

2022年 中国容器安全市场报告

2022 China Container Security Market Report

2022年中国コンテナセキュリティ市場レポート

报告标签：容器集群安全、镜像安全、运行时安全、容器隔离

报告提供的任何内容（包括但不限于数据、文字、图表、图像等）均系弗若斯特沙利文及头豹研究院独有的高度机密性文件（在报告中另行标明出处者除外）。未经弗若斯特沙利文及头豹研究院事先书面许可，任何人不得以任何方式擅自复制、再造、传播、出版、引用、改编、汇编本报告内容，若有违反上述约定的行为发生，弗若斯特沙利文及头豹研究院保留采取法律措施、追究相关人员责任的权利。弗若斯特沙利文及头豹研究院开展的所有商业活动均使用“弗若斯特沙利文”、“沙利文”、“头豹研究院”或“头豹”的商号、商标，弗若斯特沙利文及头豹研究院无任何前述名称之外的其他分支机构，也未授权或聘用其他任何第三方代表弗若斯特沙利文或头豹研究院开展商业活动。

概览说明

沙利文谨此发布中国容器安全系列报告之《2022年中国容器安全市场报告》年度报告。本报告旨在梳理梳理容器安全领域产品及服务形态，洞悉用户特点、市场存量空间及增量空间，并结合市场发展前景判断中国容器安全市场各类竞争者所处地位。

2022年第四季度，沙利文联合头豹研究院对容器安全核心产品进行了下游用户体验调查。根据下游用户调研反馈、行业专家见解及供应商专题交流，融合多维视角，输出分析成果。本市场报告提供的容器安全趋势分析亦反映出容器安全行业整体的动向。

本研究结果将通过增长指数体现竞争者维持现有市场地位的能力，通过创新指数体现竞争者进一步提高市场地位的能力。报告最终对市场排名、领导者的判断仅适用于本年度中国容器安全发展周期。

本报告所有图、表、文字中的数据均源自弗若斯特沙利文咨询（中国）及头豹研究院调查，数据均采用四舍五入，小数计一位。

01 容器安全防护的挑战

容器安全的建设仍处于起步爬坡阶段，需要克服新旧双重威胁存在未知风险、普遍的镜像漏洞、数据的低可观测性等难题，传统的安全防护手段已经难以适配云原生环境。

02 容器运行时防护方案

点对点的攻防映射在攻防演练和实践中逐渐训练出容器安全的防守技术网，但要满足用户的容器安全需求，需要构建覆盖容器使用全生命周期和全架构层次的容器安全体系框架。

03 镜像仓库安全方案

容器、镜像、镜像仓库是容器技术安全的三大核心组件。镜像仓库安全是构筑全生命周期的容器安全核心之一，其中包括版本可信、连接安全、认证和授权安全等层面。

04 产品形态发展方向

容器安全防护体系需覆盖容器技术的全生命周期，针对容器规划、安装、配置、部署、运维、处置等不同阶段，设置相应的动态应用策略。

研究框架

◆ 容器架构风险与安全挑战概述	-----	5
• 容器架构的原生安全性		
• 容器安全威胁与风险		
• 容器安全防护的挑战		
◆ 容器安全技术发展综述	-----	11
• 容器安全防护架构体系		
• 容器运行时防护方案		
• 容器仓库安全方案		
◆ 中国容器安全市场发展趋势	-----	15
• 核心特征发展方向		
• 产品形态发展方向		
◆ 中国容器安全市场竞争态势	-----	18
• 容器安全竞争力评价维度		
• 容器安全综合竞争力表现		
➢ 领导者：青藤云安全		
➢ 领导者：腾讯安全		
➢ 领导者：小佑科技		
➢ 领导者：博云		
◆ 名词解释	-----	29
◆ 方法论	-----	30
◆ 法律声明	-----	31

CONTENTS

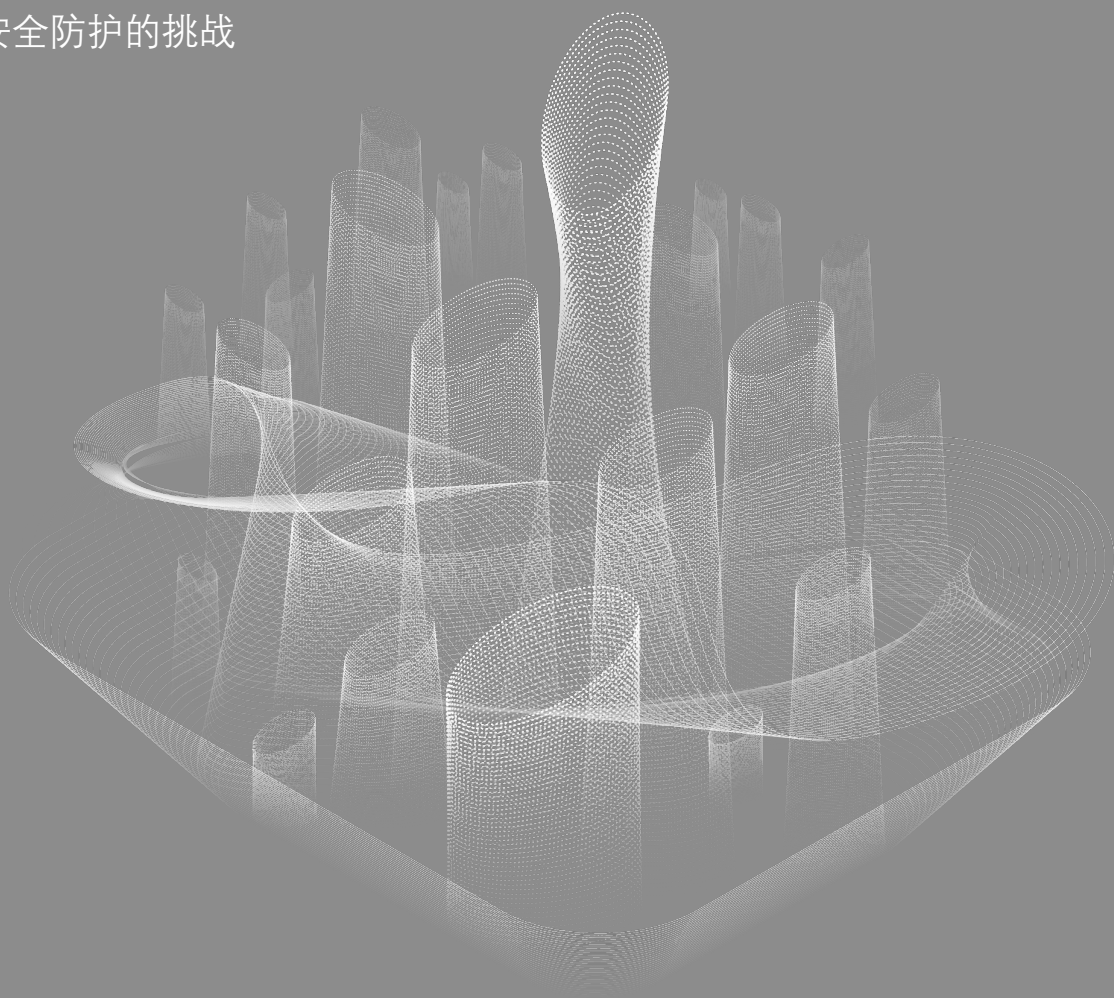
◆ Overview of container structure risks and security challenges	-----	5
• Native security feature of container		
• Container security threats and risks		
• Challenges of container security protection		
◆ Overview of the development of container security technique	-----	11
• Container security protection structure		
• Protection scheme during container operation		
• Container warehouse security plan		
◆ Development trend of container security market in China	-----	15
• Development direction of core feature		
• Development direction of product form		
◆ competitive landscape of container security market in China	-----	18
• Assessment scoring		
• Comprehensive vendor assessment – Frost Radar		
➤ Leading competitor : Qingteng		
➤ Leading competitor : Tencent		
➤ Leading competitor : DOSEC		
➤ Leading competitor : BoCloud		
◆ Terms	-----	29
◆ Methodology	-----	30
◆ Legal Statement	-----	31



