网络空间威胁对抗防御技术研讨会
暨 第十届安天网络安全冬训营

谙海横流

执行体全量识别与精细管控

安天智甲容器安全检测系统

安天 | 云安全中心

目 录

01 容器安全背景和现状

容器安全背景

在数字化飞速发展的背景下，政府、金融、互联网、制造业等也都在进行不同程度的云化，然而云原生技术（容器、微服务、kubernetes等）正在凭借着**敏捷性、可靠性、高弹性、易扩展、持续更新**等特点在蓬勃发展，成为各行业业务创新的重要推动力。

从重保的角度来看，今年8月份举行的攻防演练**明确了容器失陷的扣分标准**，每失陷一个容器扣10分。基于集群与容器的数量关系，如果整个集群被攻陷，丢分会非常严重。

早期的容器安全是缺少国内规范和标准的，近两年时间国内的标准规范也在不断完善，

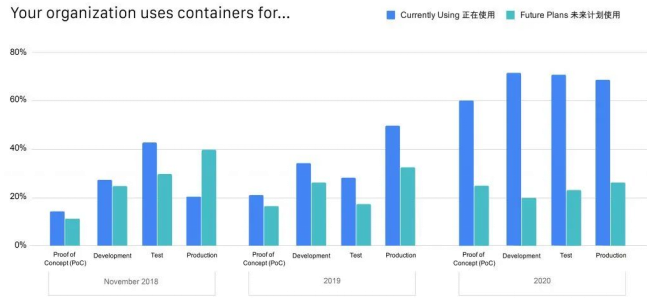
- 2020年，信通院发布容器安全标准，推出**可信容器云认证**；
- 2021年，信通院发布**云原生架构安全白皮书**；
- 2022年，CSA大中华区发布了**云原生安全技术规范（CNST）**，同时联合公安三所发布了**针对云原生和云应用的安全可信认证**。
- 2022 年，公安三所编写了等保2.0的**容器云安全增补部分**（征求意见稿）。

容器技术应用现状

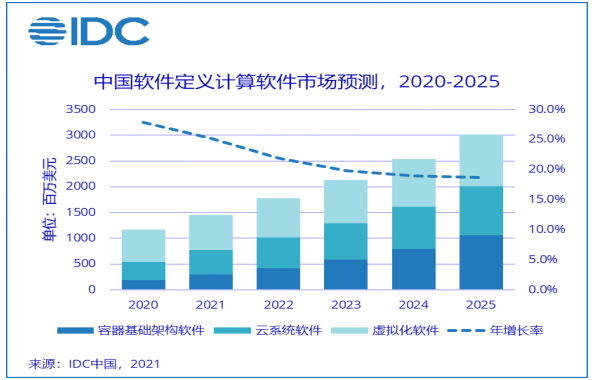
云原生市场中容器所扮演的角色也越来越重要，容器使用量持续大幅度增长成为加速企业数字化转型的利器，并且已经应用到了企业的核心业务。容器使用在所有应用类别中都有增长，包括测试、概念验证（POC）、开发和生产。

在CNCF调查报告中，94% 的调查对象至少在一个阶段使用容器。测试环境中使用容器的单位相比去年，已经增长了 150%，意味着开发和生产中容器的使用将会持续增长。开发阶段使用容器的单位已经从去年的 34% 跃升至 72%，POC 阶段的容器用户也已由 21% 增长到 60%。

同时IDC预测，容器软件市场在近几年呈爆发式增长，并且未来五年仍然会保持超过 40% 的复合增长率。到 2025 年，容器基础架构软件市场收入将与虚拟化软件市场、云系统软件市场齐平，成为近几年促使软件定义计算市场增长的新动力。



数据来源：CNCF调查报告



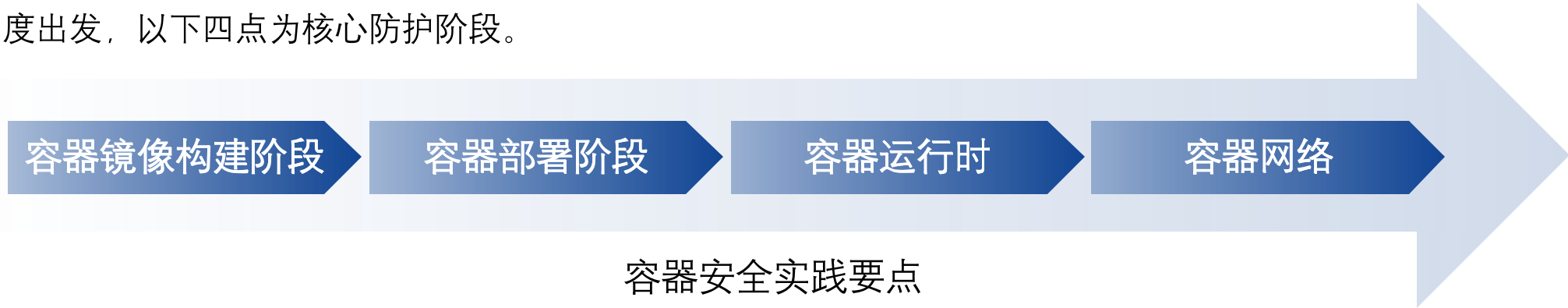
数据来源：IDC市场预测报告

02 容器安全要点

容器安全要点

容器作为支撑应用运行的重要载体为应用的运行提供了隔离与封装，也因此生成为云原生的基础设施，正因如此，其相关的**安全风险与威胁也在不断地涌现**。从容器的特性与使用方式中可体现出针对容器的**攻击面和攻击手段十分广泛**。

容器其本身**秒级启动、销毁以及持续频繁动态变化**的特征，极大的缩短了容器的生命周期，提升了容器安全防护的难度与挑战。满足企业的容器安全需求，需要**构建容器全生命周期的安全体系架构**，从攻防对抗角度出发，以下四点为核心防护阶段。



容器安全实践要点

容器安全要点

容器镜像构建阶段

随着容器越来越普及，作为容器运行基础的容器镜像自然成为了安全的第一道防线。如果容器基础镜像存在安全风险，那其潜在威胁将存在于整个容器生命周期，所带来的威胁甚至会高于外部黑客攻击。

容器部署阶段

容器镜像安全是容器安全的第一道防线，容器合规部署是第二道防线，非法启动滥用权限容器、非法引入高危容器镜像、非法引入不可信容器镜像等进入集群环境都会导致引发相关安全事故，甚至影响PaaS平台安全。

容器运行时

容器运行时安全问题中，最常见的为“容器逃逸、资源消耗”，通过漏洞逃逸出当前权限从而对宿主机或其他容器进行访问，同时导致资源耗尽，其中最常见的安全事件便为被利用进行挖矿。

