CSA GCR 6th Congress 第六届云安全联盟大中华区大会

云安全的下半场：原生安全

演讲人：刘文懋 绿盟科技

CSA GCR cloud security alliance GREATER CHINA REGION

CSA GCR 6th Congress 第六届云安全联盟大中华区大会

目　录

CONTENTS

Part1

云安全：纯安全问题

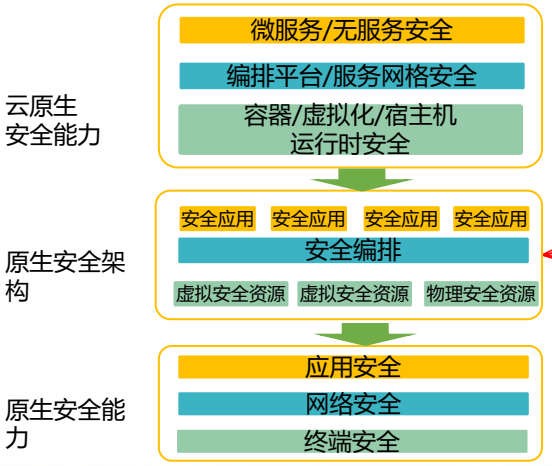
未来：云安全=纯安全

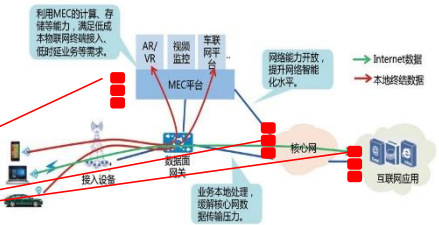
- **The Future of Network Security is in the Cloud**
  - 在2020年前，50%的企业将业务工作流放到本地需要作为异常事件进行审批。公司“无云”的策略会和现在“无网络”的策略一样少。
- **Cloud Security Becomes … Just Security**
  - 云计算与各行各业IT基础设施进一步融合，云或是基础，或是组件
- **云赋能安全+安全赋能云=纯安全**
  - 各类安全机制，将会或多或少适用于或应用云计算技术