Chap 1

容器架构风险与安全挑战概述




- 容器架构的原生安全性
- 容器的安全威胁与风险
- 容器安全防护的挑战



容器架构的原生安全性

- 如何帮助企业尽可能获得使用容器等技术带来的价值，并且降低使用容器而带来的安全代价，是容器安全赛道中的各厂商竞相追逐的目标。

面向不同虚拟环境的架构

	传统物理机架构	虚拟机架构	容器架构
定义	$\text{物理服务器} + \text{操作系统} + \text{环境配置} + \text{程序代码} = \text{应用}$	$\text{物理服务器} \times \left[\begin{array}{l} \text{操作系统} \\ \text{环境配置} \\ \text{程序代码} \end{array} \right] = \text{应用}$ <p style="text-align: center;">虚拟机封装</p>	$\left(\text{物理服务器} + \text{操作系统} \right) \times \left[\begin{array}{l} \text{环境配置} \\ \text{程序代码} \end{array} \right] = \text{应用}$ <p style="text-align: center;">容器封装</p>
理解	 <p>独栋别墅</p> <p>独立地基、独立大门 一栋楼一户人家</p>	 <p>公寓大楼</p> <p>共享地基、共享大门 一栋楼多套房、一套房一户人家 独立卫生间、独立厨房、独立宽带</p>	 <p>胶囊旅馆</p> <p>共享地基、共享大门 一套房多个隔间、一个隔间一位租户 共享卫生间、共享厨房、共享宽带</p>

来源：头豹研究院

❑ 虚拟机与容器

虚拟机架构是从操作系统层开始建立一个可以用来执行整套操作系统的沙盒独立执行环境。

容器架构是直接将一个应用程序所需的相关程序代码、函式库、环境配置文件都打包起来建立沙盒执行环境，通过“打包”和“标准化”的理念凸显对“可移植性”、“敏捷”和“弹性”的需求导向。

❑ 隔离程度

在虚拟机中的每个应用都拥有独立的内核，具备软件层完全隔离的优势。

在容器中的每个应用都是共享宿主机的内核的，仅具备进程级隔离，配置环境的不细致会让容器直接地与其他容器发生交互。

❑ 容器安全

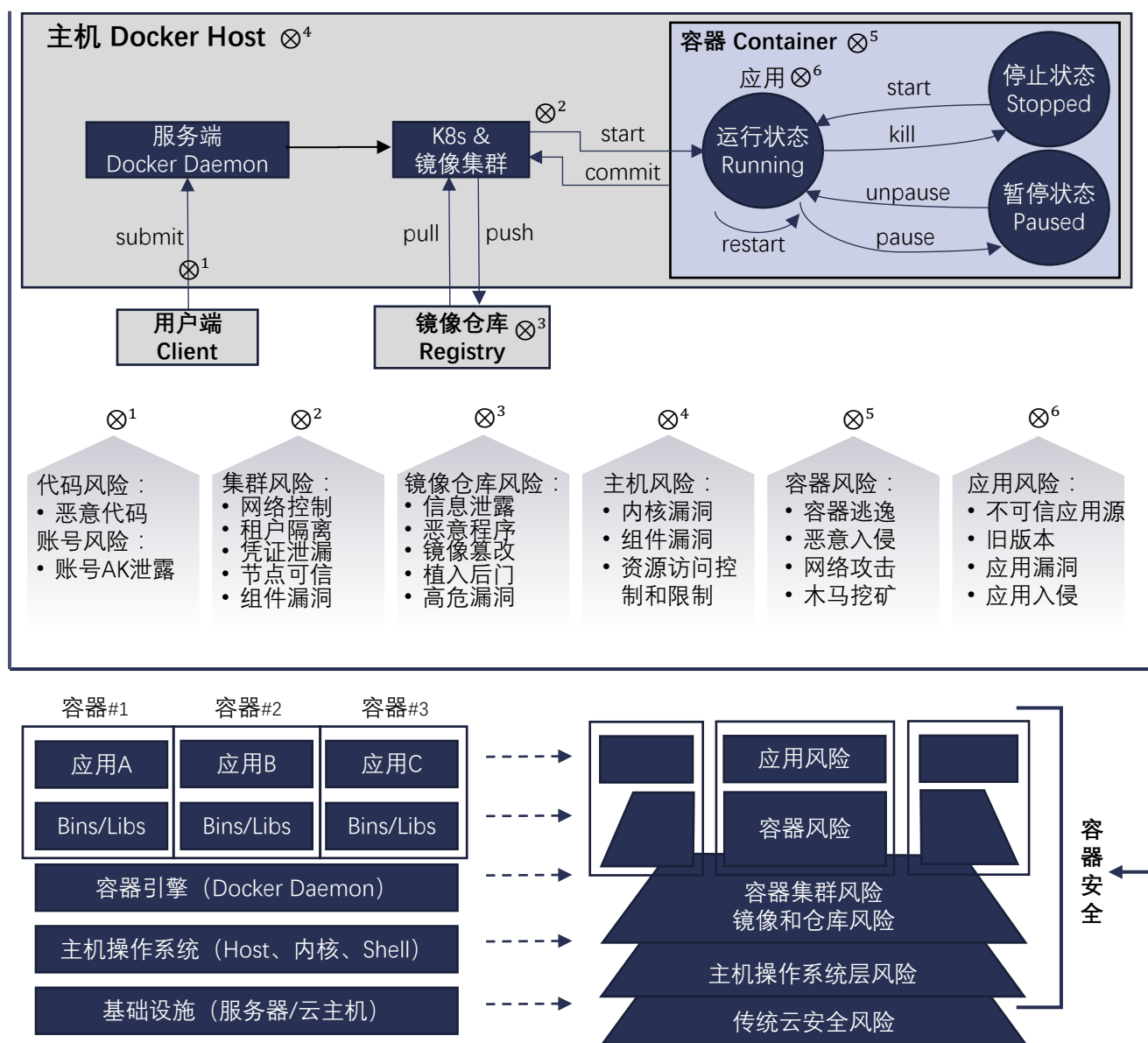
虚拟机和容器代表了两种需求，兴一利必生一弊：虽然容器在轻量化方向优势显著，但其代价则是由于资源隔离的不彻底和“打包”服务限制的数据可见性而带来的原生安全隐患。

