KubeCon | CloudNativeCon

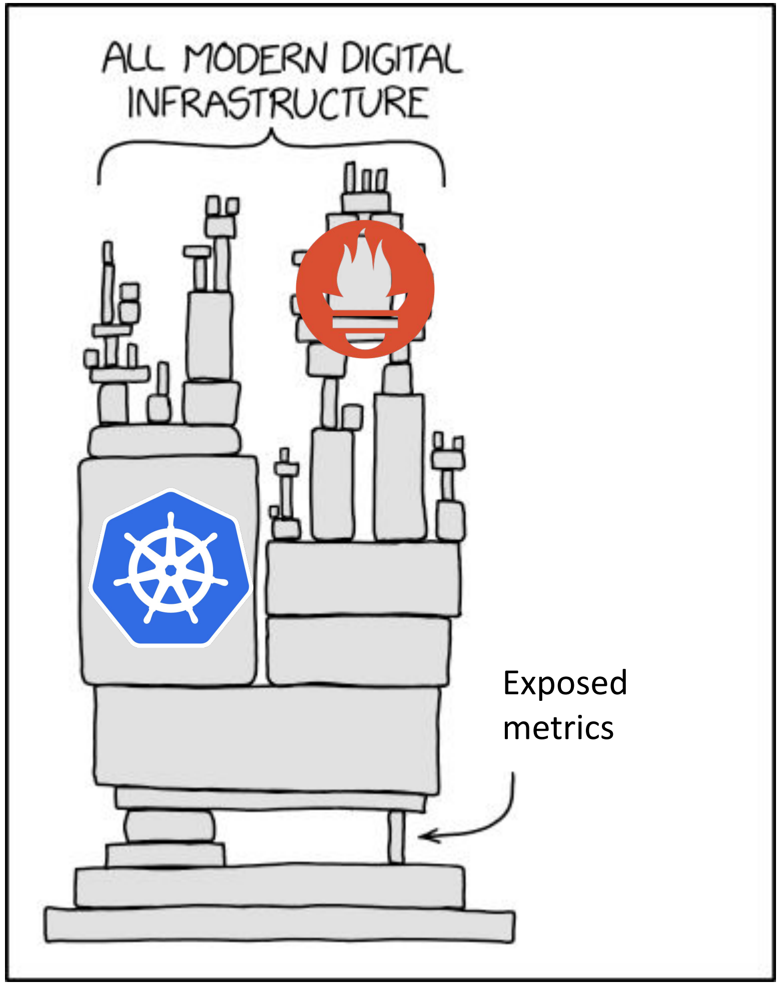
Europe 2022

WELCOME TO VALENCIA

How Attackers Use Exposed Prometheus Server to Exploit Kubernetes Clusters

Miguel Hernández & David de Torres, Sysdig

Kubernetes fingerprinting with Prometheus



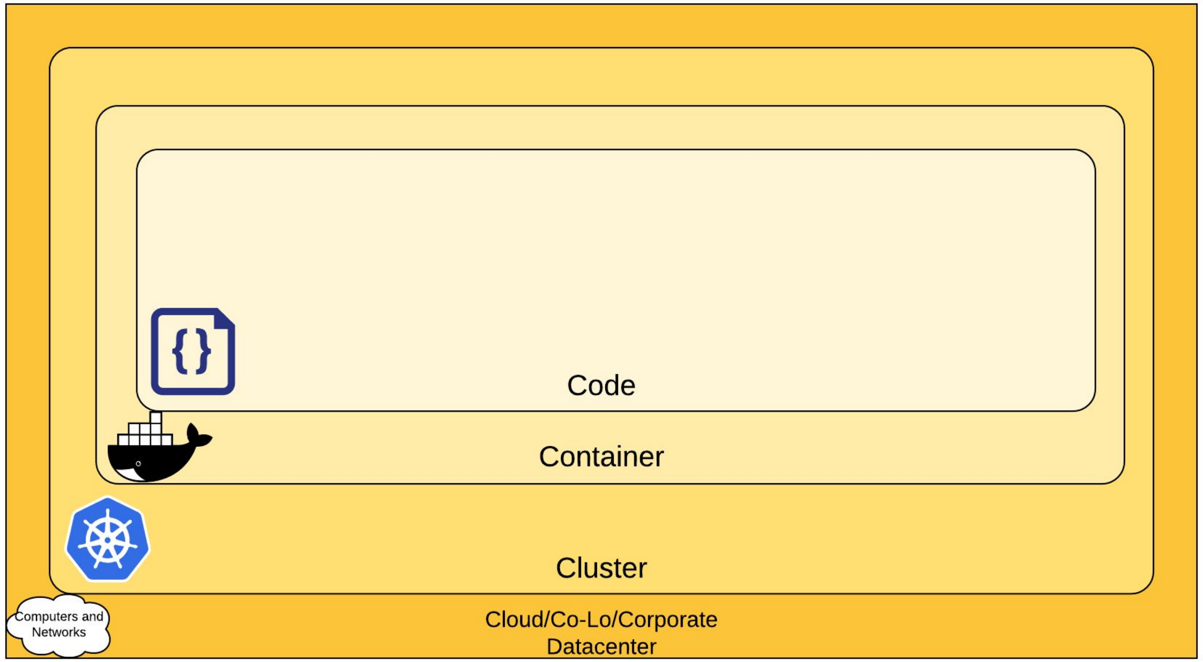
https://xkcd.com/2347/

Miguel Hernandez
Security Researcher
Sysdig
@MiguelHzBz

David de Torres
Manager of Engineering
Sysdig
@maellyssa

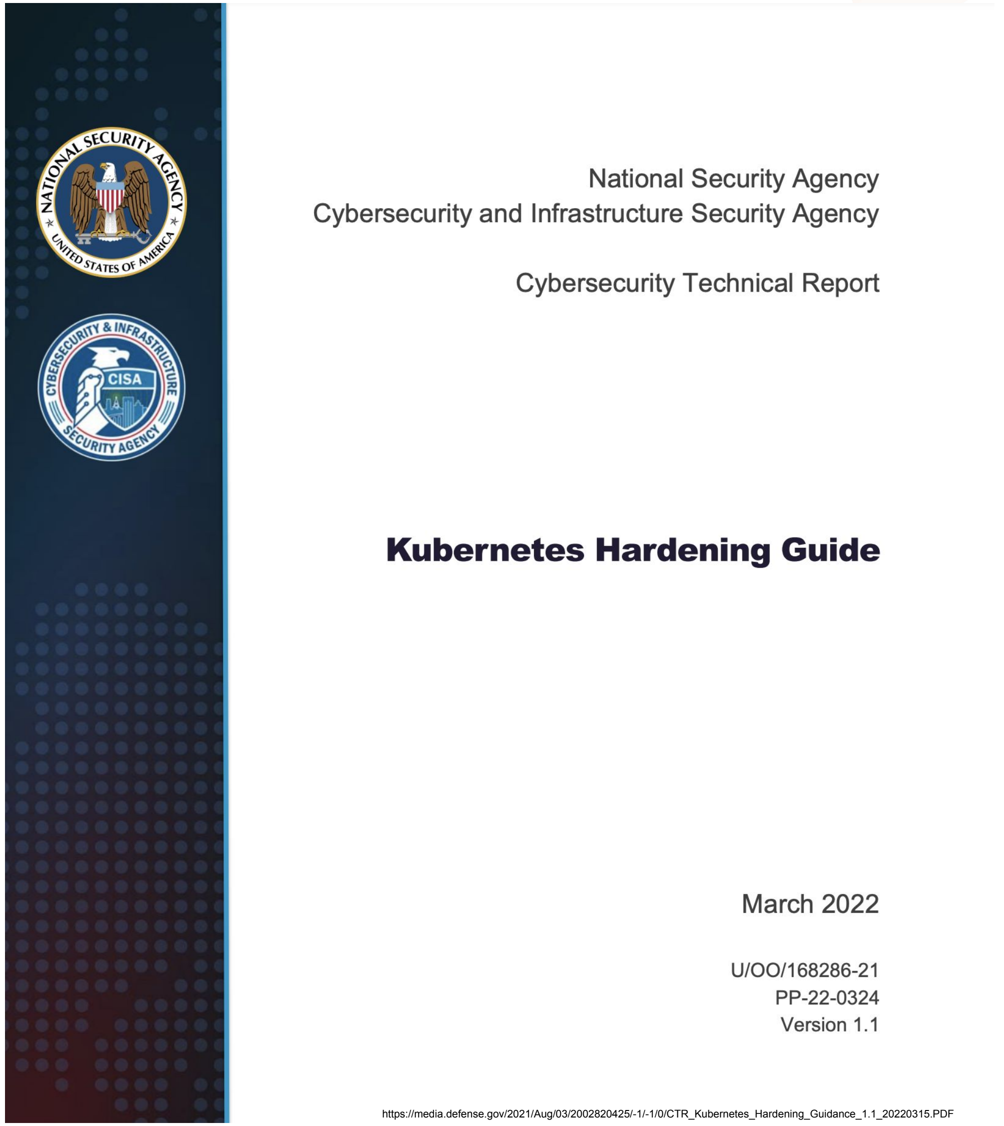
Assume you are a target, but not for free

- Follow the Kubernetes security best practices.
- Use Prometheus to monitor everything.
- But don't let the door open.



DISCLAIMER

We are not going to break and break into Kubernetes Cluster or Prometheus.



National Security Agency
Cybersecurity and Infrastructure Security Agency

Cybersecurity Technical Report

Kubernetes Hardening Guide

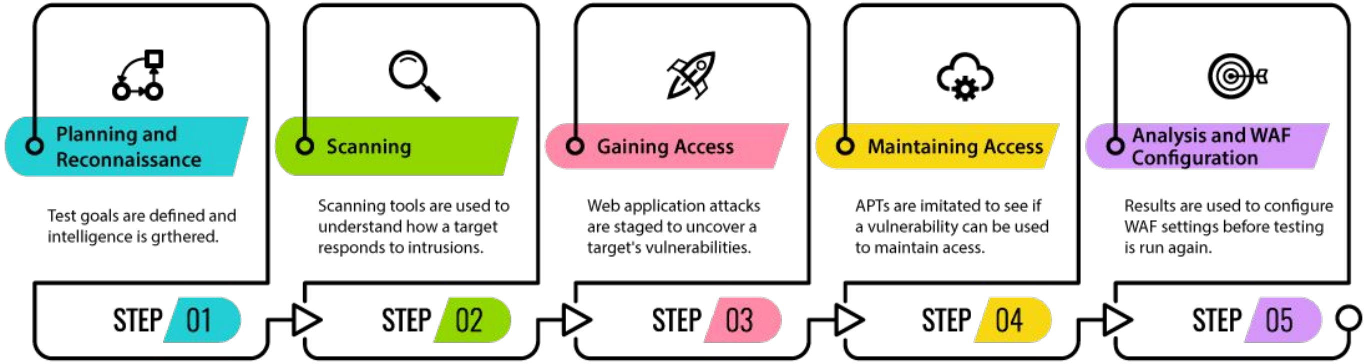
March 2022

U/OO/168286-21
PP-22-0324
Version 1.1

https://media.defense.gov/2021/Aug/03/2002820425/-1/-1/0/CTR_Kubernetes_Hardening_Guidance_1.1_20220315.PDF

Why Kubernetes fingerprinting?

The **first step** in any pentesting, ethical hacking or cybercriminal groups, is to **gather as much information as you can about the target** you want to breach.



Planning and Reconnaissance

Test goals are defined and intelligence is grthered.

STEP 01

Scanning

Scanning tools are used to understand how a target responds to intrusions.

STEP 02

Gaining Access

Web application attacks are staged to uncover a target's vulnerabilities.

STEP 03

Maintaining Access

APTs are imitated to see if a vulnerability can be used to maintain acess.

STEP 04

Analysis and WAF Configuration

Results are used to configure WAF settings before testing is run again.

