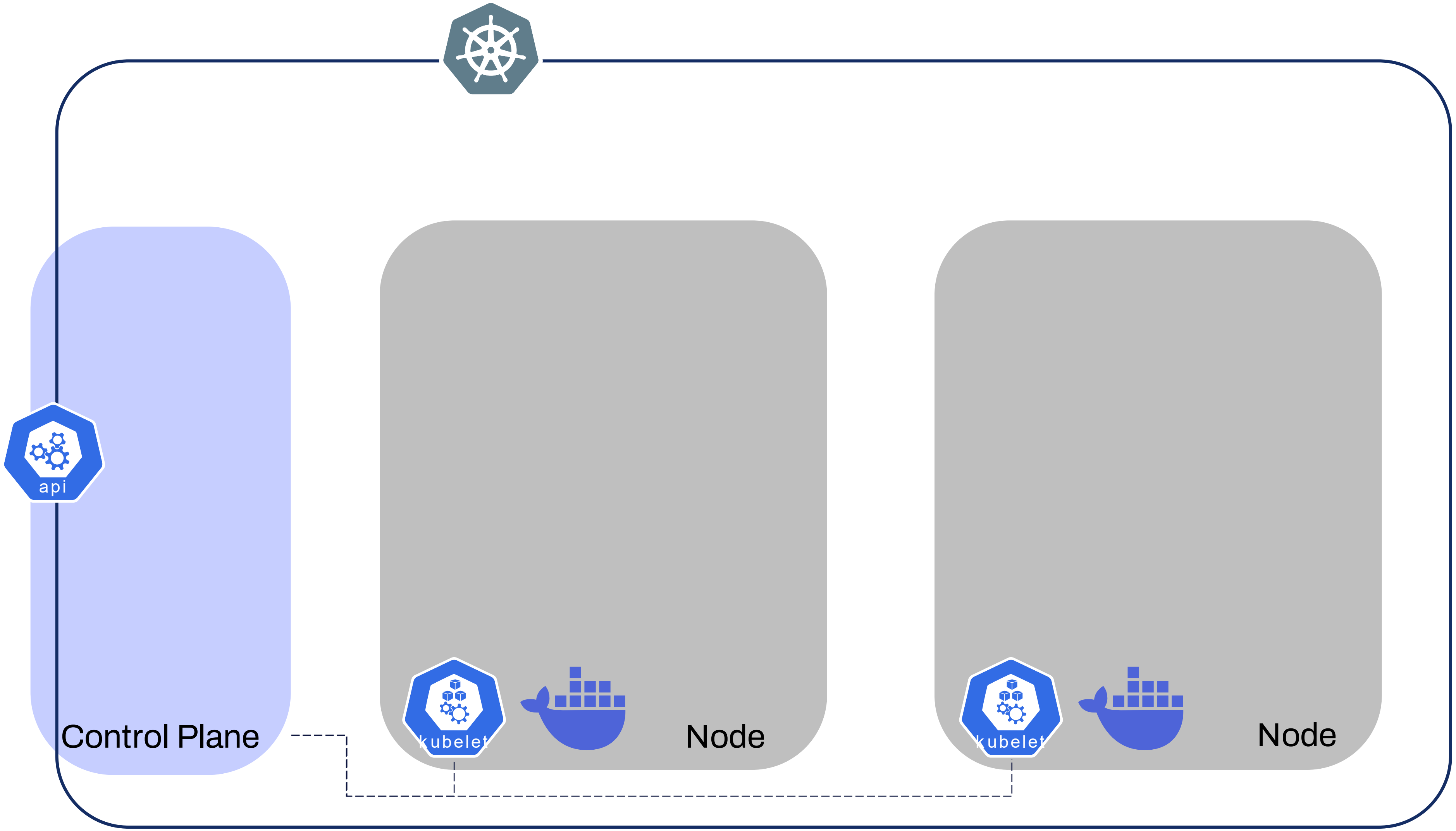


SwiftOnSecurity
@SwiftOnSecurity

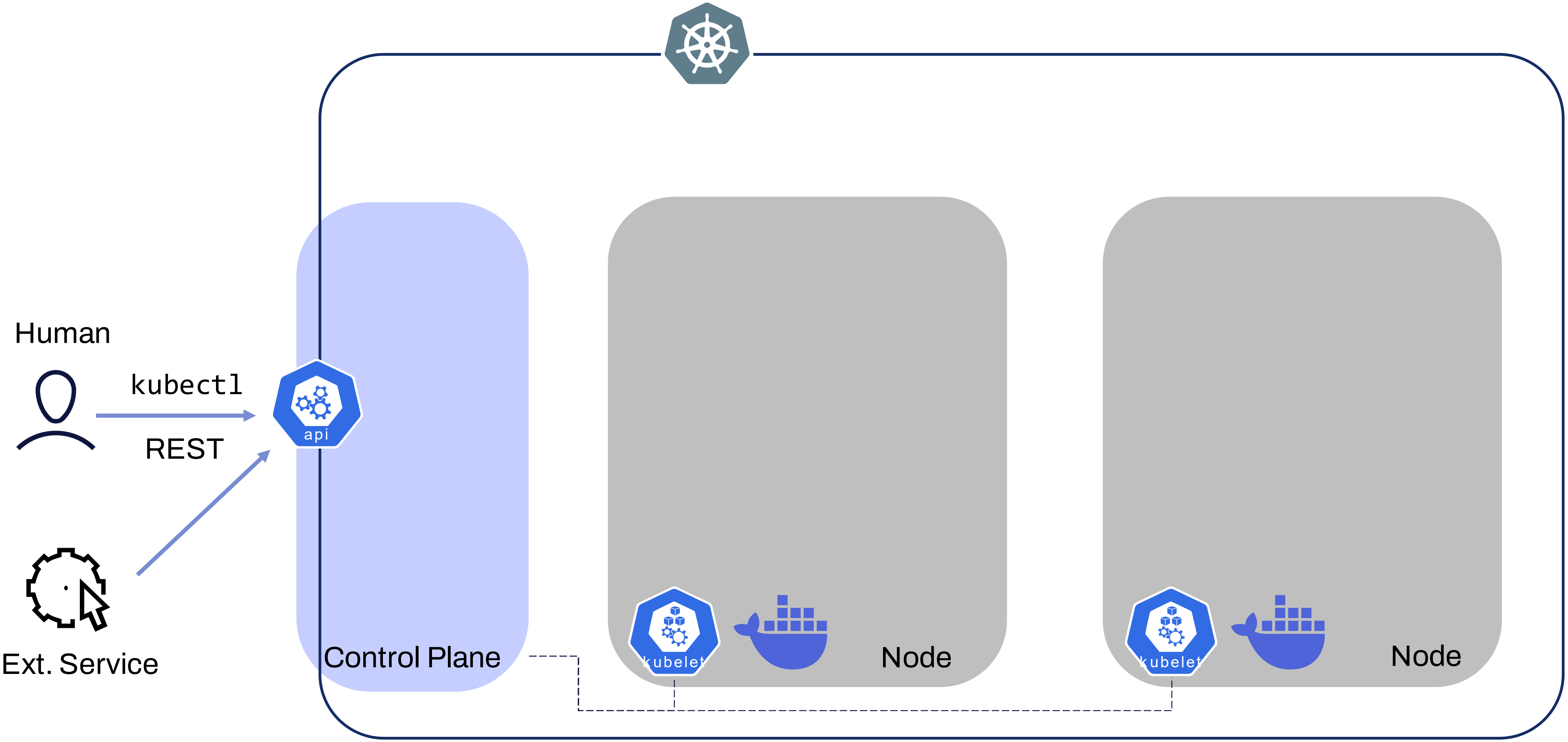
One time I tried to explain Kubernetes to someone.
Then we both didn't understand it.