在如今云原生体系向行业的不断渗透的趋势中，如何帮助企业尽可能获得使用容器等技术带来的价值，并且降低使用容器而带来的安全代价，正是容器安全赛道中的各厂商竞相追逐的目标。

容器的安全威胁与风险

- 容器的安全保障需要同时应对来自主机层、Docker层、容器层和应用层的攻击面威胁。对于企业和安全厂商而言，充分了解容器环境和攻击状况，是建立防御体系的第一步。

容器的运行机制、构成结构与风险来源



来源：Docker、腾讯安全、成都信息工程大学、Cloudman、头豹研究院

对容器的运行机制和容器架构的组成分别展开解析，与传统平台相比，容器生态系统涉及的组件、工具和代码通道更多，容器用户需要确保具有专门构建的全栈安全性，以解决容器化应用程序在构建、部署和运行的安全要求。同时，容器的快速广泛采用也创造了一个“安全左移”的机会，保护容器从开发到 CI/CD 管道再到运行时，并在开发和安全团队之间架起桥梁。

容器与镜像ATT&CK攻防对抗知识库

初始入侵 Initial Access

对外漏洞	远程服务
投毒镜像	账户泄露

- 云账号AK泄露
- 使用恶意镜像
- K8s API Server未授权访问
- K8s configfile 泄露
- Docker Daemon公网暴露
- 容器内应用漏洞入侵
- 主节点SSH登陆凭证泄露
- 私有镜像库暴露
- Dashboard暴露

下发指令 Execution

容器服务	创建后门
脚本	RCE

- 通过kubectl进入容器
- 创建后门容器
- 通过K8s控制器部署后门容器
- 利用Service Account连接API Server执行指令
- 带有SSH服务的容器
- 通过CloudShell下发指令
- Bash/cmd 命令行执行脚本

持久控制 Persistence

挂载Host	冒充镜像
计划服务	创建账号

- 部署远控客户端
- 可写挂载目录hostPath
- K8s cronjob持久化
- 在私有镜像库的镜像中植入后门修改核心组件访问权限
- 添加创建账户
- 冒充正常镜像签名
- 部署计划任务

权限提升 Privilege Escalation

容器逃逸	漏洞
账号权限	挂载目录

- 部署特权容器
- 集群binding添加用户权限
- 利用挂载目录逃逸
- 访问云资源
- 利用Linux内核漏洞逃逸
- 利用Docker漏洞逃逸
- 利用K8s漏洞进行提权
- 容器内访问docker.sock逃逸
- 利用Linux Capabilities逃逸
- Host命名空间、Cgroups滥用

躲避防御 Defense Evasion

名称伪装	路径伪装
卸载杀软	日志清理

- 容器及宿主机日志清理
- K8s Audit日志清理
- 利用系统Pod名称伪装
- 利用路径伪装
- 通过代理或匿名网络访问K8s API Server
- 卸载安全产品Agent
- 创建影子API Server
- 创建超长annotations使K8s Audit日志解析失败

窃取凭证 Credential Access

账号泄露	API凭证
配置文件	

- K8s Secret泄露
- 云产品AK泄露
- K8s Service Account凭证泄露
- 容器API凭证泄露
- 应用层API凭证泄露
- 利用K8s准入控制器窃取信息
- 窃取应用凭证配置文件

探测信息 Discovery

Kubelet API	内网扫描
私有镜像库	元数据API

- 访问K8s API Server
- 访问Kubelet API
- Cluster内网扫描
- 访问K8s Dashboard所在Pod
- 访问私有镜像库
- 访问云厂商服务接口
- 通过NodePort访问Service
- 实例元数据API

横向移动 Lateral Movement

云资源	内网渗透
宿主机	K8s组件

- 窃取凭证攻击云服务
- 窃取凭证攻击其他应用
- 通过Service Account访问K8s API
- Cluster内网渗透
- 通过挂载目录逃逸到宿主机
- 访问K8s Dashboard
- 攻击第三方K8s插件
- 访问Tiller endpoint

破坏目标 Impact

数据破坏	资源劫持
拒绝服务	

- 破坏系统及数据
- 劫持资源
- DoS攻击
- 加密勒索

来源：MITRE、阿里云、Azure、腾讯安全、青藤云安全、头豹研究院、沙利文

❑ 容器与镜像ATT&CK攻防对抗知识库

ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) 是一个攻击行为知识库和威胁建模模型。容器与镜像ATT&CK覆盖了K8s编排层、Docker容器层和应用层的攻击行为，还包括了容器相关的恶意软件威胁。

对于企业和安全厂商而言，充分了解容器环境和攻击状况，是建立防御体系的第一步。企业用户，利用容器ATT&CK模拟红蓝对抗，有助于了解K8s中的安全风险和关键攻击媒介，并且基于此有助于制定正确的检测和缓解策略来应对这些风险，提供全面的保护。

从容器ATT&CK看模拟攻击路线



来源：MITRE、安全狗、头豹研究院

❑ 一种容器攻击路线

1. 首先攻击者发现K8s暴露在外的微服务开始渗透，利用容器内的应用漏洞入侵并成功获取到容器的shell。
2. 然后探测内网环境，进行Cluster内网的扫描，探测存活主机和存活主机开放的端口，以此可以确定K8s集群对外暴露的微服务。扫描还可以得到Pod的IP，条件允许的话可以直接在容器内利用。
3. 通过扫描开放的组件的默认端口，进而发现K8s的未授权访问，在获取到shell的容器内利用kubectI用户端操纵K8s资源。
4. 利用K8s的未授权访问创建后门容器。
5. 从挂载宿主机目录并向宿主机目录写入定时的反弹shell任务。
6. 通过挂载目录逃逸到宿主机完成主机的横向移动。
7. 最后利用K8s cronjob进行持久化控制。

容器安全防护的挑战

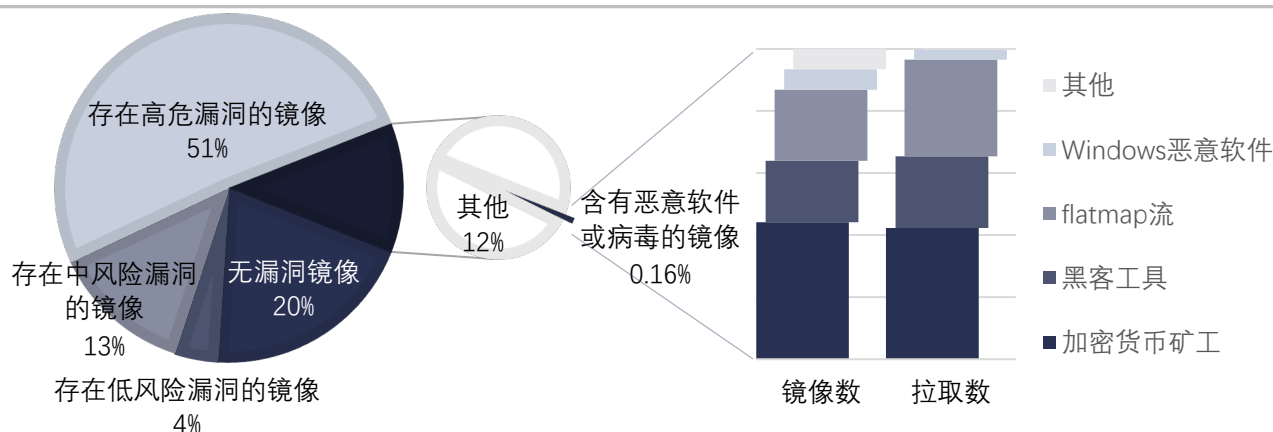
- 容器安全的建设仍处于起步爬坡阶段，需要克服新旧双重威胁存在未知风险、普遍的镜像漏洞、数据的低可观测性等难题，传统的安全防护手段已经难以适配云原生环境。