工业互联网

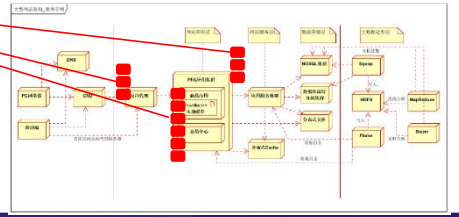
5G/边缘计算

原生安全：基于云原生、无处不在的安全





5G/边缘计算场景



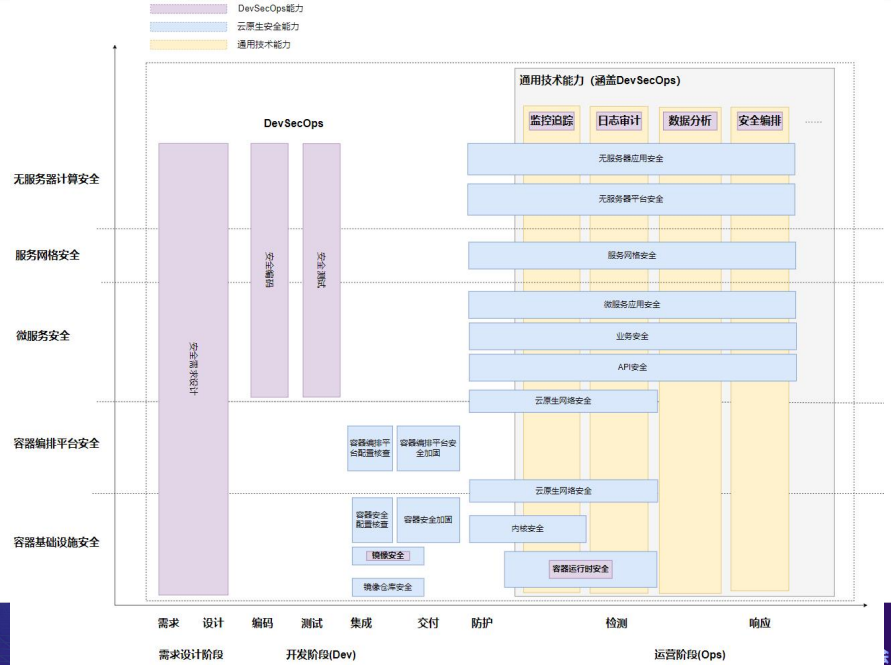
大型电商场景

Part2

云原生安全：云计算下半场

云原生的生态和安全栈


云原生
微服务/无服务
安全配置
API安全
业务安全
AI安全
编排平台/服务网格
通信安全
身份认证
微隔离
安全配置
容器运行时
安全配置
镜像安全
入侵防护
操作系统安全
仓库安全
异常检测
IaaS
虚拟化运行时
Hypervisor安全
终端安全
安全资源池
宿主机/物理环境
主机安全
网络安全
物理安全

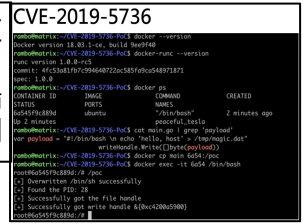

DevSecOps能力
云原生安全能力
通用技术能力
通用技术能力（涵盖DevSecOps）
监控追踪
日志审计
数据分析
安全编排
DevSecOps
无服务器计算安全
服务网格安全
微服务安全
容器编排平台安全
容器基础设施安全
无服务器应用安全
无服务器平台安全
服务网格安全
微服务应用安全
业务安全
API安全
云原生网络安全
容器编排平台配置核查
容器编排平台安全加固
云原生网络安全
容器安全配置核查
容器安全加固
内核安全
镜像安全
容器运行时安全
镜像仓库安全
需求
设计
编码
测试
集成
交付
防护
检测
响应
需求设计阶段
开发阶段(Dev)
运营阶段(Ops)

容器运行时最大风险：容器逃逸

容器逃逸

应用漏洞

CVE-2019-5736

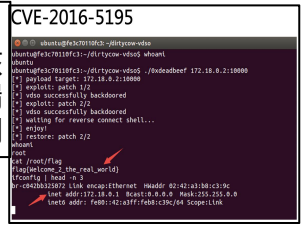


危险配置

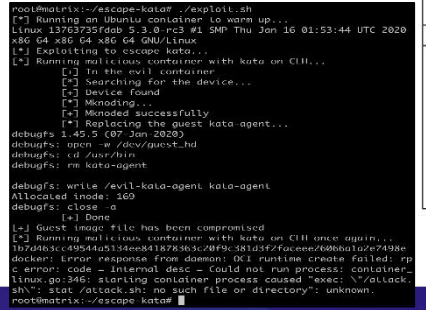
--privileged 特权模式

内核漏洞

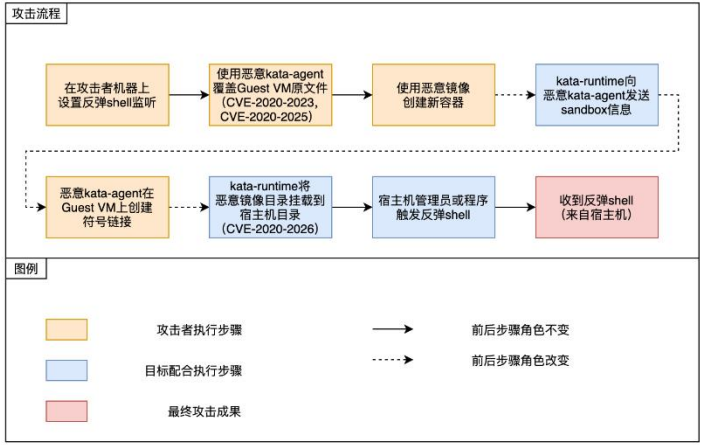
CVE-2016-5195



- Kata-containers逃逸
- 先容器逃逸，再虚拟机逃逸
- 涉及：
  - CVE-2020-2023
  - CVE-2020-2025
  - CVE-2020-2026
- 安全容器不是银弹！



安全容器逃逸



云原生攻防套件

云原生靶场

- Metarget = meta + target
- https://github.com/brant-ruan/metarget
- 安装内核漏洞：metarget cnv install cve-2016-5195
- 安装Docker漏洞：metarget cnv install cve-2019-5736
- 安装Kubernetes漏洞：metarget cnv install cve-2018-1002105