容器网络

由于K8S集群默认不做网络隔离，同时传统网络攻击手段依然有效，黑客便可利用失陷容器对集群发起渗透。容器内部网络环境错综复杂，也会给企业带来“网络关系梳理难，策略控制难”

典型威胁与案例-镜像构建

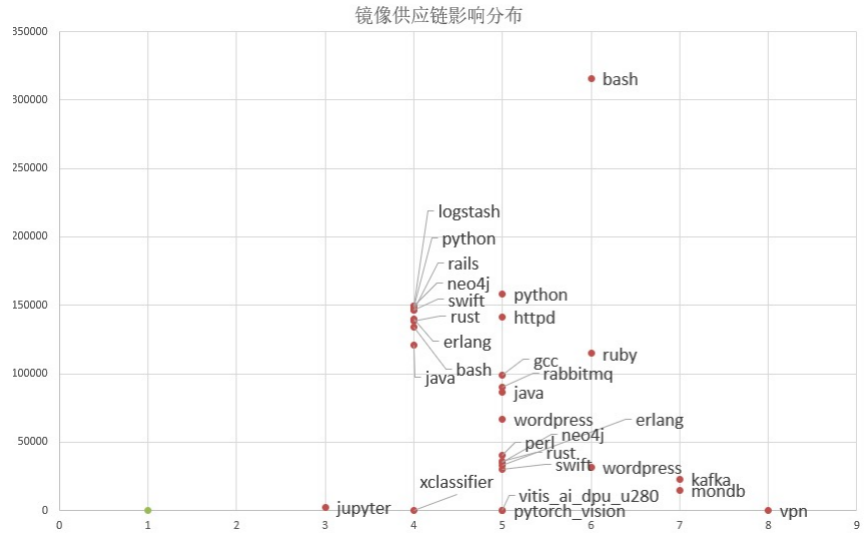
早在2020年，某安全团队发现有攻击者**在 Docker 容器中注入木马**，进行劫持挖矿，并**使用 Docker Hub 分发恶意镜像**。一个名为 azurenql 的账号从 2019 年开始一直处于活动状态，并**托管了六个用于包含挖掘 Monero 的恶意木马**。镜像中的挖矿木马通过使用 ProxyChains 和 Tor 的网络匿名化工具来逃避网络检测。账户上托管的镜像累计拉取次数**超过 200 万**。

黑产镜像

- 制作恶意黑产镜像、伪装躲藏
- 蠕虫传播
- 入侵利用
- 诱导下载
- 持续感染

恶意镜像**4-5个月**就可以达到**10w左右的下载量**

最大的黑产团伙镜像下载量**超过1.8亿**



数据来源：腾讯云鼎实验室

典型威胁与案例-容器部署

随着DevOps的流行，容器部署阶段也需要密切关注是否合规，是否有**高危配置**、**高危挂载**、**滥用特权**等不合规行为，若发现不及时，各类风险将会被带入集群内部，可能直接导致宿主机和整个集群被攻击者渗透，当发现不合规配置需要及时进行阻断、告警，防止威胁进一步扩散。

一般攻击者通过各种扫描工具探测暴露在外部的容器API，发现后会在系统内**部署恶意容器**，同时拉取容器镜像，并在部署时**指定"Privileged"参数**，使得容器可以**进行逃逸并执行相关挖矿程序**，并且所执行的这类ELF 格式的加密货币挖矿程序会进行各种各样的伪装躲避检查。例如**伪装成 nginx 的应用程序**，来让人以为它只是 HTTP 服务器，同时会隐藏在某些恶意镜像当中，如果发现不及时会被耗光资源导致业务崩溃。

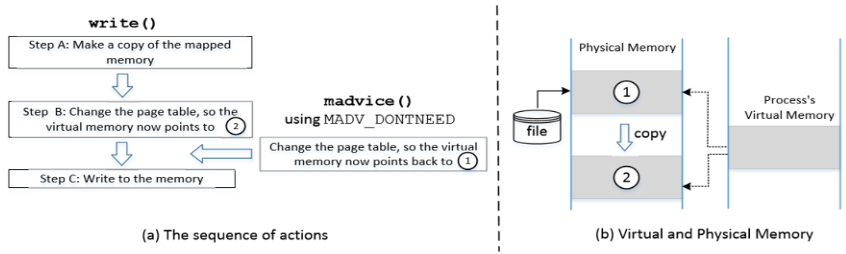
典型威胁与案例-容器运行时

runC容器逃逸漏洞(CVE-2019-5736)、Docker cp 命令(CVE-2019-14271)、脏牛漏洞-Dirty COW(CVE-2016-5195)都是已公开的容器逃逸CVE，其中Dirty Cow是公开后影响范围最广和最深的漏洞之一，脏牛漏洞之所以如此著名，主要在于其历史悠久且影响范围较广。脏牛漏洞自2007年9月 linux kernel 2.6.22 被引入，直到2018年之后才被彻底修复，其中存在10余年时间，影响在此之间的所有基于其中版本范围的Linux发行版。

虽然容器环境有命名空间进行隔离，但容器运行环境与宿主机共享系统内核，攻击者可通过此类漏洞轻松逃逸出容器环境，获取宿主机权限。容器环境可能由于各种原因引入不安全容器镜像并成功运行，引发容器逃逸事件发生，导致容器集群失陷，所以需要深入对容器运行行为进行监控，防止恶意行为发生。

漏洞成因

Dirty-COW 漏洞



漏洞利用过程

1. 启动两个线程
2. 线程1：使用write系统调用写入映射的内存
3. 线程2：丢弃映射内存的私有副本
4. 让线程1与线程2相互竞争，以便他们能影响输出从而向只读文件写入数据
5. 竞态条件触发漏洞