容器安全的难题和挑战

从容器的攻击路径中可以体现，针对容器可选择的攻击面和攻击手段十分广泛，同时通过横向移动可以轻松扩大入侵价值。相较于攻击视角的“低门槛高收益”的特性，容器在安全防守的视角中却是“荆棘载途”。容器安全的建设仍处于起步爬坡阶段，需要克服几大难题：

1. 新旧双重威胁，未知风险。包括漏洞利用、暴力破解、权限提升在内的传统攻击手段对容器一样奏效。同时，云原生使用的一系列技术也意味着容器存在大量未知的风险隐患，黑客也不断衍生出新的攻击手段，如投毒镜像、容器逃逸、集群API调用等，容器防护需要为容器环境创新定制专门的防御策略。
2. 镜像漏洞普遍。由于镜像的本质是静态存档文件，容器必须在上游的镜像中进行更新并重新部署。容器环境常见风险是使用的镜像版本存在漏洞而使部署的容器存在漏洞。Docker Hub中的镜像普遍存在不同程度的漏洞，这些有安全隐患的漏洞都有可能被恶意利用。

Docker中镜像的漏洞风险现况与恶意镜像构成



来源：Prevasio, Docker, 头豹研究院

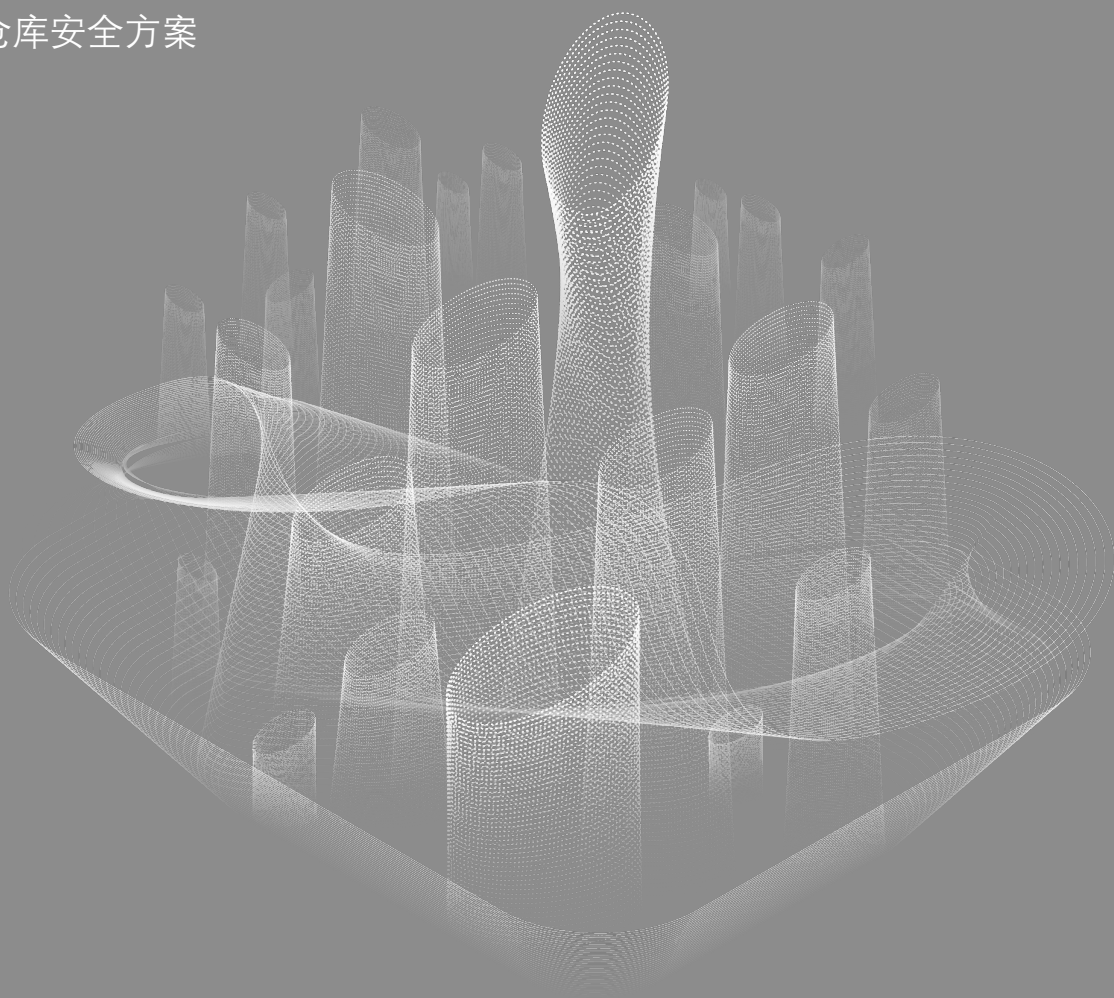
3. 数据可观测性低，攻击溯源难。容器的生命周期短，动态变化快，超过50%容器从上线到下架的整个生命周期不超过1天。同时容器的轻量化部署原理也使主机上的可以承载上百个容器的同时运行，集群内部的网络流量和通信端口总量大幅增加。容器化环境的容器应用部署密度和容器变化频率都远高于传统环境，攻击威胁的检测、追踪和溯源的难度显著。