Unwatch 5 | Unstar 162 | Fork 31

Name	Class	Type	CVSS 3.x	Status
cve-2018-15664	docker	container_escape	7.5	✅
cve-2019-13139	docker	command_execution	8.4	✅
cve-2019-14271	docker	container_escape	9.8	✅
cve-2020-15257	docker/containerd	container_escape	5.2	✅
cve-2019-5736	docker/runc	container_escape	8.6	✅
cve-2021-30465	docker/runc	container_escape	7.6	✅
cve-2017-1002101	kubernetes	container_escape	9.6	✅
cve-2018-1002105	kubernetes	privilege_escalation	9.8	✅
cve-2019-11253	kubernetes	denial_of_service	7.5	✅
cve-2019-9512	kubernetes	denial_of_service	7.5	✅
cve-2019-9514	kubernetes	denial_of_service	7.5	✅

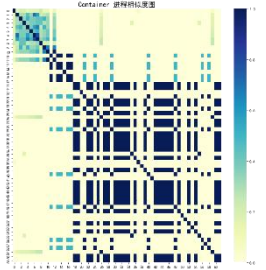
云原生安全评估套件

模块名	涉及组件	模块类型	模块描述
cve-2016-5195	Kernel	攻击利用	利用条件竞争漏洞进行提权
cve-2017-1002101	Kubernetes	攻击利用	使用subPath挂载机制来创建挂载宿主机文件系统的Pod
cve-2018-1002105	Kubernetes	攻击利用	通过向Kubernetes API Server构造请求实现权限提升
cve-2019-5736	Docker/runc	攻击利用	通过修改runc的二进制文件导致提权
cve-2020-15257	Docker/containerd	攻击利用	通过通过containerd-shim API实现提权
kata_container_escape	Kata containers	攻击利用	
docker.sock	Docker.sock	攻击利用	
privileged_container	特权容器	攻击利用	
docker_api_rce	Docke/api	攻击利用	
mount_procfs	Procfs挂载	攻击利用	
ptrace_escape	Ptrace逃逸	攻击利用	
cgroup_escape	Cgroup逃逸	攻击利用	
cve-2020-8554	Kubernetes	后渗透	有创建和编辑服务
cve-2020-8558	Kube-proxy	后渗透	利用Kub
backdoor_pod	Kubernetes	后渗透	
shadow_api_server	Kubernetes	后渗透	
deployment_Cronjob	Kubernetes	后渗透	

异常行为检测(workload)

基于行为基线的异常检测

容器进程相似度



容器异常行为

	msg	event_type	process	user	cpu	process_len	container_name
0	原进程出现了新用户	01-02	bash	abc	0	4	k8s_data-dispatcher_data-dispatcher-deployment...
12	新进程启动，进程名长度过长异常	02-04	/usr/bin/python_abc	root	0	10	k8s_action-dispatcher_action-dispatcher-deploy...
7	原进程出现了新路径	01-03	/abc/bin/sh	root	0	2	k8s_zookeeper_zookeeper-1_default_9998eacf-85d...
11	原进程CPU偏高	01-01	/pause	root	100	5	k8s_POD_data-dispatcher-deployment-64d9f7f59d-...

基于行为特征的异常检测

逃逸行为检测规则

```
- macro: container
  condition: (container.id != host)

- rule: New madvise System Call
  desc: >
    Detect new madvise in container.
  condition: >
    evt.type = "madvise" and evt.dir=> and container
  output: name=%proc.name exe=%proc.exe user=%user.name cwd=%proc.cwd
exeline=%proc.exeline pid=%proc.pid ppid=%proc.ppid
        ctr_name=%container.name ctr_id=%container.id
image_name=%container.image image_id=%container.image.id
  priority: INFO
  tags: [container, nsfocus]
```