STEP 05

https://www.eccouncil.org/ethical-hacking/

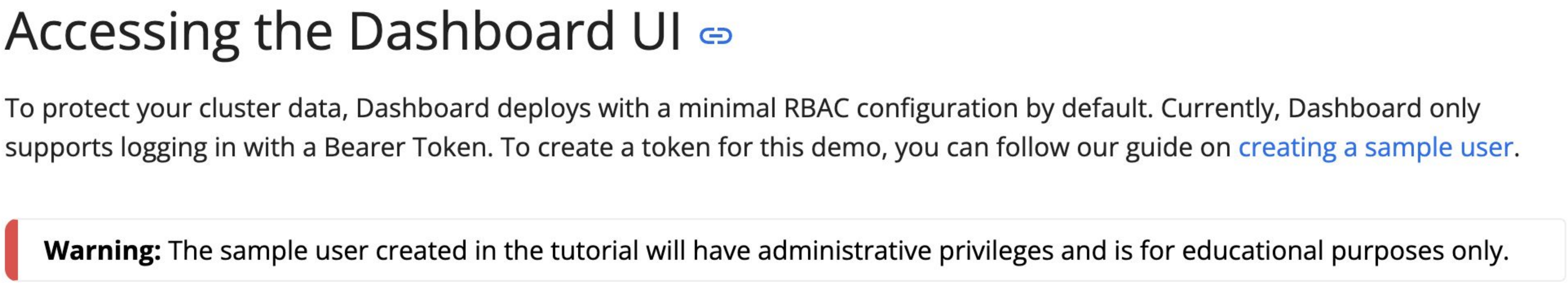
Why? Simple, to know **what technique** to use or the **appropriate tools** to achieve intrusion and evasion of defense systems.

Information on versions inside the cluster can map to CVE and vulnerabilities that can be exploited.

Information on applications, tools and architectures can be used for competitors.

Kubernetes in the wild

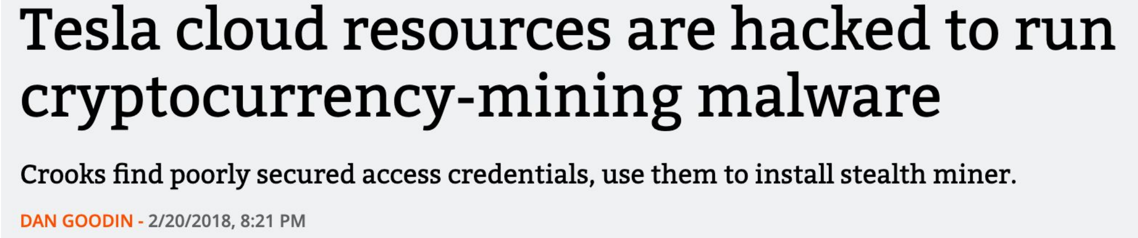
https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/



Accessing the Dashboard UI

To protect your cluster data, Dashboard deploys with a minimal RBAC configuration by default. Currently, Dashboard only supports logging in with a Bearer Token. To create a token for this demo, you can follow our guide on creating a sample user.

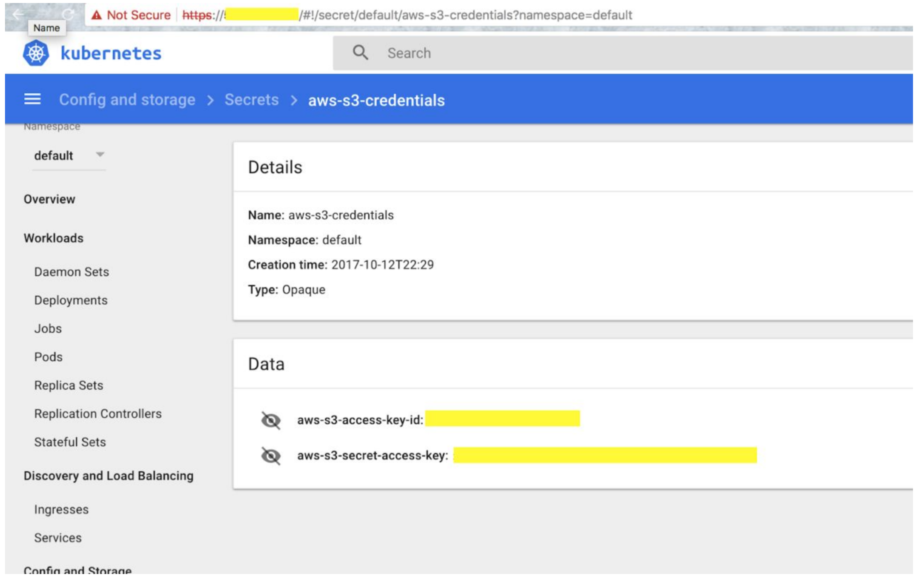
Warning: The sample user created in the tutorial will have administrative privileges and is for educational purposes only.



Tesla cloud resources are hacked to run cryptocurrency-mining malware

Crooks find poorly secured access credentials, use them to install stealth miner.

DAN GOODIN - 2/20/2018, 8:21 PM



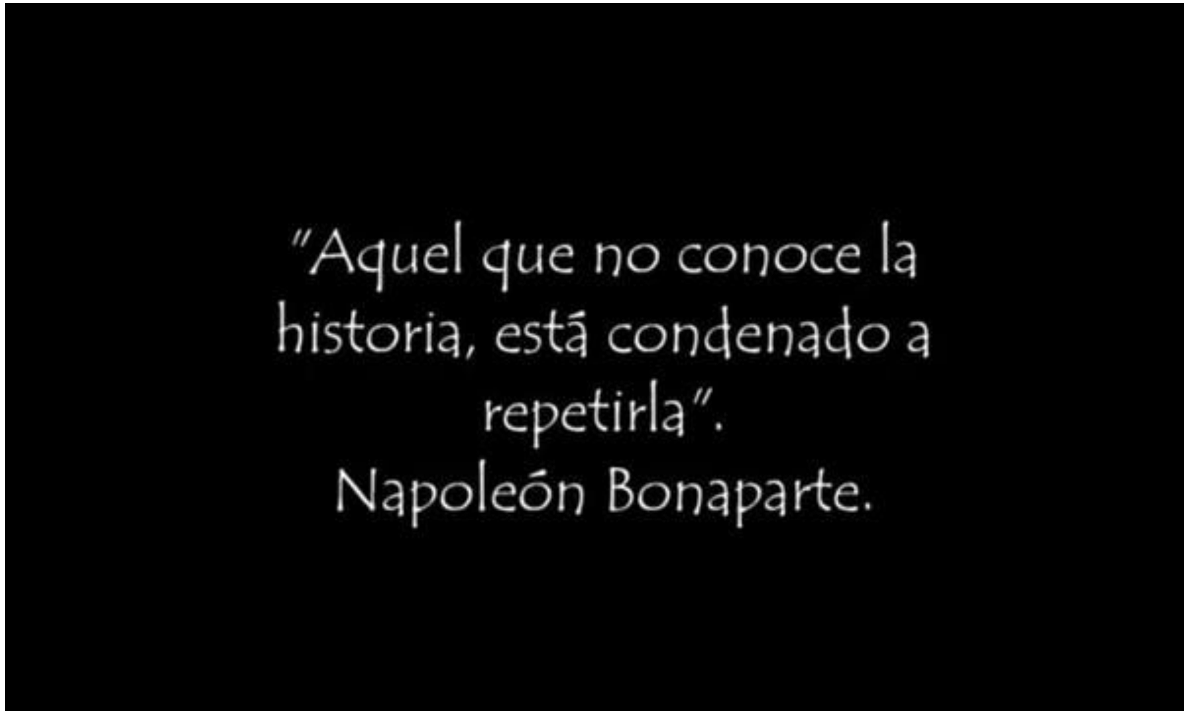
⚠ Not Secure | ~~https~~://[redacted]/#!/secret/default/aws-s3-credentials?namespace=default

Details

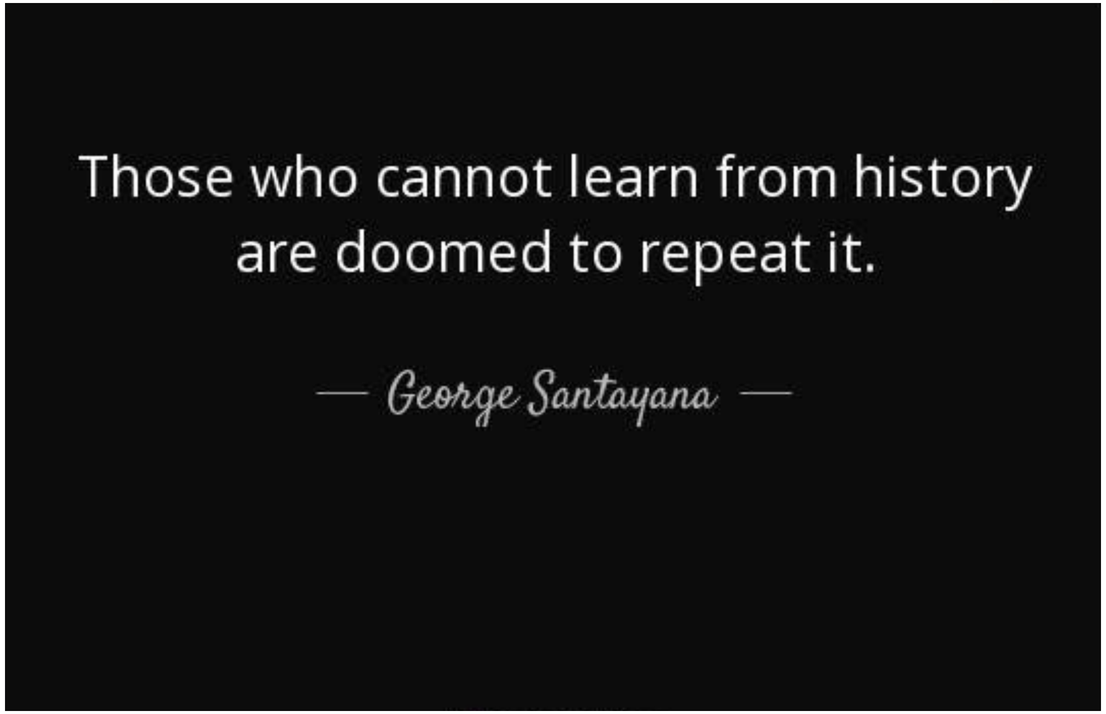
Name: aws-s3-credentials
Namespace: default
Creation time: 2017-10-12T22:29
Type: Opaque

Data

aws-s3-access-key-id: [redacted]
aws-s3-secret-access-key: [redacted]



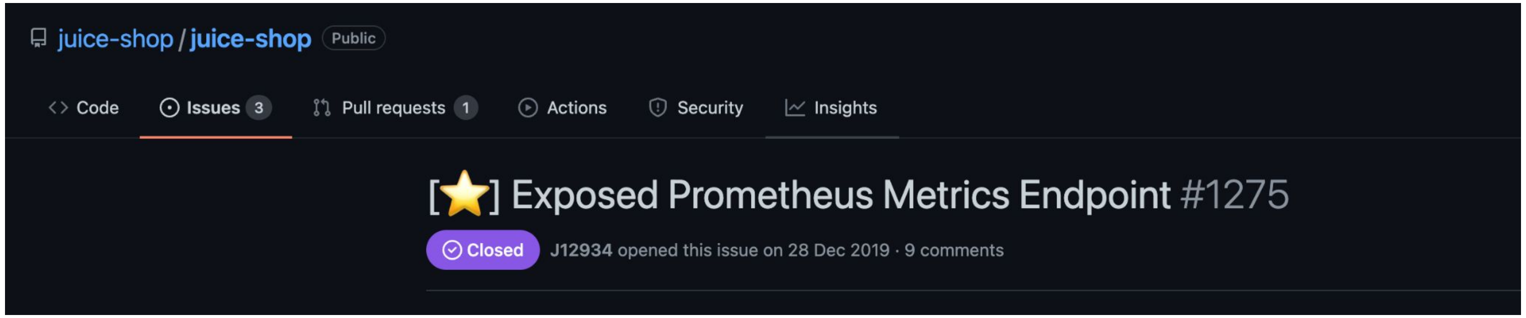
"Aquel que no conoce la historia, está condenado a repetirla".
Napoleón Bonaparte.



Those who cannot learn from history are doomed to repeat it.