4:40 PM · Aug 6, 2019

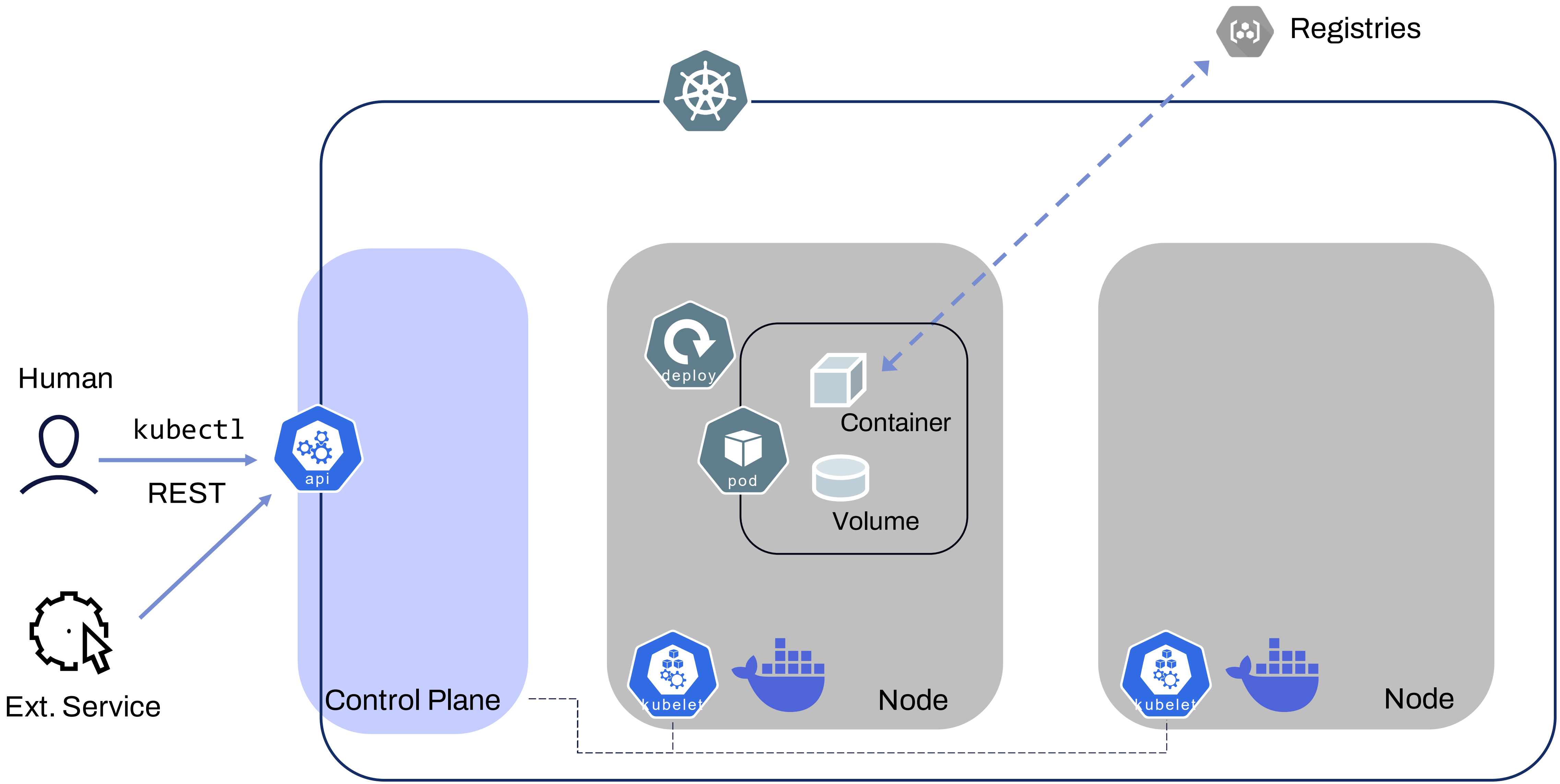
A View to a Cluster



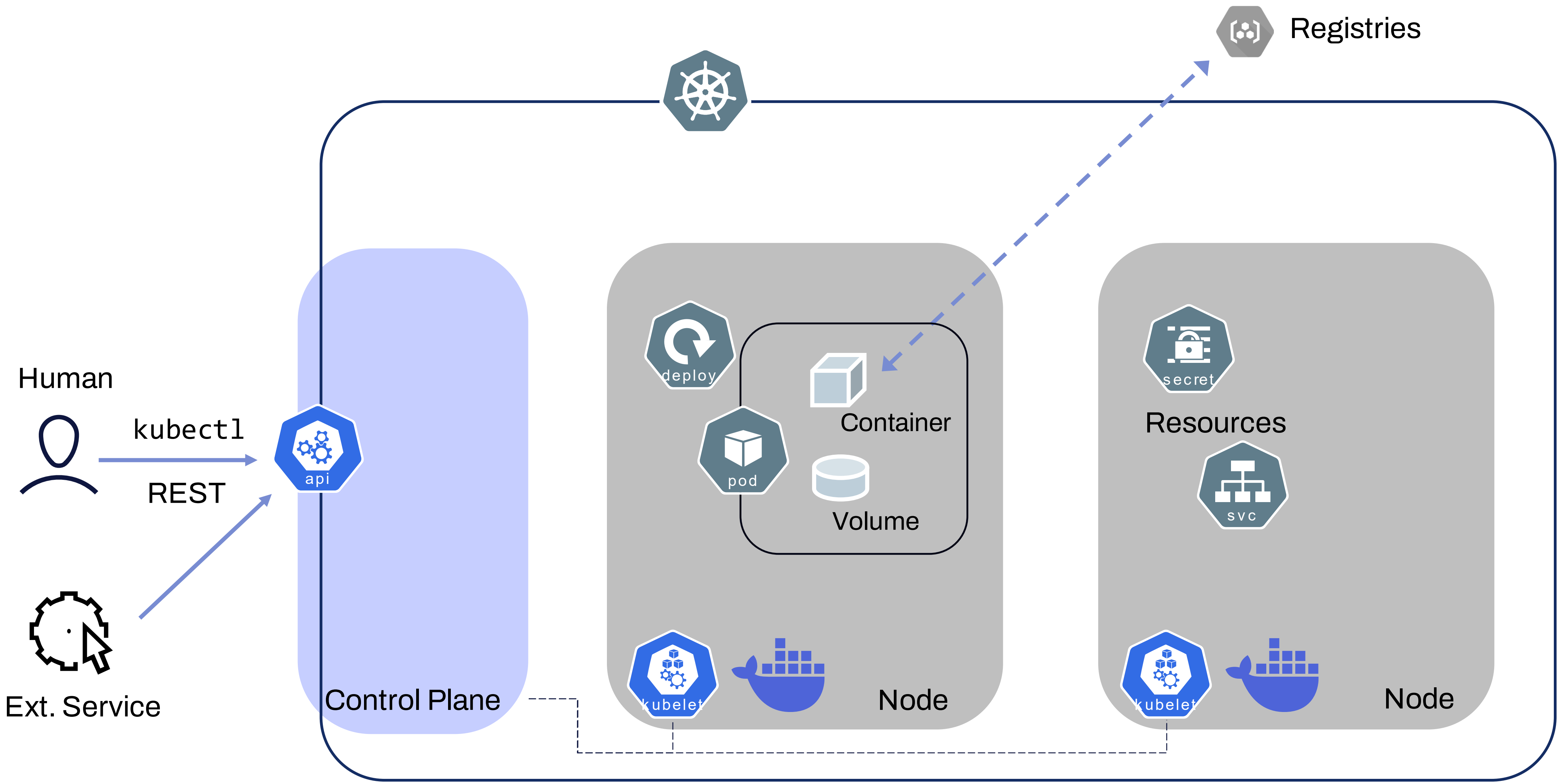
A View to a Cluster



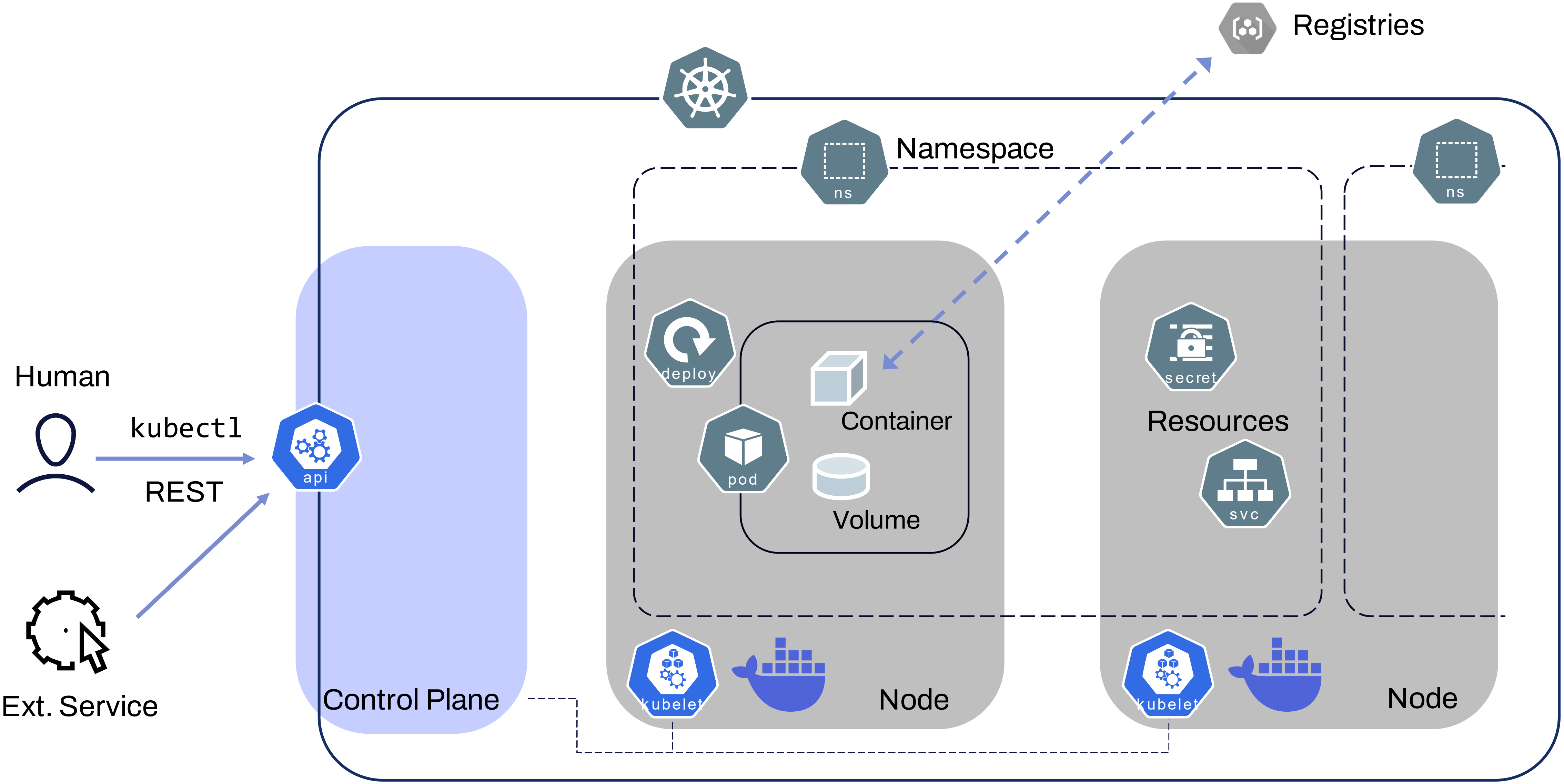
A View to a Cluster



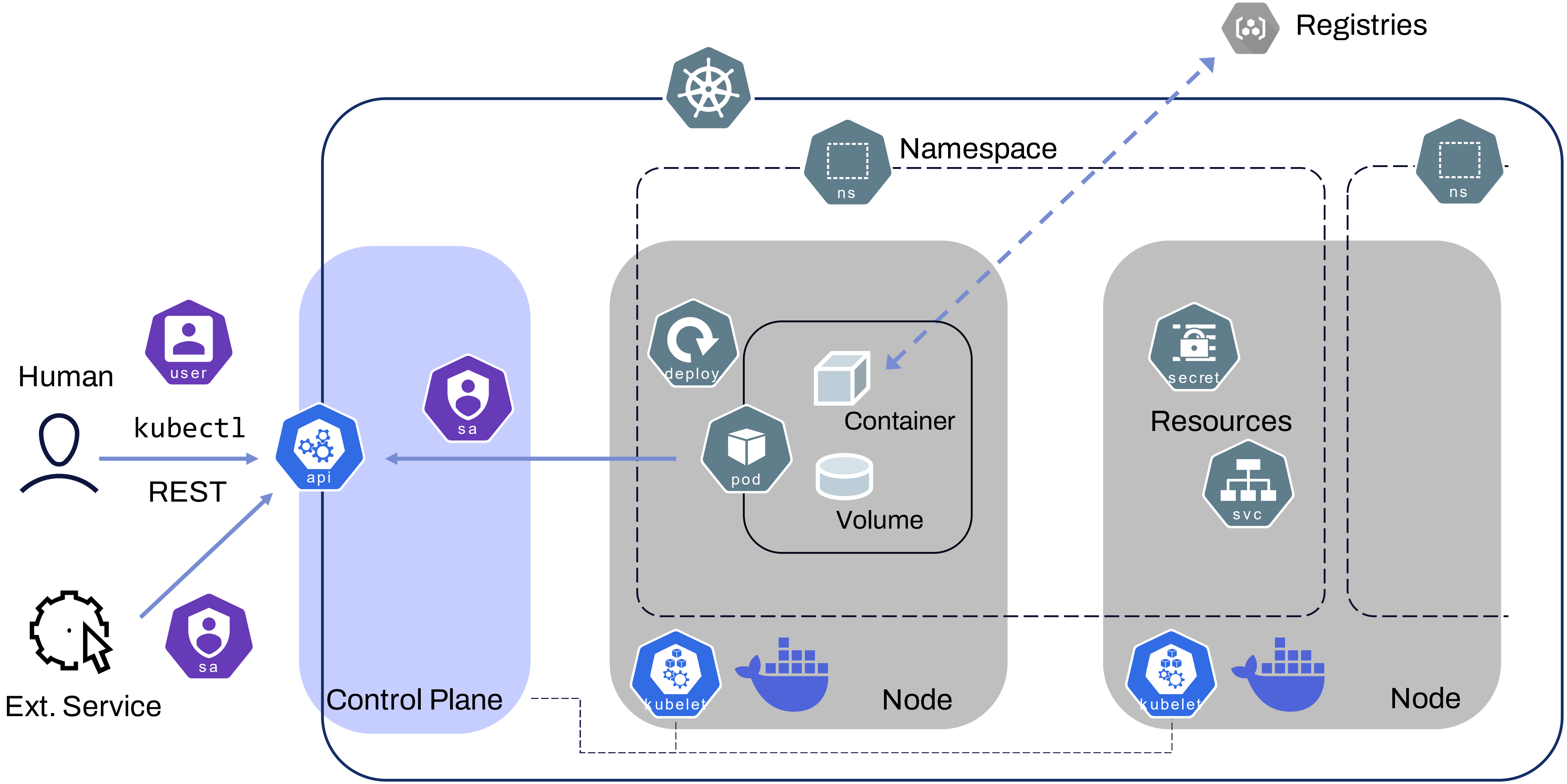
A View to a Cluster



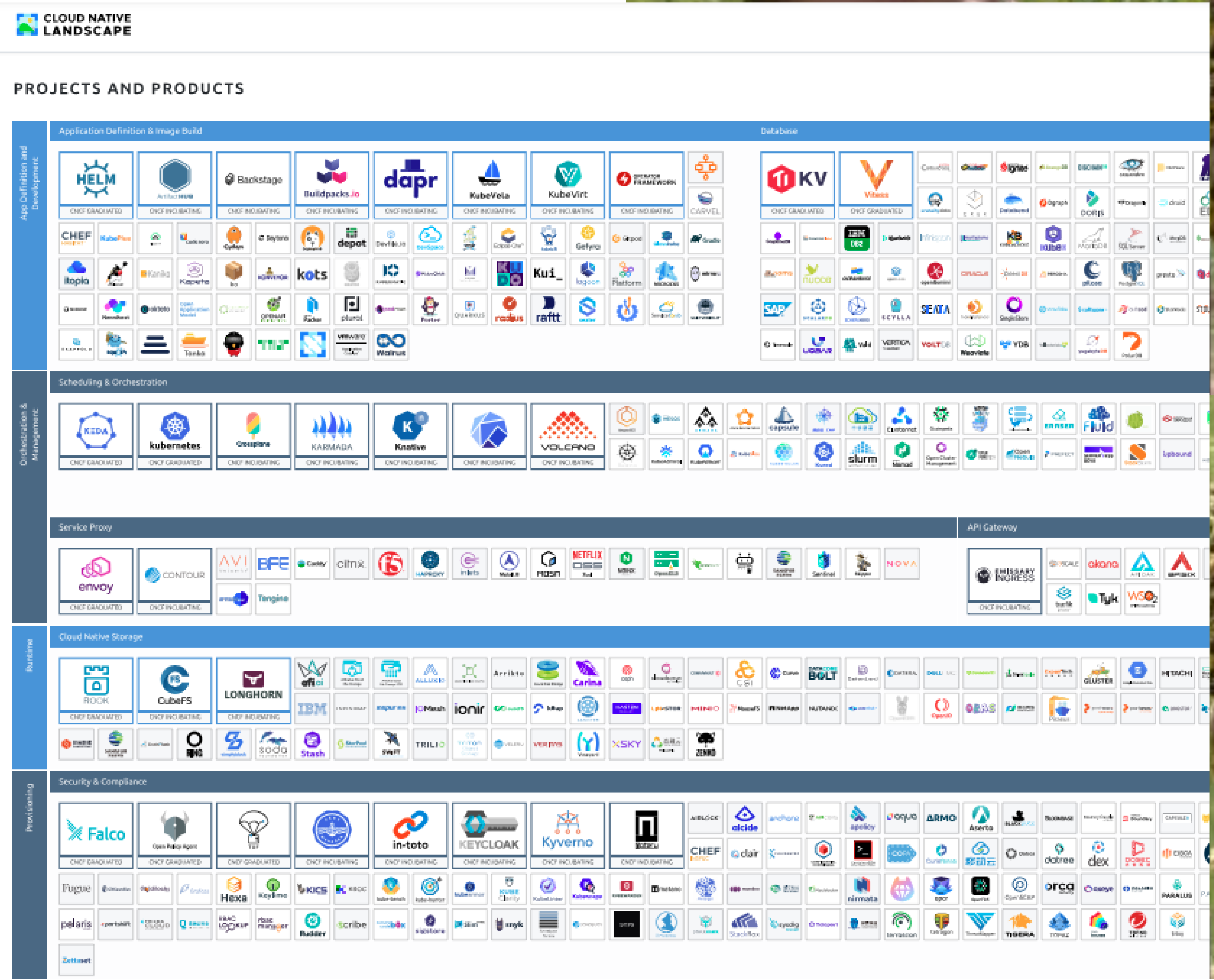
A View to a Cluster



A View to a Cluster



Simple enough



CLOUD NATIVE LANDSCAPE

PROJECTS AND PRODUCTS

App Definition and Development

Application Definition & Image Build

Database

Orchestration & Management

Scheduling & Orchestration

Service Proxy

API Gateway

Runtime

Cloud Native Storage

Provisioning

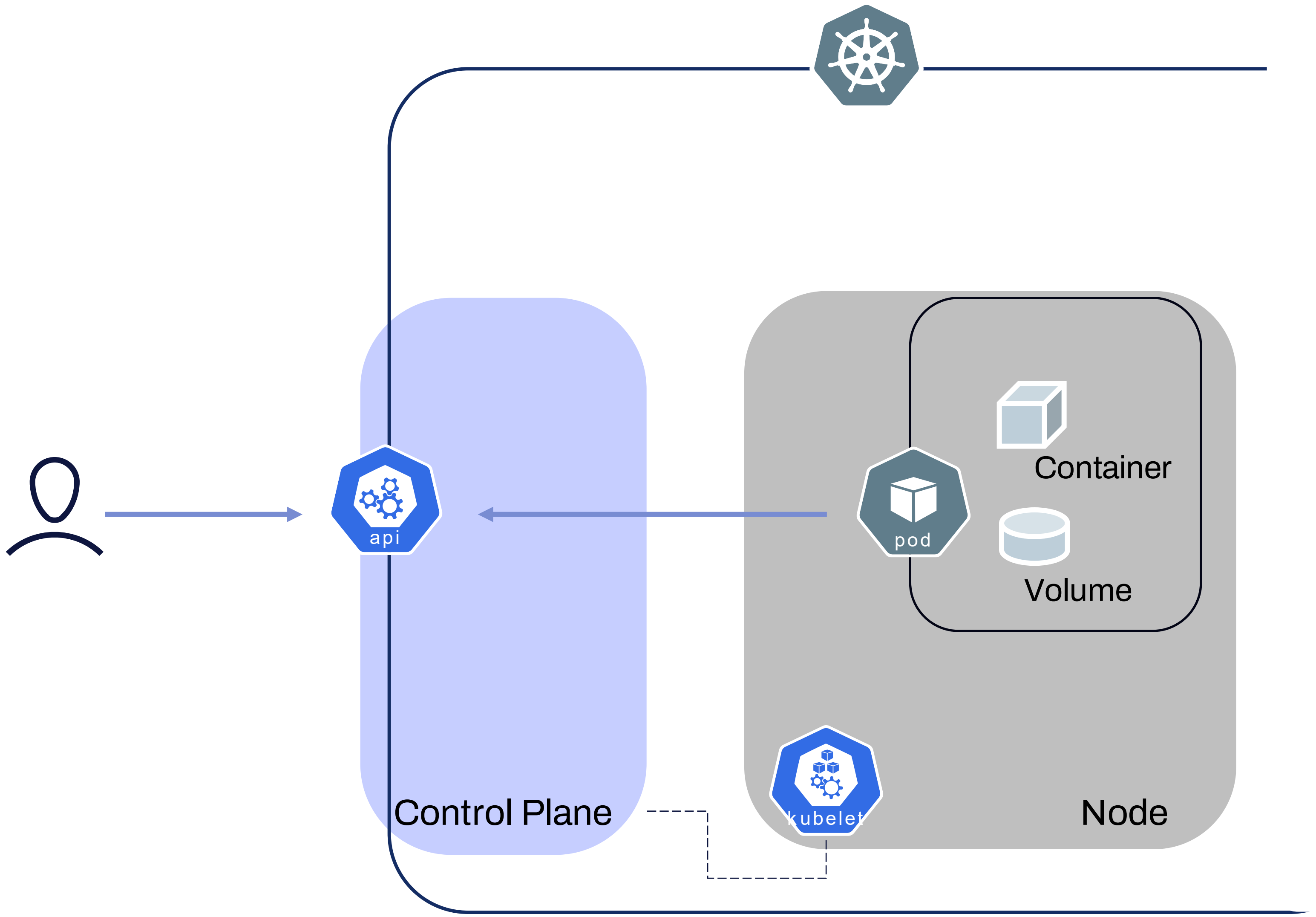
Security & Compliance

Threat Modelling

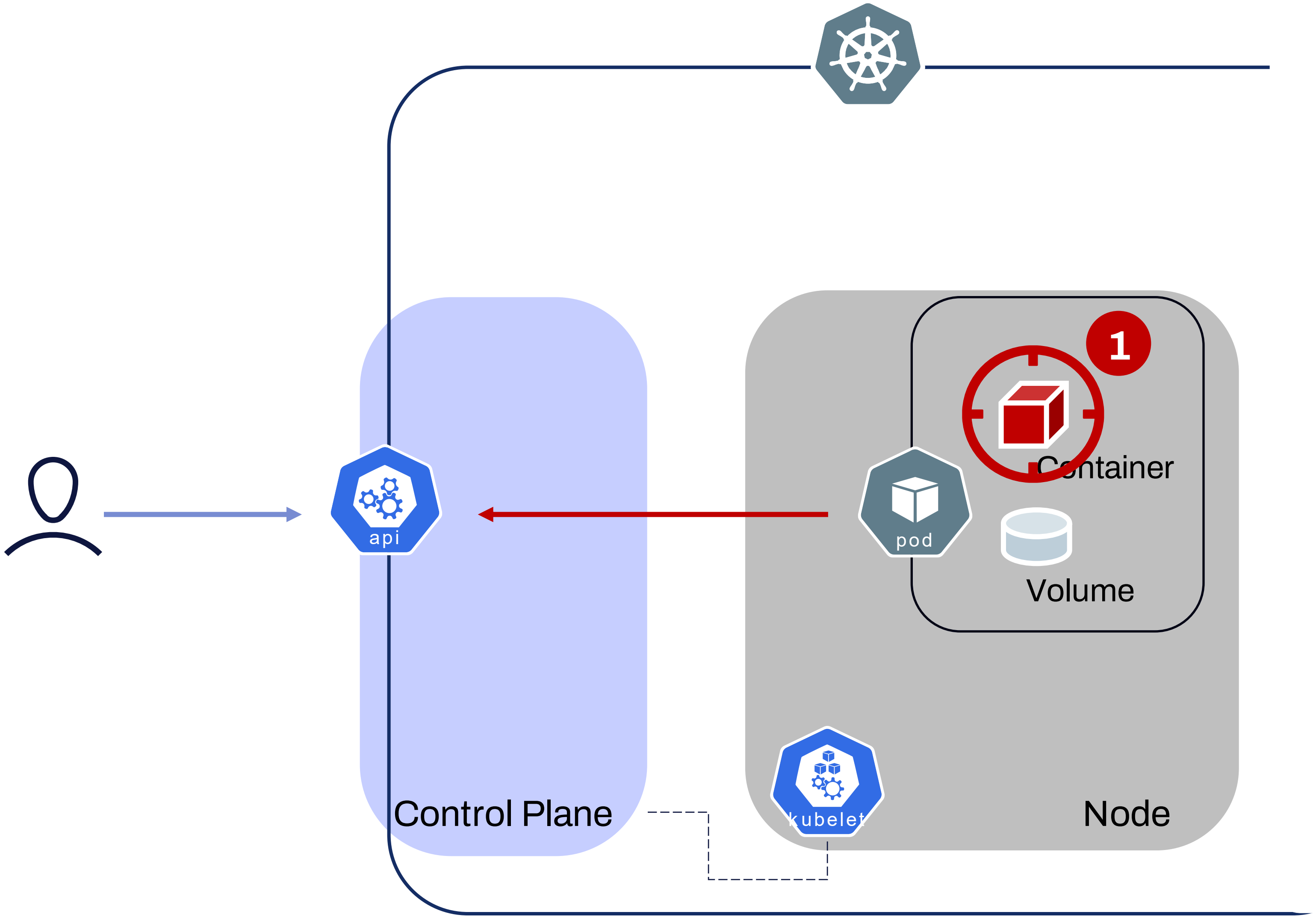
Kubernetes
Attack Simulation

Kubernetes
Attack Detection

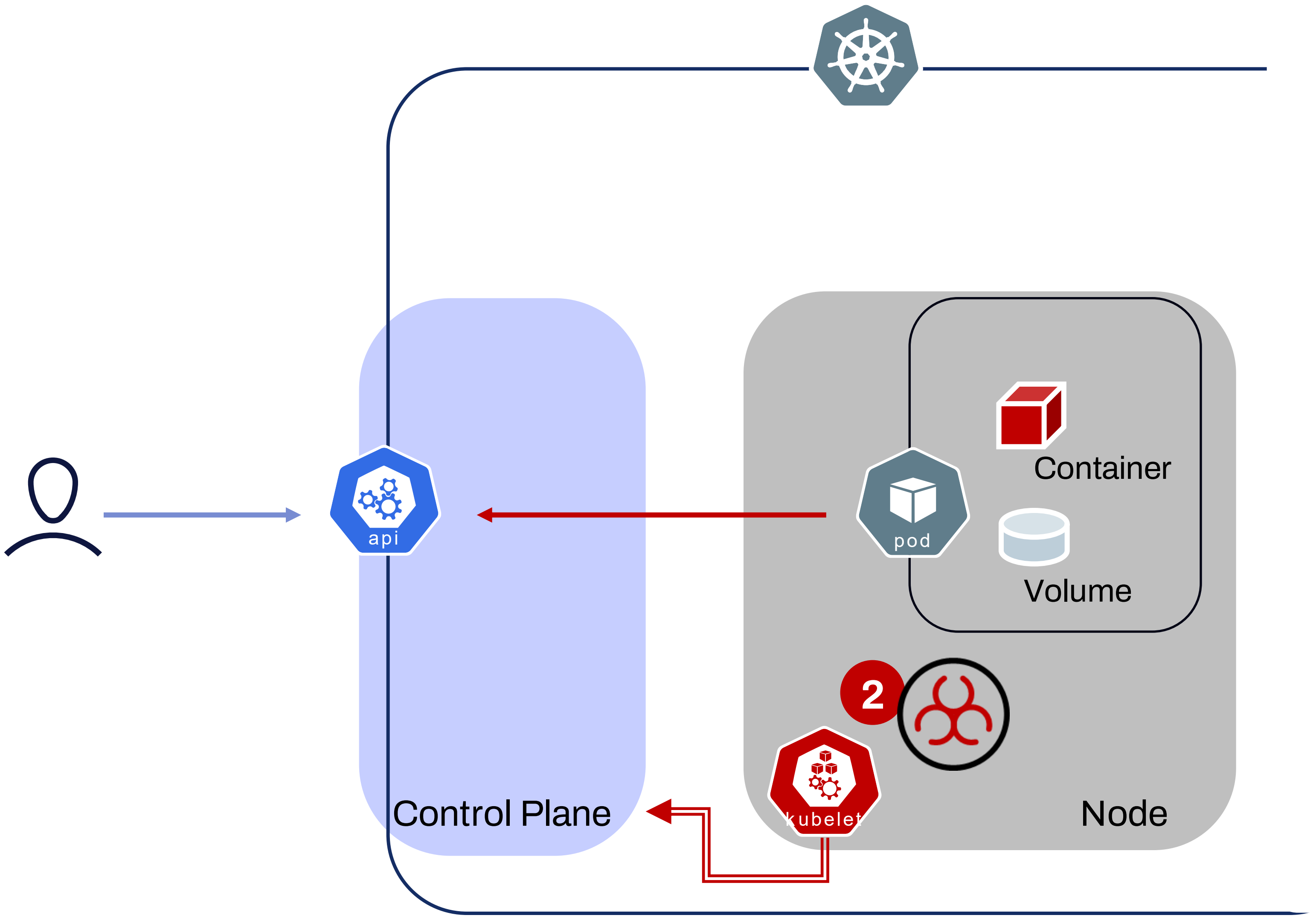
Attack Surfaces



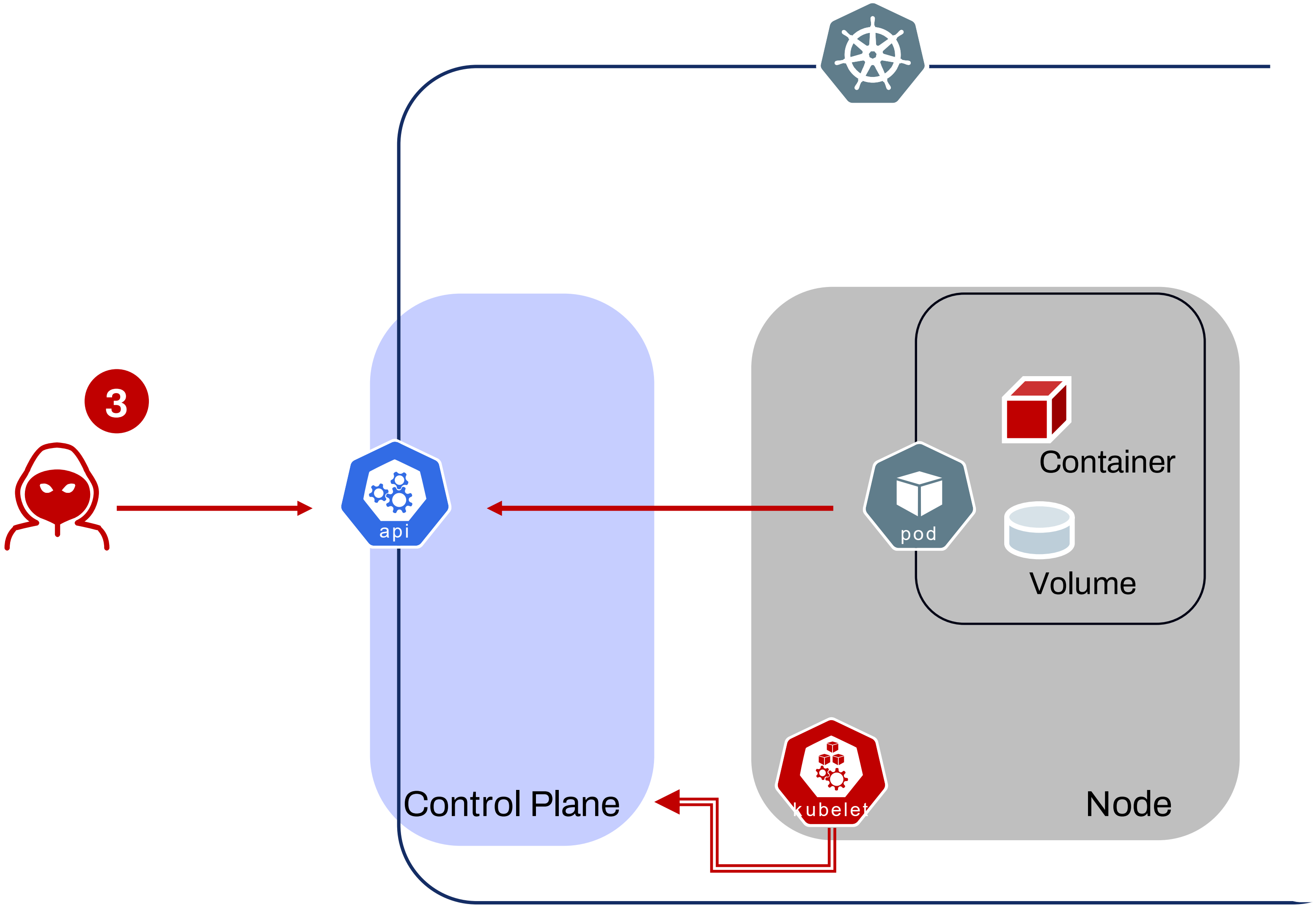
Attack Surfaces



Attack Surfaces



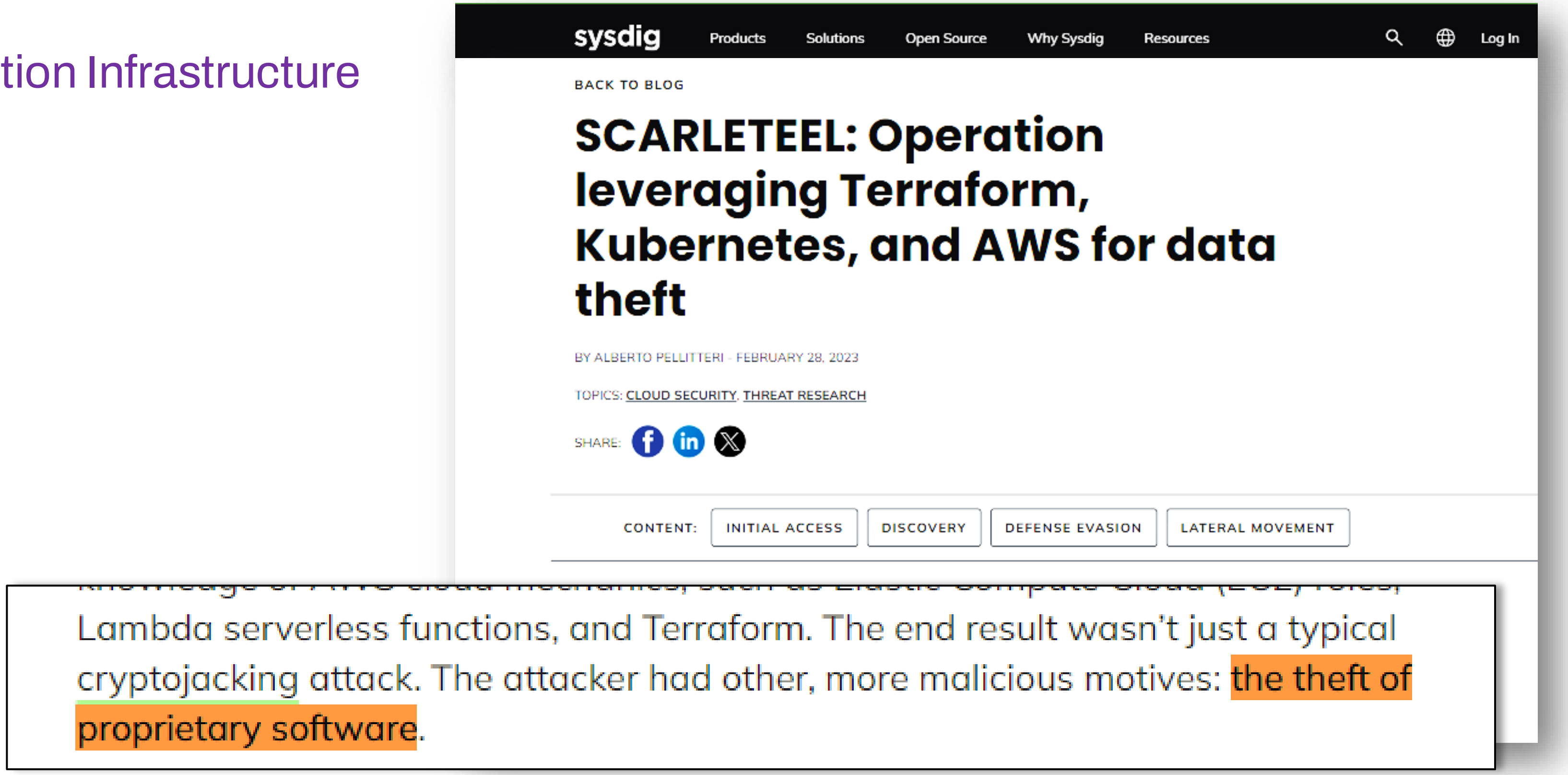
Attack Surfaces



Attacker Incentives

Attacker Incentives

- Orchestration Platform = Application Infrastructure



SCARLETEEL: Operation leveraging Terraform, Kubernetes, and AWS for data theft

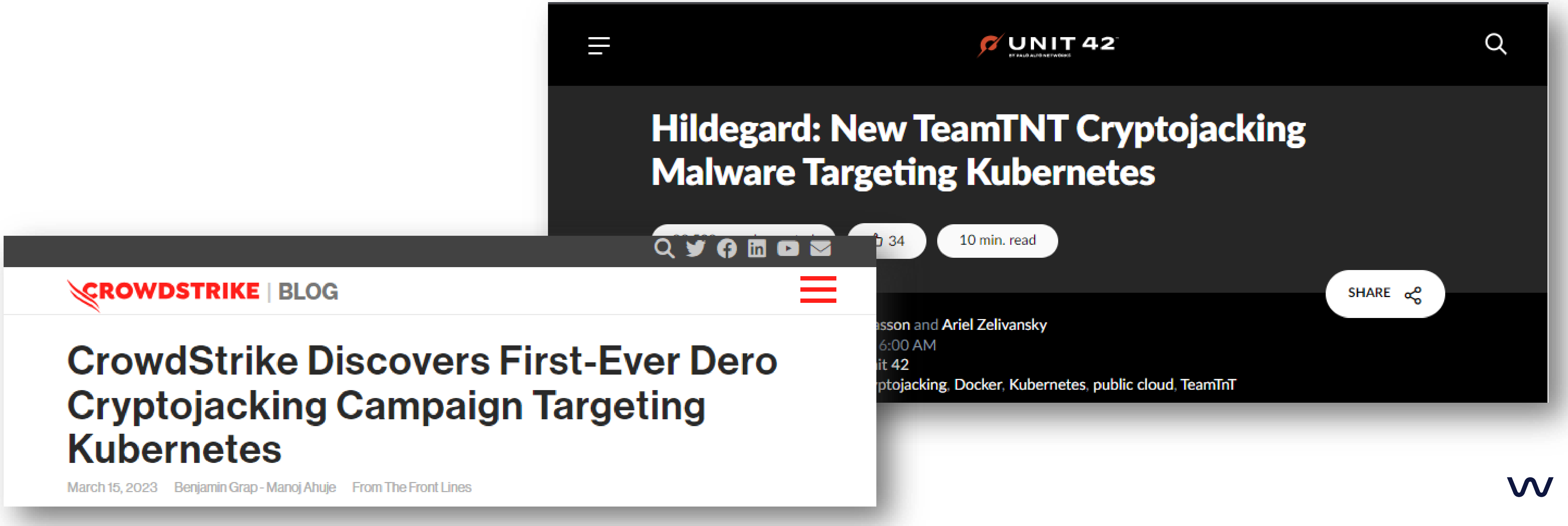
BY ALBERTO PELLITTERI · FEBRUARY 28, 2023

TOPICS: CLOUD SECURITY, THREAT RESEARCH

knowledge of AWS cloud mechanics, such as Elastic Compute Cloud (EC2) roles, Lambda serverless functions, and Terraform. The end result wasn't just a typical cryptojacking attack. The attacker had other, more malicious motives: the theft of proprietary software.

Attacker Incentives

- Orchestration Platform = Application Infrastructure
- Compute Resources = Hardware for Mining



UNIT 42

Hildegard: New TeamTNT Cryptojacking Malware Targeting Kubernetes

34

10 min. read

SHARE

asson and Ariel Zelivansky