逃逸行为检测结果

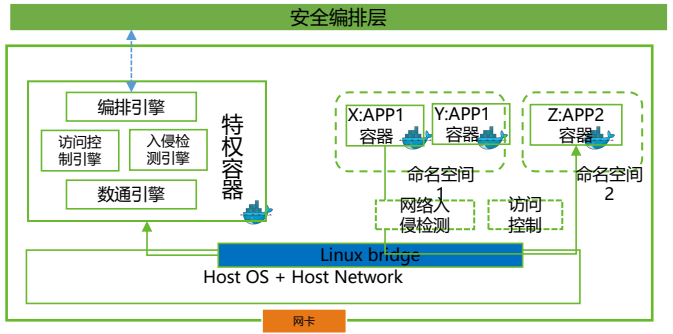
```
+ ~ grep "Notice madvise name=0xdeadbeef" ./cve-2016-5195-exp.log  | head -n 1
10:55:37.616951232: Notice madvise name=0xdeadbeef exe=./0xdeadbeef user=root cwd=/ exeline=./0xdeadbeef 172.17.0.3:10000 pid=13024 ppid=12958 ctr_name=dirtycow ctr_id=b4a846745b46 image_name=ubuntu image_id=72300a873c2ca11c70d0c8642177ce76ff69ae04d61a5813ef58d40ff66e3e7c
+ ~ grep "Notice ptrace name=0xdeadbeef" ./cve-2016-5195-exp.log  | head -n 1
10:55:37.616988760: Notice ptrace name=0xdeadbeef exe=./0xdeadbeef user=root cwd=/ exeline=./0xdeadbeef 172.17.0.3:10000 pid=13028 ppid=13027 ctr_name=dirtycow ctr_id=b4a846745b46 image_name=ubuntu image_id=72300a873c2ca11c70d0c8642177ce76ff69ae04d61a5813ef58d40ff66e3e7c
+ ~ grep "Notice madvise name=0xdeadbeef" ./cve-2016-5195-exp.log  | wc -l
219
+ ~ grep "Notice ptrace name=0xdeadbeef" ./cve-2016-5195-exp.log  | wc -l
791
```

```
'2020-05-22 03:15:22,480 - root - INFO - race condition detector: CVE-2016-5195 is enough abnormal for (u'victim-2', u'489592185a60', 16978)'
'2020-05-22 03:15:22,480 - root - INFO - race condition detector: certain alarm for CVE-2016-5195, (u'victim-2', u'489592185a60', 16978)'
'2020-05-22 03:15:22,486 - root - INFO - victim-2异常攻击行为:发现攻击行为，场景为CVE-2016-5195:1590117312.33:{"image_id": "4e5021d210f65ebe915670c7009120120cc0a303b9020859285170dc1b8c04bd", "container_id": "489592185a60", "syscall_seq": ["madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "ptrace", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "madvise", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "ptrace", "madvise", "ptrace", "madvise", "ptrace", "madvise", "madvise", "madvise"], "ppid": 14813, "monitor_time": "1590117312.33", "cmd": "./0xdeadbeef 172.17.0.2 10001", "pid": 16978, "delay_count": null}'
```

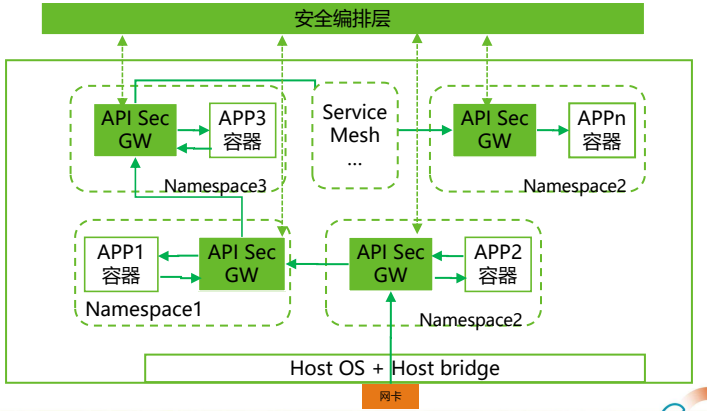
异常行为检测(network)

安全云原生化绝不仅是安全设备容器化，还有部署轻量化、功能简单化，策略编排化，能力云原生化

容器网络安全(微隔离、访问控制、入侵检测)



云原生应用安全(API、应用层攻击)



Part3

原生安全：未来的安全

未来的重大产业升级：新基建?