Chap 2

容器安全技术发展综述

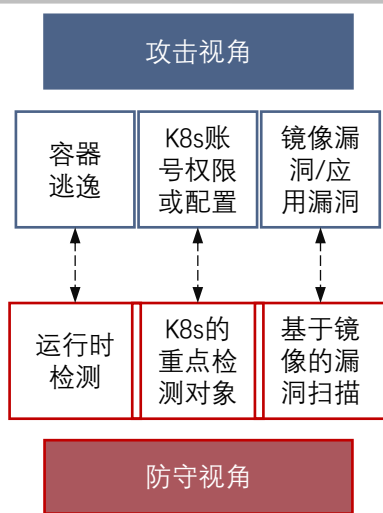
- 容器安全防护架构体系
- 容器运行时防护方案
- 镜像仓库安全方案



容器安全防护架构体系

- 点对点的攻防映射在攻防演练和实践中逐渐训练出容器安全的防守技术网，但要满足用户的容器安全需求，需要构建覆盖容器使用全生命周期和全架构层次的容器安全体系框架。

容器安全的攻防视角



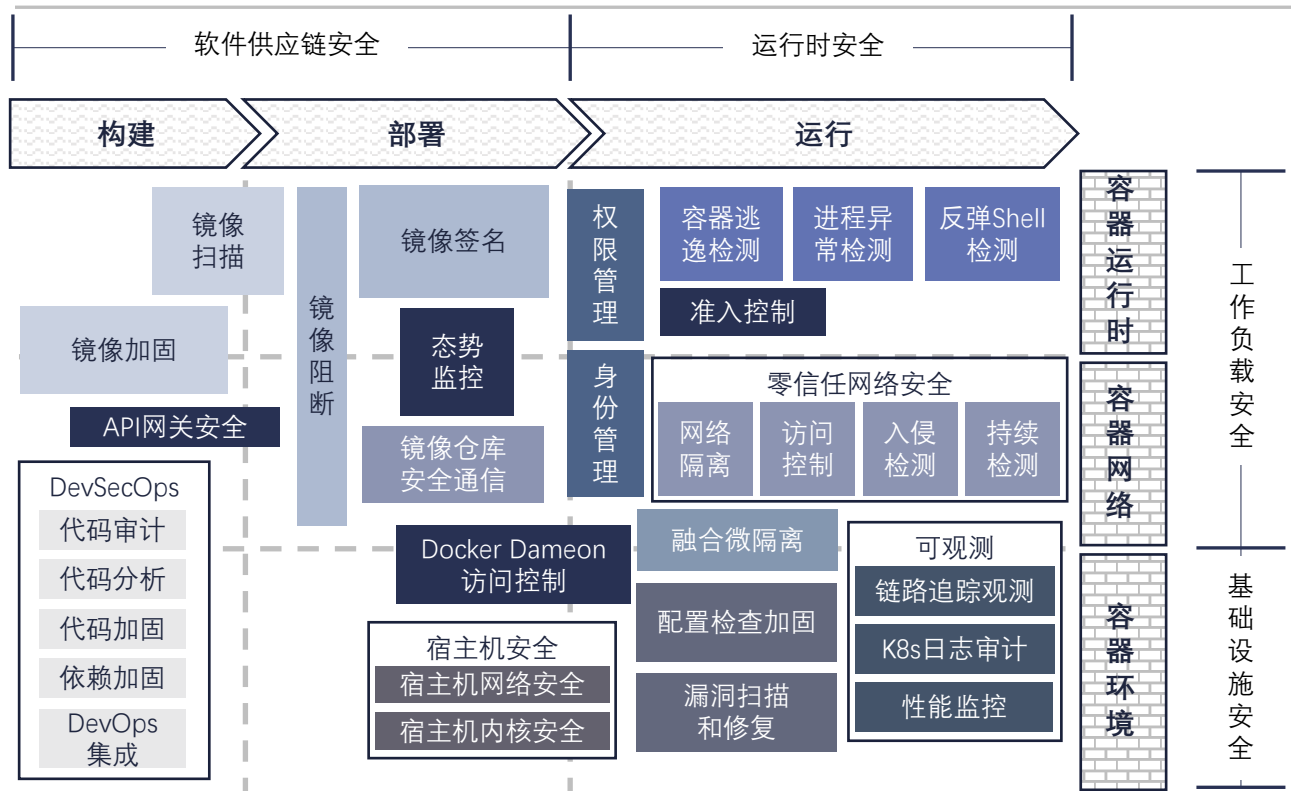
从攻击视角到防御视角

容器安全体系的建立，最初是攻防映射的点对点关系，常见攻防视角有：

- 针对镜像漏洞/应用漏洞的攻击。防守方可以通过基于版本与漏洞库的比对完成基于镜像的漏洞扫描。
- 利用暴露的K8s配置或账号发起入侵。防守方可以通过针对主流的攻击场景进行检测识别，同时关注账号安全层面的异常状况。
- 容器逃逸会对容器集群有很大的危害。防守方可以通过对攻击场景的理解建立黑白名单完成运行时监测。

点对点的攻防映射在攻防演练和实践中逐渐训练出容器安全的防守技术网，但要满足用户的容器安全需求，需要构建覆盖容器使用全生命周期和全架构层次的容器安全体系框架。

全生命周期和全架构的容器安全体系



来源：火山引擎、NeuVector、腾讯安全、青藤云安全、头豹研究院、沙利文

容器运行时防护方案

- 容器运行时安全是容器安全与传统安全相比最考验安全供应商在面对容器环境威胁如何提供创新有效产品技术的方面。其中包含了运行时监控、容器集群隔离、融合微隔离、运行时权限配置、持续开发和部署等基础能力。

关于容器运行时安全体系的建立，防守策略可以从攻守视角拆分：

- 攻击方视角中应用行为监测机制实时检测攻击方的扫描、暴力破解、反弹shell、提权、Web后门、异常链接、挂马等攻击行为；
- 防守方视角中建立文件操作、网络访问、系统调用、资产画像等基线，用以缩小攻击面。完备运行时监控、运行时权限配置、容器集群隔离、微隔离等功能模块的正常运行预防攻击的发生。

容器运行时威胁防护安全策略的主流基础策略和创新策略

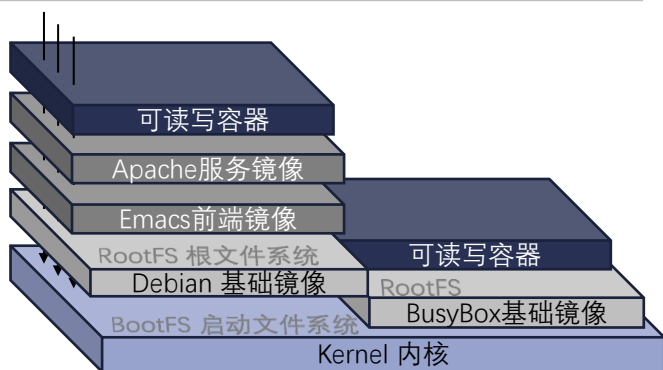


来源：头豹研究院，沙利文

镜像仓库安全方案

- 容器、镜像、镜像仓库是容器技术安全的三大核心组件。镜像仓库安全是构筑全生命周期的容器安全核心之一，其中包括版本可信、连接安全、认证和授权安全等层面。

容器基础架构与镜像的作用



来源：Docker, 头豹研究院

□ 镜像安全

中层镜像层，主要包含上层程序的代码和运行程序所需的系统环境，容器要运行需要先构建镜像，镜像定义了容器运行的内容和运行程序所需的系统环境，包含软件包、版本号等，容器的镜像安全直接关系到容器的安全运行。