6:00 AM
it 42
ptojacking, Docker, Kubernetes, public cloud, TeamTnT

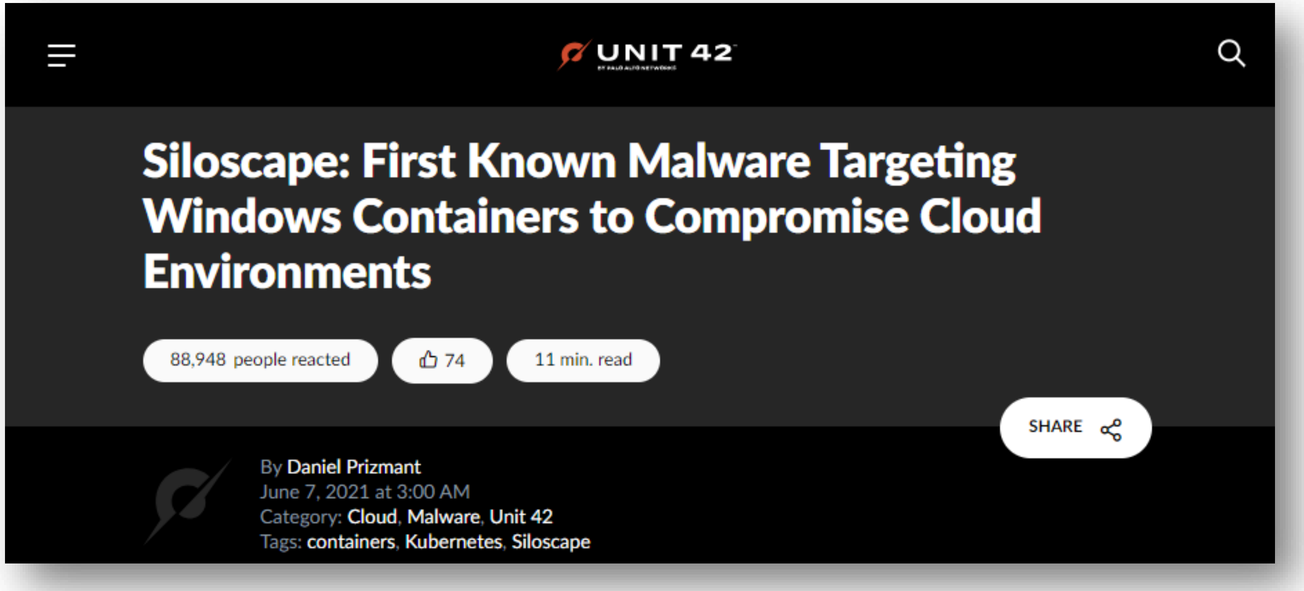
CROWDSTRIKE | BLOG

CrowdStrike Discovers First-Ever Dero Cryptojacking Campaign Targeting Kubernetes

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

Attacker Incentives

- Orchestration Platform = Application Infrastructure
- Compute Resources = Hardware for Mining
- Entrypoint to Cloud

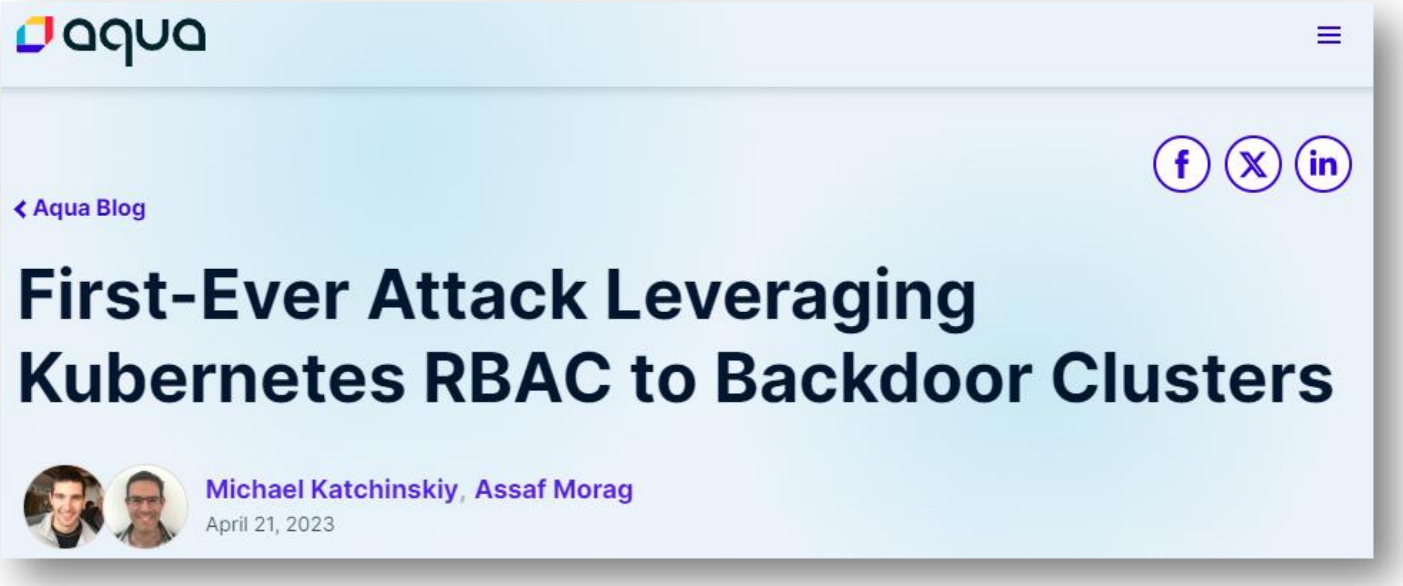


Siloscape: First Known Malware Targeting Windows Containers to Compromise Cloud Environments

By Daniel Prizmant
June 7, 2021 at 3:00 AM
Category: Cloud, Malware, Unit 42
Tags: containers, Kubernetes, Siloscape

Attacker Incentives

- Orchestration Platform = Application Infrastructure
- Compute Resources = Hardware for Mining
- Entrypoint to Cloud
- Persistent access for Espionage



< Aqua Blog

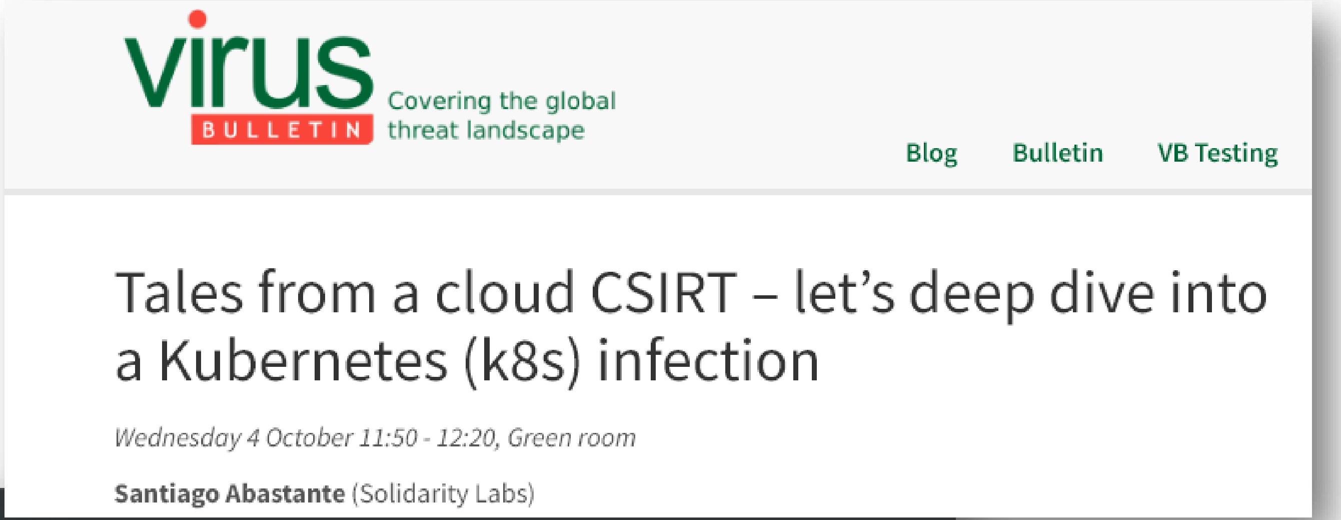
First-Ever Attack Leveraging Kubernetes RBAC to Backdoor Clusters

Michael Katchinskiy, Assaf Morag

April 21, 2023

Attacker Incentives

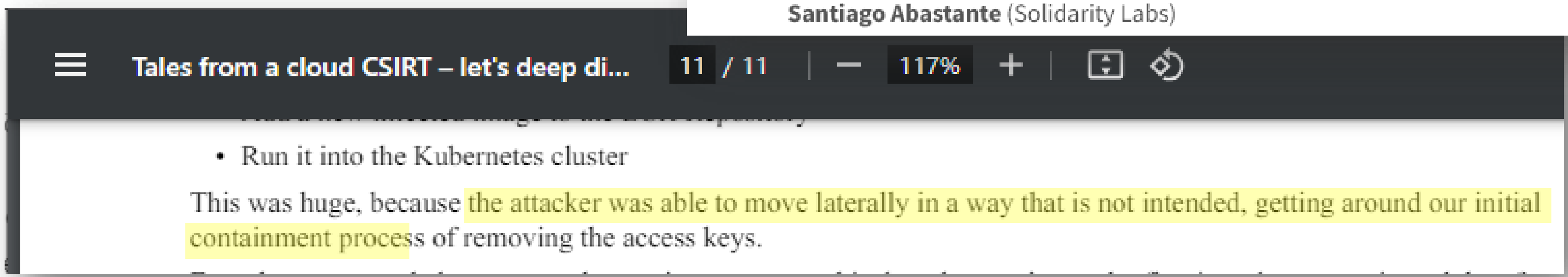
- Orchestration Platform = Application Infrastructure
- Compute Resources = Hardware for Mining
- Entrypoint to Cloud
- Persistent access for Espionage
- Hiding place to Evade Containment



Tales from a cloud CSIRT – let's deep dive into a Kubernetes (k8s) infection

Wednesday 4 October 11:50 - 12:20, Green room

Santiago Abastante (Solidarity Labs)



- Run it into the Kubernetes cluster

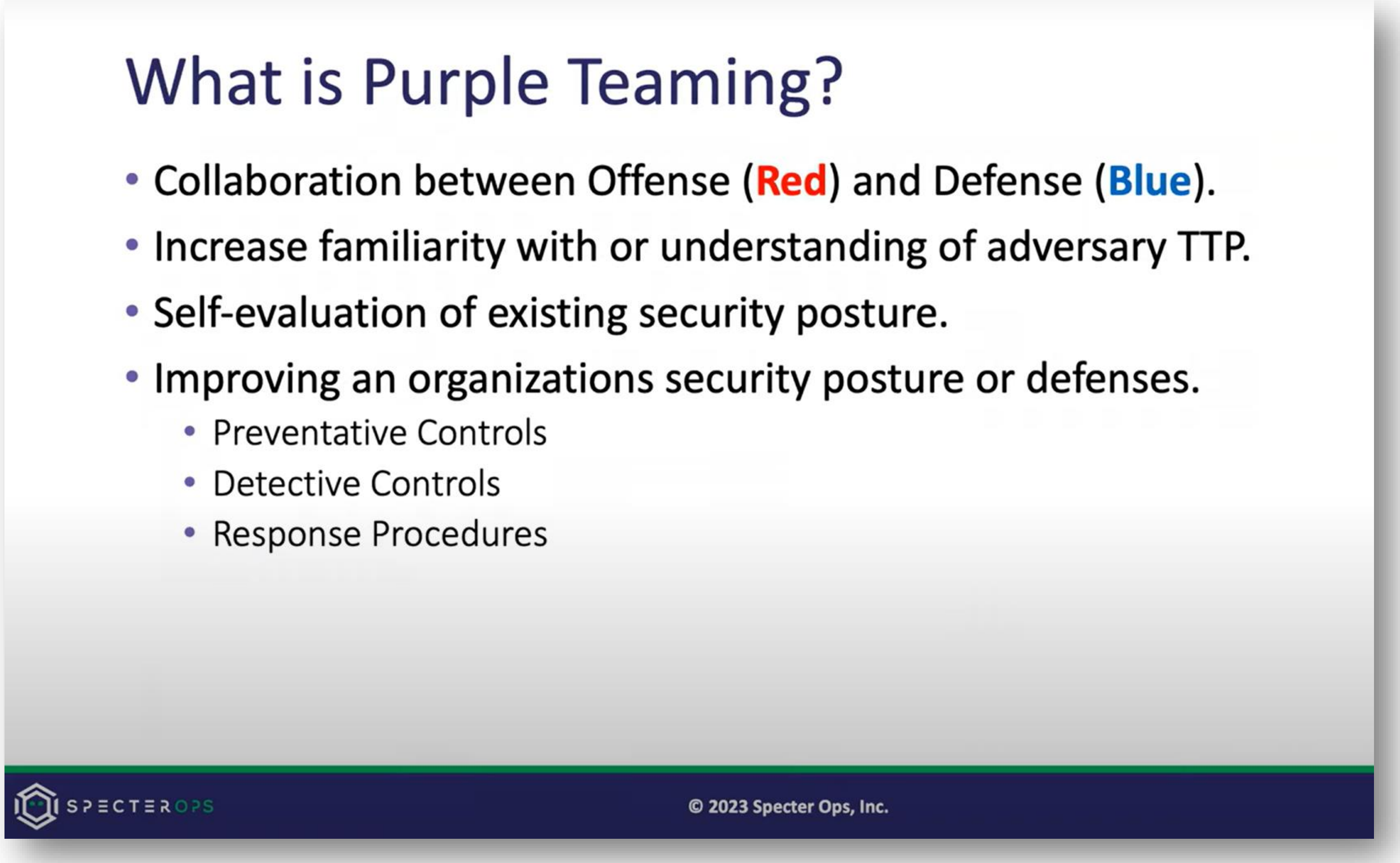
This was huge, because the attacker was able to move laterally in a way that is not intended, getting around our initial containment process of removing the access keys.

Threat Modelling

Kubernetes Attack Simulation

Kubernetes Attack Detection

Demo



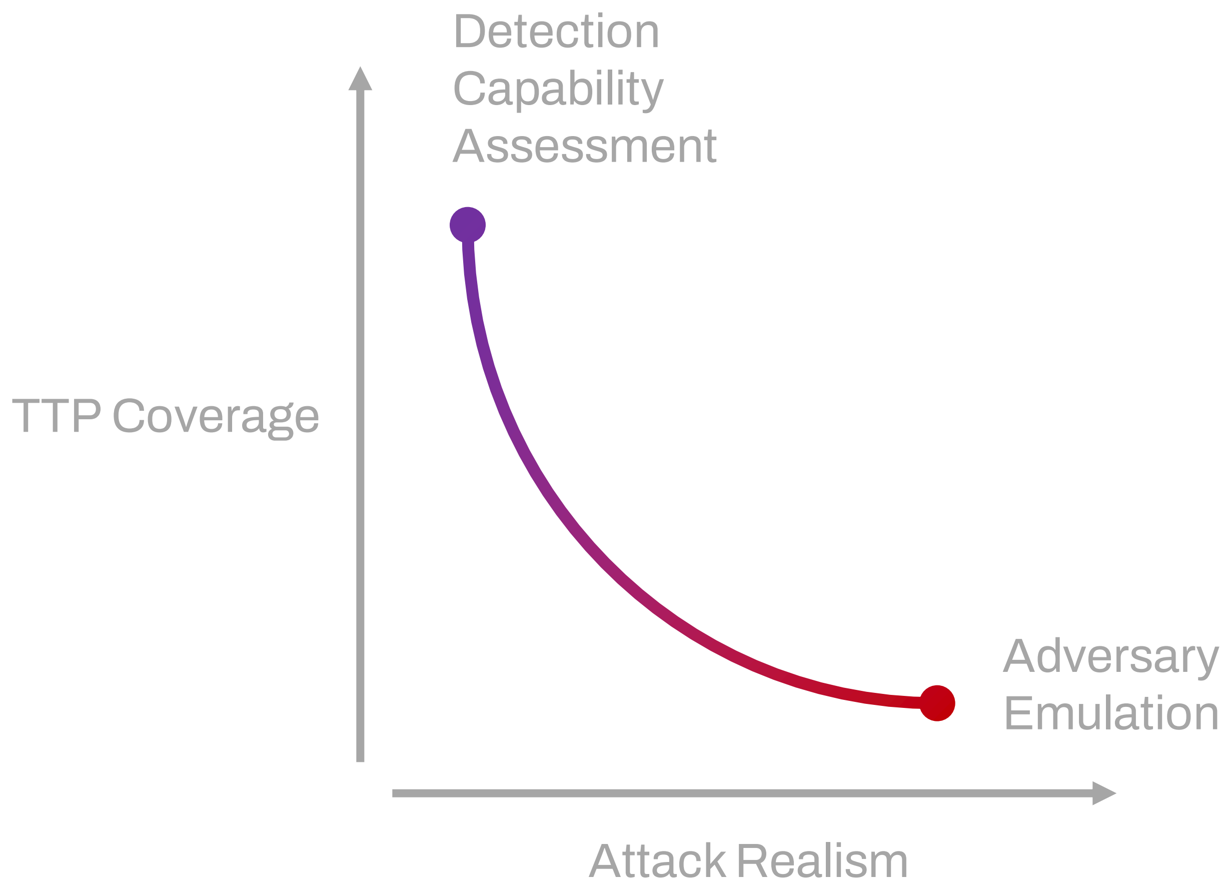
What is Purple Teaming?