2020年4月20日，国家发改委终于给出了权威说法。新型基础设施主要包括三方面内容：

一是信息基础设施，包括以5G、物联网、工业互联网、卫星互联网为代表的通信网络基础设施，以人工智能、云计算、区块链等为代表的新技术基础设施，以数据中心、智能计算中心为代表的算力基础设施等。

二是融合基础设施，主要指深度应用互联网、大数据、人工智能等技术，支撑传统基础设施转型升级，进而形成的融合基础设施，比如，智能交通基础设施、智慧能源基础设施等。

三是创新基础设施。主要是指支撑科学研究、技术开发、产品研制的具有公益属性的基础设施，比如，重大科技基础设施、科教基础设施、产业技术创新基础设施等。

边缘计算

- **多接入边缘计算（MEC)是指在网络边缘侧提供智能服务且支持多租户的开放平台**

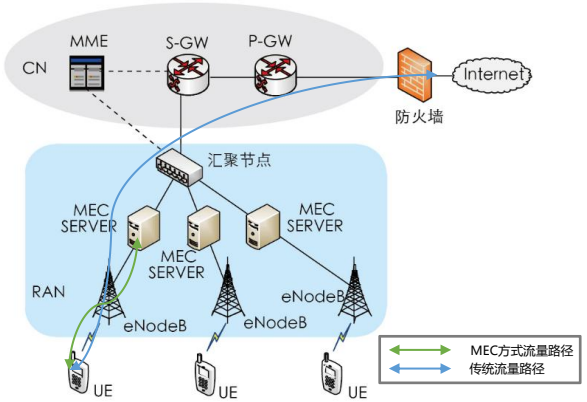
- 距终端用户一步之遥，降低服务时延
- 减轻网络负载
- 资源受限，轻量级
- 基于容器、微服务

01 网络基础设施
- 拒绝服务攻击
- 中间人攻击
- 伪网关

02 边缘数据中心
- 物理损坏
- 隐私泄露
- 权限升级

03 核心基础设施
- **隐私泄露**
- **服务篡改**
- **伪基础设施**

04 虚拟化基础设施
- 拒绝服务
- 资源误用
- 隐私泄露



来源： MEC研究进展与应用场景探讨，MEC研究进展与应用场景探讨

来源：Roman R , Lopez J , Mambo M . Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges

主流开源边缘计算平台

KubeEdge

公司：华为
分类：
分类依据一：物联网边缘计算为主
分类依据二：边缘网关或边缘云
Github链接：
https://github.com/kubeedge

OpenNESS

OpenNESS
OPEN NETWORK EDGE SERVICES SOFTWARE

公司：Intel
分类：
分类依据一：多接入边缘计算
分类依据二：边缘云或云边缘
Github链接：
https://github.com/open-ness

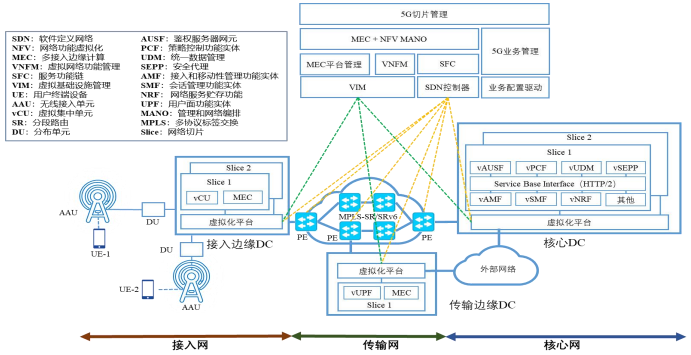
StarlingX

STARLINGX

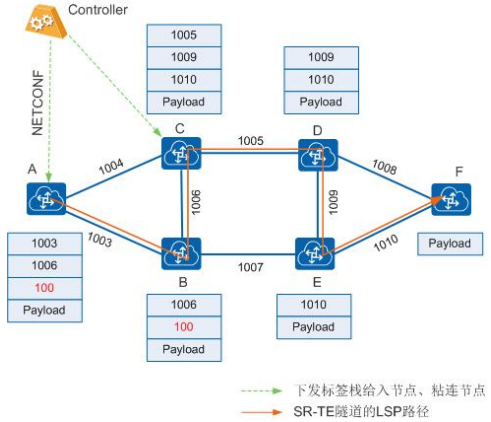
公司：Intel和WindRiver
分类：
分类依据一：多接入边缘计算
分类依据二：边缘云或云边缘
Opendev链接：
https://opendev.org/starlingx