— George Santayana —

But Prometheus is only metrics…

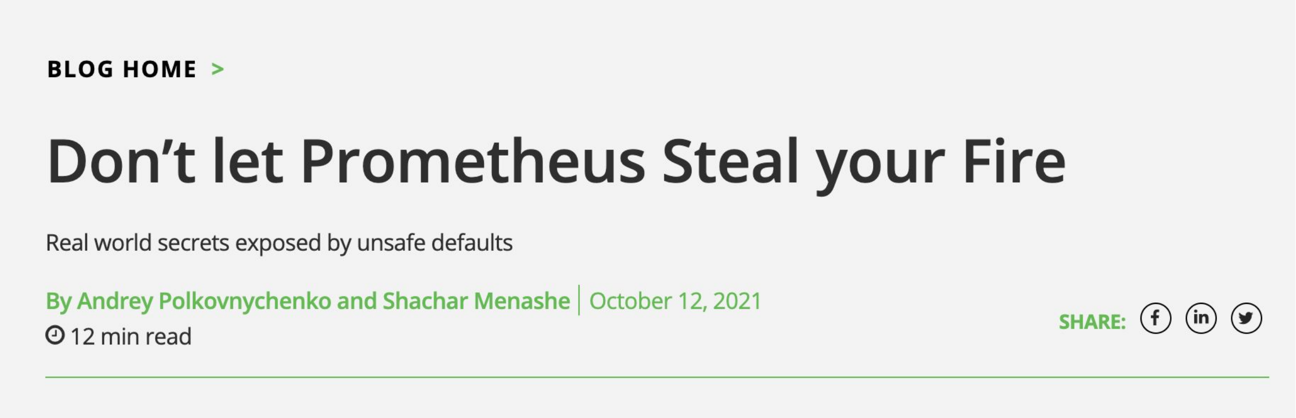


juice-shop / juice-shop Public

[⭐] Exposed Prometheus Metrics Endpoint #1275

Closed J12934 opened this issue on 28 Dec 2019 · 9 comments

https://github.com/juice-shop/juice-shop/issues/1275



BLOG HOME >

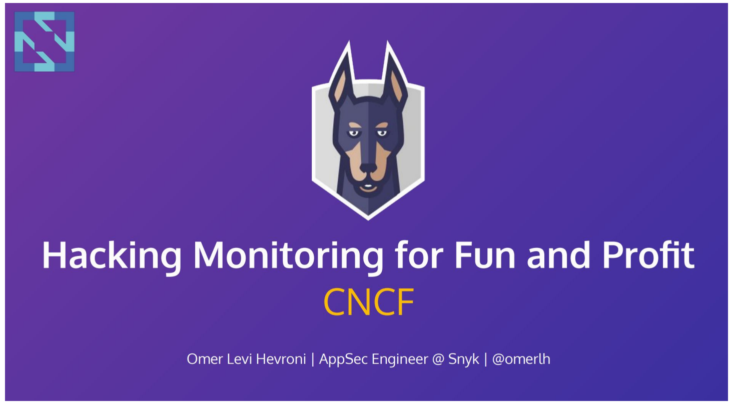
Don’t let Prometheus Steal your Fire

Real world secrets exposed by unsafe defaults

By Andrey Polkovnychenko and Shachar Menashe | October 12, 2021

12 min read

https://jfrog.com/blog/dont-let-prometheus-steal-your-fire/



Hacking Monitoring for Fun and Profit

CNCF

Omer Levi Hevroni | AppSec Engineer @ Snyk | @omerlh

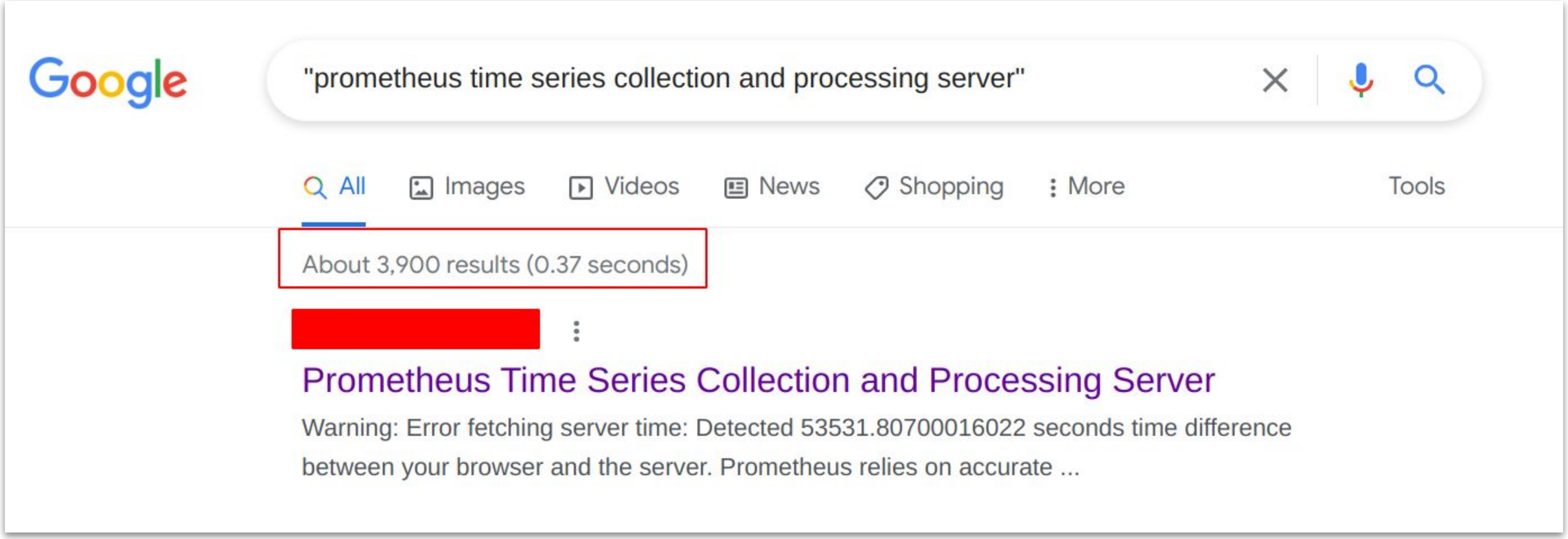
https://www.cncf.io/online-programs/a-look-at-how-hackers-exploit-prometheus-grafana-fluentd-jaeger-more/

Prometheus in the wild

Prometheus collects and stores its metrics as time series data, i.e. metrics information is stored with the timestamp at which it was recorded, alongside optional key-value pairs called labels.

Prometheus allows (and recommends) using basic authentication, but **not enabled by default**: https://prometheus.io/docs/operating/security/

Exposing open Prometheus endpoints to the Internet is a bad idea… and **as every bad idea, it's highly adopted**:



"prometheus time series collection and processing server"

About 3,900 results (0.37 seconds)

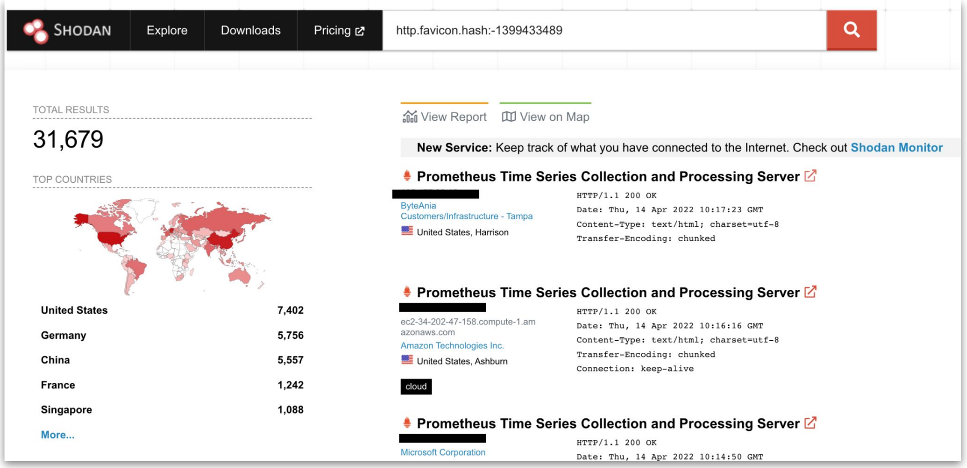
[redacted]

Prometheus Time Series Collection and Processing Server

Warning: Error fetching server time: Detected 53531.80700016022 seconds time difference between your browser and the server. Prometheus relies on accurate ...

More Prometheus in the wild

Shodan -> favicons (https://github.com/sansatart/scrapts/blob/master/shodan-favicon-hashes.csv)



http.favicon.hash:-1399433489

TOTAL RESULTS

31,679

TOP COUNTRIES

United States	7,402
Germany	5,756
China	5,557
France	1,242
Singapore	1,088

More...

View Report   View on Map

New Service: Keep track of what you have connected to the Internet. Check out **Shodan Monitor**

Prometheus Time Series Collection and Processing Server

[redacted]
ByteAnia
Customers/Infrastructure - Tampa
United States, Harrison

```
HTTP/1.1 200 OK
Date: Thu, 14 Apr 2022 10:17:23 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
```

Prometheus Time Series Collection and Processing Server

[redacted]
ec2-34-202-47-158.compute-1.amazonaws.com
Amazon Technologies Inc.
United States, Ashburn
cloud

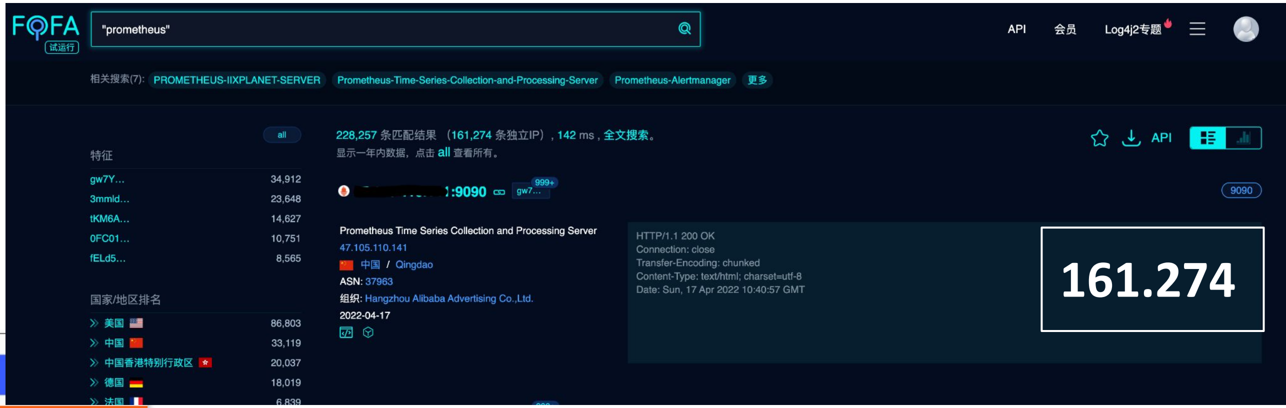
```
HTTP/1.1 200 OK
Date: Thu, 14 Apr 2022 10:16:16 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
```

Prometheus Time Series Collection and Processing Server

[redacted]
Microsoft Corporation

```
HTTP/1.1 200 OK
Date: Thu, 14 Apr 2022 10:14:50 GMT
```

Fofa (https://fofa.info/)



"prometheus"

相关搜索(7): PROMETHEUS-IIXPLANET-SERVER Prometheus-Time-Series-Collection-and-Processing-Server Prometheus-Alertmanager 更多

228,257 条匹配结果 （161,274 条独立IP），142 ms，全文搜索。
显示一年内数据，点击 all 查看所有。

特征

gw7Y...	34,912
3mmld...	23,648
tKM6A...	14,627
0FC01...	10,751
fELd5...	8,565

国家/地区排名

美国	86,803
中国	33,119
中国香港特别行政区	20,037
德国	18,019
法国	6,839

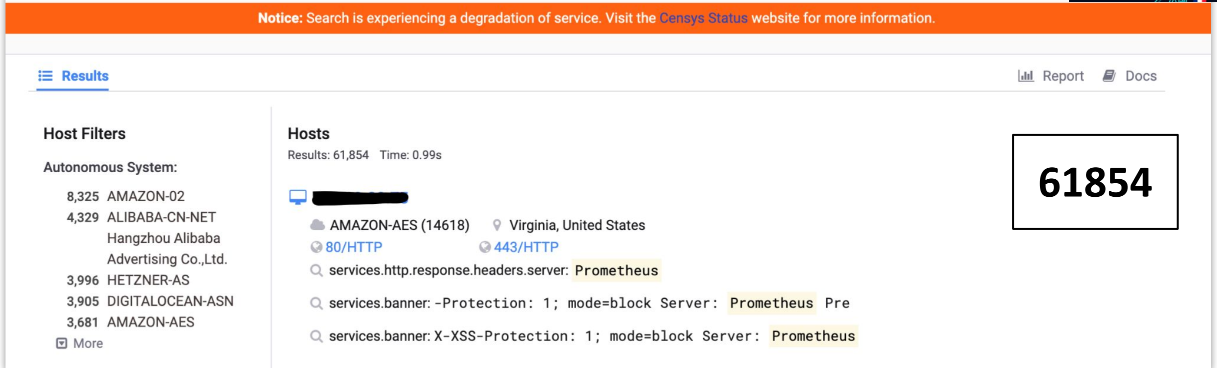
[redacted]:9090 gw7... 999+

Prometheus Time Series Collection and Processing Server
47.105.110.141
中国 / Qingdao
ASN: 37963
组织: Hangzhou Alibaba Advertising Co.,Ltd.
2022-04-17

```
HTTP/1.1 200 OK
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=utf-8
Date: Sun, 17 Apr 2022 10:40:57 GMT
```

161.274

Censys (https://search.censys.io/)



prometheus

Notice: Search is experiencing a degradation of service. Visit the Censys Status website for more information.