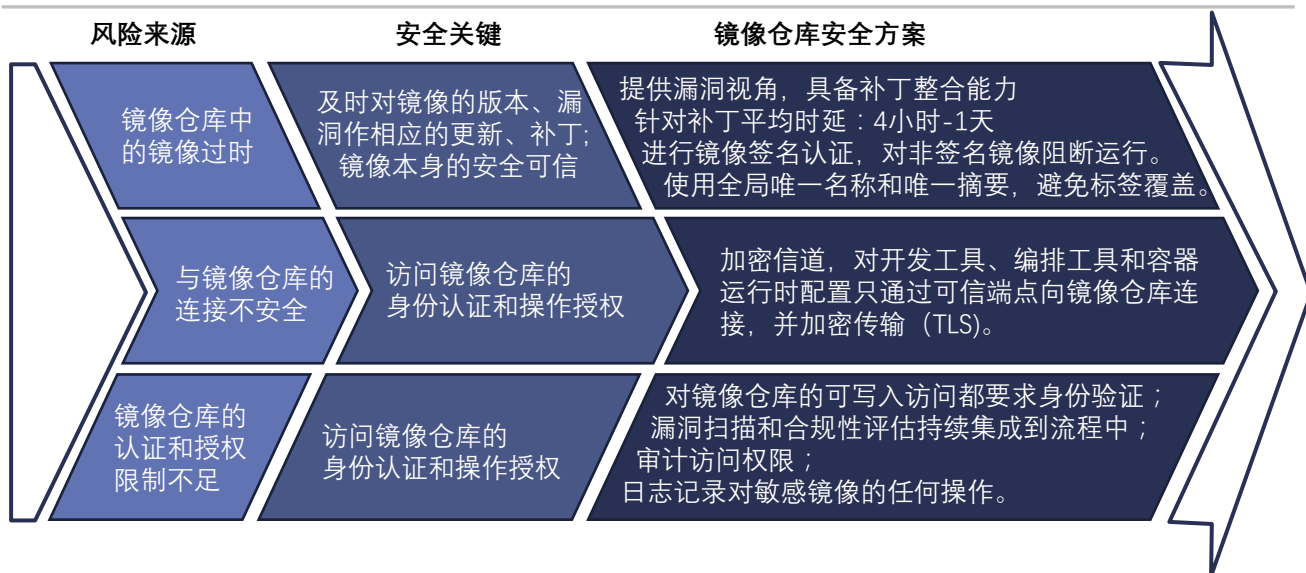
容器安全对基础镜像提供的安全验证手段通常已经包括镜像软件漏洞扫描、镜像中病毒扫描、镜像敏感信息扫描、镜像配置合规扫描等方面。

□ 镜像仓库的安全诉求

Docker Hub由于免费开源的特性，长期是存储镜像的主流选择，但由于有限的存储空间、有限的上传/拉取速度，加之作为公有云服务器不适合上传企业内部的敏感开发项目代码，越来越多的企业用户正在基于Registry构建私有镜像仓库。

虽然使用Registry使容器用户可以在全面管理控制镜像的存储、分配，紧密集成到内部开发流程等方面获得优势，但是对镜像仓库的安全性保障却面临更高的挑战。在Registry 1.0代版本中，用户可以随意篡改Layer中的文件，尽管Docker Registry 2.0版本也在安全性上做了诸多优化，但依然需要警惕安全性。

镜像仓库的安全策略



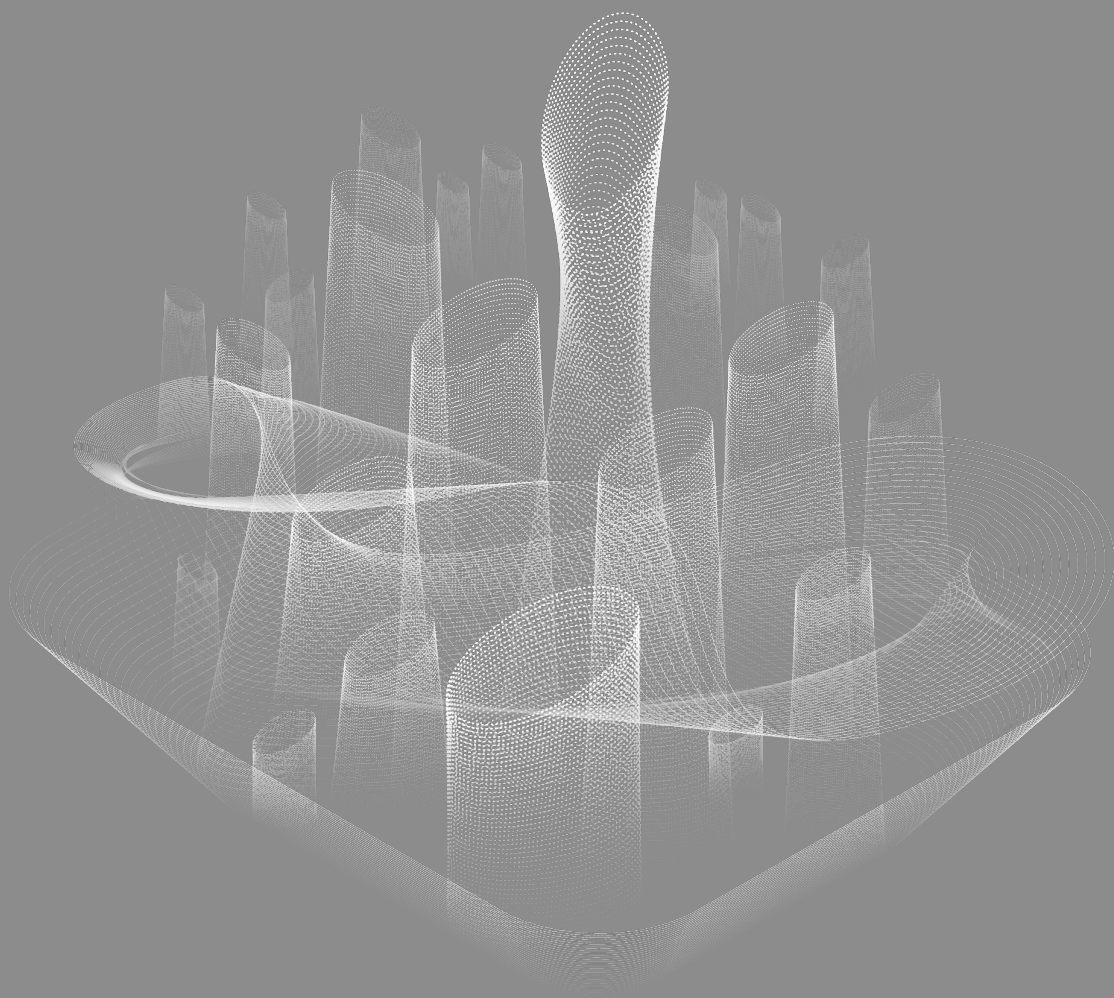
来源：青藤云安全, 头豹研究院、沙利文



Chap 3

中国容器安全市场发展趋势

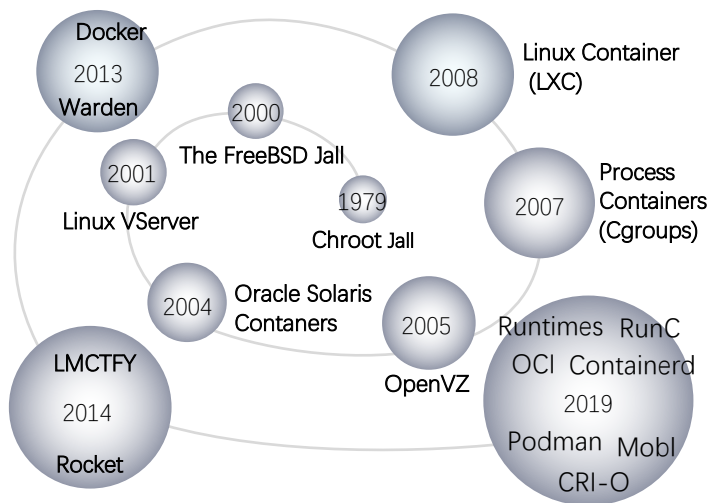
- 核心特征发展方向
- 产品形态发展方向



核心特征发展方向

- 受共享内核特点的影响，容器在多租户场景的应用存在诸多安全风险，未来需要服务商通过细化隔离层级、拓展隔离维度等方式，确保容器在不同应用模式下实现用户资产安全性。