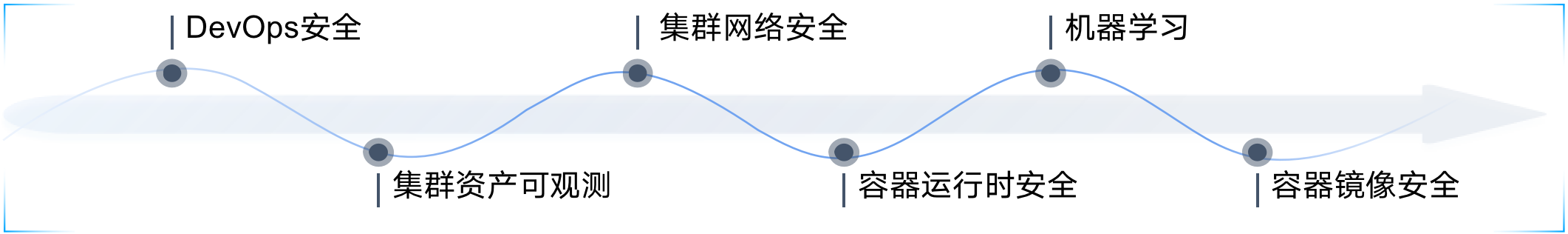
03 安天容器安全产品

产品介绍

安天智甲容器安全检测系统（CNSP）定位于云原生安全领域，致力于提供专业、全面的容器安全防护能力。

产品采用轻量化“一键部署”，无需人为在集群节点手动安装Agent，实现接入即防护。

通过DevOps安全、**集群资产可观测**、**集群网络安全**、**容器运行时安全**、**机器学习**、**容器镜像安全**六大核心功能为容器安全提供全生命周期一站式服务，全面解决安全防护需求，为客户高效提升安全防护能力。



产品架构

1 自动构建

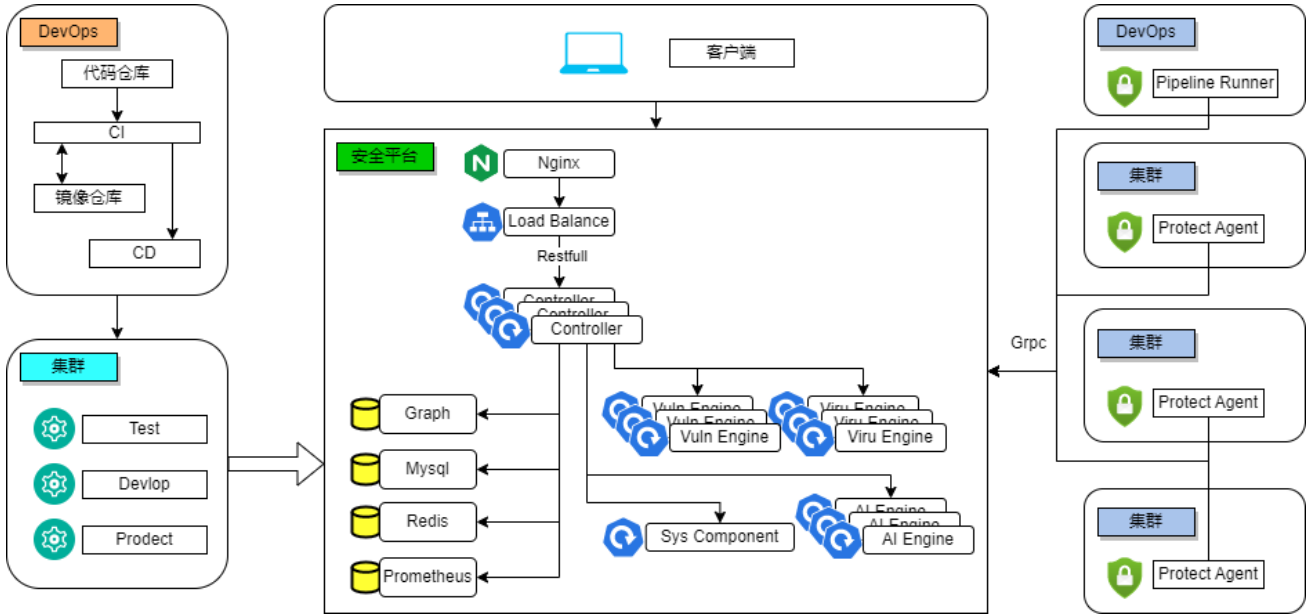
自动构建容器镜像、产品安装包、产品升级包等，快速独立发布迭代，提升软件开发、发布效率