Results   Report   Docs

Host Filters

Autonomous System:

8,325	AMAZON-02
4,329	ALIBABA-CN-NET Hangzhou Alibaba Advertising Co.,Ltd.
3,996	HETZNER-AS
3,905	DIGITALOCEAN-ASN
3,681	AMAZON-AES

More

Hosts

Results: 61,854   Time: 0.99s

[redacted]
AMAZON-AES (14618)   Virginia, United States
80/HTTP   443/HTTP
services.http.response.headers.server: Prometheus
services.banner: -Protection: 1; mode=block Server: Prometheus Pre
services.banner: X-XSS-Protection: 1; mode=block Server: Prometheus

61854

What will we us to fingerprint Kubernetes?

Two of the most widely used exporters offer most of the information that we need:

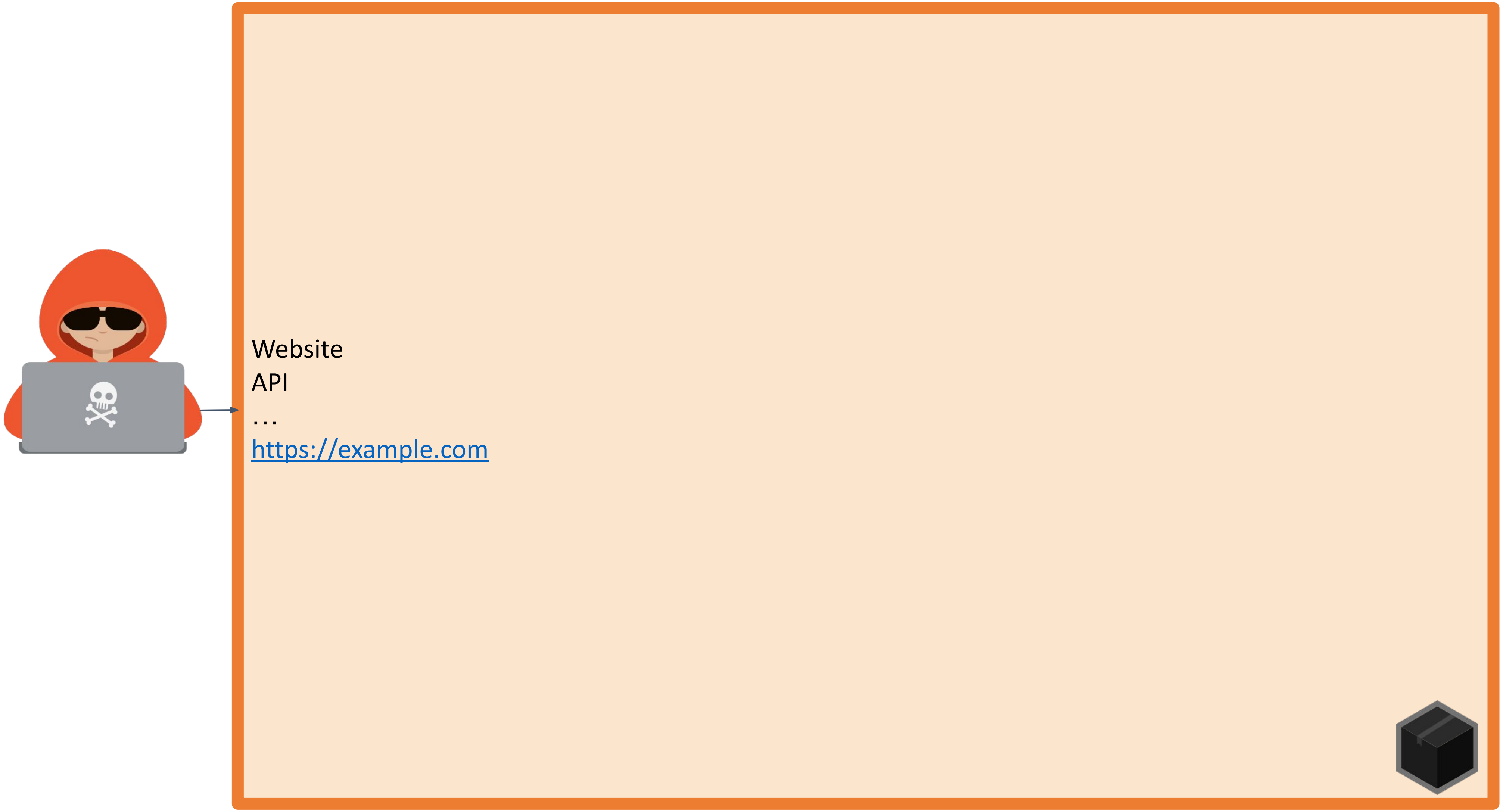
Node Exporter

- Physical infrastructure
- Network interfaces

Kube State Metrics

- Host OS & kernel
- Kubernetes components
- Hostnames and network topology
- Logical hierarchy
- Secrets location
- Applications (and versions) deployed

Scenario - NotARealCompanyForSure ©



Website
API
…
https://example.com

Scenario - NotARealCompanyForSure ©

Website
API
…
https://example.com

Fingerprinting Physical Infrastructure

Node Exporter:
`node_dmi_info`

bios_vendor:
- SeaBIOS
- Amazon EC2

bios_version:
- seabios-1.9.1-qemu-project.org
- 8f19b21
- 1.0

bios_release:
- 1.0

bios_date:
- 10/16/2017
- 04/01/2014

chassis_asset_tag:
- Amazon EC2

chassis_vendor:
- Amazon EC2
- Alibaba Cloud

system_vendor:
- Tencent Cloud
- Amazon EC2
- Alibaba Cloud

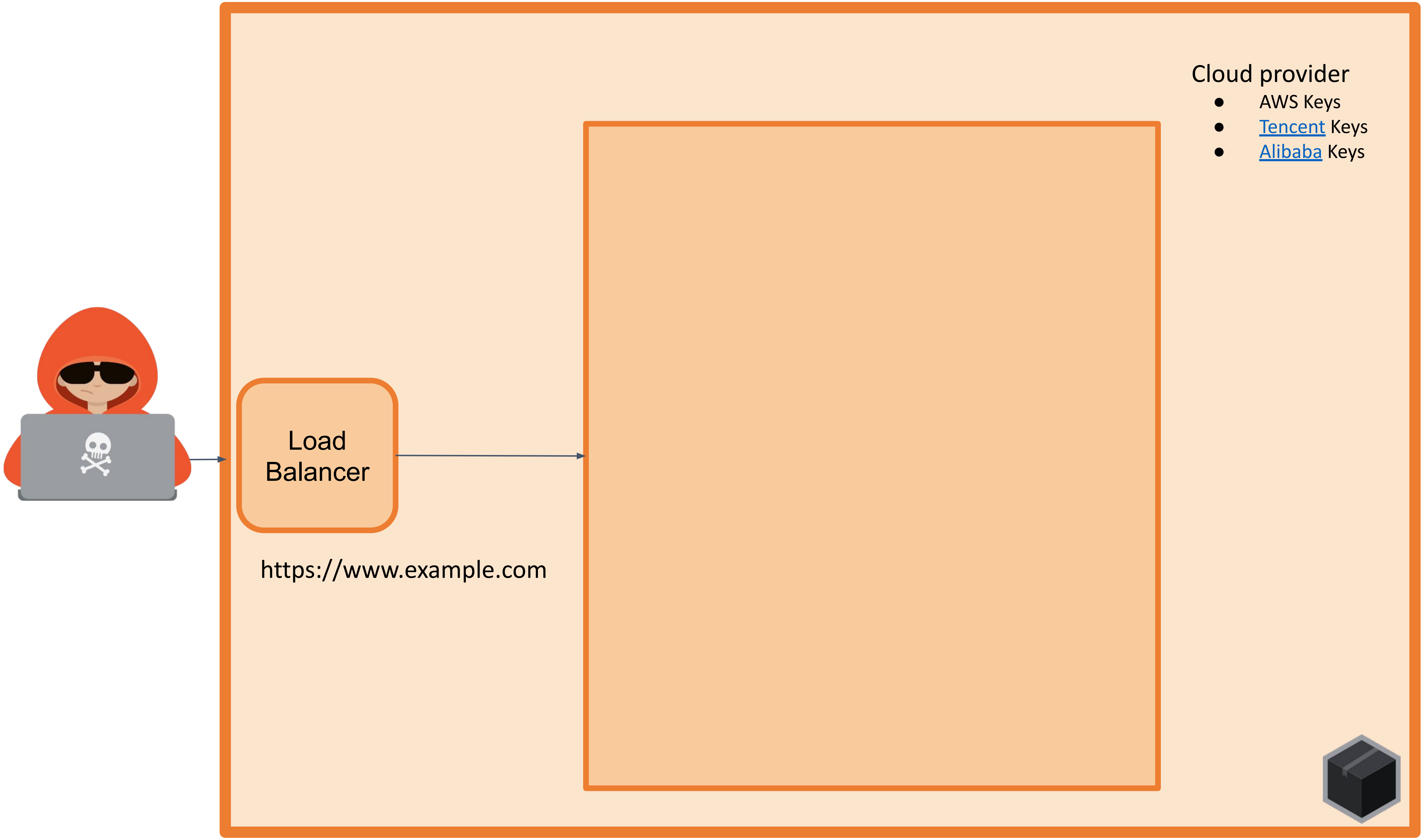
product_name:
- m5.xlarge
- Alibaba Cloud ECS

product_version:
- pc-i440fx-2.1

board_vendor:
- Amazon EC2

board_asset_tag:
- i-00280f617XXXXX

board_vendor:
- Smdbmds
- Amazon EC2



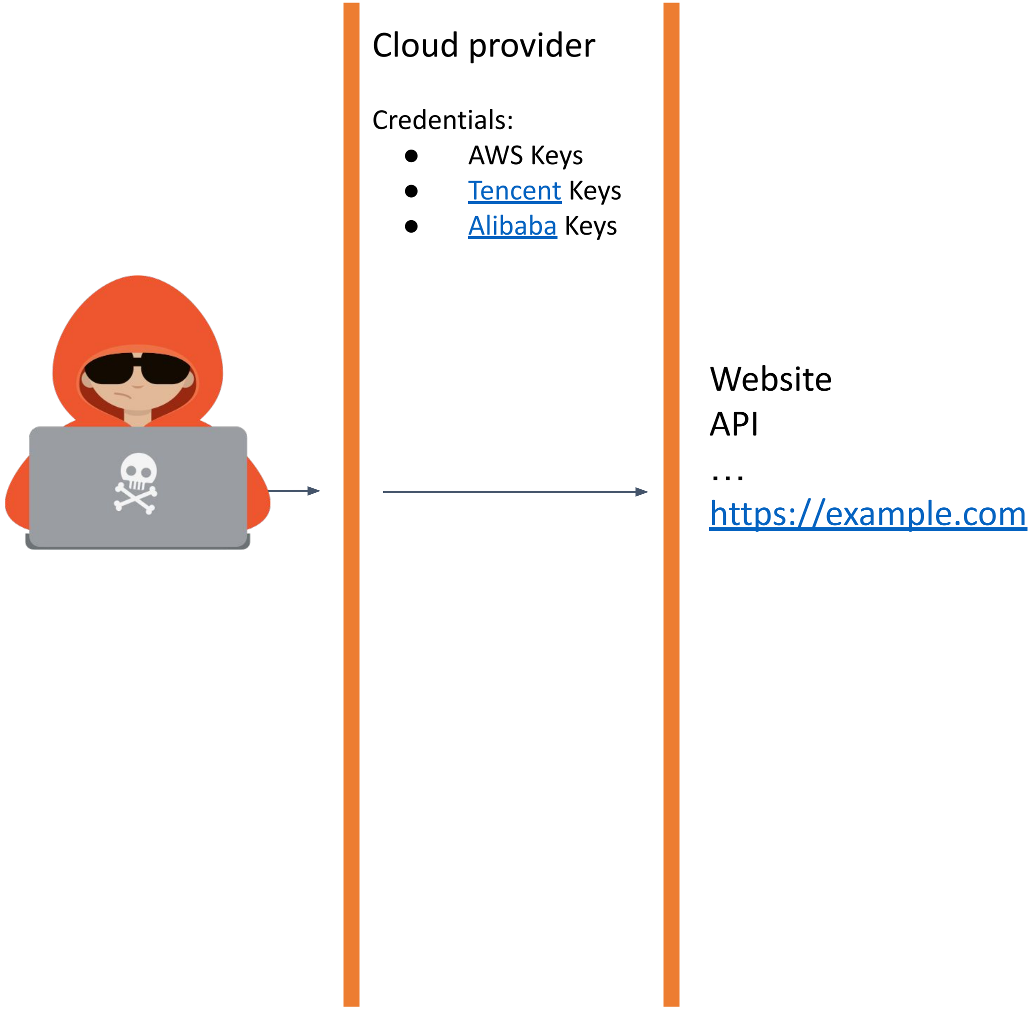
Cloud provider

- AWS Keys
- Tencent Keys
- Alibaba Keys

Load Balancer

https://www.example.com

Scenario - NotARealCompanyForSure ©



Cloud provider

Credentials:

- AWS Keys
- Tencent Keys
- Alibaba Keys

Website
API
…
https://example.com

Fingerprinting network interfaces

Node Exporter:

```
    node_network_info{device=~'eth.+'}
```

```
{
    address="06:d5:XX:XX:XX:XX",
    broadcast="ff:ff:ff:ff:ff:ff",
    device="eth0",
    instance="172.31.XX.XX:9100",
    instance_az="us-west-2a",
    instance_id="i-XXXXX",
    instance_name="XXX-XXX",
    instance_type="c5.xlarge",
    instance_vpc="vpc-XXXXXXX",
    job="ec2_instances",
    operstate="up"
}
```

Fingerprinting network topology

KSM:

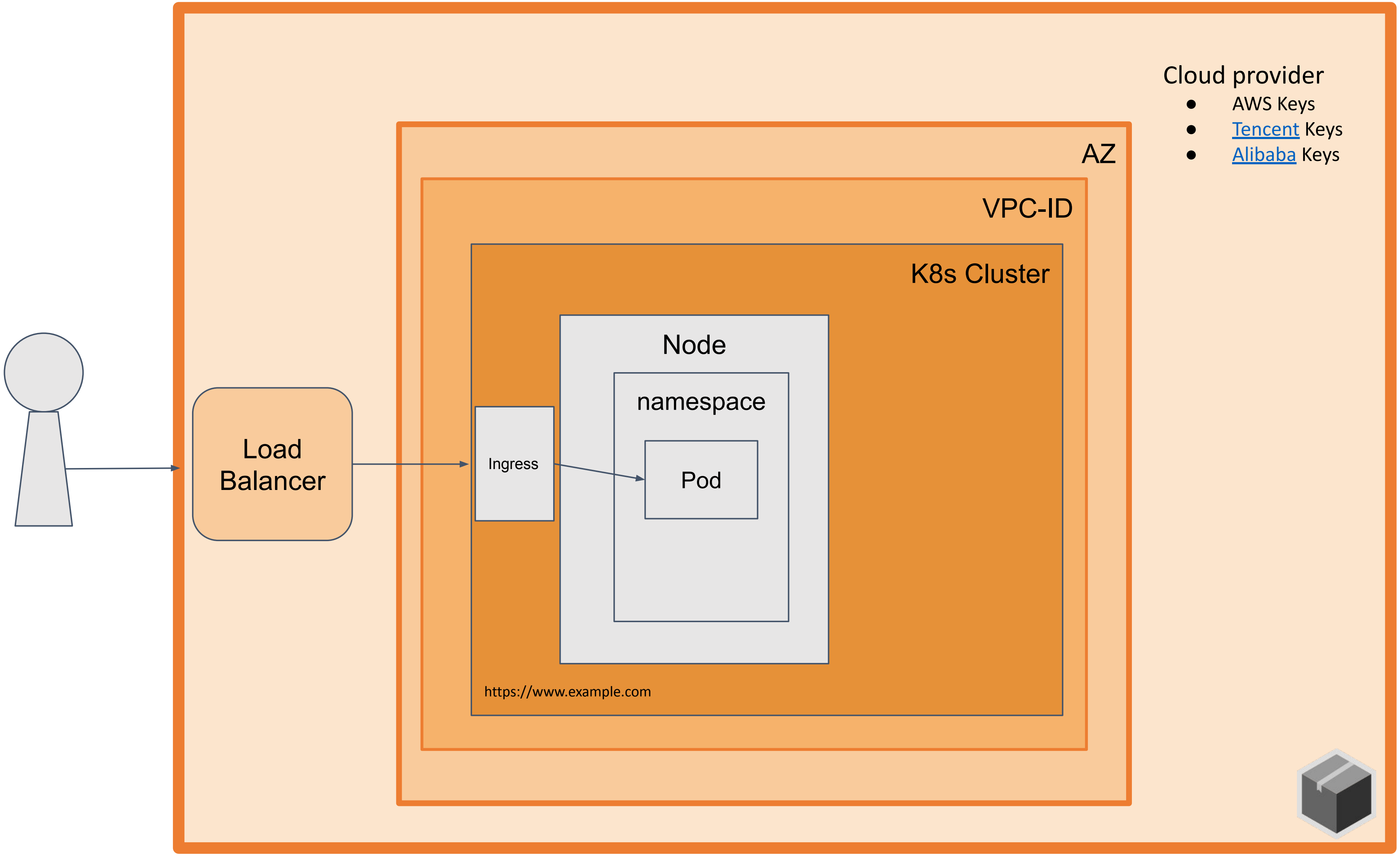
```
    kube_node_info

    kube_service_info * on (service) group_left group by
        (service,type)(kube_service_spec_type{type="LoadBalancer"})

    kube_ingress_info
```

Node hostname

Services in the cluster (specially load-balancers)

- namespace
- cluster IP
- node
- (application behind the service can be guessed by name of service/namespace)

Ingresses in the cluster



Cloud provider

- AWS Keys
- Tencent Keys
- Alibaba Keys

Scenario - NotARealCompanyForSure ©

Cloud provider

Credentials:

- AWS Keys
- Tencent Keys
- Alibaba Keys

Networking

- Load Balancer
- Region & AZ
- VPC
- Instance IP & ID

K8s Cluster

Topology

- Cluster IP
- Namespaces
- Nodes
- Ingress

Website

API

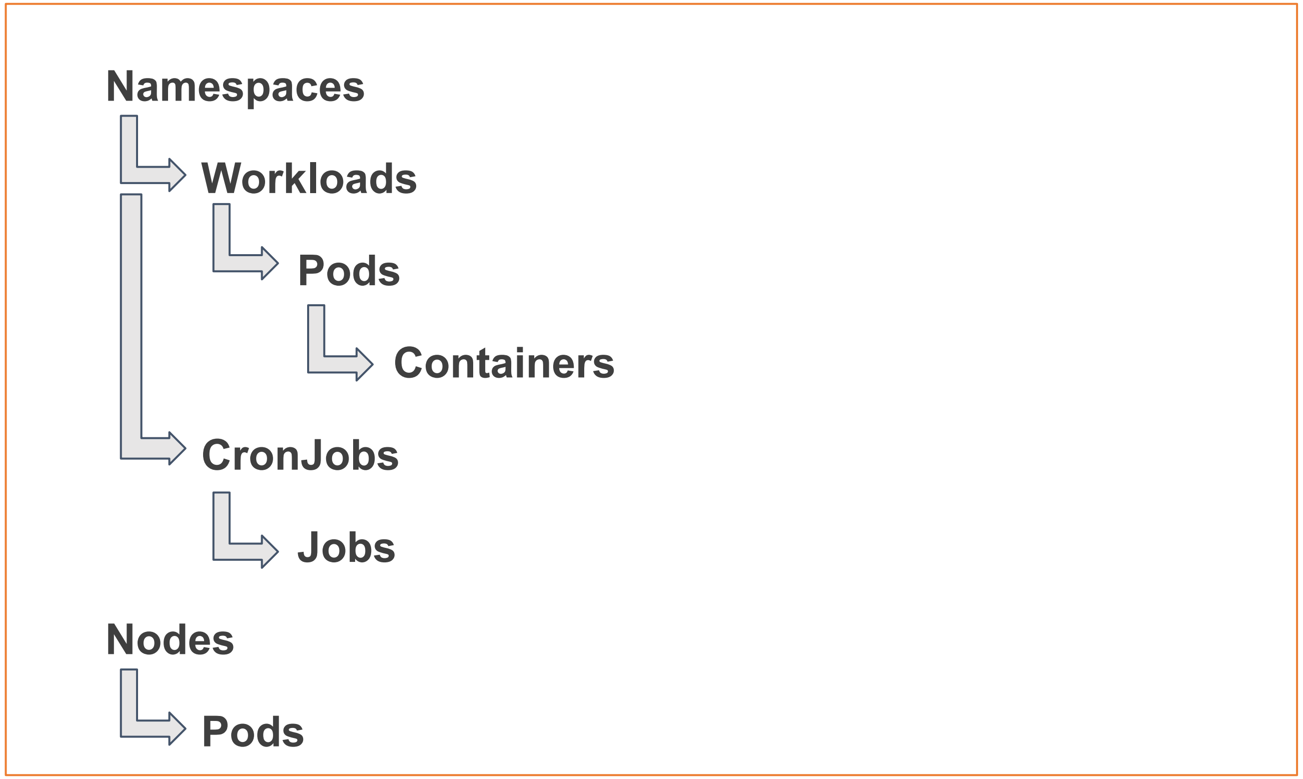
…

https://example.com

Fingerprinting Kubernetes hierarchy

KSM:

```
    kube_namespace_status_phase

    kube_deployment_spec_replicas
    kube_daemonset_status_desired_number_scheduled
    kube_statefulset_replicas
    kube_replicaset_spec_replicas

    kube_pod_info

    kube_pod_container_info

    kube_cronjob_info

    kube_job_info
```



Namespaces
- Workloads
  - Pods
    - Containers
- CronJobs
  - Jobs

Nodes
- Pods

Fingerprinting Kubernetes Control Plane

Kubernetes:

`kubernetes_build_info`

Component

- API-server
- controller-manager
- kube-proxy…

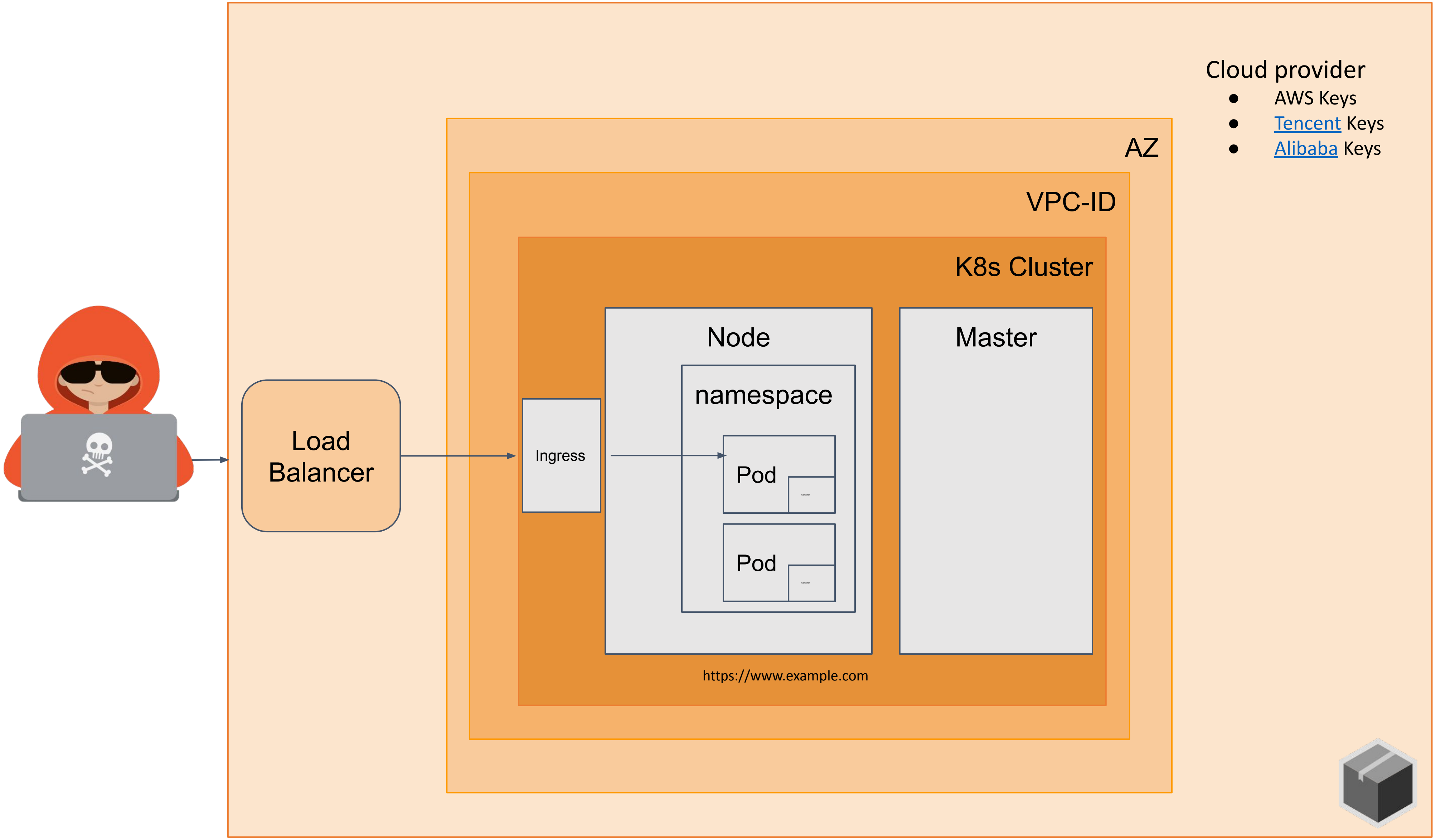
Major, minor version

git version

git commit

build_date

go_version



Cloud provider

- AWS Keys
- Tencent Keys
- Alibaba Keys

AZ

VPC-ID

K8s Cluster

Load Balancer

Ingress

Node

namespace

Pod

Pod

Master

https://www.example.com

Scenario - NotARealCompanyForSure ©

Cloud provider

Credentials:

- AWS Keys
- Tencent Keys
- Alibaba Keys

Networking

- Load Balancer
- Region & AZ
- VPC
- Instance IP & ID

K8s Cluster

Components:

- Kube-proxy
- Kube-admin
- Kubelet

Topology

- Cluster IP
- Namespaces
- Nodes
- Ingress

Known Vulnerabilities:

- CVE-2020-8554
- CVE-2020-8558
- CVE-2020-8559
- CVE-2021-25735
- CVE-2021-25737
- CVE-2021-25741

Website
API
…
https://example.com

Fingerprinting OS & Kernel

KSM Exporter:

`kube_node_info`

os_image:

- Ubuntu 18.04.4 LTS
- Ubuntu 20.04.3 LTS
- CentOS Linux 7 (Core)
- Tencent Linux 2.4

kernel_version:

- 5.11.0-1027-aws
- 4.15.0-142-generic
- 4.14.105-19-0020.1
- 3.10.0-1160.59.1.el7.x86_64

Scenario - NotARealCompanyForSure ©

Cloud provider

Credentials:

- AWS Keys
- Tencent Keys
- Alibaba Keys

Networking

- Load Balancer
- Region & AZ
- VPC
- Instance IP & ID

K8s Cluster

Components:

- Kube-proxy
- Kube-admin
- Kubelet

Topology

- Cluster IP
- Namespaces
- Nodes

Known Vulnerabilities:

- CVE-2020-8554
- CVE-2020-8558
- CVE-2020-8559
- CVE-2021-25735
- CVE-2021-25737
- CVE-2021-25741

Node

- Kernel
- OS
- Go version
- Git version

Known Vulnerabilities:

- CVE-2022-0847 - dirty pipe (Kernel Linux)
- CVE-2022-0185
- USN-3833-1: Linux kernel (AWS) vulnerabilities
  - CVE-2018-18955
- CVE-2021-3156

Website
API
…
https://example.com

Applications versions

KSM:

`kube_pod_container_info`

Custom:

`prometheus_build_info`

pod (app name)

image name + tag + sha256

- docker.io/library/cassandra:3.11.6
- sha256:5aa8400b4b3b794b5eba85f79b75a9ed9326e41428ae3a9d6b91cd731f2cf768

Prometheus version

Scenario - NotARealCompanyForSure ©

Cloud provider

Credentials:

- AWS Keys
- Tencent Keys
- Alibaba Keys

Networking

- Load Balancer
- Region & AZ
- VPC
- Instance IP & ID

K8s Cluster

Components:

- Kube-proxy
- Kube-admin
- Kubelet

Topology

- Cluster IP
- Namespaces
- Nodes

Known Vulnerabilities:

- CVE-2020-8554
- CVE-2020-8558
- CVE-2020-8559
- CVE-2021-25735
- CVE-2021-25737
- CVE-2021-25741

Node

- Kernel
- OS
- Go version
- Git version
- Docker

Known Vulnerabilities:

- CVE-2022-0847 - dirty pipe (Kernel Linux)
- CVE-2022-0185
- USN-3833-1: Linux kernel (AWS) vulnerabilities
  - CVE-2018-18955
- CVE-2021-3156

Pod / Container

Registry:

- docker.io

Image:

- Image-id

Service

- Service-example
  - Website
  - API
  - …
  - https://example.com

Known Vulnerabilities:

- CVE-2021-44521 - Cassandra
- https://mariadb.com/kb/en/security/ - RCE
- CVE-2020-28035
- Wordpress
- CVE-2018-16850 - PostgreSQL
- CVE-2019-11043 - PHP
- CVE-2021-44228 - Log4j
- CVE-2022-22963 - Spring Cloud
- CVE-2020-13942 - Apache unomi

Locating Kubernetes secrets

KSM:

```
    kube_secret_info
    kube_secret_type

    kube_secret_annotations
```

Namespace

Secret name

Type

- Opaque
- service-account-token…

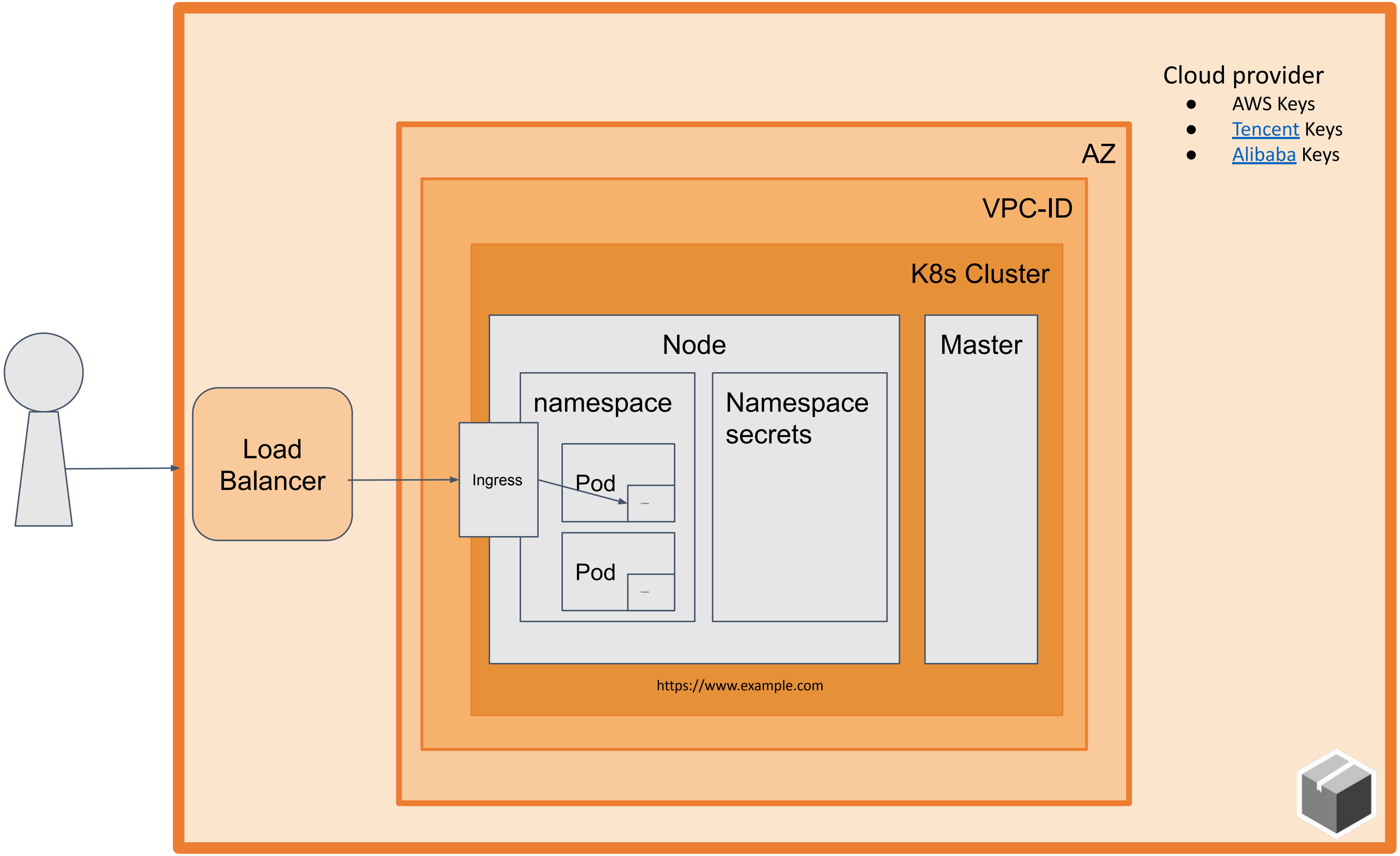
Kubectl last applied info (leak)

Application (application that uses the secret can be usually guessed by the name of secret/namespace)

```
kube_secret_annotations{kubectl_kubernetes_io_last_applied_configuration != ""}
```

Cloud provider

- AWS Keys
- Tencent Keys
- Alibaba Keys



Scenario - NotARealCompanyForSure ©

Cloud provider

Credentials:

- AWS Keys
- Tencent Keys
- Alibaba Keys

Networking

- Load Balancer
- Region & AZ
- VPC
- Instance IP & ID

K8s Cluster

Components:

- Kube-proxy
- Kube-admin
- Kubelet

Topology

- Cluster IP
- Namespaces
- Nodes

Known Vulnerabilities:

- CVE-2020-8554
- CVE-2020-8558
- CVE-2020-8559
- CVE-2021-25735
- CVE-2021-25737
- CVE-2021-25741

Node

- Kernel
- OS
- Go version
- Git version
- Docker

Known Vulnerabilities:

- CVE-2022-0847 - dirty pipe (Kernel Linux)
- CVE-2022-0185
- USN-3833-1: Linux kernel (AWS) vulnerabilities
  - CVE-2018-18955
- CVE-2021-3156

Pod / Container

Registry:

- docker.io

Image:

- Image-id

Service

- Service-example
  - Website
  - API
  - …
  - https://example.com

Known Vulnerabilities:

- CVE-2021-44521 - Cassandra
- https://mariadb.com/kb/en/security/ - RCE
- CVE-2020-28035
- Wordpress
- CVE-2018-16850 - PostgreSQL
- CVE-2019-11043 - PHP
- CVE-2021-44228 - Log4j
- CVE-2022-22963 - Spring Cloud
- CVE-2020-13942 - Apache unomi