容器技术生态的发展历程



来源：FAUN，头豹研究院

隔离性升级确保容器安全性

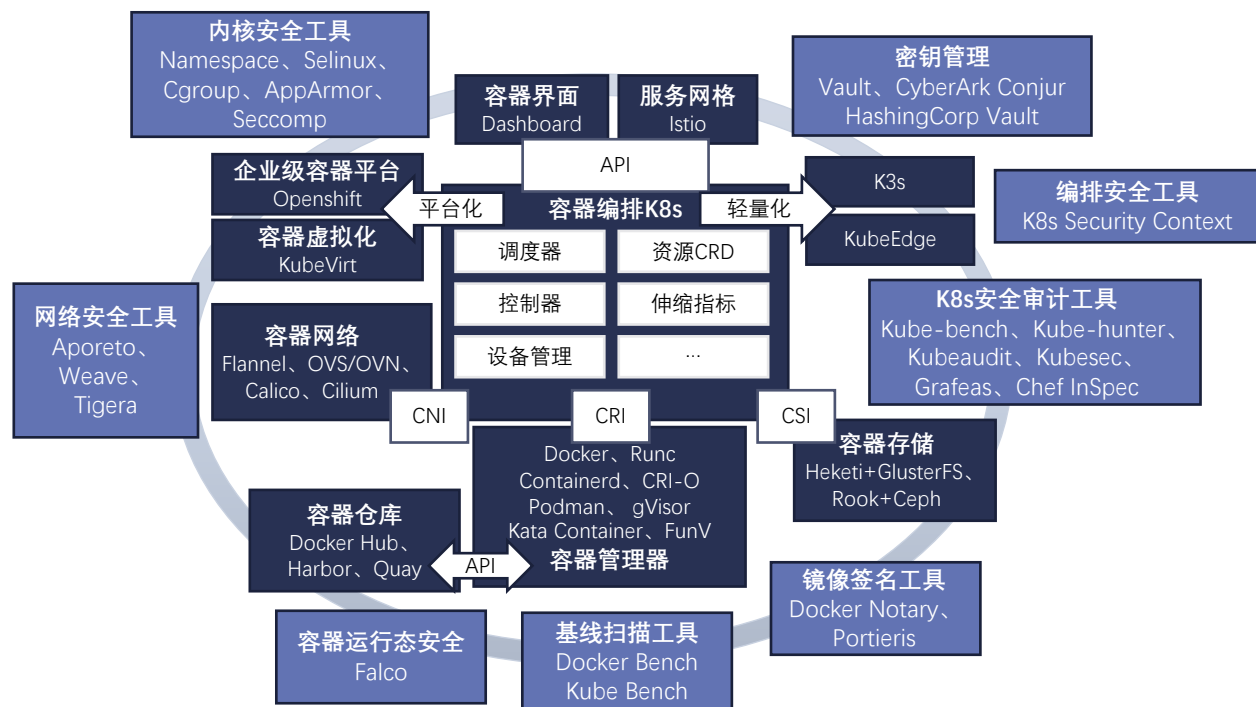
共享内核的特征使得容器技术在安全性方面存在缺陷，容器应用场景因此受到限制。云原生产品更多应用于单租户场景，在多租户模式下，容器特征无法确保运行时隔离、网络隔离、镜像隔离等安全性。

隔离颗粒度进一步细化，隔离层深化

容器应用安全涉及运行时隔离、镜像隔离、网络隔离、磁盘存储隔离等；为细化和深化隔离程度，服务商可通过提供网络策略隔离、存储隔离、镜像引用隔离等模式确保容器用户资产安全性。

此外，容器技术栈与开源安全工具的融合应用将有效提升隔离策略在网络、镜像、存储等层面的渗透和协同。

容器生态技术栈与开源安全工具

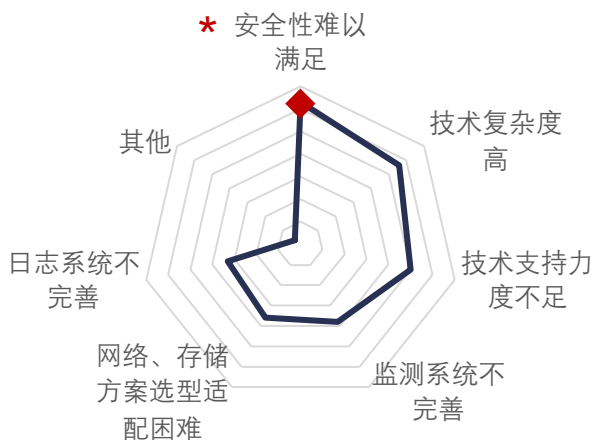


来源：CSDN，头豹研究院

产品形态发展方向

- 容器安全防护体系需覆盖容器技术的全生命周期，针对容器规划、安装、配置、部署、运维、处置等不同阶段，设置相应的动态应用策略。

用户使用容器技术时存在的问题



□ 容器生命周期安全

在对用户使用容器体验的调研中，90%以上的用户表达了对容器安全性天生存在缺陷的担忧，容器应用场景受到限制。容器与的宿主机共享内核，且容器技术本身建立在 Linux Namespace 和 Linux Cgroups 两项关键技术之上，而 Linux 内核本身所产生的漏洞会导致容器逃逸。容器运行时，攻击者可通过恶意镜像和修改容器配置入侵容器。在容器的配置、部署和运维等不同阶段，实际业务线具有针对性的需求，安全规划需渗透容器运行全生命周期。

容器全生命周期安全因素考虑及安全规划

	潜在问题	对应策略
研究阶段	传统开发和安全策略不适用于容器	结合容器的特点，对事件响应取证、漏洞管理策略规划等进行调整
设计阶段	路径检查未覆盖文件和数据	容器环境下，在对事件做出响应时，通过特定存储供应商进行内容的离线检查
测试和部署阶段	缺乏内省能力，隔离性不足	于部署前查验身份、连接、应用、管理、技术的安全性，对传统安全控制和技术配置进行调整
运维阶段	事件响应计划未同步，管理策略不适用	确保容器运行团队充分了解自身在事件响应计划中的位置；掌握容器所有者和敏感度级别，集成相关数据
处置阶段	容器部署和运行记录难以保存、取证	对容器和镜像的销毁进行规划，完成处置前数据提取、容器撤销或删除行为