2 微服务

减少业务组件耦合性；标准化发布服务，使得业务更加易于管理与变更；立体运营、监控服务状态，提升产品高可用、高并发能力

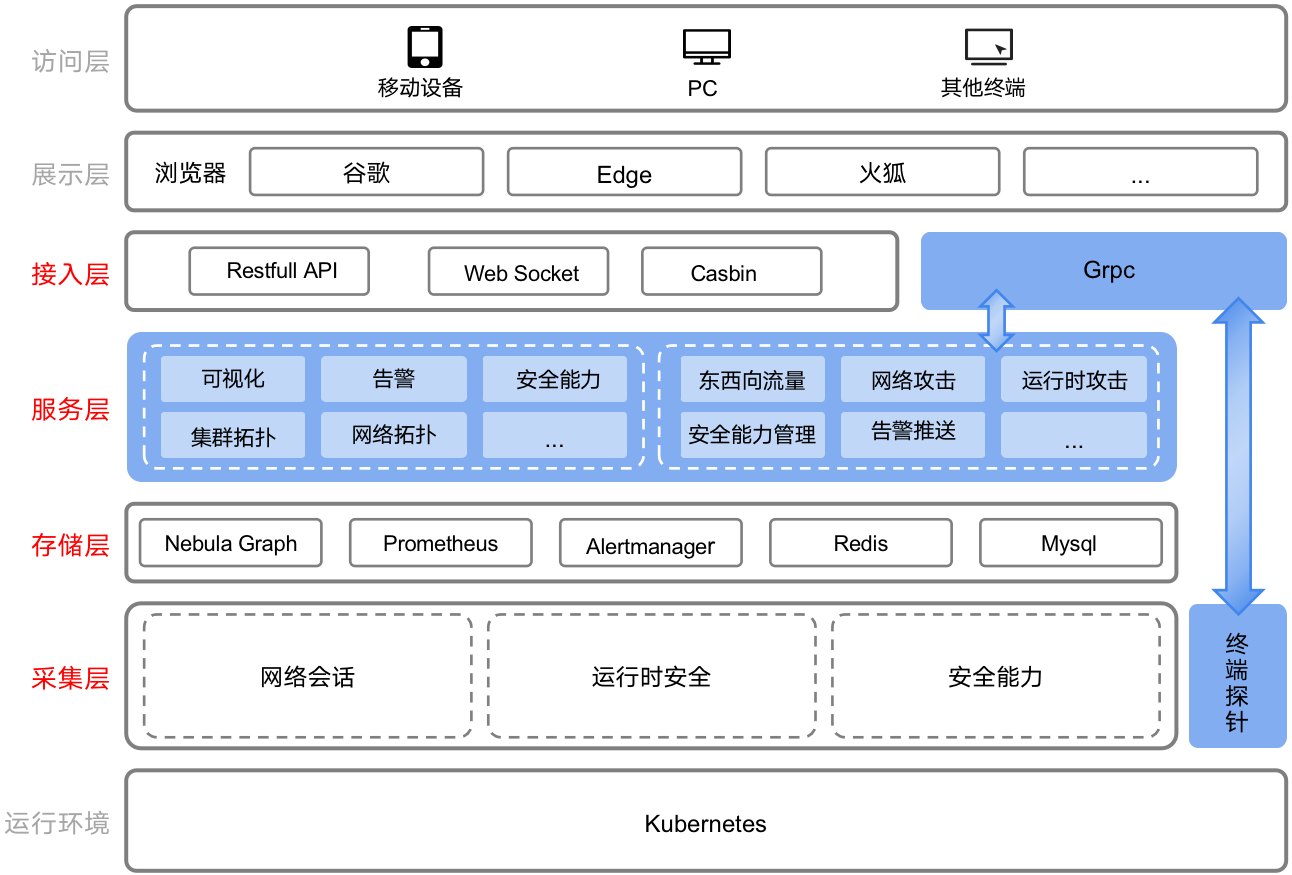
3 自动部署

安全探针自动部署，可跟随集群扩缩容动态部署



产品组件架构

安天智甲容器安全检测系统（CNSP）构建于K3S平台，主要由4层组件协作实现防护能力



- **接入层**

认证、授权接入请求，为展示层提供数据，同时提供查询和推送能力

- **服务层**

提供产品核心能力，主要包括安全相关扫描能力、安全分析能力、安全运营能力、告警推送能力、AI分析能力等

- **存储层**

提供不同类型数据持久化能力，包括图数据库、缓存数据库、关系型数据库

- **采集层**

使用探针收集被防护集群数据传回服务层，或根据预设策略进行安全防护

安全防护闭环

Step1

资产识别

识别集群内部所有资产类型（POD、容器、镜像等），对资产信息梳理，进行可视化展示。

Step2

风险监测

实时监测各类资产风险状态，包括网络攻击、运行时攻击、漏洞、病毒、WebShell、敏感文件等。

Step3

威胁预警

当检测到集群资产产生风险时，可通过邮箱、企业微信、钉钉第一时间进行预警，协助客户快速进行风险处置。

Step4

策略防护

结合预设策略，平台自动对所发生的风险进行处置，保证对客户资产进行全方位防护不被入侵。

Step5

动态安全防护

对当前风险处置完毕后，平台会自动对本次事件进行分析学习，并对同类型资产下发相应防护策略，从而形成动态防护，防护能力相较于传统的静态规则有了大幅提升。

Step6

主动防御

在不断对各类资产下发动态防护策略后，对于攻击面的防护进一步进行完善，对可能发生的攻击进行预判，最终由过去的被动防御形成主动防御从而完成产品安全防护闭环。

传统EDR+NDR方式，对于云原生环境防护的挑战

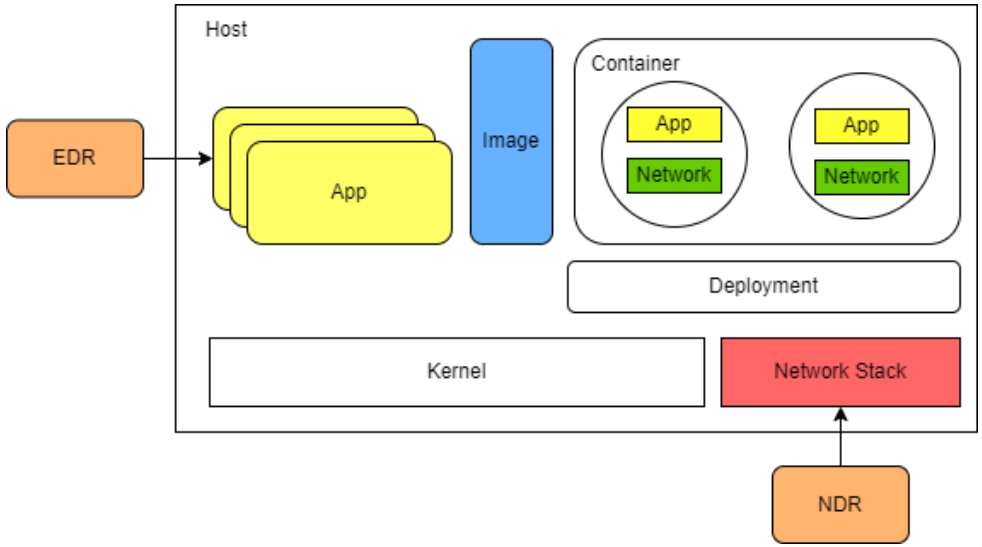
随着技术的迭代、发展，传统方式主要以EDR或EDR+NDR方式解决云原生环境中的安全问题，但这并不能在敏捷性较强的容器环境发挥较大优势。

容器生命周期较短

容器环境难以具备固定的网络属性，同时容器PaaS平台的VPC网络形态也多样导致适配的难度较大

容器运行环境相对独立

容器镜像会带入较大风险，另外容器运行环境与宿主机文件系统相互隔离，EDR检测能力会因此受限



容器运行环境与主机环境相互隔离

- EDR收集进程行为能力可能失效
- 需要有效区分PID、USER、IPC命名空间

容器网络与主机网络相互隔离

- NDR分析网络流量能力可能失效
- 需要有效区分Network命名空间
- 需要兼容容器网络属性易变场景

容器文件系统与主机文件系统相互隔离

- 进程权限、文件路径等信息需要分别处理
- 需要有效区分容器挂载点

安天的容器安全防护思路

全面云原生

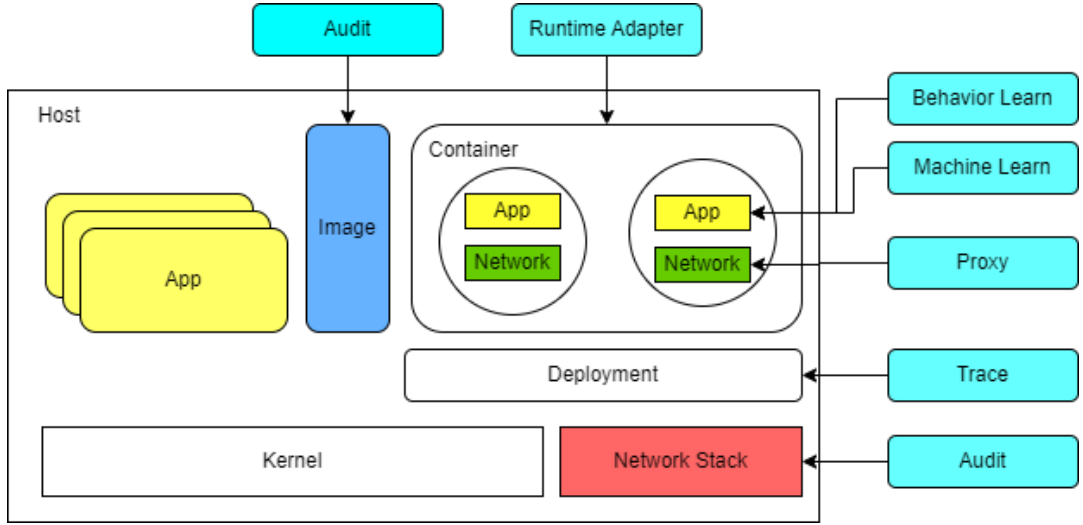
- 安全左移保障DevOps制品安全
- 实时跟踪集群资产状态，平台内部还原资产拓扑
- 动态扩缩容适应不同防护场景