Kubernetes Secrets

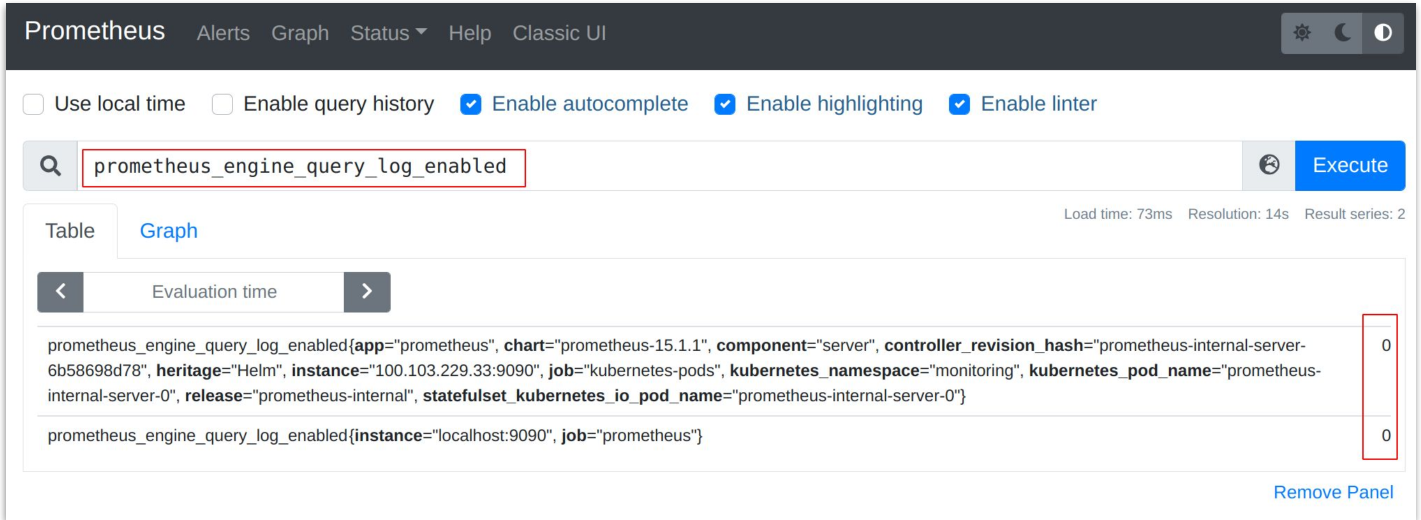
- Service auth tokens

Logging queries in Prometheus

Prometheus allows query logging… but it's **not enabled by default**.

You can check if loggin is enabled by querying this metric:

```
prometheus_engine_query_log_enabled
```

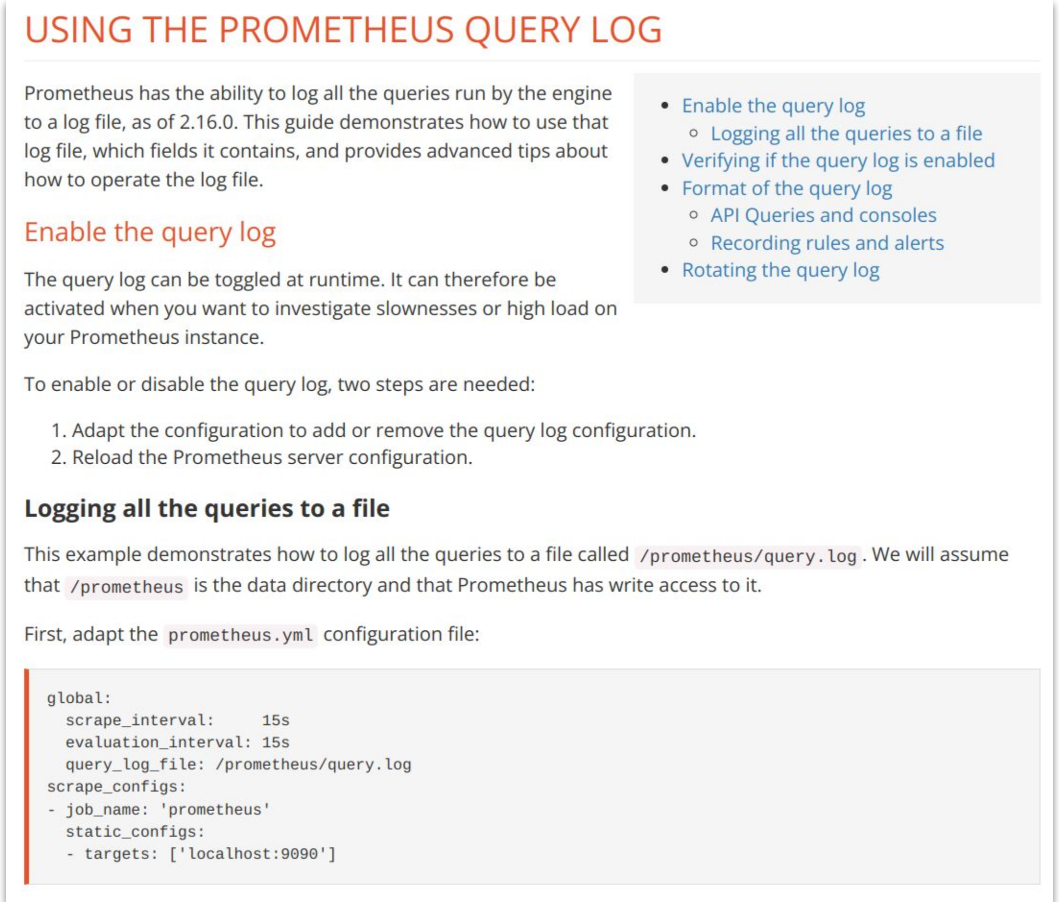


□ Use local time □ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

```
prometheus_engine_query_log_enabled
```

Load time: 73ms Resolution: 14s Result series: 2

```
prometheus_engine_query_log_enabled{app="prometheus", chart="prometheus-15.1.1", component="server", controller_revision_hash="prometheus-internal-server-6b58698d78", heritage="Helm", instance="100.103.229.33:9090", job="kubernetes-pods", kubernetes_namespace="monitoring", kubernetes_pod_name="prometheus-internal-server-0", release="prometheus-internal", statefulset_kubernetes_io_pod_name="prometheus-internal-server-0"}    0
prometheus_engine_query_log_enabled{instance="localhost:9090", job="prometheus"}    0
```



USING THE PROMETHEUS QUERY LOG

Prometheus has the ability to log all the queries run by the engine to a log file, as of 2.16.0. This guide demonstrates how to use that log file, which fields it contains, and provides advanced tips about how to operate the log file.

- Enable the query log
  - Logging all the queries to a file
- Verifying if the query log is enabled
- Format of the query log
  - API Queries and consoles
  - Recording rules and alerts
- Rotating the query log

Enable the query log

The query log can be toggled at runtime. It can therefore be activated when you want to investigate slownesses or high load on your Prometheus instance.

To enable or disable the query log, two steps are needed:

1. Adapt the configuration to add or remove the query log configuration.
2. Reload the Prometheus server configuration.

Logging all the queries to a file

This example demonstrates how to log all the queries to a file called `/prometheus/query.log`. We will assume that `/prometheus` is the data directory and that Prometheus has write access to it.

First, adapt the `prometheus.yml` configuration file:

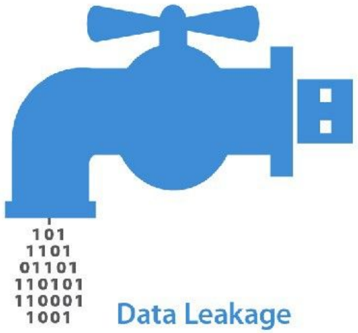
```
global:
  scrape_interval:     15s
  evaluation_interval: 15s
  query_log_file: /prometheus/query.log
scrape_configs:
- job_name: 'prometheus'
  static_configs:
  - targets: ['localhost:9090']
```

https://prometheus.io/docs/guides/query-log/

Real History

Now, the attacker prepares the journey and the intrusion target.

In this fictitious examples, the attacker might want to access the data leak, use your machines for cryptomining or encrypt the victim's data (ransomware). With this knowledge of Prometheus exposed, the attacker uses the specific technique for each case.



https://miro.medium.com/max/750/1*TSX7fu85EwGEdnhA-Sv4cA.jpeg

https://cd.blokt.com/wp-content/uploads/2018/02/crypto-mining-e1518714481556.jpg

Leak data scenario - Attacker Path

Cloud provider

Credentials:

- AWS Keys
- Tencent Keys
- Alibaba Keys

Networking

- Load Balancer
- Region & AZ
- VPC
- Instance IP & ID

K8s Cluster

Components:

- Kube-proxy
- Kube-admin
- Kubelet

Topology

- Cluster IP
- Namespaces
- Nodes

Known Vulnerabilities:

- CVE-2020-8554
- CVE-2020-8558
- CVE-2020-8559
- CVE-2021-25735
- CVE-2021-25737
- CVE-2021-25741

Node

- Kernel
- OS
- Go version
- Git version
- Docker

Known Vulnerabilities:

- CVE-2022-0847 - dirty pipe (Kernel Linux)
- CVE-2022-0185
- USN-3833-1: Linux kernel (AWS) vulnerabilities
  - CVE-2018-18955
- CVE-2021-3156

Pod / Container

Registry:

- docker.io

Image:

- Image-id

Service

- Service-example
  - Website
  - API
  - com

Known Vulne

- CVE-2021-44521 - Cassandra
- https://mariadb.com/kb/en/security/ - RCE
- CVE-2020-28035
- Wordpress
- CVE-2018-16850 - PostgreSQL
- CVE-2019-11043 - PHP
- CVE-2021-44228 - Log4j
- CVE-2022-22963 - Spring Cloud
- CVE-2020-13942 - Apache unomi

Kubernetes Secrets

- Service auth tokens

Cryptomining scenario - Attacker Path

Cloud provider

Credentials: 3

- AWS Keys
- Tencent Keys
- Alibaba Keys

Networking

- Load Balancer
- Region & AZ
- VPC
- Instance IP & ID

K8s Cluster

Components:

- Kube-proxy
- Kube-admin
- Kubelet

Topology

- Cluster IP
- Namespaces
- Nodes

Known Vulnerabilities:

- CVE-2020-8554
- CVE-2020-8558
- CVE-2020-8559
- CVE-2021-25735
- CVE-2021-25737
- CVE-2021-25741

Node

- Kernel
- OS
- Go version
- Git version
- Docker

Known Vulnerabilities: 2

- CVE-2022-0847 - dirty pipe (Kernel Linux)
- CVE-2022-0185
- USN-3833-1: Linux kernel (AWS) vulnerabilities
  - CVE-2018-18955
- CVE-2021-3156

https://dirtypipe.cm4all.com/
https://github.com/arget13/DDexec

Pod / Container

Registry:

- docker.io

Image:

- Image-id

Service

- Service-example
  - Website
  - API
  - …
  - https://example.com

Known Vulnerabilities:

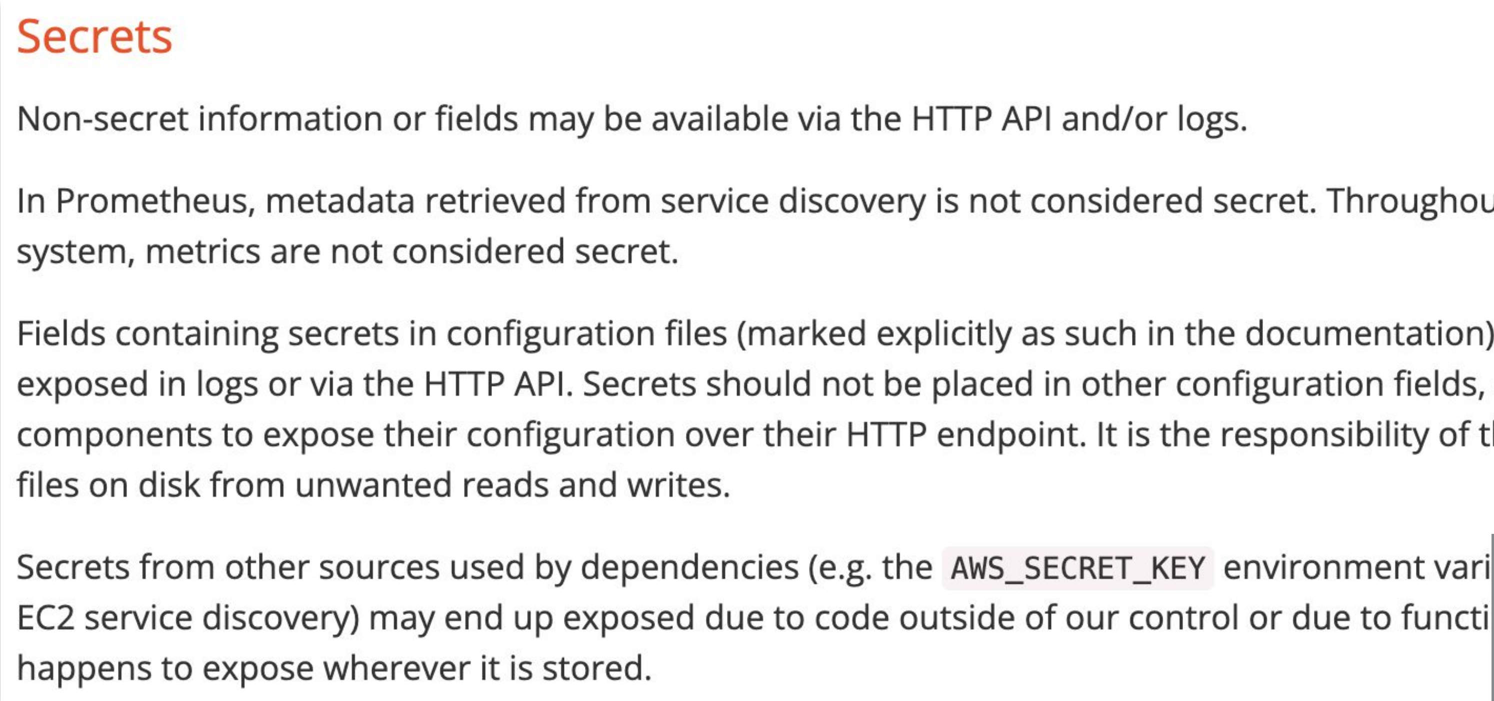
- CVE-2021-44521 - Cassandra
- https://mariadb.com/kb/en/security/ - RCE
- CVE-2020-28035
- Wordpress
- CVE-2018-16850 - PostgreSQL
- CVE-2019-11043 - PHP
- CVE-2021-44228 - Log4j 1
- CVE-2022-22963 - Spring Cloud
- CVE-2020-13942 - Apache unomi