- Collaboration between Offense (**Red**) and Defense (**Blue**).
- Increase familiarity with or understanding of adversary TTP.
- Self-evaluation of existing security posture.
- Improving an organizations security posture or defenses.
  - Preventative Controls
  - Detective Controls
  - Response Procedures

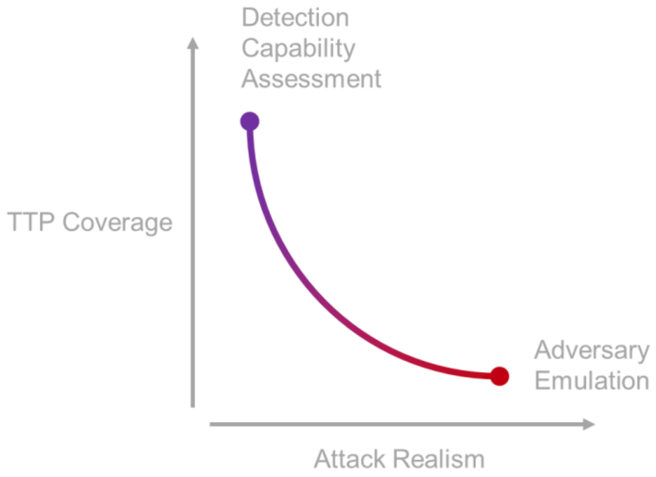
© 2023 Specter Ops, Inc.

Specter Ops | Purple Teaming (Black Hat USA 2023 Booth Talk)

Shades of Purple



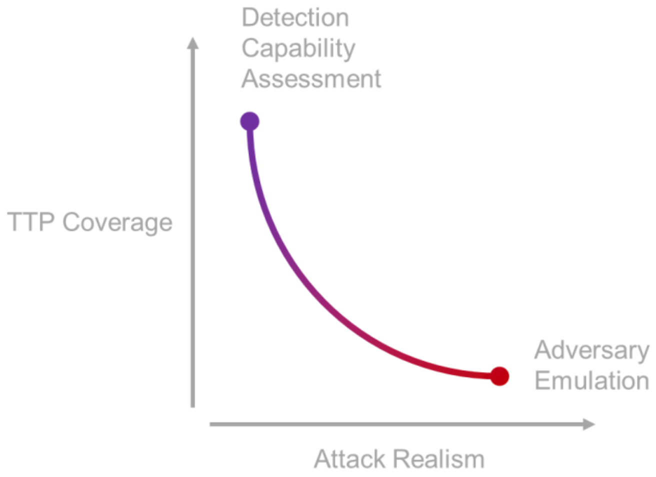
Planning the Exercise



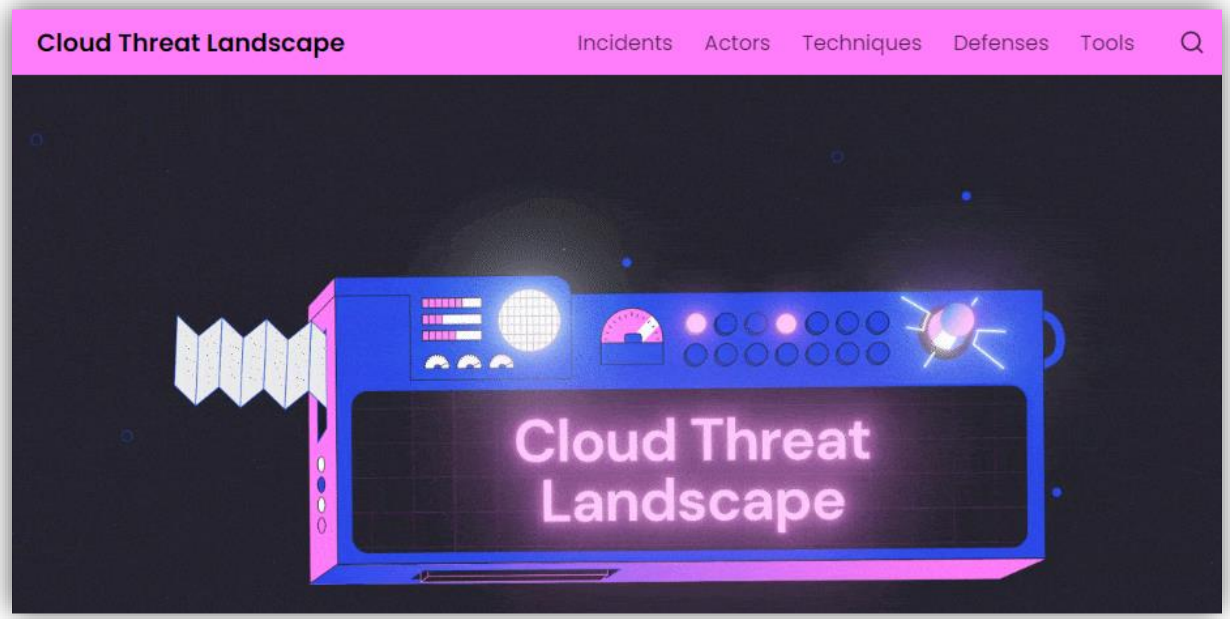
TI-driven

Planning the Exercise

TI-driven

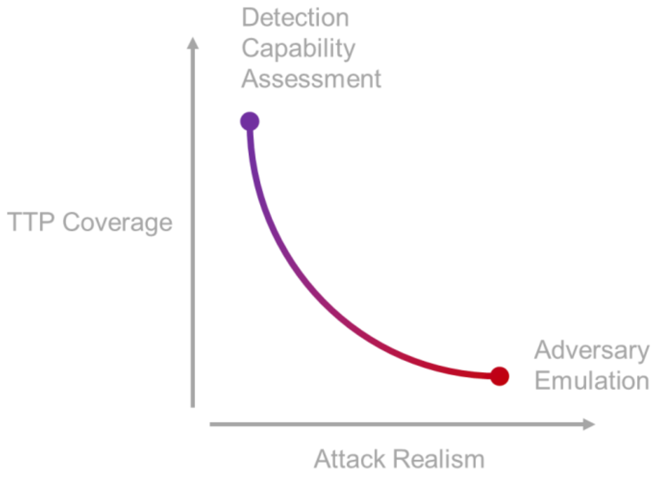


Select Campaign of interest

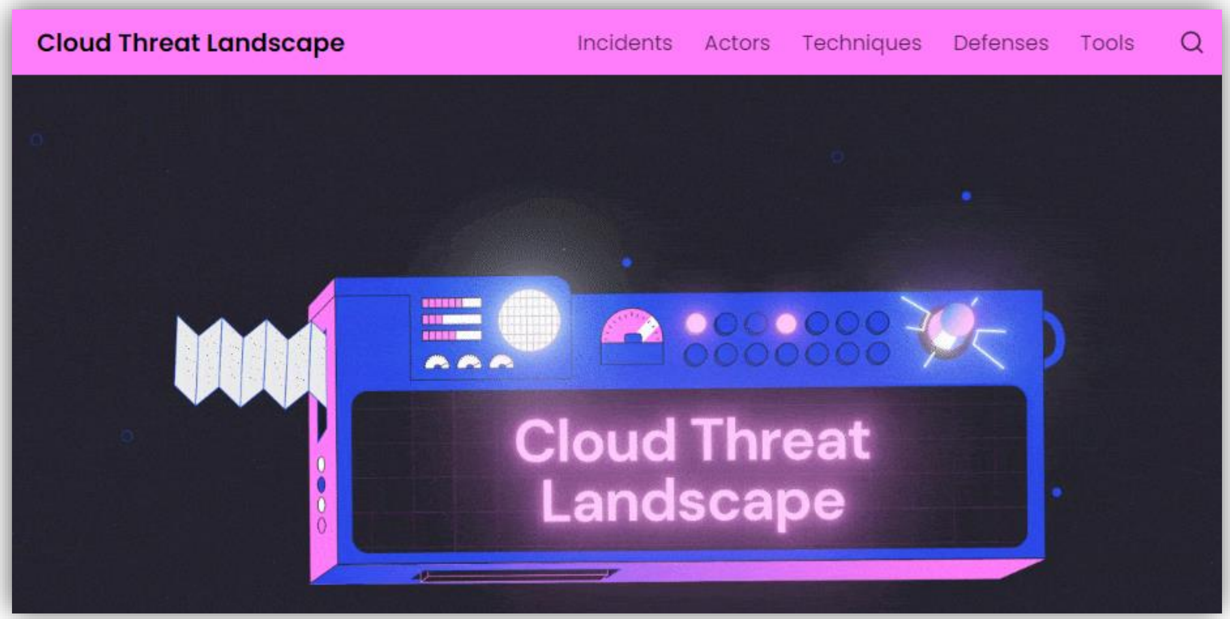


Planning the Exercise

TI-driven

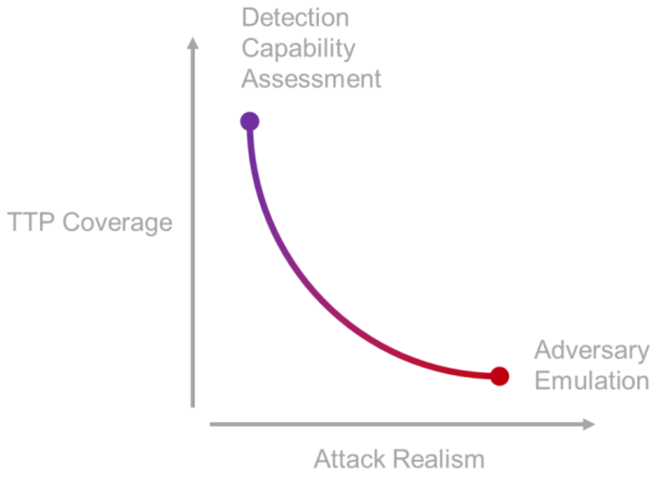


Select Campaign of interest → Gather Threat Intelligence



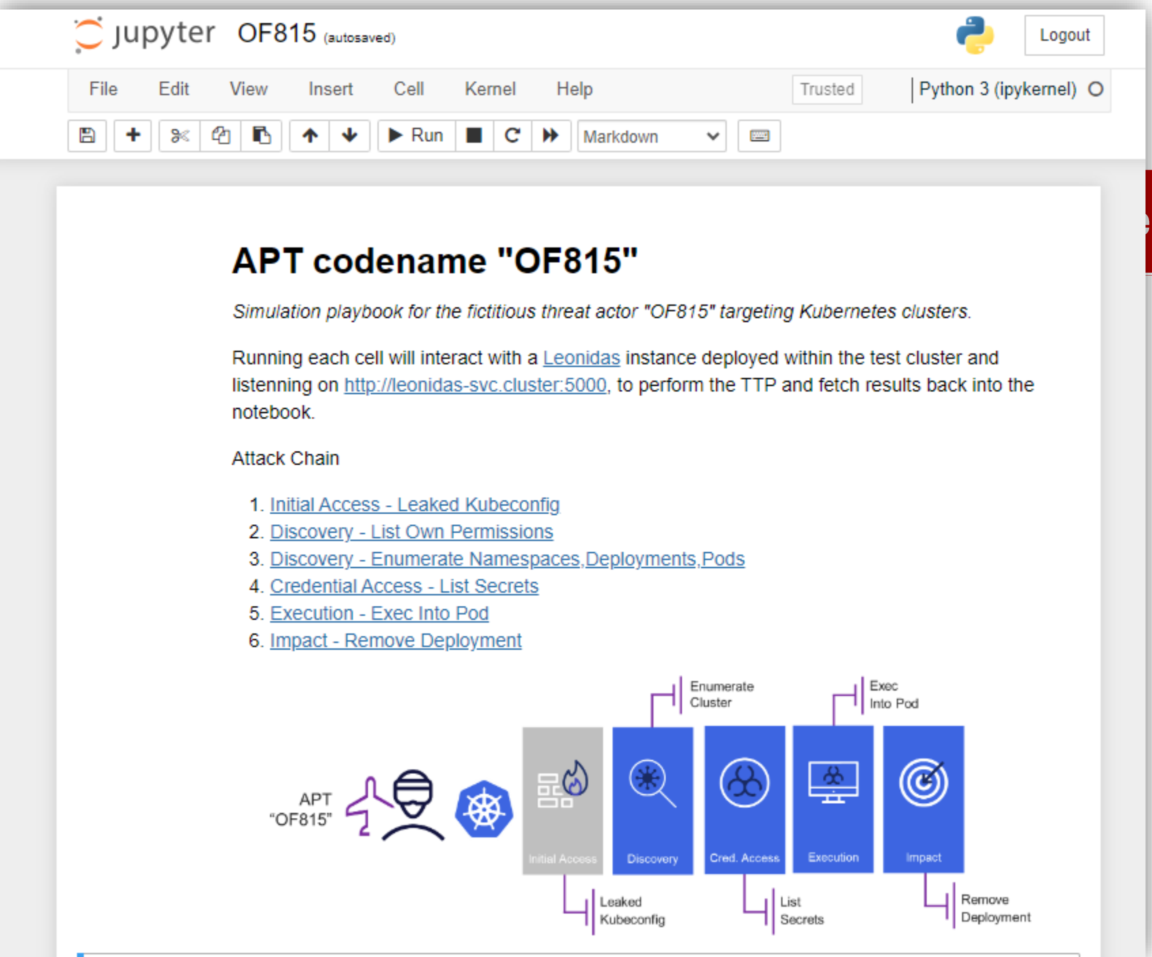
Planning the Exercise

TI-driven



lligence

Re-produce Attack Chain



APT codename "OF815"

Simulation playbook for the fictitious threat actor "OF815" targeting Kubernetes clusters.

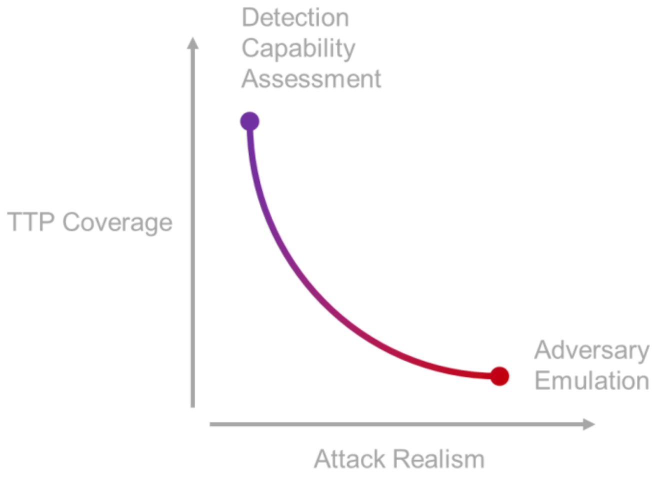
Running each cell will interact with a Leonidas instance deployed within the test cluster and listenning on http://leonidas-svc.cluster:5000, to perform the TTP and fetch results back into the notebook.

Attack Chain

1. Initial Access - Leaked Kubeconfig
2. Discovery - List Own Permissions
3. Discovery - Enumerate Namespaces,Deployments,Pods
4. Credential Access - List Secrets
5. Execution - Exec Into Pod
6. Impact - Remove Deployment

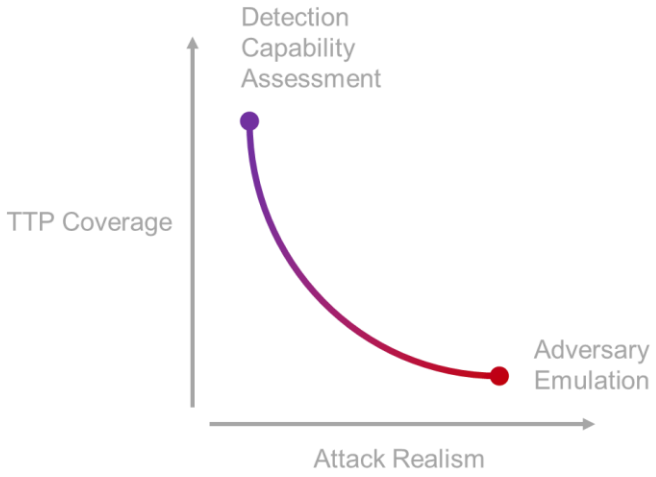
Planning the Exercise

Breadth-first

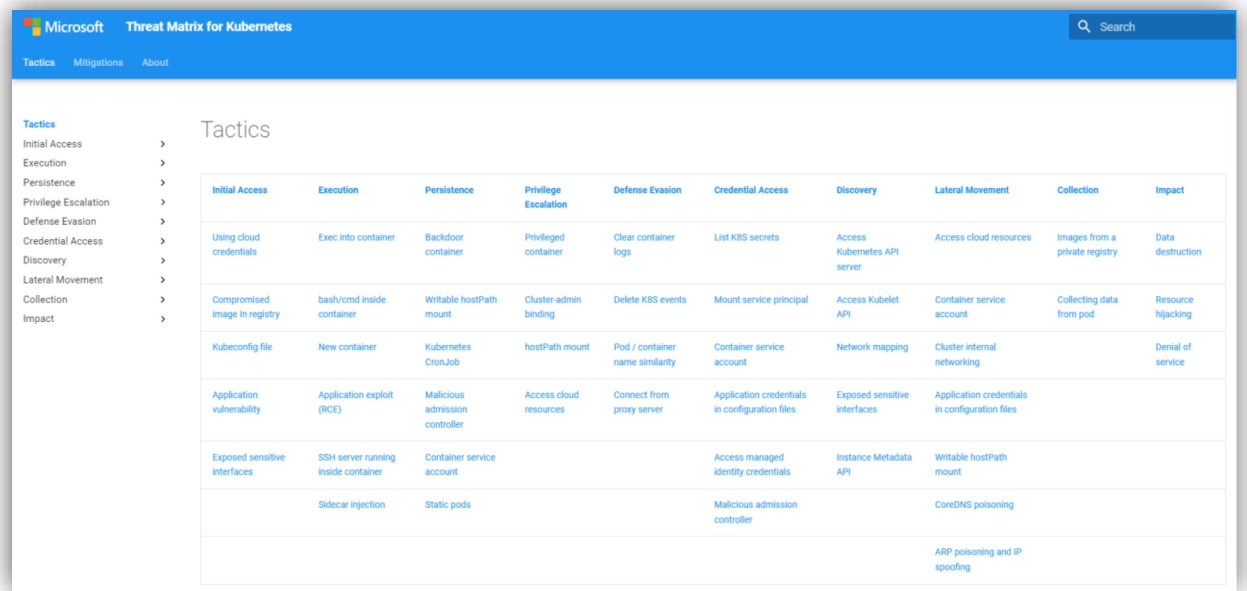


Planning the Exercise

Breadth-first

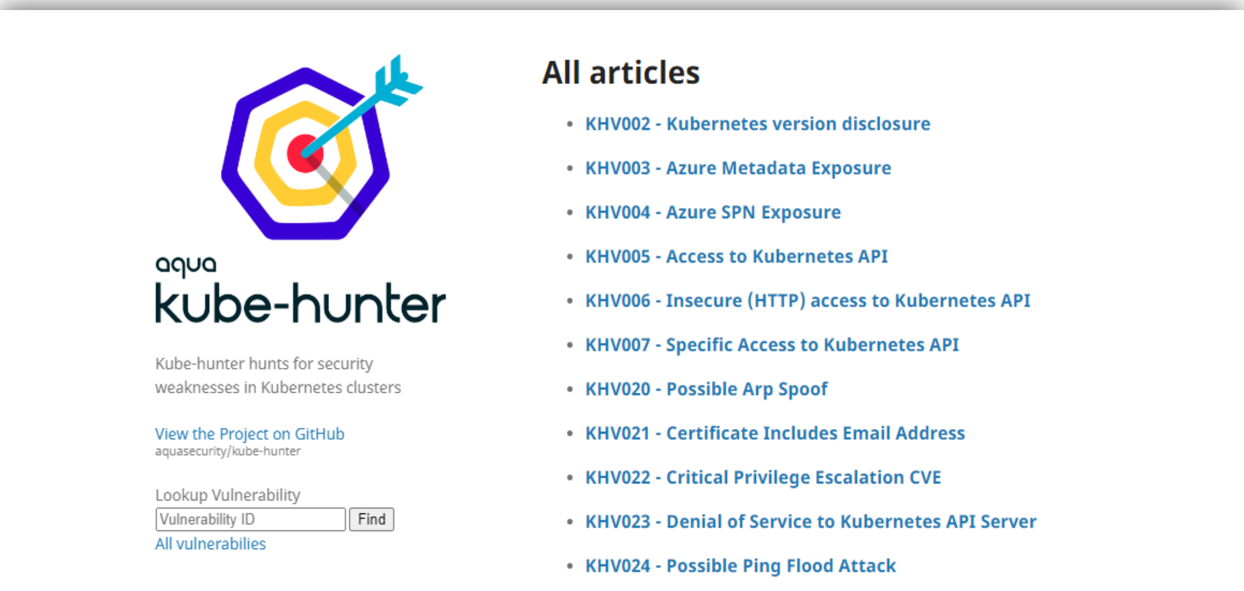


Select TTPs



Tactics

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 Kubernetes API server	Access cloud resources	Images from a private registry	Data destruction
Compromised image In registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account	Collecting data from pod	Resource hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	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	Application credentials in configuration files	Exposed sensitive interfaces	Application credentials in configuration files		
Exposed sensitive interfaces	SSH server running inside container	Container service account			Access managed identity credentials	Instance Metadata API	Writable hostPath mount		
	Sidecar injection	Static pods			Malicious admission controller		CoreDNS poisoning		
							ARP poisoning and IP spoofing		



Kube-hunter hunts for security weaknesses in Kubernetes clusters

View the Project on GitHub
aquasecurity/kube-hunter

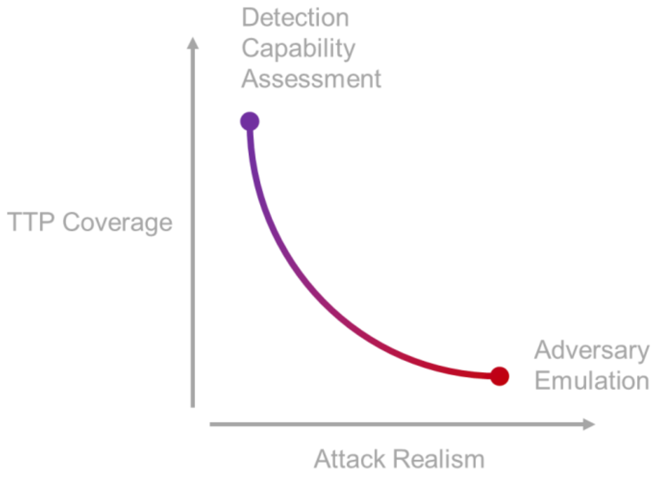
Lookup Vulnerability
Vulnerability ID [Find]
All vulnerabilies