ML&零信任防护

- 自动学习容器行为形成行为基线
- 具备认证、授权的容器网络微隔离能力
- 采用ML算法改善安全事件误报、漏报问题

可观测性风险溯源

- 实时跟踪集群资产风险及其事件
- 智能推荐防护策略，形成防护闭环
- 风险资产、风险事件溯源



04 关键技术实践

DevOps安全

云原生安全包括的安全左移是相对传统安全检查内容提早对源代码或中间制品做检测，将缺陷和风险修复的成本尽可能地降低，同时将效率及质量尽可能地提高，以实现更全面的防护。

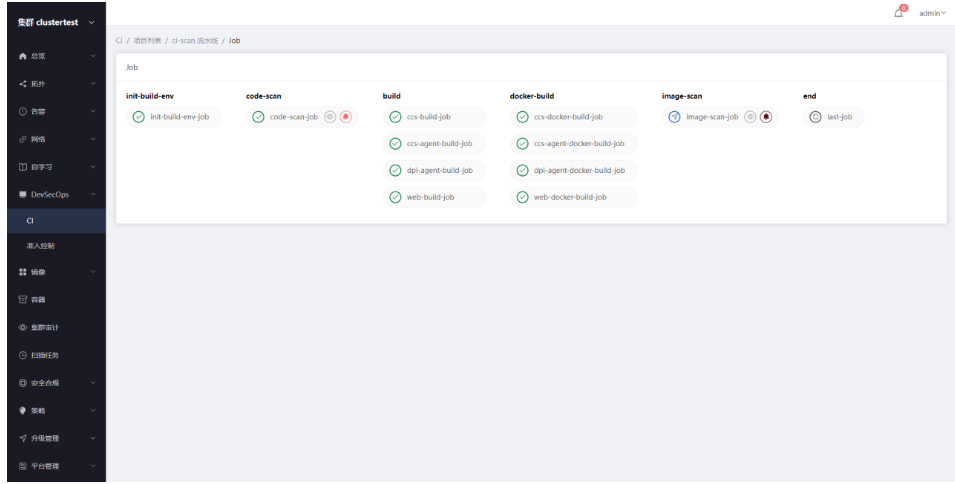
安天智甲容器安全检测系统（CNSP）可根据需求在DevOps的关键阶段进行安全检查，并支持在平台可视化流水线状态，发现异常后可及时发出告警或者阻断。通过层层扫描、策略限制，最大程度保障容器及集群环境安全。

检查点1	检查点2	检查点3	检查点4
第一个检查点在代码编译阶段可对源代码进行扫描；	第二个检查点在二进制编译完成后进行检查；	第三个检查点在打包完容器镜像至上传仓库阶段进行检查；	第四个检查点在服务部署阶段，通过安全策略限制有安全隐患的容器部署到集群内部；