开源边缘平台安全的探索

5GC是新一代ICT网络



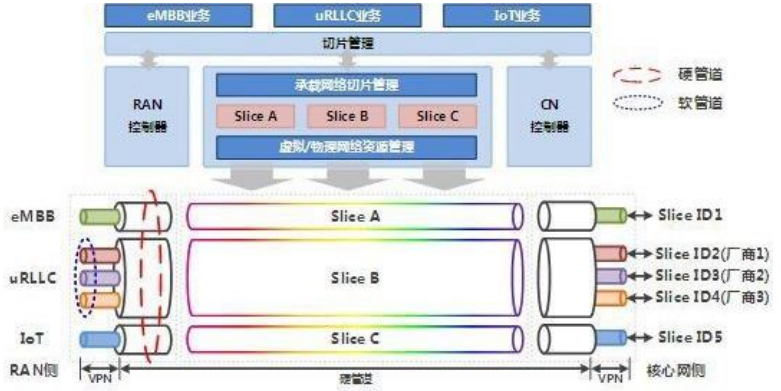
独立组网架构之核心网：网元NFV



5G将采用分段路由（Segment Routing）技术

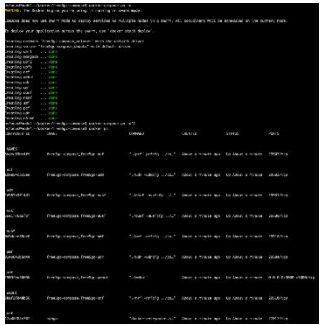
数据面切片：轻量级安全资源池

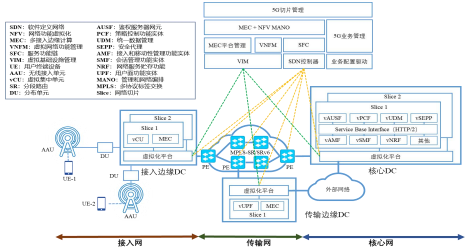
- 支持多种切片隔离技术：**面向不同应用资源和流量的安全防护机制隔离**
- 实现子切片弹性伸缩和低时延：**安全防护应满足安全强度和业务时延的要求**
- 实现物理网络和切片网络的端到端统一控制和管理：**统一安全资源池和服务链**



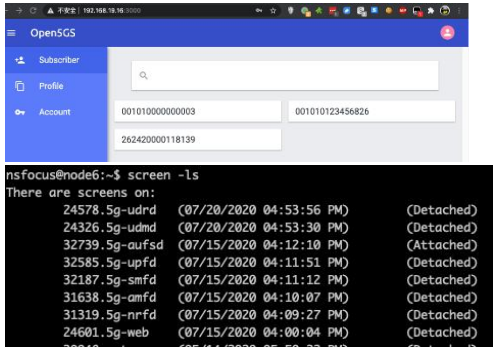
来源： https://www.sdnlab.com/23430.html

控制面网元：微服务化





free5gc



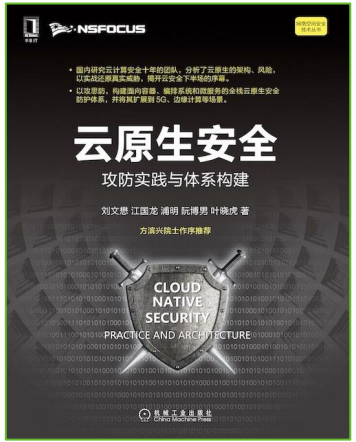
open5gs

Some takeaways

- 云计算已无处不在，下半场云原生已经开场
- 安全厂商需要构建全栈、基于云原生的原生安全能力，适应新业务、新场景的发展
- 一些云原生的安全风险，已经影响原生安全产品，比如蜜罐...
- 攻防永远是安全的主线，云原生攻防最终会变成传统攻防


NSFOCUS
NeuVector
2018 绿盟科技
容器安全技术报告


2020 | 巨人肩后的专家
云原生安全技术报告
NSFOCUS
移动云


NSFOCUS
云原生安全
攻防实践与体系构建
刘文懋 江国龙 浦明 阮博男 叶晓虎 著
方滨兴院士作序推荐
CLOUD
NATIVE
SECURITY
PRACTICE AND ARCHITECTURE
机械工业出版社


CSA 云安全联盟标准
CSA GCR XXXX—XXXX
云原生安全标准
英文名称
Cloud Native Security Standards

2022

敬请期待...

CSA GCR 6th Congress 第六届云安全联盟大中华区大会

CSA GCR cloud security alliance® GREATER CHINA REGION

THANK YOU.