All articles

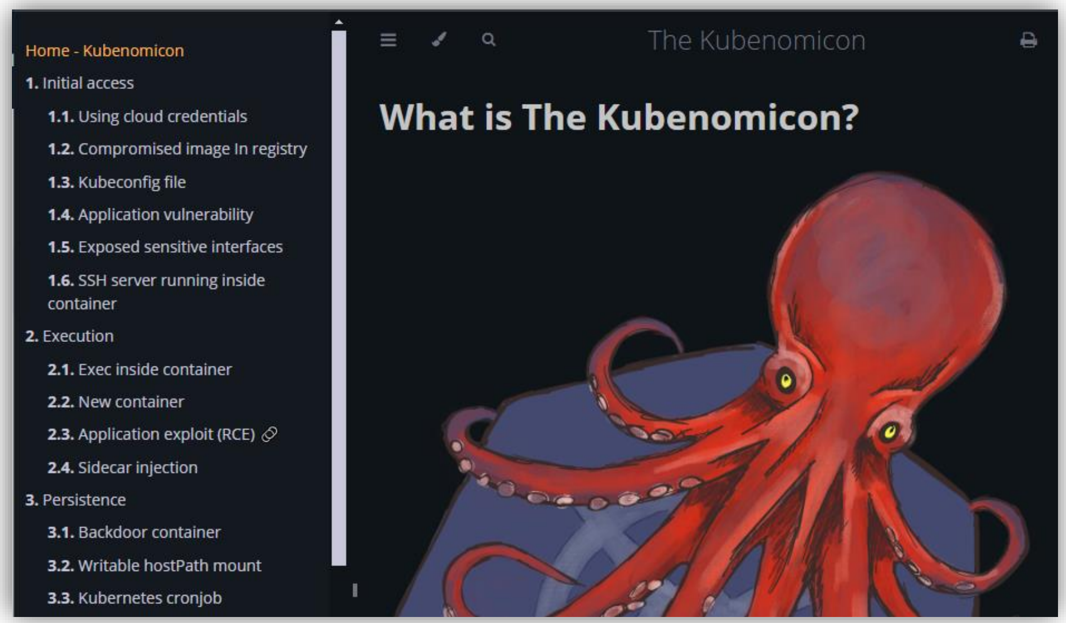
- KHV002 - Kubernetes version disclosure
- KHV003 - Azure Metadata Exposure
- KHV004 - Azure SPN Exposure
- KHV005 - Access to Kubernetes API
- KHV006 - Insecure (HTTP) access to Kubernetes API
- KHV007 - Specific Access to Kubernetes API
- KHV020 - Possible Arp Spoof
- KHV021 - Certificate Includes Email Address
- KHV022 - Critical Privilege Escalation CVE
- KHV023 - Denial of Service to Kubernetes API Server
- KHV024 - Possible Ping Flood Attack

Planning the Exercise

Breadth-first



Select TTPs → Design Test Cases



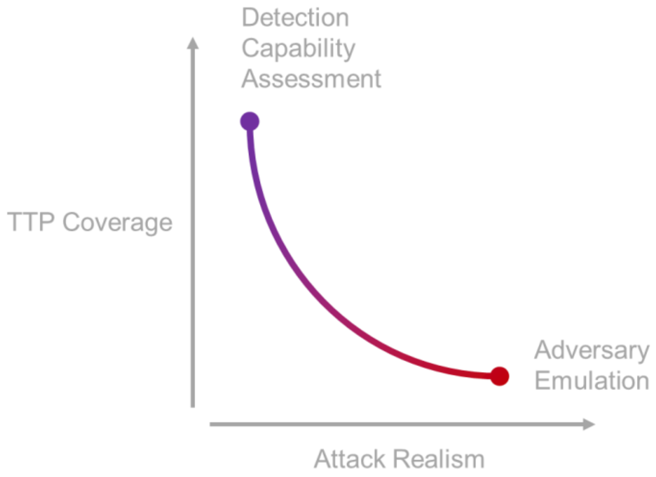
Home - Kubenomicon

1. Initial access
   - 1.1. Using cloud credentials
   - 1.2. Compromised image In registry
   - 1.3. Kubeconfig file
   - 1.4. Application vulnerability
   - 1.5. Exposed sensitive interfaces
   - 1.6. SSH server running inside container
2. Execution
   - 2.1. Exec inside container
   - 2.2. New container
   - 2.3. Application exploit (RCE)
   - 2.4. Sidecar injection
3. Persistence
   - 3.1. Backdoor container
   - 3.2. Writable hostPath mount
   - 3.3. Kubernetes cronjob

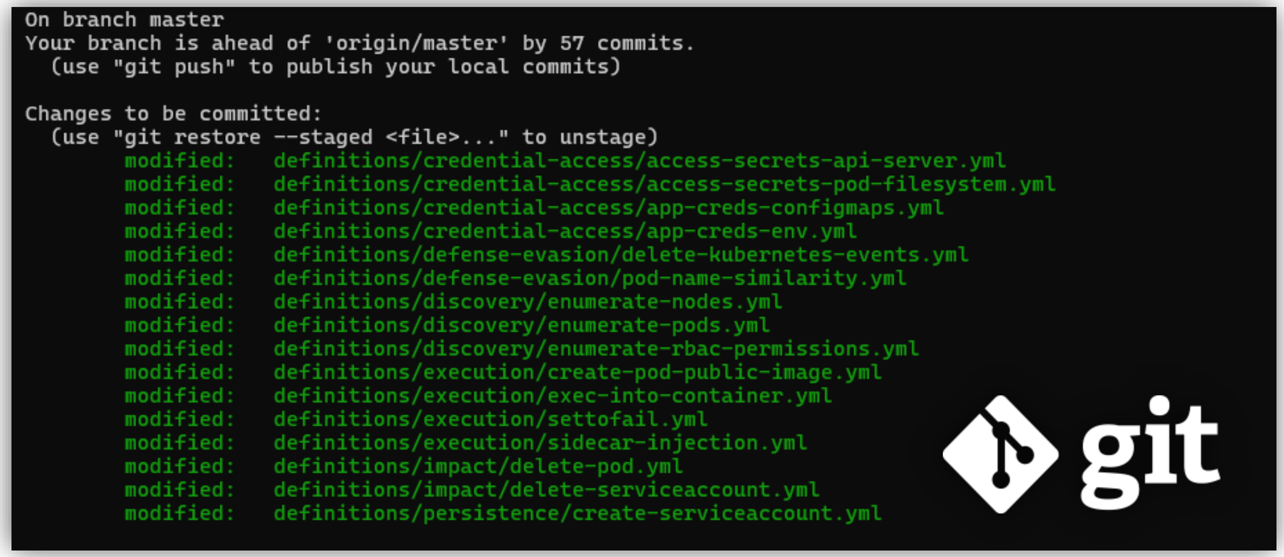
What is The Kubenomicon?

Planning the Exercise

Breadth-first



Select TTPs → Design Test Cases → Maintain, Expand, Repeat



```
On branch master
Your branch is ahead of 'origin/master' by 57 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   definitions/credential-access/access-secrets-api-server.yml
        modified:   definitions/credential-access/access-secrets-pod-filesystem.yml
        modified:   definitions/credential-access/app-creds-configmaps.yml
        modified:   definitions/credential-access/app-creds-env.yml
        modified:   definitions/defense-evasion/delete-kubernetes-events.yml
        modified:   definitions/defense-evasion/pod-name-similarity.yml
        modified:   definitions/discovery/enumerate-nodes.yml
        modified:   definitions/discovery/enumerate-pods.yml
        modified:   definitions/discovery/enumerate-rbac-permissions.yml
        modified:   definitions/execution/create-pod-public-image.yml
        modified:   definitions/execution/exec-into-container.yml
        modified:   definitions/execution/settofail.yml
        modified:   definitions/execution/sidecar-injection.yml
        modified:   definitions/impact/delete-pod.yml
        modified:   definitions/impact/delete-serviceaccount.yml
        modified:   definitions/persistence/create-serviceaccount.yml
```

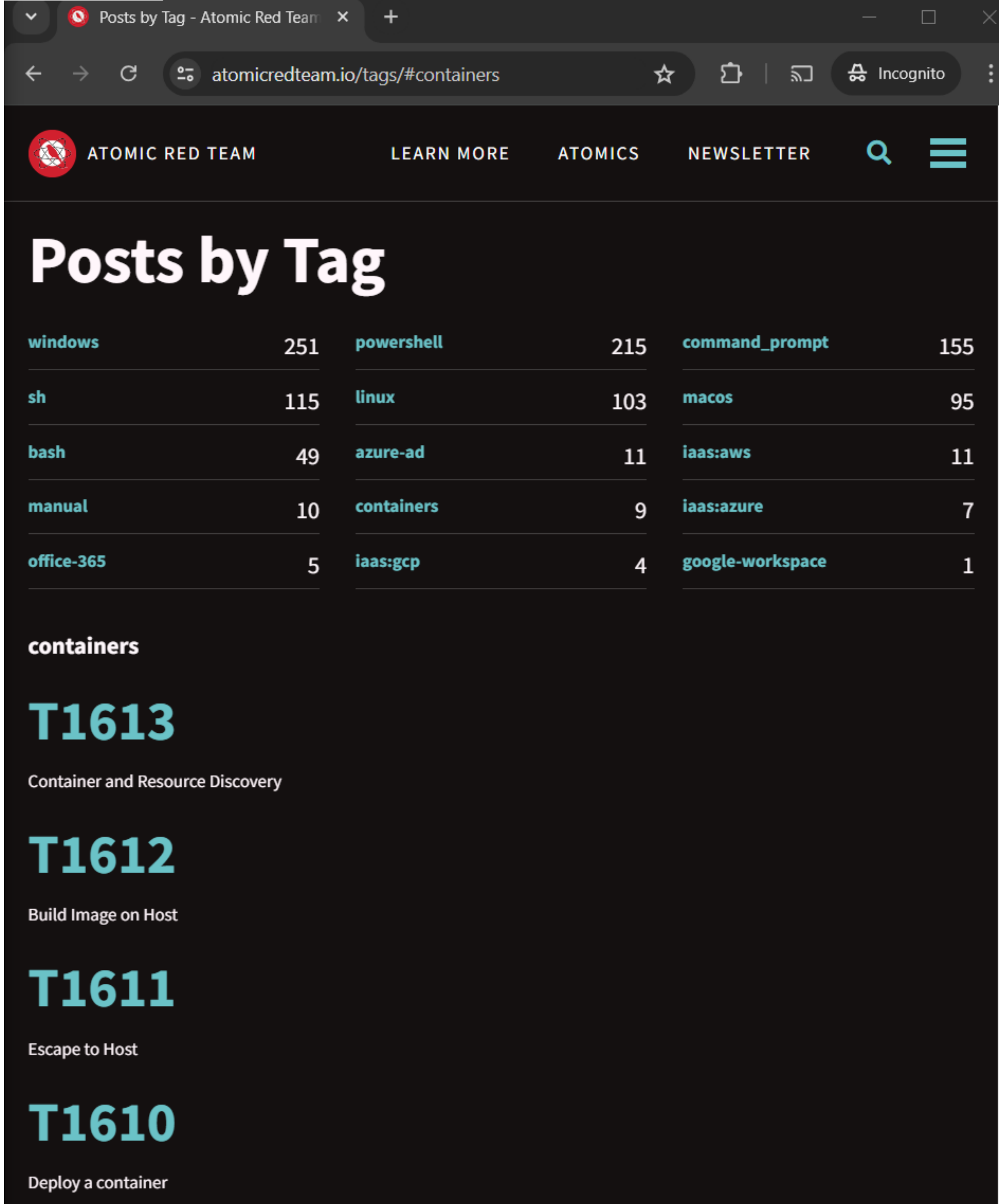
Execution

- K8S Attack Simulation Tools / Frameworks

Execution

- K8S Attack Simulation Tools / Frameworks

  1. Atomic Red Team


Posts by Tag - Atomic Red Tean
atomicredteam.io/tags/#containers
Incognito
ATOMIC RED TEAM
LEARN MORE
ATOMICS
NEWSLETTER

Posts by Tag

windows	251	powershell	215	command_prompt	155
sh	115	linux	103	macos	95
bash	49	azure-ad	11	iaas:aws	11
manual	10	containers	9	iaas:azure	7
office-365	5	iaas:gcp	4	google-workspace	1

containers

T1613

Container and Resource Discovery

T1612

Build Image on Host

T1611

Escape to Host

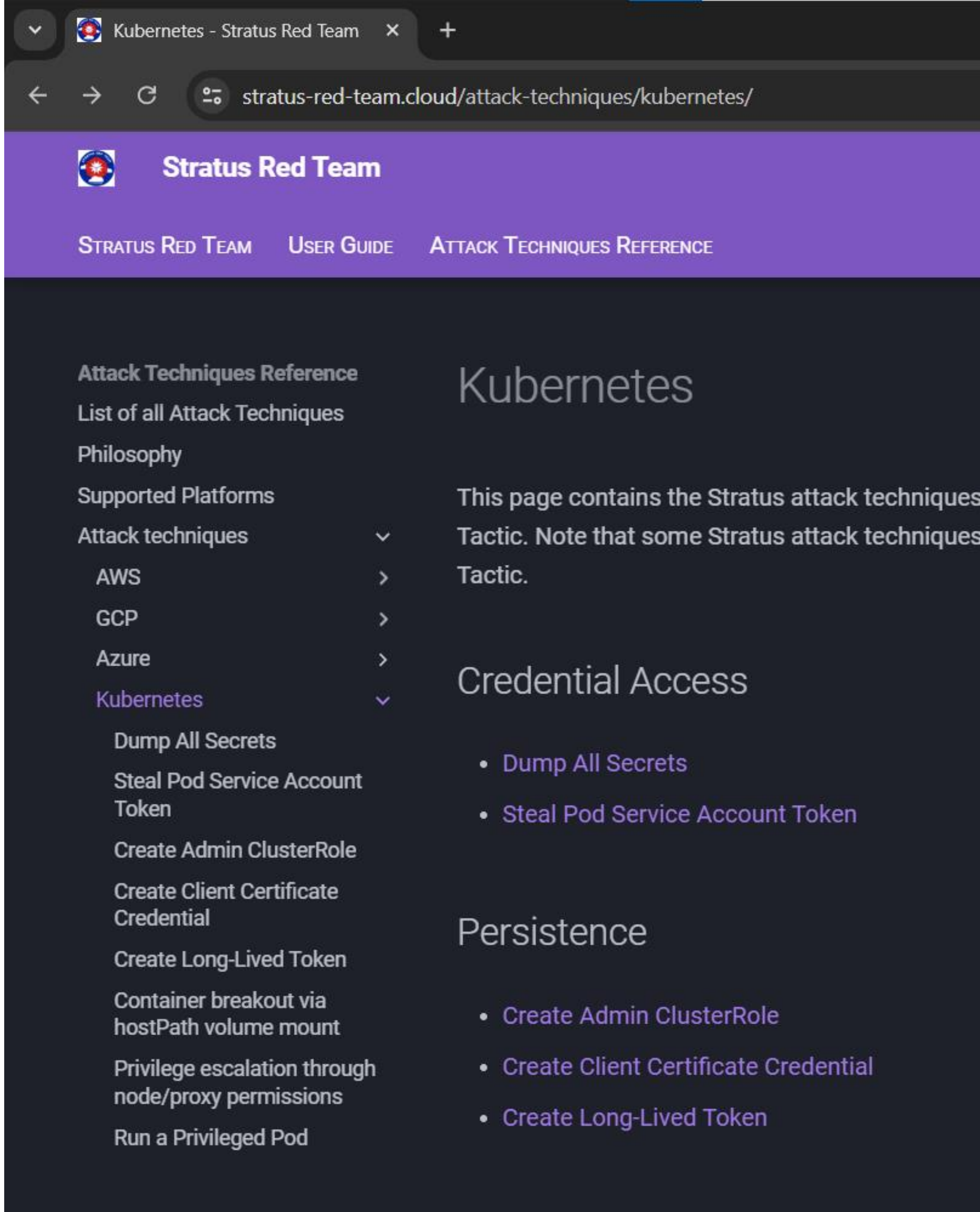
T1610

Deploy a container

Execution

- K8S Attack Simulation Tools / Frameworks

  1. Atomic Red Team
  2. Stratus Red Team



Kubernetes - Stratus Red Team
stratus-red-team.cloud/attack-techniques/kubernetes/
Stratus Red Team
Stratus Red Team
User Guide
Attack Techniques Reference

Attack Techniques Reference

List of all Attack Techniques

Philosophy

Supported Platforms

Attack techniques

- AWS
- GCP
- Azure
- Kubernetes
  - Dump All Secrets
  - Steal Pod Service Account Token
  - Create Admin ClusterRole
  - Create Client Certificate Credential
  - Create Long-Lived Token
  - Container breakout via hostPath volume mount
  - Privilege escalation through node/proxy permissions
  - Run a Privileged Pod

Kubernetes

This page contains the Stratus attack techniques
Tactic. Note that some Stratus attack techniques
Tactic.

Credential Access

- Dump All Secrets
- Steal Pod Service Account Token

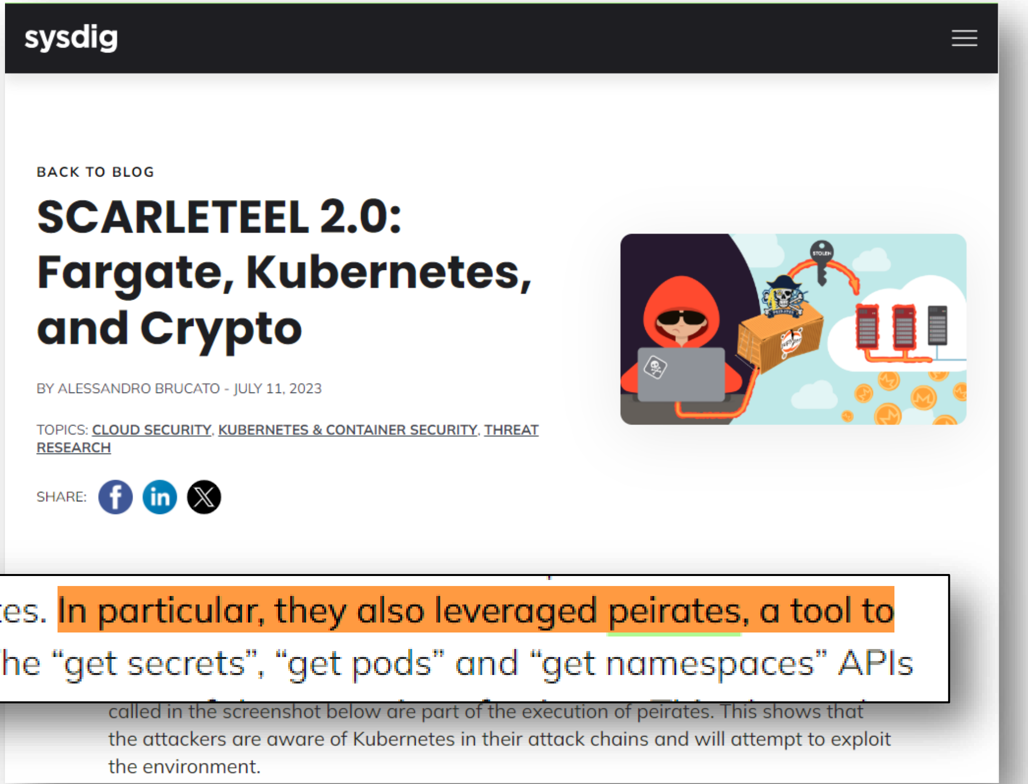
Persistence

- Create Admin ClusterRole
- Create Client Certificate Credential
- Create Long-Lived Token

Execution

- K8S Attack Simulation Tools / Frameworks

  1. Atomic Red Team
  2. Stratus Red Team
  3. Peirates



sysdig

BACK TO BLOG

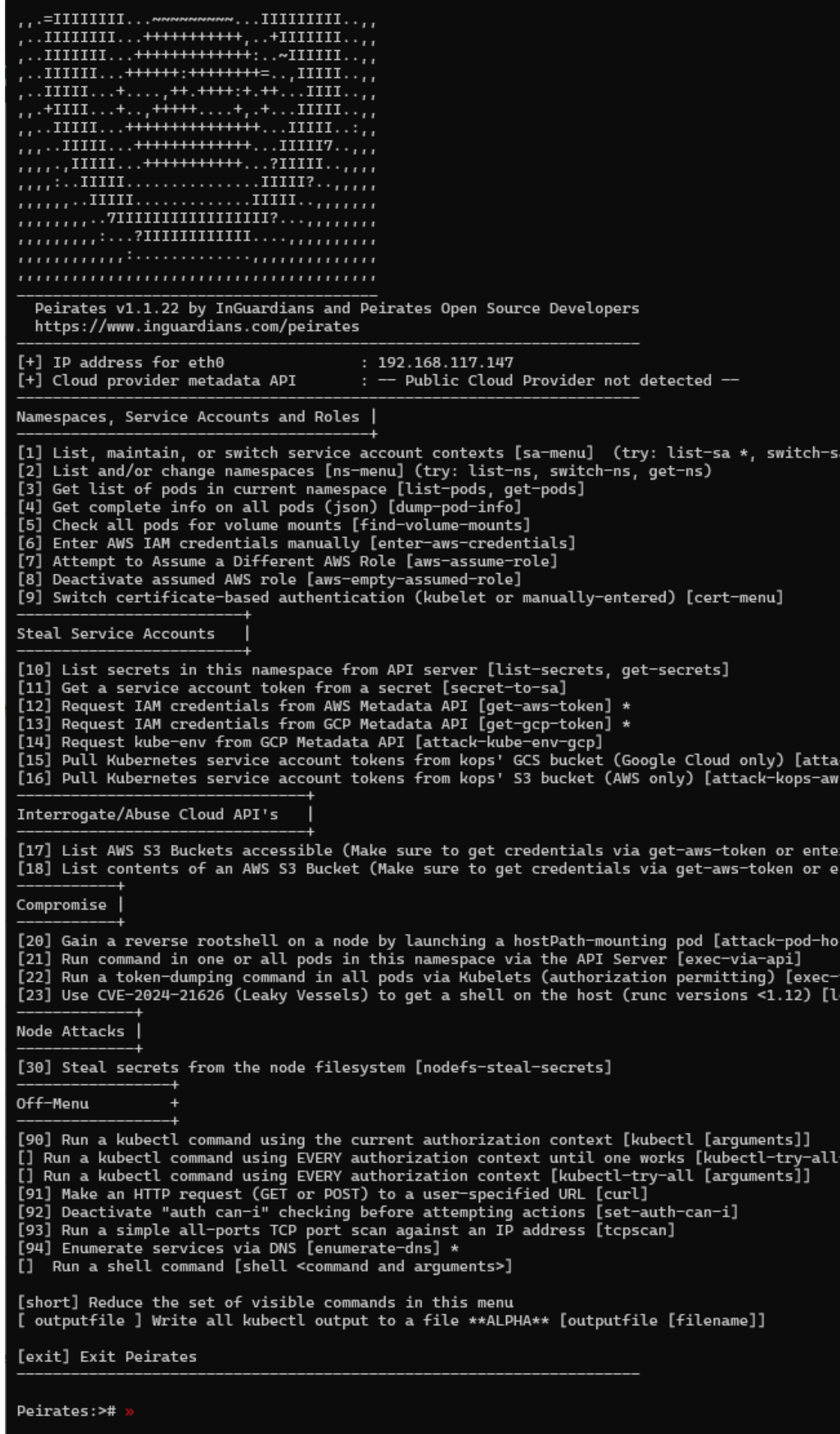
SCARLETEEL 2.0: Fargate, Kubernetes, and Crypto

BY ALESSANDRO BRUCATO - JULY 11, 2023

TOPICS: CLOUD SECURITY, KUBERNETES & CONTAINER SECURITY, THREAT RESEARCH

SHARE:

including targeting Kubernetes. In particular, they also leveraged peirates, a tool to further exploit Kubernetes. The “get secrets”, “get pods” and “get namespaces” APIs called in the screenshot below are part of the execution of peirates. This shows that the attackers are aware of Kubernetes in their attack chains and will attempt to exploit the environment.