DevOps安全

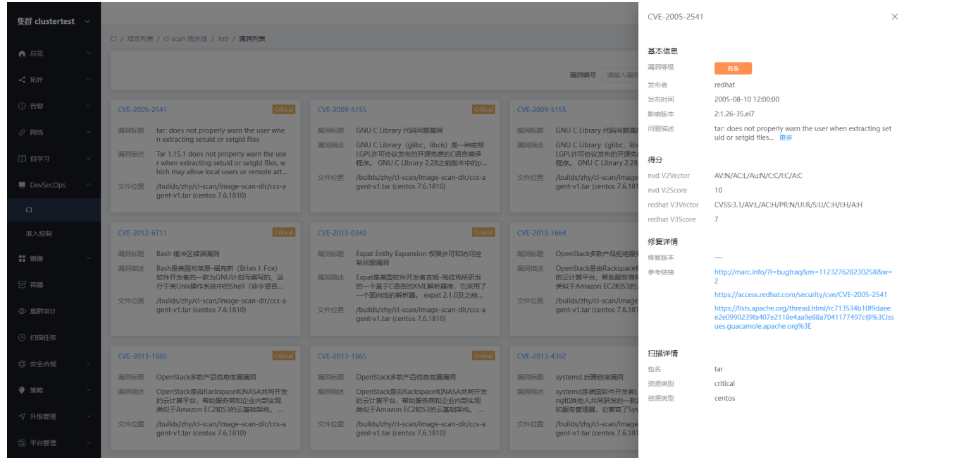
可视化流水线状态：

实时跟踪流水线运行状态，包括流水线不同阶段运行任务，并在对应阶段标注风险状态，更直观的可视化DevOps流水线运行状态和风险状态。



指导&建议：

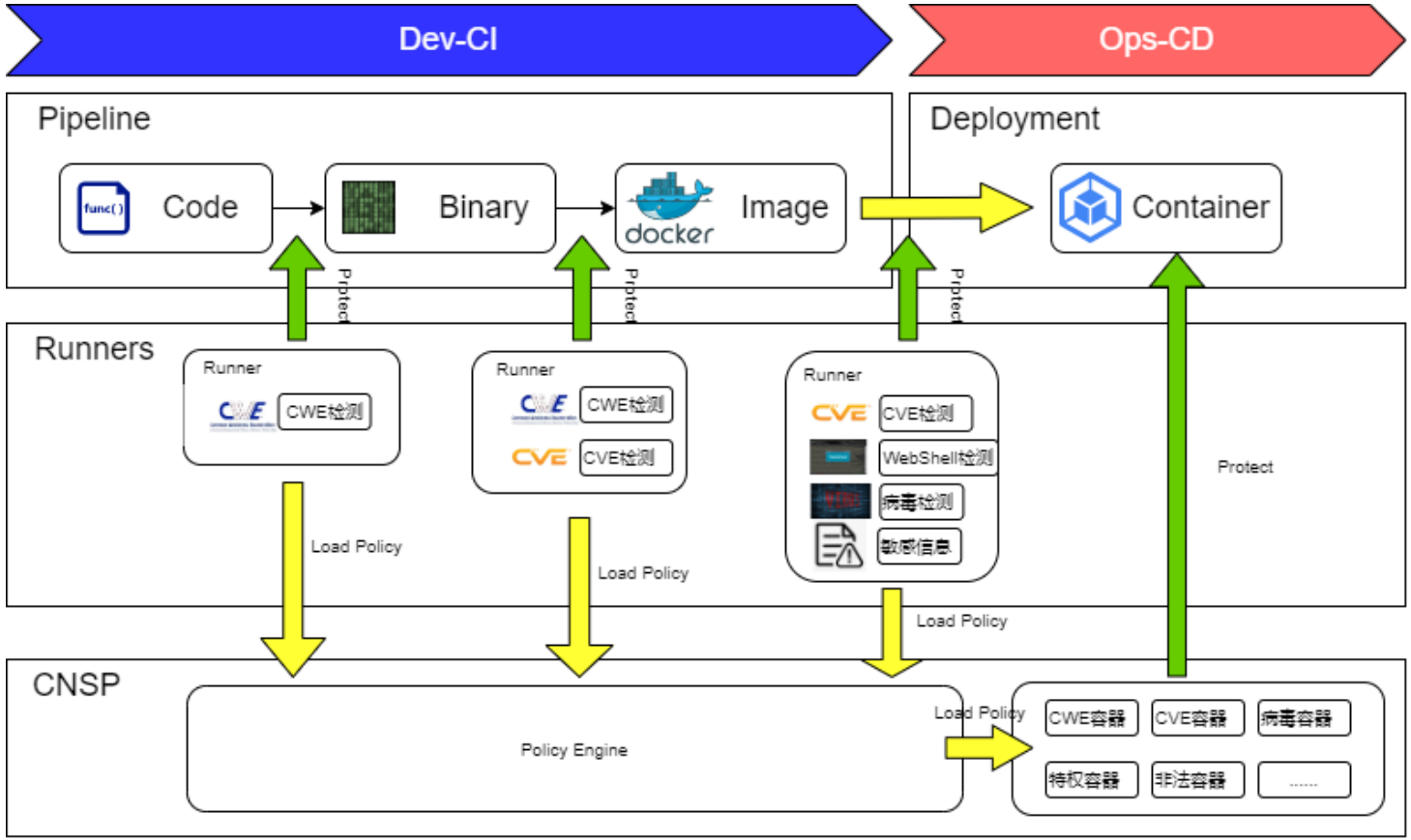
源代码审计和容器镜像扫描阶段可根据扫描结果给出风险原因以及对应的解决方案，包括CWE、CVE相关说明



DevOps CD联动：

云原生持续部署阶段可根据持续集成阶段扫描结果进行资产管控，防止高危镜像部署到机器环境。持续部署阶段可多维度根据资产情况进行限制或告警，如：阻断特定CVE容器镜像、特权容器等

DevOps安全



工作流程

CI防护

1. 流水线加入CNSP检测Runner
2. 检测Runner启动自动拉取策略
3. 检测Runner回传检测结果
4. 检测Runner根据策略决策后续流程

CD防护

1. CNSP中控启动WebHook服务器
2. 被防护集群添加Validating配置接入CNSP
3. Trace集群事件由kube-apiserver传入CNSP
4. CNSP根据策略决策后续动作

资产可观测性

安天智甲容器安全检测系统（CNSP）通过实现资产可观测性，帮助客户达到**更快自动化识别和解决问题**，快速了解集群内部状态。产品从集群内部出发，通过对集群资产的**风险状态、进程、告警、日志等数据**进行智能分析，提供完整的**可视化安全风险视图**，当集群中某一资产出现风险，那么该资产关联的所有资源将会同时会被关联该风险内容，无论在哪一阶段均可快速溯源至风险源头，同时结合平台监测、告警等服务，从而实现**故障诊断、风险溯源、自动化解决问题**。

资产数据源分析		平台服务		产品能力	
进程行为	操作文件	资产可视化	风险监测	故障诊断	风险溯源
网络行为	资产拓扑	威胁告警	事件分析	自动化	风险预判
.....		.....		.....	

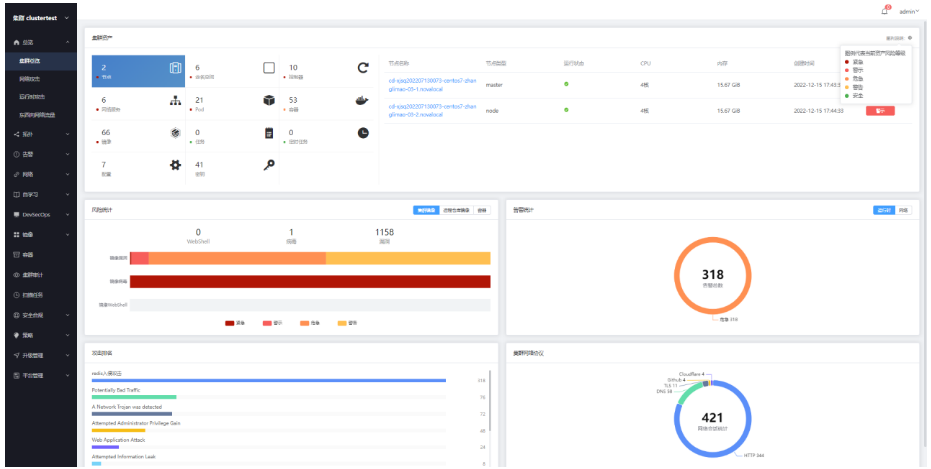
在此之上，安天智甲容器安全检测系统智能分析风险事件上下文内容，结合机器学习与动态防护能力发现以前不知道且未来仍可能发生的威胁并对其进行提前布防。从而真正做到**【预判并解决"未知且不确定的问题"】**！