https://github.com/kozmer/log4j-shell-poc

Kubernetes Secrets

- Service auth tokens

Prometheus secrets



Secrets

Non-secret information or fields may be available via the HTTP API and/or logs.

In Prometheus, metadata retrieved from service discovery is not considered secret. Throughout the Prometheus system, metrics are not considered secret.

Fields containing secrets in configuration files (marked explicitly as such in the documentation) will not be exposed in logs or via the HTTP API. Secrets should not be placed in other configuration fields, as it is common fo components to expose their configuration over their HTTP endpoint. It is the responsibility of the user to protect files on disk from unwanted reads and writes.

Secrets from other sources used by dependencies (e.g. the `AWS_SECRET_KEY` environment vari EC2 service discovery) may end up exposed due to code outside of our control or due to functi happens to expose wherever it is stored.

Ransomware scenario - Attacker Path

Cloud provider

Credentials:

- AWS Keys
- Tencent Keys
- Alibaba Keys

Networking

- Load Balancer
- Region & AZ
- VPC
- Instance IP & ID

K8s Cluster

Components:

- Kube-proxy
- Kube-admin
- Kubelet

Topology

- Cluster IP
- Namespaces
- Nodes

Known Vulnerabilities:

2

- CVE-2020-8554
- CVE-2020-8558
- CVE-2020-8559
- CVE-2021-25735
- CVE-2021-25737
- CVE-2021-25741

Node

- Kernel
- OS
- Go version
- Git version
- Docker

Known Vulnerabilities:

- CVE-2022-0847 - dirty pipe (Kernel Linux)
- CVE-2022-0185
- USN-3833-1: Linux kernel (AWS) vulnerabilities
  - CVE-2018-18955
- CVE-2021-3156

Pod / Container

Registry:

- docker.io

Image:

- Image-id

Service

- Service-example
  - Website
  - API
  - …
  - https://example.com

Known Vulnerabilities:

1

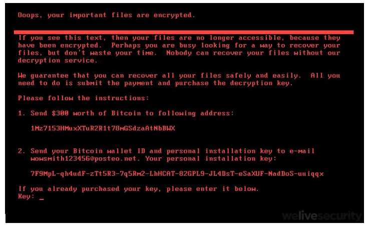
- CVE-2021-44521 - Cassandra
- https://mariadb.com/kb/en/security/ - RCE
- CVE-2020-28035
- Wordpress
- CVE-2018-16850 - PostgreSQL
- CVE-2019-11043 - PHP
- CVE-2021-44228 - Log4j
- CVE-2022-22963 - Spring Cloud
- CVE-2020-13942 - Apache unomi
- …

https://github.com/hktalent/spring-spel-0day-poc

3

Kubernetes Secrets

- Service auth tokens

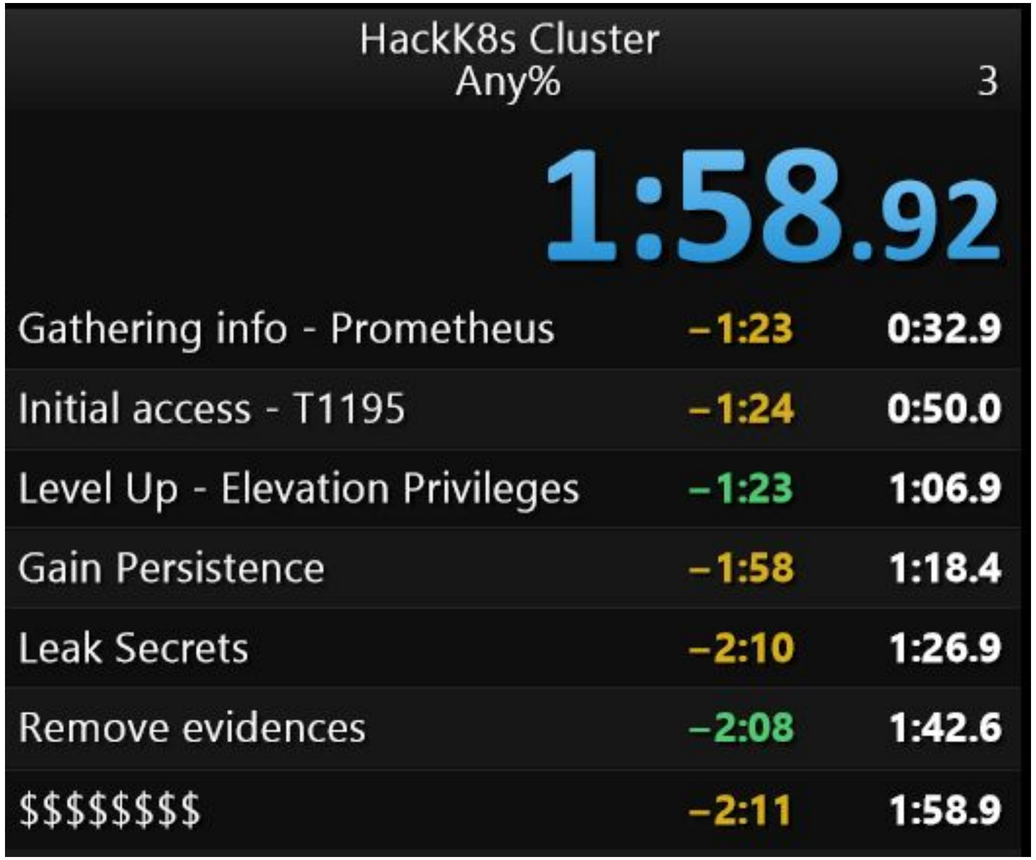


```
Ooops, your important files are encrypted.

If you see this text, then your files are no longer accessible, because they
have been encrypted.  Perhaps you are busy looking for a way to recover your
files, but don't waste your time.  Nobody can recover your files without our
decryption service.

We guarantee that you can recover all your files safely and easily.  All you
need to do is submit the payment and purchase the decryption key.

Please follow the instructions:

1. Send $300 worth of Bitcoin to following address:

   1Mz7153HMuxXTuR2R1t78mGSdzaAtNbBWX


2. Send your Bitcoin wallet ID and personal installation key to e-mail
   wowsmith123456@posteo.net. Your personal installation key:

   7F9MpL-qh4udF-zTt5R3-7q5Rm2-LhHCAT-82GPL9-JL4DsT-eSaXUF-NadDoS-uuiqqx

If you already purchased your key, please enter it below.
Key: _
```

Summary

We could think that metrics are not important in a security perspective, but we show that’s not true.

It’s also important to mention that the proper services Kubernetes or Prometheus advise of the problems to expose their data to the world



HackK8s Cluster
Any% 3

1:58.92

Gathering info - Prometheus	−1:23	0:32.9
Initial access - T1195	−1:24	0:50.0
Level Up - Elevation Privileges	−1:23	1:06.9
Gain Persistence	−1:58	1:18.4
Leak Secrets	−2:10	1:26.9
Remove evidences	−2:08	1:42.6
$$$$$$$$	−2:11	1:58.9

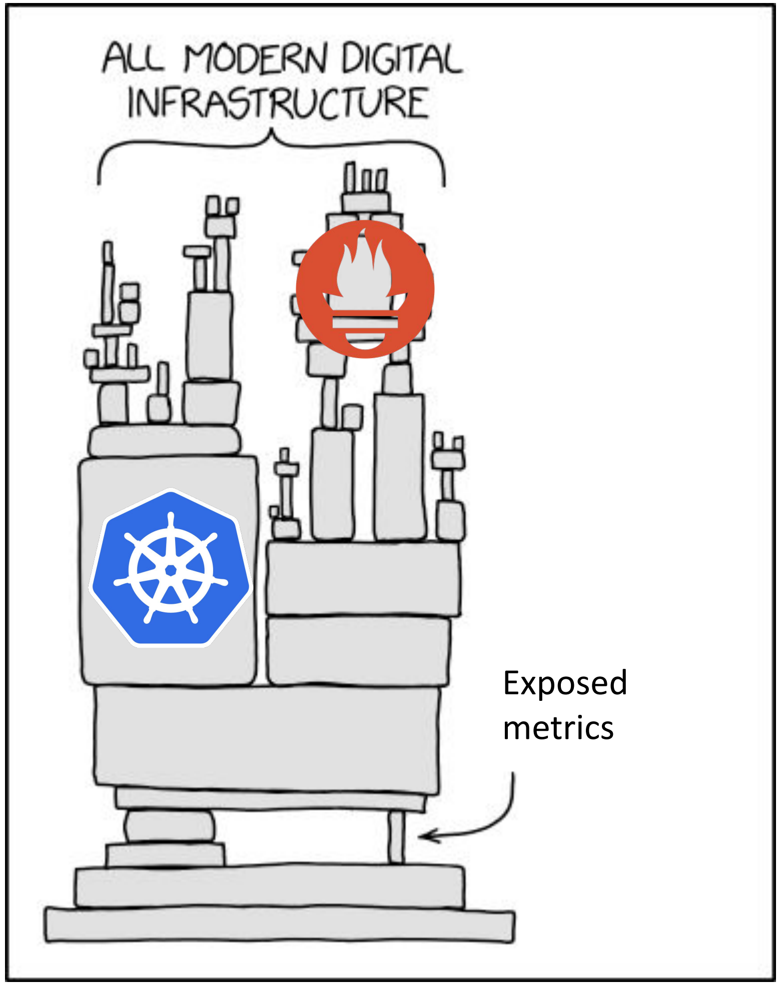
Recommendations

Today, if we follow security best practices in every part of our chain, we are safe from most security incidents.

We will have to continue to fight with new vulnerabilities that impact our services and also, not least, a plan against insiders. But let's at least make things difficult.

- Secure your Cloud provider with Principle of least privilege.
  - **https://www.cisa.gov/uscert/ncas/current-activity/2020/01/24/nsa-releases-guidance-mitigating-cloud-vulnerabilities**
- Secure your Cluster Kubernetes
  - **https://media.defense.gov/2021/Aug/03/2002820425/-1/-1/0/CTR_Kubernetes_Hardening_Guidance_1.1_20220315.PDF**
- Secure the Host / OS
  - **https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-123.pdf**
- Secure the containers
  - **https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-190.pdf**
- Secure your code
  - **https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-95.pdf**
- Secure your Prometheus Metrics!
  - **https://prometheus.io/docs/operating/security/#prometheus**

Kubernetes fingerprinting with Prometheus



https://xkcd.com/2347/

Miguel Hernandez
Security Researcher
Sysdig
@MiguelHzBz

David de Torres
Manager of Engineering
Sysdig
@maellyssa

How Attackers Use Exposed Prometheus Server to Exploit Kubernetes Clusters

Miguel Hernández & David de Torres, Sysdig