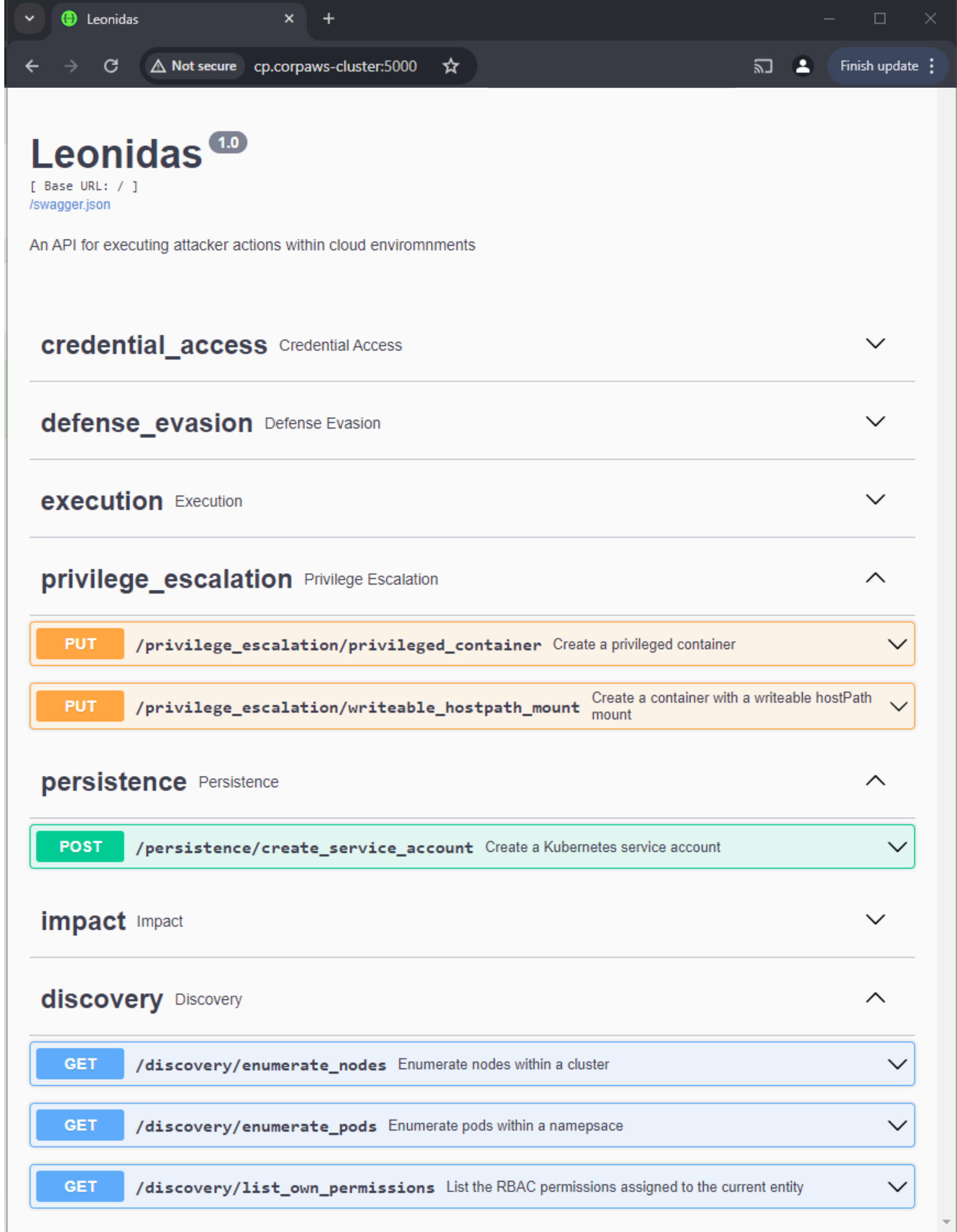


```
,,.=IIIIIIII...~~~~~~~~~~...IIIIIIIII..,,
,..IIIIIIII...+++++++++++,..+IIIIIII..,,
,..IIIIIII...+++++++++++++:..~IIIIII..,,
,..IIIIII...++++++:++++++++=..,IIIII..,,
,..IIIII...+....,++.++++:+.++...IIII..,,
,,.+IIII...+..,+++++....+,.+...IIIII..,,
,,..IIIII...+++++++++++++++...IIIII..:,,
,,,..IIIII...+++++++++++++...IIIII7..,,,
,,,,.,IIIII...++++++++++++...?IIIII..,,,,
,,,,:..IIIII...............IIIII?..,,,,,
,,,,,,..IIIII.............IIIII..,,,,,,,
,,,,,,,,..7IIIIIIIIIIIIIIIII?...,,,,,,,,
,,,,,,,,,:...?IIIIIIIIIIII....,,,,,,,,,,
,,,,,,,,,,,,:..............,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
----------------------------------------
  Peirates v1.1.22 by InGuardians and Peirates Open Source Developers
  https://www.inguardians.com/peirates
-----------------------------------------------------------------------
[+] IP address for eth0              : 192.168.117.147
[+] Cloud provider metadata API      : -- Public Cloud Provider not detected --
-----------------------------------------------------------------------
Namespaces, Service Accounts and Roles |
---------------------------------------+
[1] List, maintain, or switch service account contexts [sa-menu]  (try: list-sa *, switch-s
[2] List and/or change namespaces [ns-menu] (try: list-ns, switch-ns, get-ns)
[3] Get list of pods in current namespace [list-pods, get-pods]
[4] Get complete info on all pods (json) [dump-pod-info]
[5] Check all pods for volume mounts [find-volume-mounts]
[6] Enter AWS IAM credentials manually [enter-aws-credentials]
[7] Attempt to Assume a Different AWS Role [aws-assume-role]
[8] Deactivate assumed AWS role [aws-empty-assumed-role]
[9] Switch certificate-based authentication (kubelet or manually-entered) [cert-menu]
-------------------------+
Steal Service Accounts   |
-------------------------+
[10] List secrets in this namespace from API server [list-secrets, get-secrets]
[11] Get a service account token from a secret [secret-to-sa]
[12] Request IAM credentials from AWS Metadata API [get-aws-token] *
[13] Request IAM credentials from GCP Metadata API [get-gcp-token] *
[14] Request kube-env from GCP Metadata API [attack-kube-env-gcp]
[15] Pull Kubernetes service account tokens from kops' GCS bucket (Google Cloud only) [atta
[16] Pull Kubernetes service account tokens from kops' S3 bucket (AWS only) [attack-kops-aw
--------------------------------+
Interrogate/Abuse Cloud API's   |
--------------------------------+
[17] List AWS S3 Buckets accessible (Make sure to get credentials via get-aws-token or ente
[18] List contents of an AWS S3 Bucket (Make sure to get credentials via get-aws-token or e
-----------+
Compromise |
-----------+
[20] Gain a reverse rootshell on a node by launching a hostPath-mounting pod [attack-pod-ho
[21] Run command in one or all pods in this namespace via the API Server [exec-via-api]
[22] Run a token-dumping command in all pods via Kubelets (authorization permitting) [exec-
[23] Use CVE-2024-21626 (Leaky Vessels) to get a shell on the host (runc versions <1.12) [l
-------------+
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 [arguments]]
[] Run a kubectl command using EVERY authorization context until one works [kubectl-try-all
[] Run a kubectl command using EVERY authorization context [kubectl-try-all [arguments]]
[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] *
[]  Run a shell command [shell <command and arguments>]

[short] Reduce the set of visible commands in this menu
[ outputfile ] Write all kubectl output to a file **ALPHA** [outputfile [filename]]

[exit] Exit Peirates
-----------------------------------------------------------------------

Peirates:># »
```

Execution

- K8S Attack Simulation Tools / Frameworks

  1. Atomic Red Team
  2. Stratus Red Team
  3. Peirates
  4. Leonidas for K8S



Leonidas 1.0

[Base URL: /]

/swagger.json

An API for executing attacker actions within cloud enviromnments

credential_access Credential Access

defense_evasion Defense Evasion

execution Execution

privilege_escalation Privilege Escalation

PUT /privilege_escalation/privileged_container Create a privileged container

PUT /privilege_escalation/writeable_hostpath_mount Create a container with a writeable hostPath mount

persistence Persistence

POST /persistence/create_service_account Create a Kubernetes service account

impact Impact

discovery Discovery

GET /discovery/enumerate_nodes Enumerate nodes within a cluster

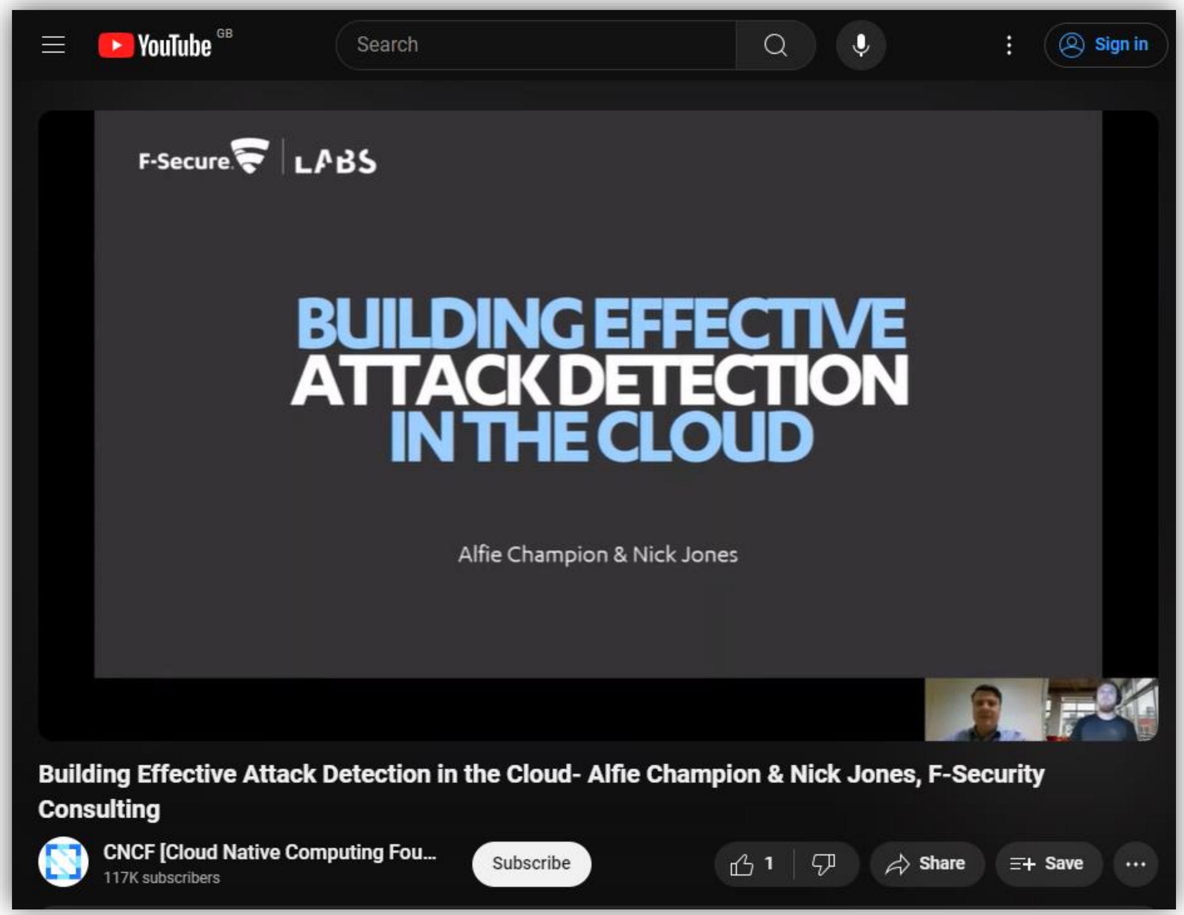
GET /discovery/enumerate_pods Enumerate pods within a namespace

GET /discovery/list_own_permissions List the RBAC permissions assigned to the current entity

Leonidas

https://github.com/WithSecureLabs/leonidas

- Extensible
- Easy to write attack test cases
- Attacks- & Detections-as-Code
- Permission management
- REST API / Scripting-friendly

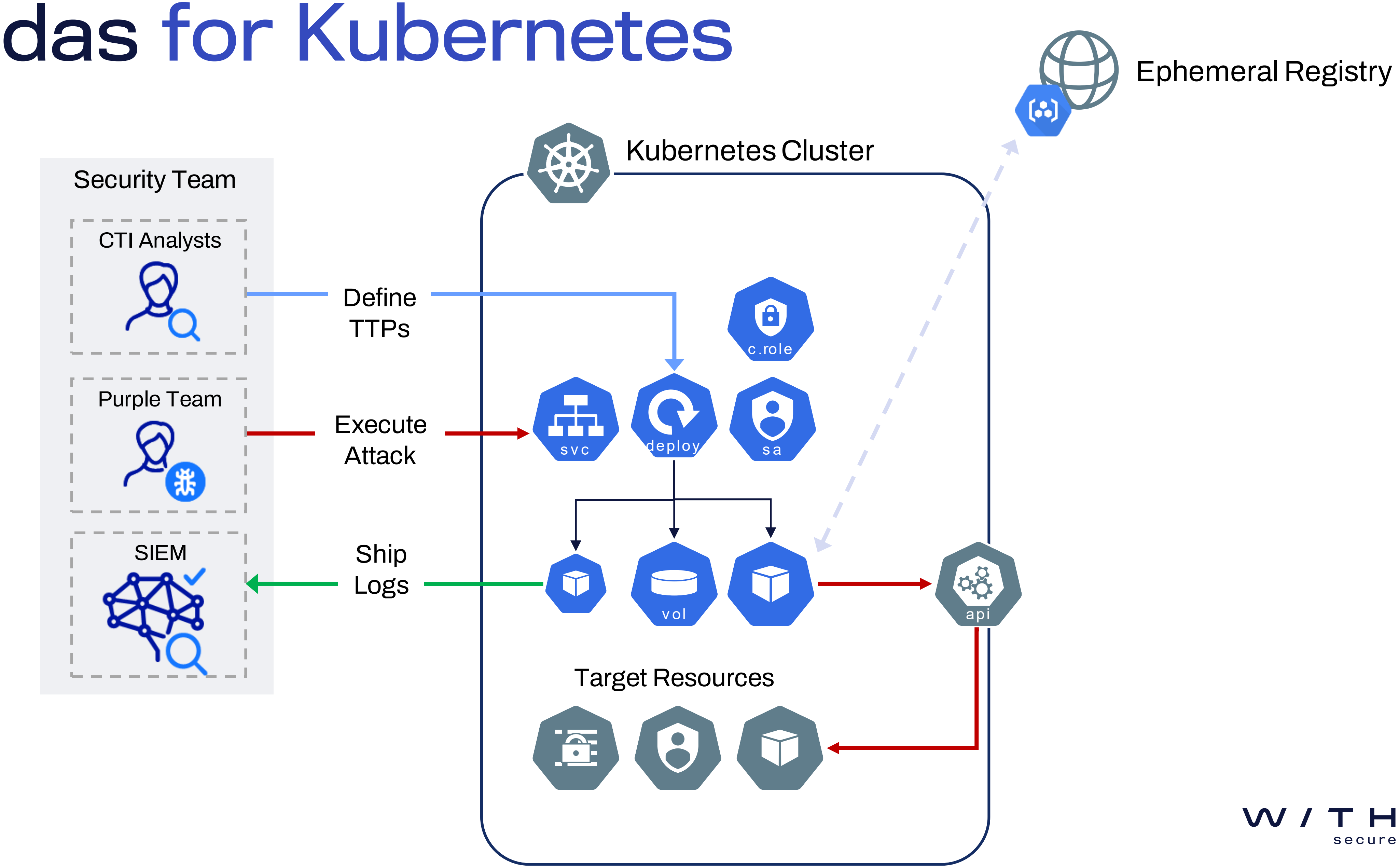

YouTube GB
Search
Sign in
F-Secure
LABS
BUILDING EFFECTIVE ATTACK DETECTION IN THE CLOUD
Alfie Champion & Nick Jones

Building Effective Attack Detection in the Cloud- Alfie Champion & Nick Jones, F-Security Consulting

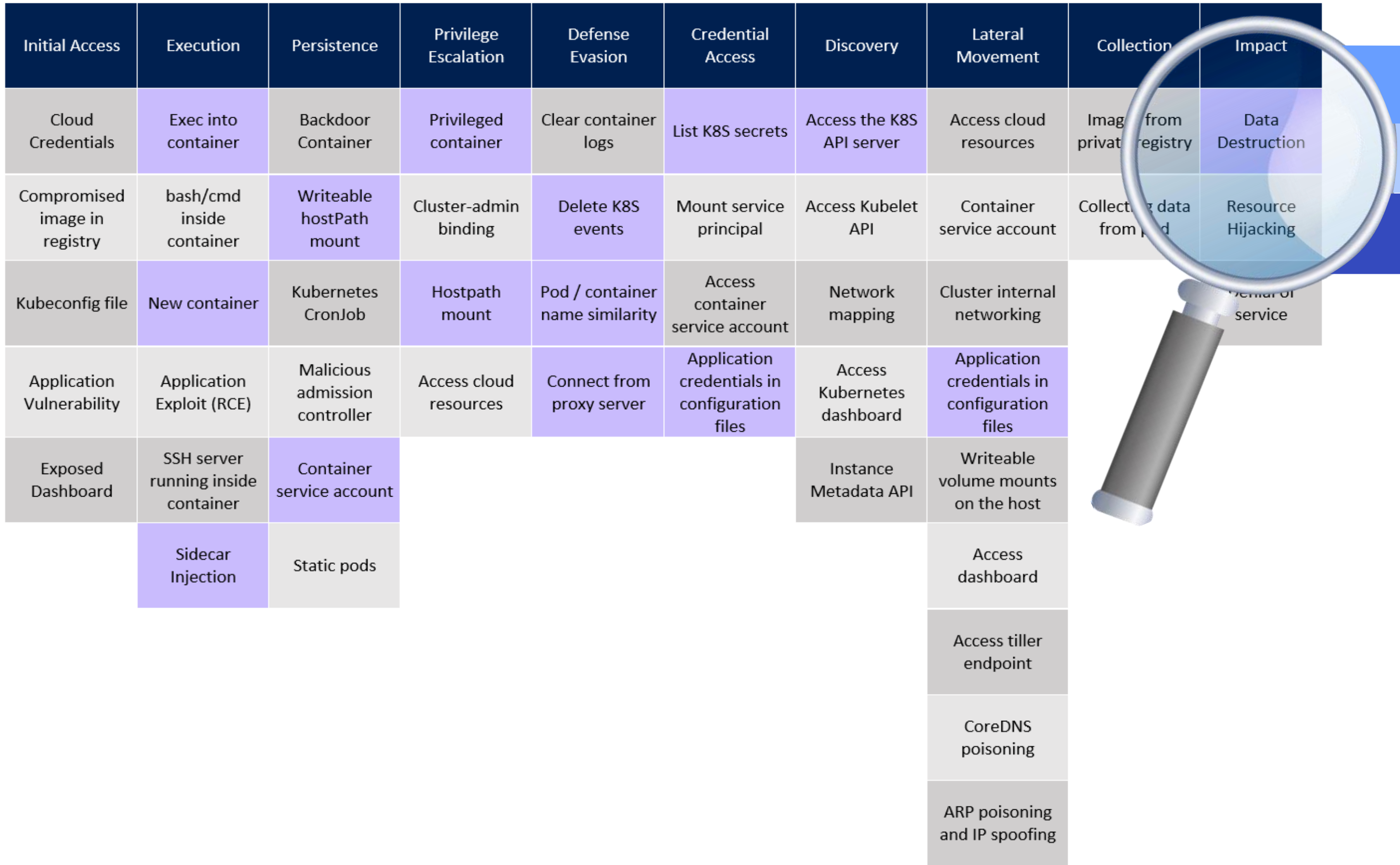
CNCF [Cloud Native Computing Fou...
117K subscribers
Subscribe
1
Share
Save