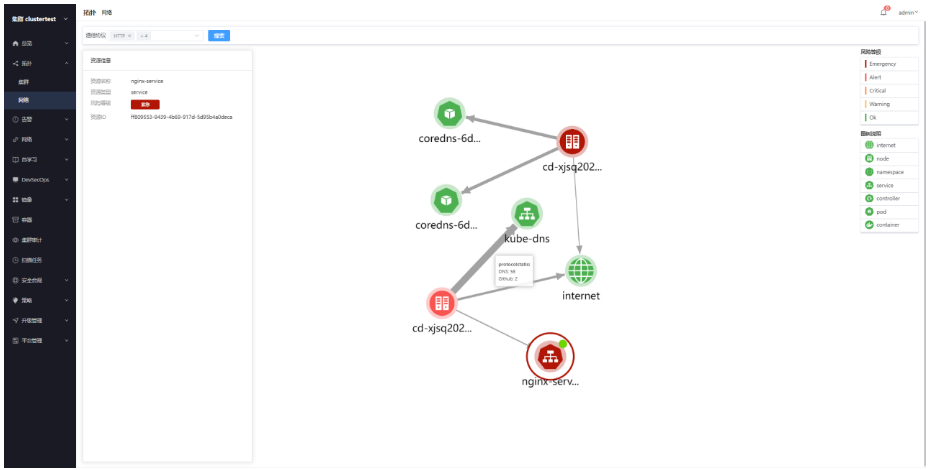
传统监控、可视化告诉我们哪些资产在工作或不工作

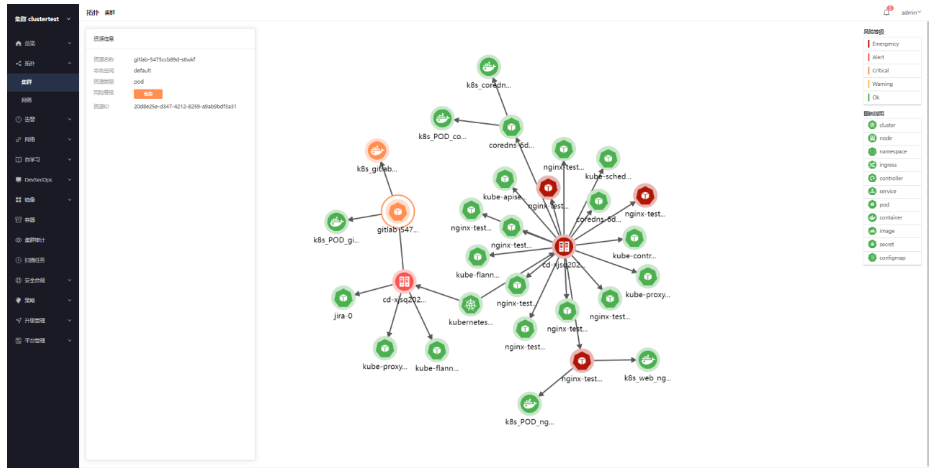
可观测性告诉我们资产为什么不工作、如何解决、未来哪里仍可能出什么问题。

资产可观测性

01 实时收集集群资产生命周期变化情况	02 实时更新集群资产风险等级
03 根据集群资产风险情况动态关联相关资产	04 集群视角可视化集群资产父子关系
05 主机视角可视化集群资产父子关系	06 可视化资产拓扑风险等级
07 集群东西向网络流量可视化	08 自动识别资产服务







资产可观测性

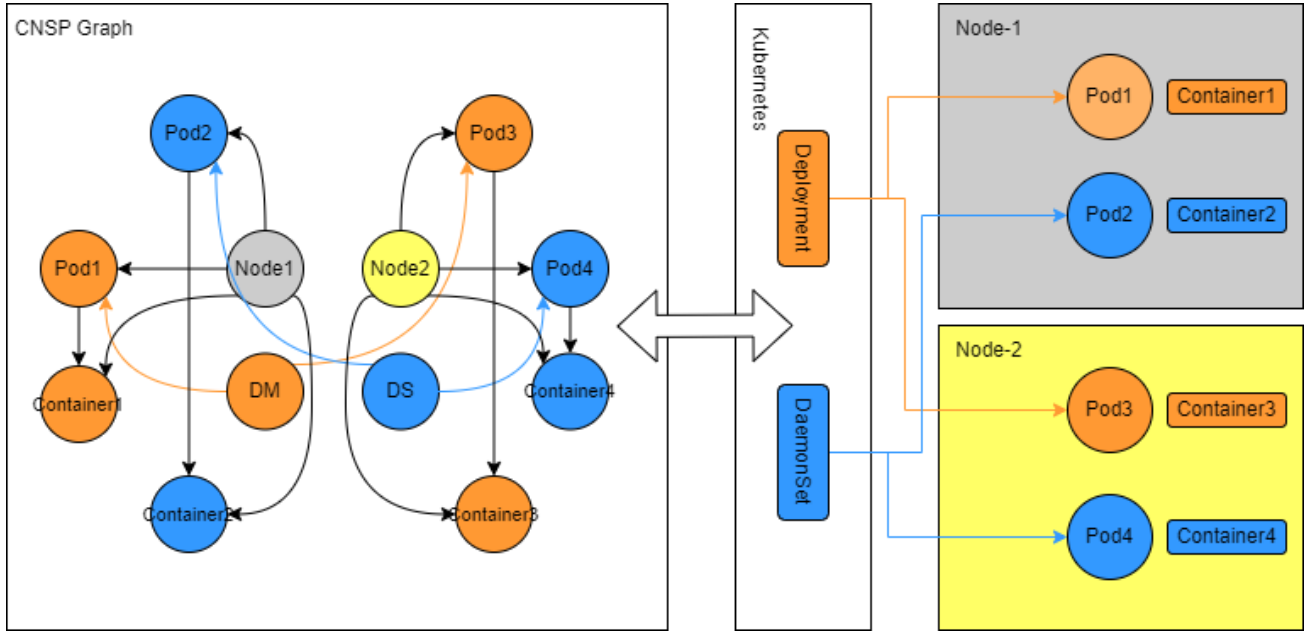
容器集群资产种类繁多，其关系也错综复杂，CNSP采用图数据库梳理其中关系，并结合图数据结构特性建立拓扑关系。

实时跟踪资产生命周期&拓扑关系

1. 通过集群kube-apiserver监听集群事件
2. 根据资产状态更新图库节点状态
3. 根据资产关系更新图库节点关系

实时更新资产风险状态

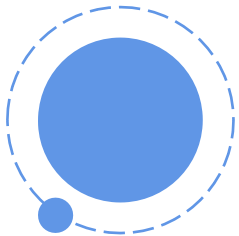
1. 监听CNSP告警事件更新图库节点风险等级
2. 根据风险资产广度遍历检索威胁事件影响范围
3. 根据风险资产深度遍历溯源威胁事件入侵流程



集群网络流量审计

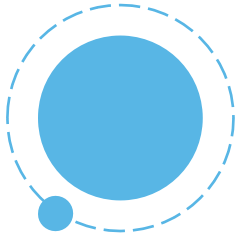
应用微服务化大幅增加了内部网络流量和服务通信端口总量，同时工作负载快速动态伸缩，增加了安全监控和保护的难度。传统防火墙由于部署场景限制很难适应这种动态伸缩的工作负载间的网络流量和异常行为。