容器全生命周期安全解决方案案例：青藤云安全基于宿主机Agent开发解决方案

1、构建阶段

容器编码阶段带入安全思维，实现源头控制；利用代码审计工具检测漏洞；通过精简和加固镜像的方式减少攻击面；在投产前对镜像进行漏洞扫描，并且对镜像仓库采取周期性扫描措施。

2、分发阶段

(a)传输中防篡改：通过设置多重签名、进行镜像校验等方式确保镜像未被篡改；(b)访问控制：镜像仓库、编排工具等模块集成在统一认证平台中（如LDAP）；(c)镜像使用习惯：避免使用未漏扫、未加固的镜像。

3、运行阶段

(a)运行时环境安全：确保运行时安全配置、Docker安全配置；(b)持续性安全：确保网络与容器连接安全，网络连接加密、周期性入侵检测、运行时进行漏扫、恶意容器隔离等。

来源：头豹研究院

名词解释

- ◆ **Docker**: Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的镜像中，然后发布到任何流行的 Linux 或 Windows 操作系统的机器上，也可以实现虚拟化。容器是完全使用沙箱机制，相互之间不会有任何接口。
- ◆ **镜像**： Docker 镜像 是一个特殊的文件系统，除了提供容器运行时所需的程序、库、资源、配置等文件外，还包含了一些为运行时准备的一些配置参数(如匿名卷、环境变量、用户等)。镜像不包含任何动态数据，其内容在构建之后也不会被改变。
- ◆ **仓库**：镜像构建完成后，可以很容易在当前宿主机上运行，但是，如果需要在其它服务器上使用这个镜像，我们就需要一个集中的存储、分发镜像的服务， Docker Registry 就是这样的服务。
- ◆ **K8s**： K8s 是 Kubernetes 的缩写， Kubernetes 是一个开源的，用于管理云平台中多个主机上的容器化的应用， Kubernetes 的目标是让部署容器化的应用简单并且高效(powerful)， Kubernetes 提供了应用部署，规划，更新，维护的一种机制。
- ◆ **容器编排**：让开发运维人员或自动化工具，能够从镜像仓库中获取镜像，将这些镜像部署到容器中，并管理正在运行的容器。
- ◆ **Kappa架构**：针对Lambda架构的需要维护两套程序等以上缺点， Kappa架构的核心思想是通过改进流计算系统来解决数据全量处理的问题，使得实时计算和批处理过程使用同一套代码。
- ◆ **Scatter/Gather模型**：在多个缓冲区上实现一个简单的I/O操作，比如从通道中读取数据到多个缓冲区，或从多个缓冲区中写入数据到通道。
- ◆ **MapReduce模型 (Hadoop)**：通过把对数据集的大规模操作分发给网络上的每个节点实现可靠性；每个节点会周期性的返回它所完成的工作和最新的状态。
- ◆ **Massively Parallel Processing (MPP)**：采用Shared-nothing架构，每个节点使用单独资源，拥有最佳运行环境。流水线执行无需等待，数据内存存储，无磁盘IO。
- ◆ **Multidimensional OLAP (MOLAP)**：基于直接支持多维数据和操作的本机逻辑模型。数据物理上存储在多维数组中，并且使用定位技术来访问它们。
- ◆ **Relational OLAP (ROLAP)**：将分析用的多维数据存储存储在关系数据库中。这种方式依赖SQL语言实现传统OLAP的切片和切块功能，本质上，切片和切块等动作都等同于在SQL语句中添加“WHERE”子句。
- ◆ **Hybrid OLAP (HOLAP)**：通过允许同时使用多维数据库 (MDDB) 和关系数据库 (RDBMS) 作为数据存储来弥合这两种产品的技术差距。

方法论

- ◆ 头豹研究院布局中国市场，深入研究10大行业，54个垂直行业的市场变化，已经积累了近50万行业研究样本，完成近10,000多个独立的研究咨询项目。
- ◆ 研究院依托中国活跃的经济环境，从纵深防御、快速响应、轻量化部署等领域着手，研究内容覆盖整个行业的发展周期，伴随着行业中企业的创立，发展，扩张，到企业走向上市及上市后的成熟期，研究院的各行业研究员探索和评估行业中多变的产业模式，企业的商业模式和运营模式，以专业的视野解读行业的沿革。
- ◆ 研究院融合传统与新型的研究方法，采用自主研发的算法，结合行业交叉的大数据，以多元化的调研方法，挖掘定量数据背后的逻辑，分析定性内容背后的观点，客观和真实地阐述行业的现状，前瞻性地预测行业未来的发展趋势，在研究院的每一份研究报告中，完整地呈现行业的过去，现在和未来。
- ◆ 研究院密切关注行业发展最新动向，报告内容及数据会随着行业发展、技术革新、竞争格局变化、政策法规颁布、市场调研深入，保持不断更新与优化。
- ◆ 研究院秉承匠心研究，砥砺前行的宗旨，从战略的角度分析行业，从执行的层面阅读行业，为每一个行业的报告阅读者提供值得品鉴的研究报告。

法律声明

- ◆ 本报告著作权归头豹所有，未经书面许可，任何机构或个人不得以任何形式翻版、复刻、发表或引用。若征得头豹同意进行引用、刊发的，需在允许的范围内使用，并注明出处为“头豹研究院”，且不得对本报告进行任何有悖原意的引用、删节或修改。
- ◆ 本报告分析师具有专业研究能力，保证报告数据均来自合法合规渠道，观点产出及数据分析基于分析师对行业的客观理解，本报告不受任何第三方授意或影响。
- ◆ 本报告所涉及的观点或信息仅供参考，不构成任何证券或基金投资建议。本报告仅在相关法律许可的情况下发放，并仅为提供信息而发放，概不构成任何广告或证券研究报告。在法律许可的情况下，头豹可能会为报告中提及的企业提供或争取提供投融资或咨询等相关服务。
- ◆ 本报告的部分信息来源于公开资料，头豹对该等信息的准确性、完整性或可靠性不做任何保证。本报告所载的资料、意见及推测仅反映头豹于发布本报告当日的判断，过往报告中的描述不应作为日后的表现依据。在不同时期，头豹可发出与本报告所载资料、意见及推测不一致的报告或文章。头豹均不保证本报告所含信息保持在最新状态。同时，头豹对本报告所含信息可在不发出通知的情形下做出修改，读者应当自行关注相应的更新或修改。任何机构或个人应对其利用本报告的数据、分析、研究、部分或者全部内容所进行的一切活动负责并承担该等活动所导致的任何损失或伤害。

深度研究小组负责人

李庆

☎ 13149946576

✉ livia.li@frostchina.com

主笔分析师

胡俊杰、贾雁

☎ 18576027961

✉ jackey.hu@frostchina.com

🌐 www.frostchina.com ; www.leadleo.com

📺 <https://space.bilibili.com/647223552>

📱 <https://weibo.com/u/7303360042>

©弗若斯特沙利文咨询（中国）

©头豹研究院