W/TH secure

Leonidas for Kubernetes



Leonidas for Kubernetes



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Impact
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 private registry	Data Destruction
Compromised image in registry	bash/cmd inside container	Writeable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account	Collecting data from pod	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	Application credentials in configuration files	Access Kubernetes dashboard	Application credentials in configuration files		
Exposed Dashboard	SSH server running inside container	Container service account				Instance Metadata API	Writeable volume mounts on the host		
	Sidecar Injection	Static pods					Access dashboard		
							Access tiller endpoint		
							CoreDNS poisoning		
							ARP poisoning and IP spoofing		

```
kubectl delete events
```

```
find /var/run/secrets/
```

```
kubectl -f /tmp/custom.yml apply
```

https://microsoft.github.io/Threat-Matrix-for-Kubernetes/

Threat Modelling

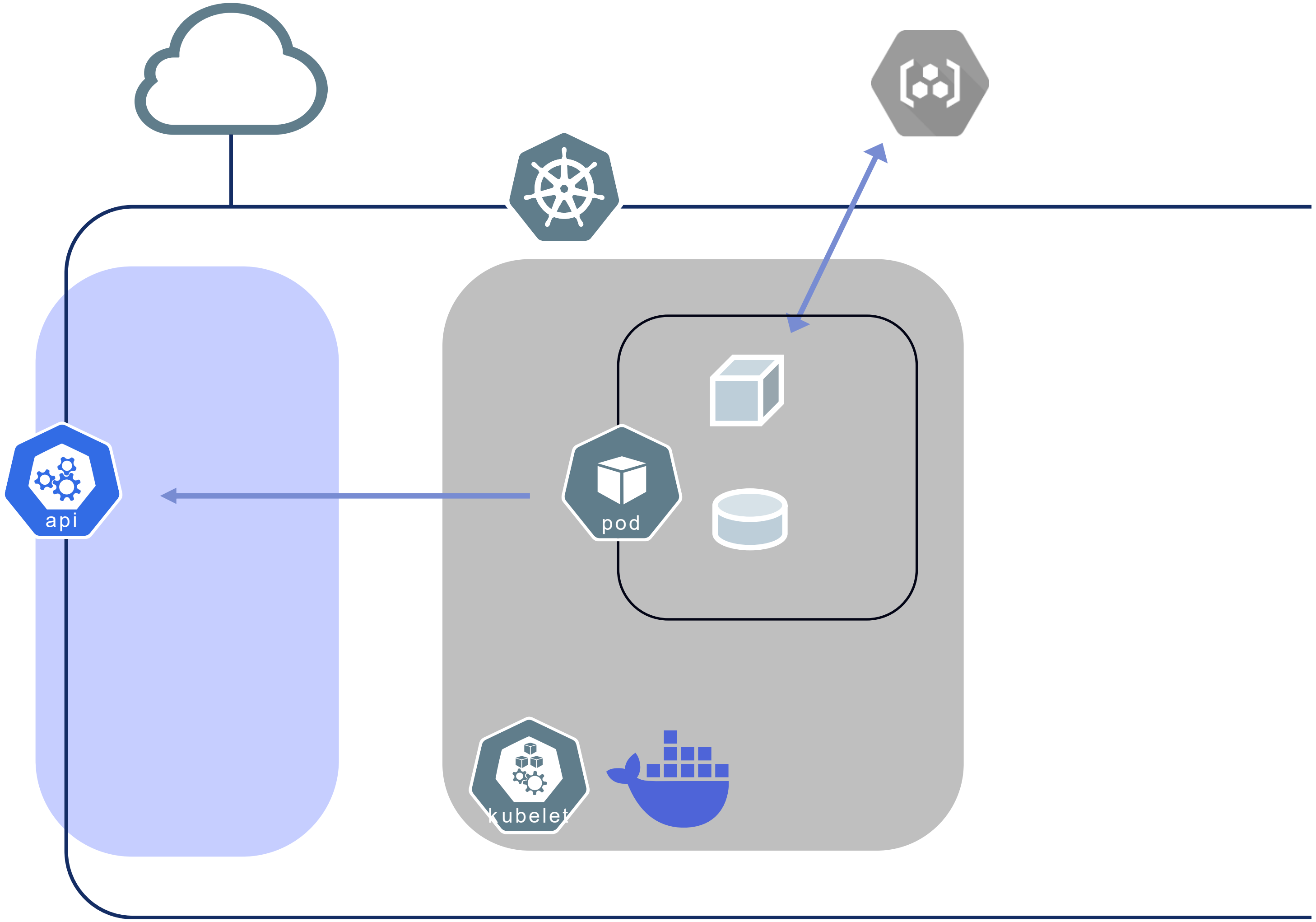
Kubernetes
Attack Simulation

Kubernetes
Attack Detection

Demo

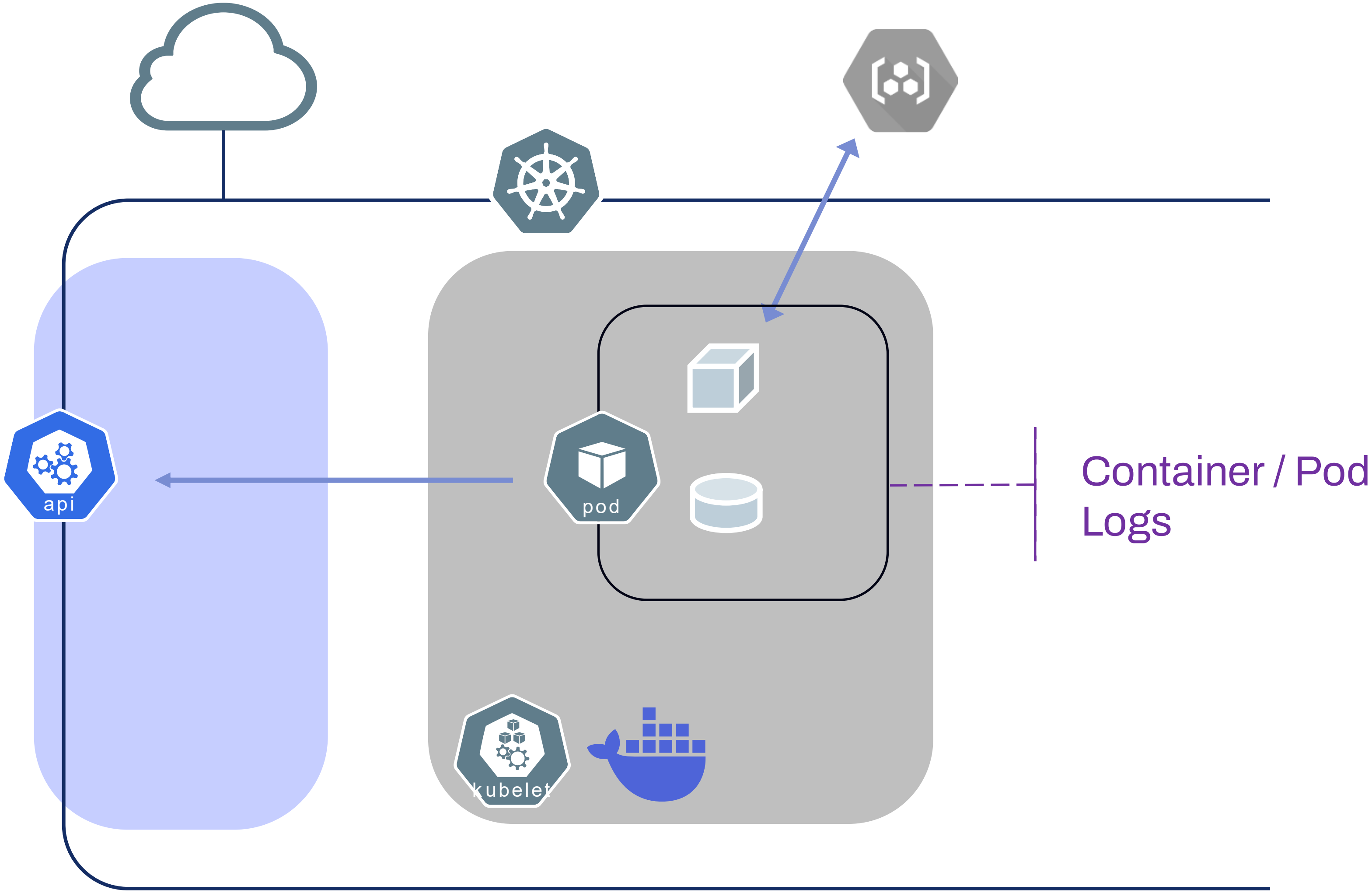
Log Sources

https://kubernetes.io/docs/concepts/cluster-administration/logging/



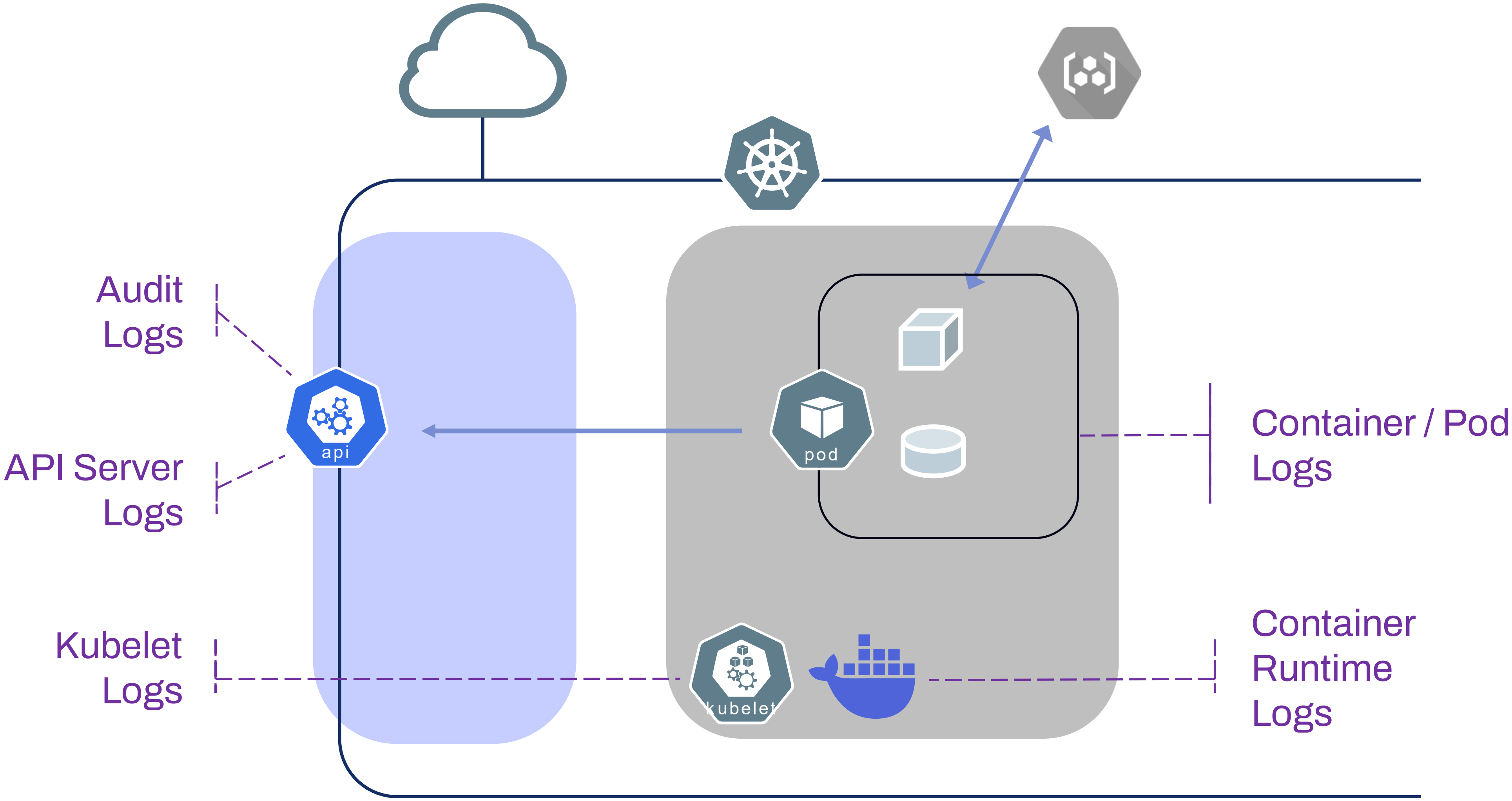
Log Sources

https://kubernetes.io/docs/concepts/cluster-administration/logging/



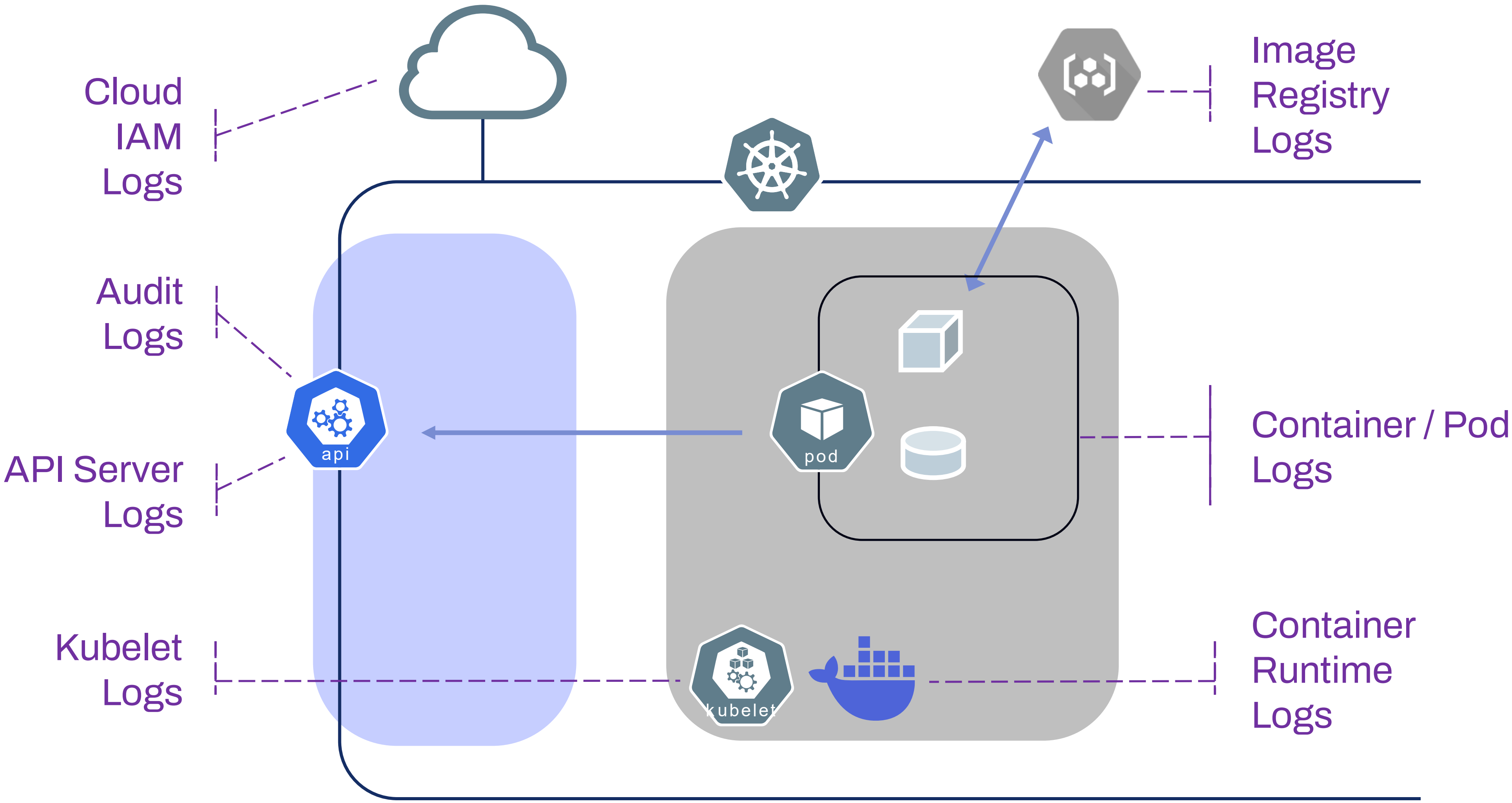
Log Sources

https://kubernetes.io/docs/concepts/cluster-administration/logging/



Log Sources

https://kubernetes.io/docs/concepts/cluster-administration/logging/



Cloud IAM Logs

Audit Logs

API Server Logs

Kubelet Logs

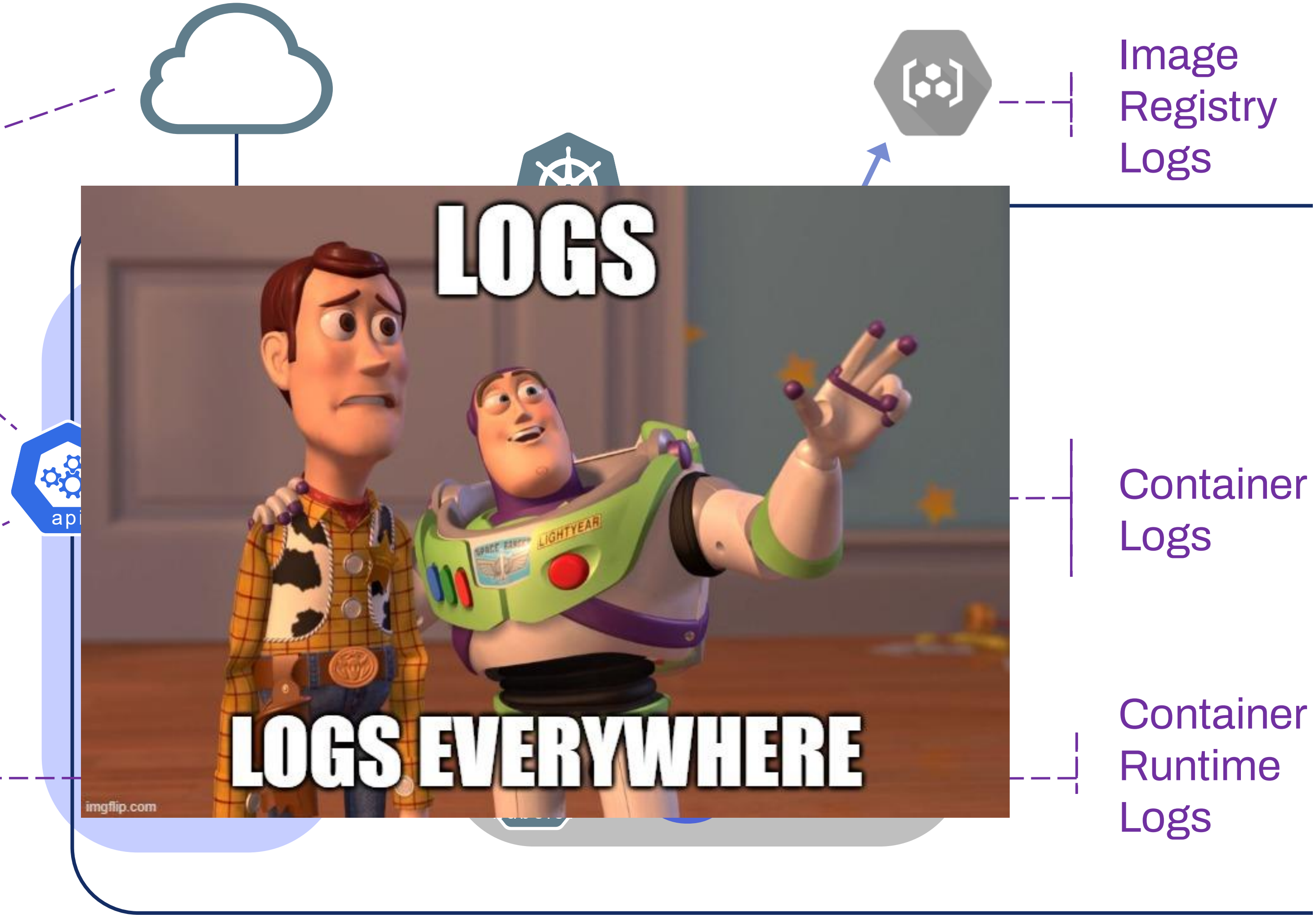
Image Registry Logs

Container / Pod Logs

Container Runtime Logs

Log Sources

https://kubernetes.io/docs/concepts/cluster-administration/logging/



Cloud IAM Logs

Audit Logs

API Server Logs

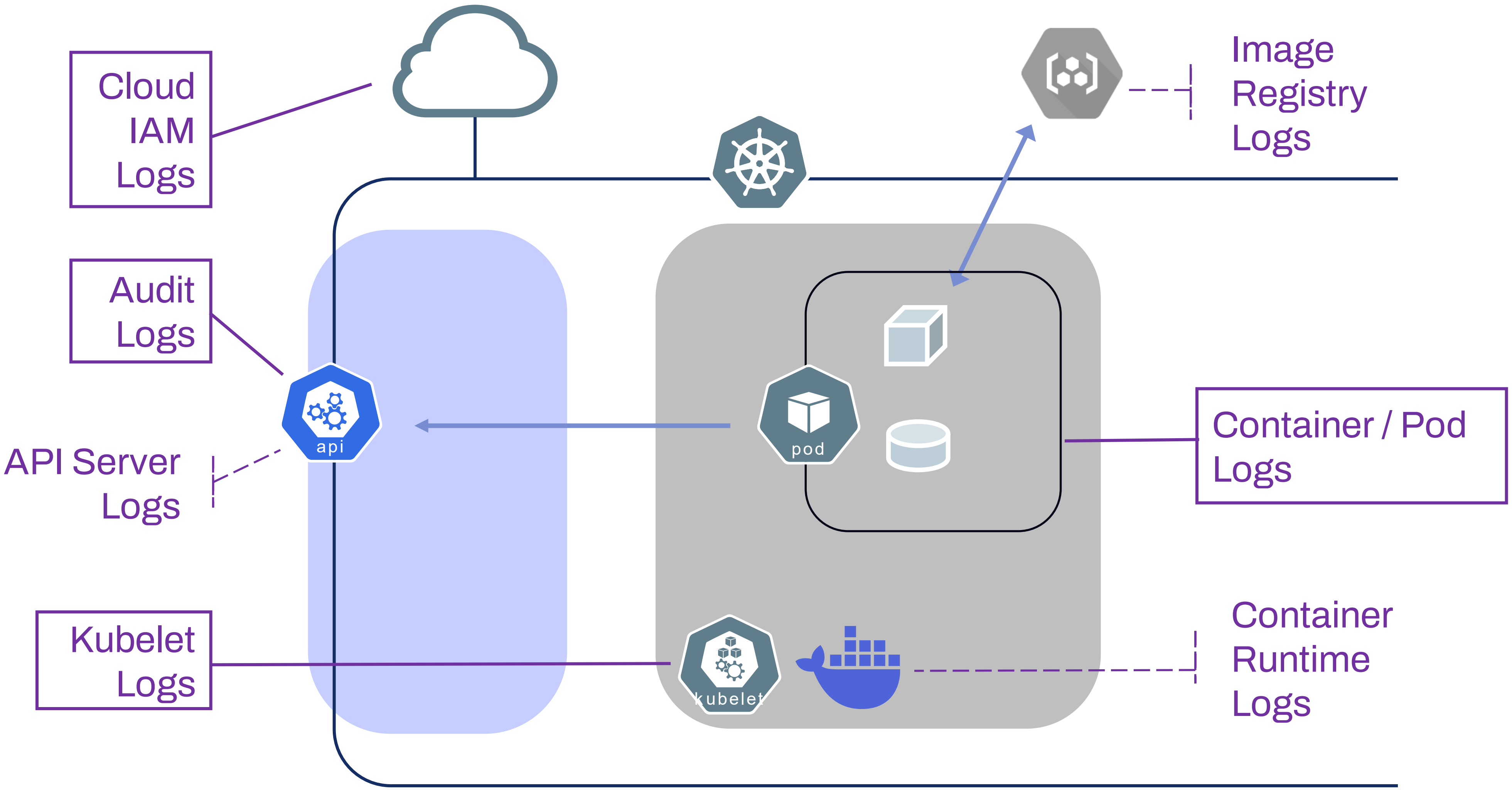
Kubelet Logs

Image Registry Logs

Container / Pod Logs

Container Runtime Logs

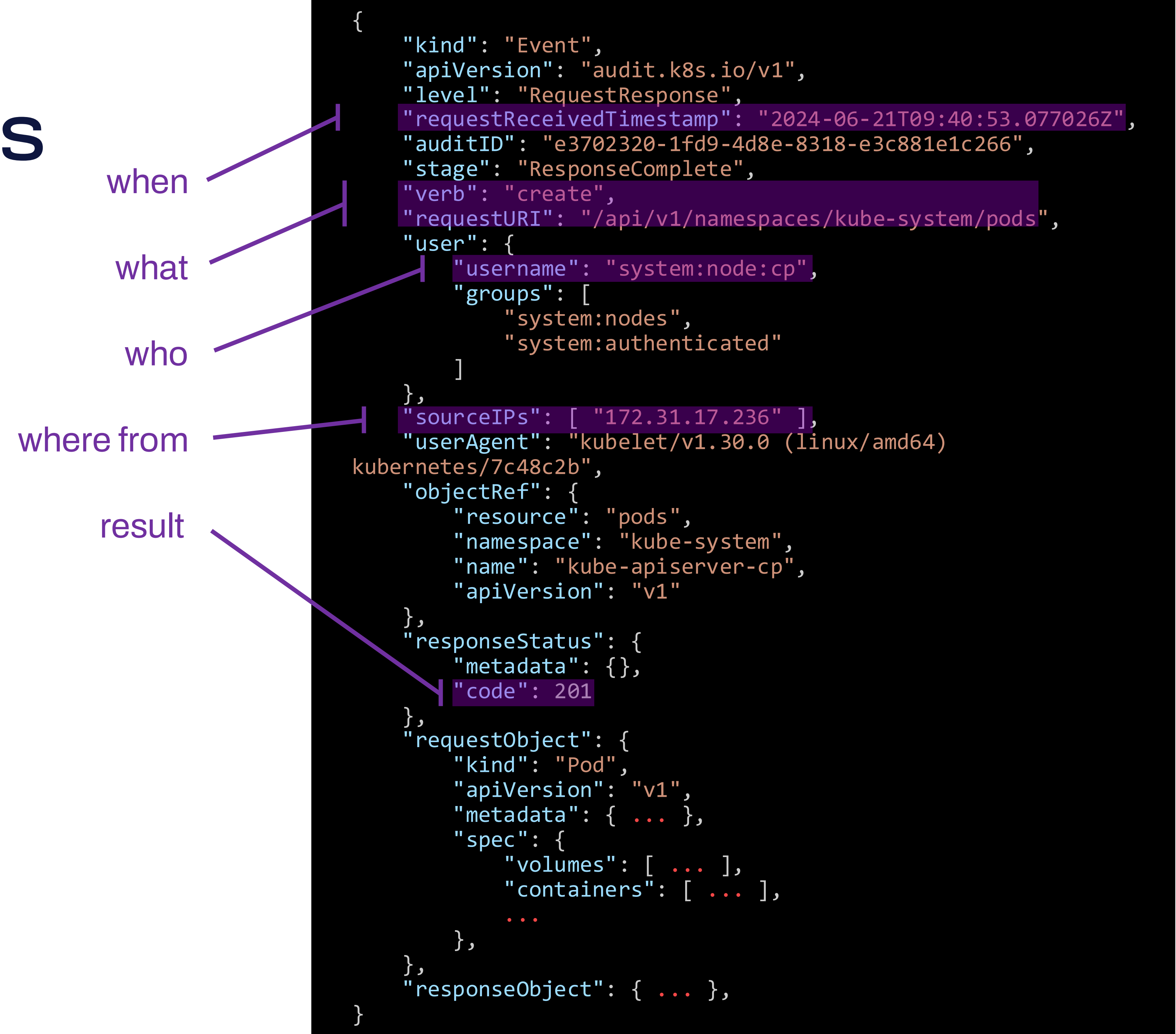
Security-relevant Logs



K8S Audit Logs

- “Access Logs” of API Server

- *Not enabled by default*
- *There are ways to evade them*



when

what

who

where from

result

```
{
    "kind": "Event",
    "apiVersion": "audit.k8s.io/v1",
    "level": "RequestResponse",
    "requestReceivedTimestamp": "2024-06-21T09:40:53.077026Z",
    "auditID": "e3702320-1fd9-4d8e-8318-e3c881e1c266",
    "stage": "ResponseComplete",
    "verb": "create",
    "requestURI": "/api/v1/namespaces/kube-system/pods",
    "user": {
        "username": "system:node:cp",
        "groups": [
            "system:nodes",
            "system:authenticated"
        ]
    },
    "sourceIPs": [ "172.31.17.236" ],
    "userAgent": "kubelet/v1.30.0 (linux/amd64)
kubernetes/7c48c2b",
    "objectRef": {
        "resource": "pods",
        "namespace": "kube-system",
        "name": "kube-apiserver-cp",
        "apiVersion": "v1"
    },
    "responseStatus": {
        "metadata": {},
        "code": 201
    },
    "requestObject": {
        "kind": "Pod",
        "apiVersion": "v1",
        "metadata": { ... },
        "spec": {
            "volumes": [ ... ],
            "containers": [ ... ],
            ...
        },
    },
    "responseObject": { ... },
}
```

K8S Audit Logs

- Audit Policy YAML

- Configurable **Verbosity** level - per Event
  1. None
  2. Metadata
  3. Request
  4. RequestResponse

- File / Webhook **Backend**
  - Agents like Filebeat can then pipe them into the SIEM

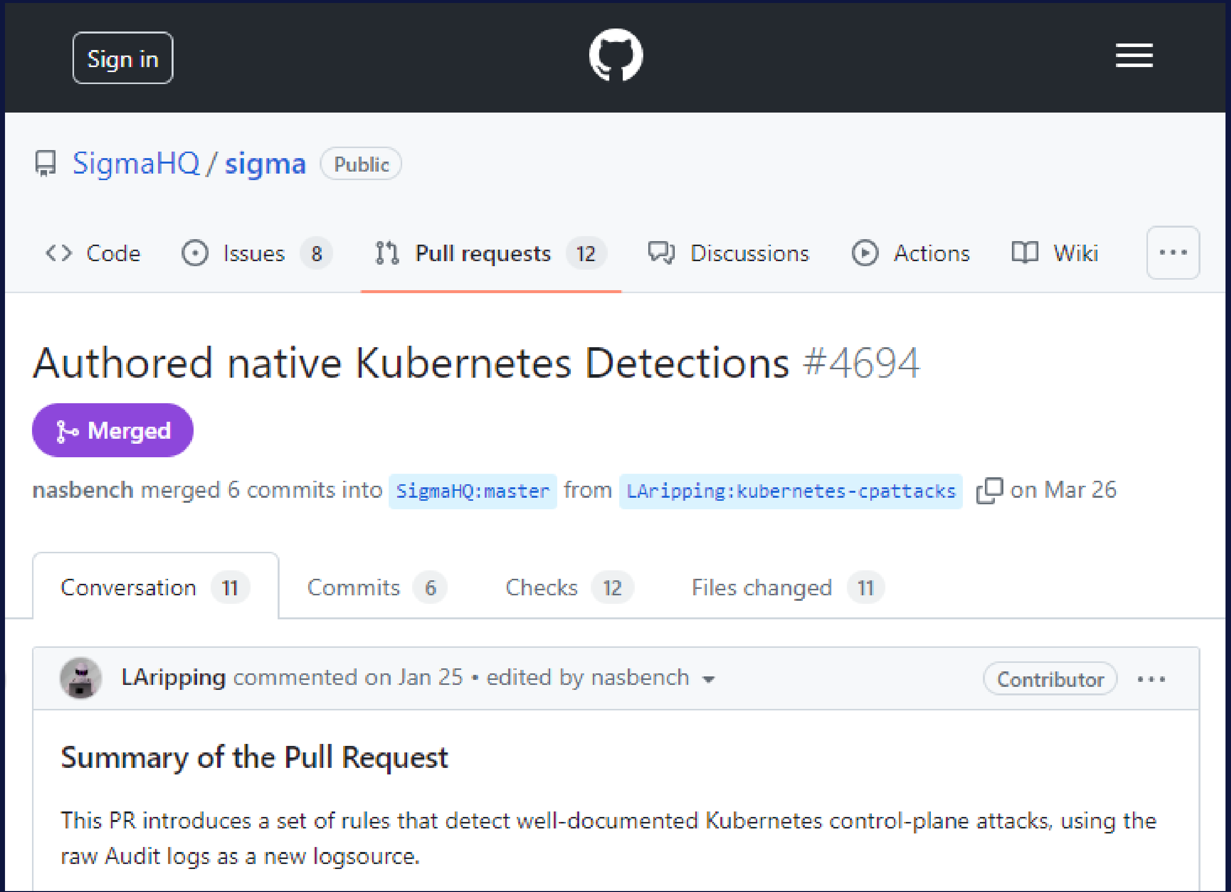
- Caveat: *Not customizable in managed clusters!*

```yaml
apiVersion: audit.k8s.io/v1
kind: Policy
rules:
  # Don't log these read-only URLs
  - level: None
    nonResourceURLs
      - /healthz*
      - /version
      - /swagger*

  # Secrets, ConfigMaps, TokenRequest and TokenReviews
can contain sensitive & binary data,
  - level: Request
    resources:
      - group: ""
        resources: ["secrets", "configmaps",
"serviceaccounts/token"]
      - group: authentication.k8s.io
        resources: ["tokenreviews"]
    omitStages:
      - "RequestReceived"

  ...

  # Default level for all other requests.
  - level: Metadata
    omitStages:
      - "RequestReceived"
```

https://github.com/kubernetes/kubernetes/blob/master/cluster/gce/gci/configure-helper.sh

K8S Detection Engineering: Control Plane

```
title: Kubernetes Secrets Enumeration
description: Detects enumeration of Kubernetes secrets.
tags:
    - attack.t1552.007
logsource:
    category: application
    product: kubernetes
    service: audit
detection:
    level: low
    condition: selection
    selection:
        verb: 'list'
        objectRef.resource: 'secrets’
```



Authored native Kubernetes Detections #4694

Merged

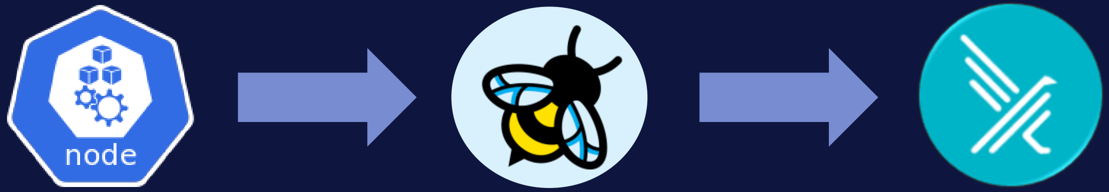
nasbench merged 6 commits into SigmaHQ:master from LAripping:kubernetes-cpattacks on Mar 26

LAripping commented on Jan 25 • edited by nasbench

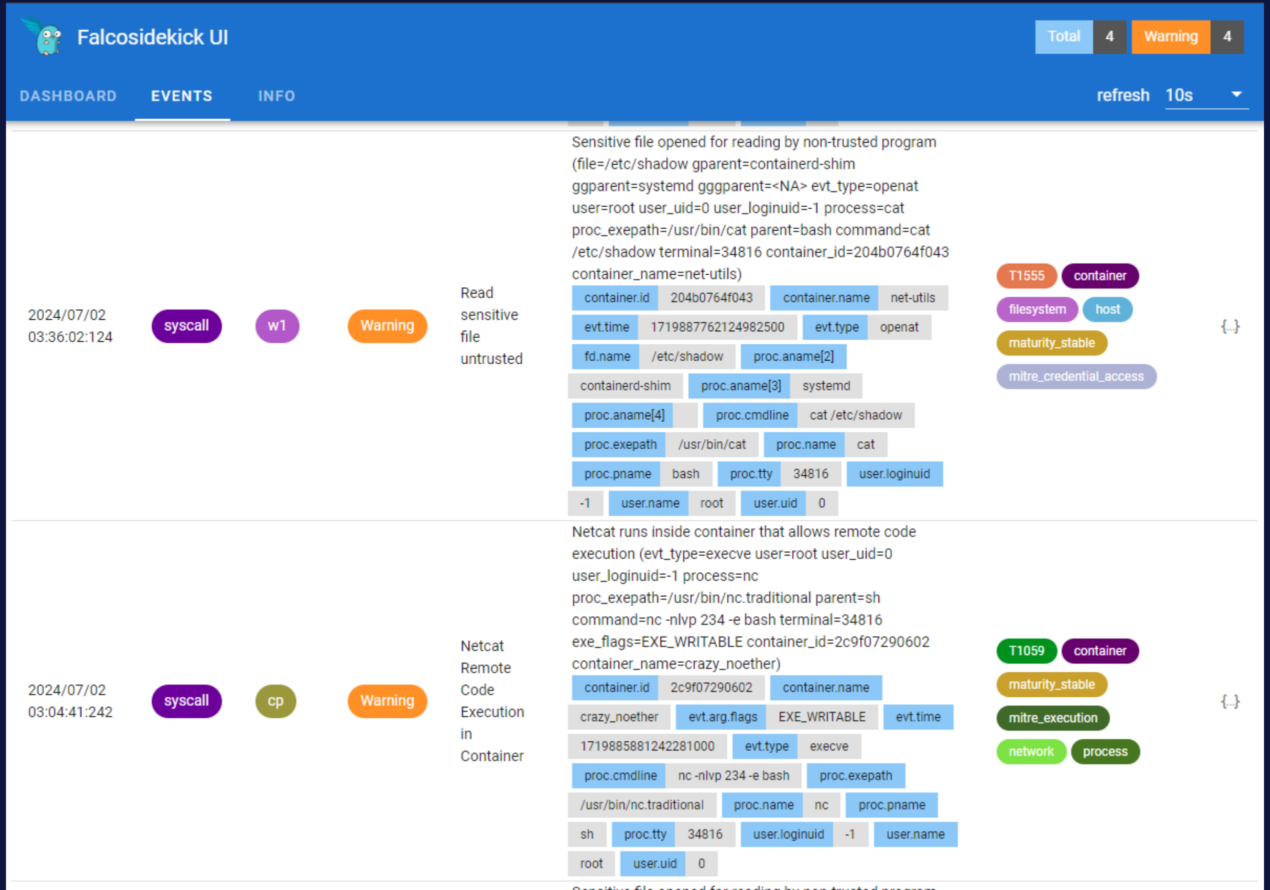
Summary of the Pull Request

This PR introduces a set of rules that detect well-documented Kubernetes control-plane attacks, using the raw Audit logs as a new logsource.

K8S Detection Engineering: Kernel Level



```yaml
- rule: Netcat Remote Code Execution in Container
  desc: >
    Netcat Program runs inside container that allows remote
    code execution and may be utilized as a part of a variety
    of reverse shell payloads
    https://github.com/swisskyrepo/PayloadsAllTheThings/.
  condition: >
    spawned_process
    and container
    and ((proc.name = "nc" and (proc.cmdline contains " -e"))
  priority: WARNING
  tags: [T1059, mitre_execution, container, network, process]
```



2024/07/02 03:36:02:124 syscall w1 Warning Read sensitive file untrusted

Sensitive file opened for reading by non-trusted program (file=/etc/shadow gparent=containerd-shim ggparent=systemd gggparent=<NA> evt_type=openat user=root user_uid=0 user_loginuid=-1 process=cat proc_exepath=/usr/bin/cat parent=bash command=cat /etc/shadow terminal=34816 container_id=204b0764f043 container_name=net-utils)

container.id 204b0764f043 container.name net-utils evt.time 1719887762124982500 evt.type openat fd.name /etc/shadow proc.aname[2] containerd-shim proc.aname[3] systemd proc.aname[4] proc.cmdline cat /etc/shadow proc.exepath /usr/bin/cat proc.name cat proc.pname bash proc.tty 34816 user.loginuid -1 user.name root user.uid 0

T1555 container filesystem host maturity_stable mitre_credential_access

{..}

2024/07/02 03:04:41:242 syscall cp Warning Netcat Remote Code Execution in Container

Netcat runs inside container that allows remote code execution (evt_type=execve user=root user_uid=0 user_loginuid=-1 process=nc proc_exepath=/usr/bin/nc.traditional parent=sh command=nc -nlvp 234 -e bash terminal=34816 exe_flags=EXE_WRITABLE container_id=2c9f07290602 container_name=crazy_noether)

container.id 2c9f07290602 container.name crazy_noether evt.arg.flags EXE_WRITABLE evt.time 1719885881242281000 evt.type execve proc.cmdline nc -nlvp 234 -e bash proc.exepath /usr/bin/nc.traditional proc.name nc proc.pname sh proc.tty 34816 user.loginuid -1 user.name root user.uid 0

T1059 container maturity_stable mitre_execution network process

{..}

Sensitive file opened for reading by non-trusted program

Kubernetes
Attack Simulation

Kubernetes
Attack Detection

Demo

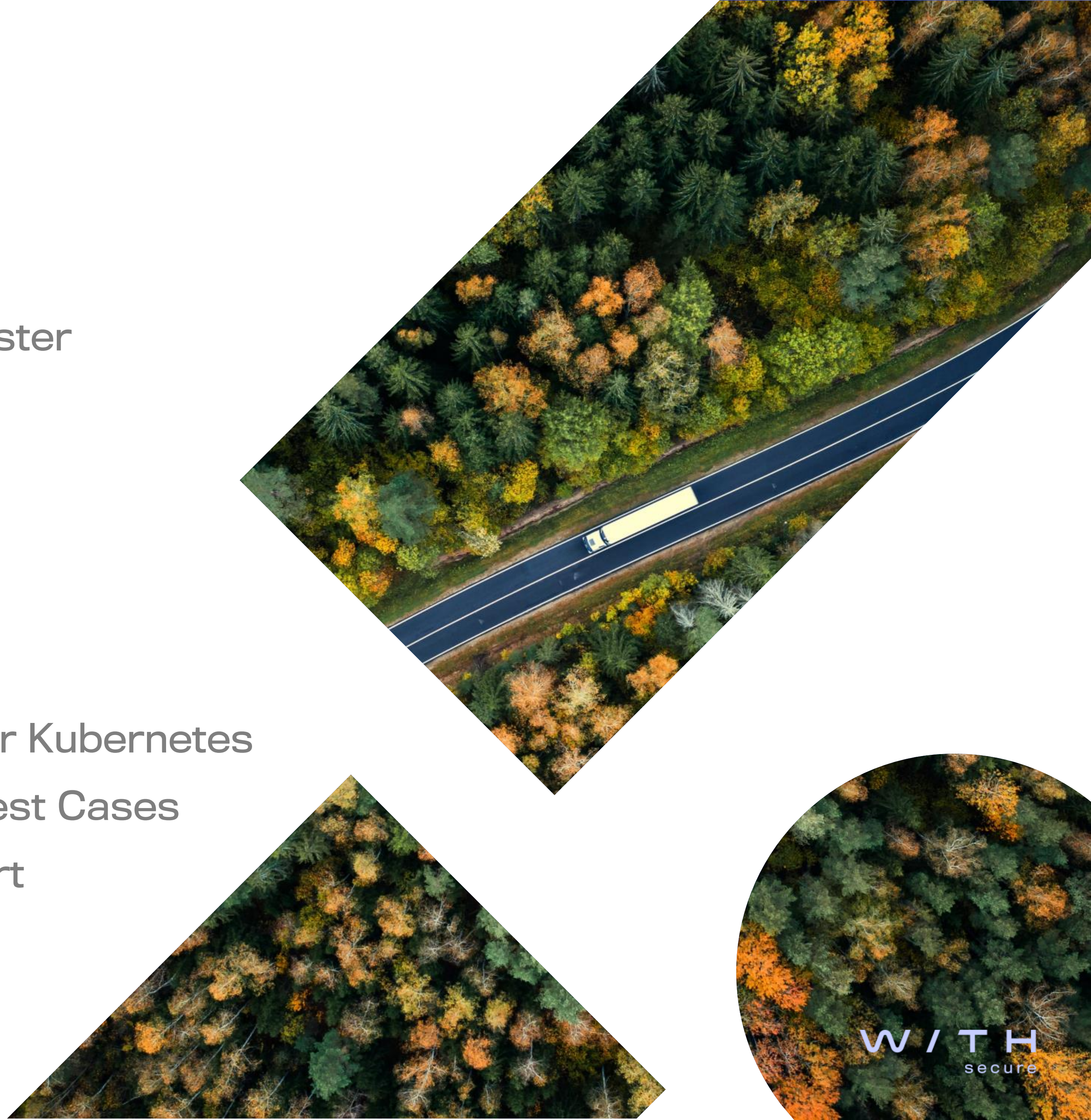
K8S Attack Simulation with Leonidas

Takeaways

- Understand the threats to your cluster
- Simulate adversaries proactively
- Build defences collaboratively

Contributions

- **Simulation Framework** Leonidas for Kubernetes
- **Attack Definitions** 17 Kubernetes Test Cases
- **Detection Signatures** Sigma support



Thank You

@LAripping

@WithSecure

References