传统解决方案



- 重点防护南北向网络流量
- SDN方式牵引东西向网络流量
- 基于规则的方式匹配恶意网络流量
- 只能基于网络包信息建立相关流表
- 由于架构变化较难适应云原生环境
- 实施成本高

安天解决方案



- ✓ Sidecar方式防护网络流量
- ✓ NodeAgent方式防护网络流量
- ✓ 基于零信任方案实时保护集群资产
- ✓ 结合集群资产风险情况动态调整相关流表
- ✓ 可根据应用场景灵活选择合适的部署方案
- ✓ Saas化部署，动态横向扩展

集群网络流量审计

安天智甲容器安全检测系统内置WAF、IDS、IPS检测能力，可根据场景需要部署对应安全能力，同时提供代理和旁路模式可供选择。

1

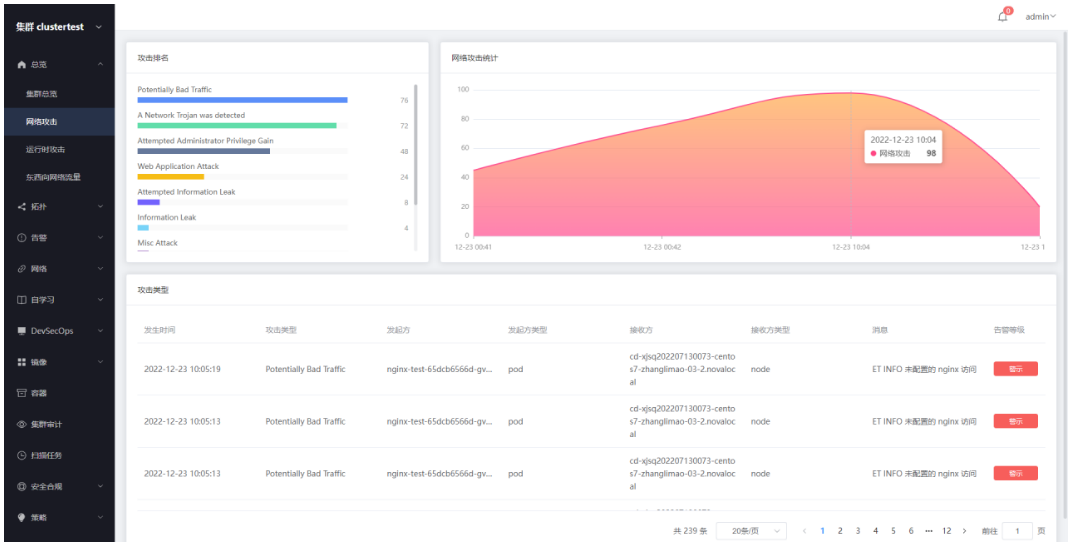
SIdeCar

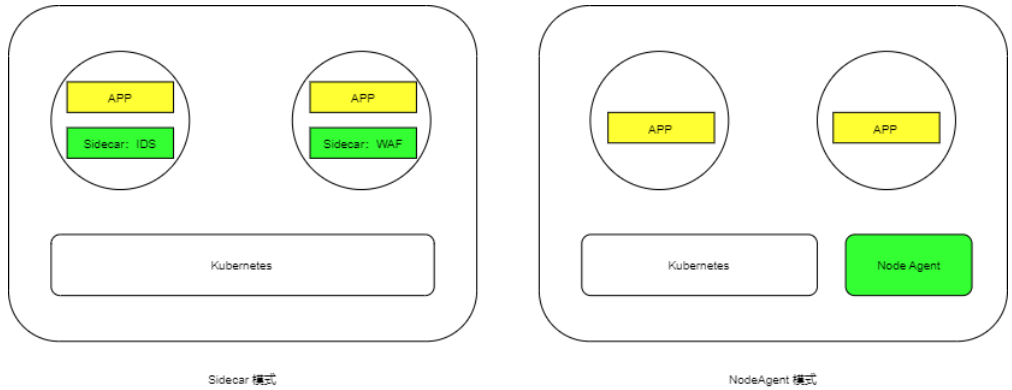
精准防护指定资产，安全能力可跟随集群资产动态迁移

2

NodeAgent

全面审计基于主机的网络流量，结合微隔离能力实现智能策略阻断功能





集群网络流量审计-SideCar

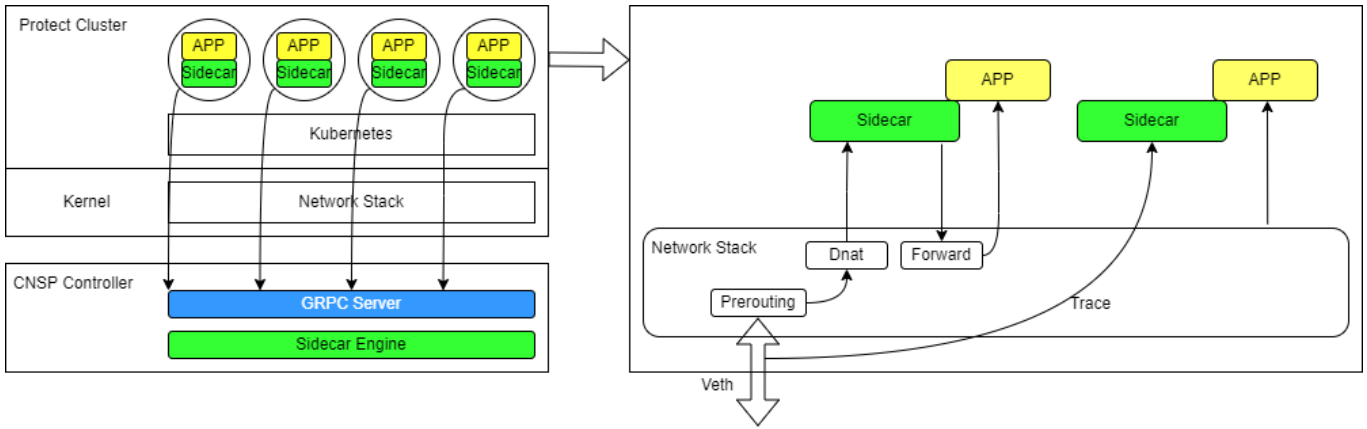
SideCar方案会根据APP编排文件动态添加Sidecar容器配置，让SideCar容器与业务容器运行在同一网络命名空间，然后代理或旁路业务容器网络流量，实现东西向网络流量审计能力。

代理模式

使用DNAT技术重定向业务网络流量到SideCar容器，SIdeCar处理完后通过代理或转发方式把流量转到业务容器，实现代理功能

旁路模式

Trace容器网络Veth接口网络流量到SideCar容器进行审计，不会对正常业务网络流量造成影响



SideCar方案优点

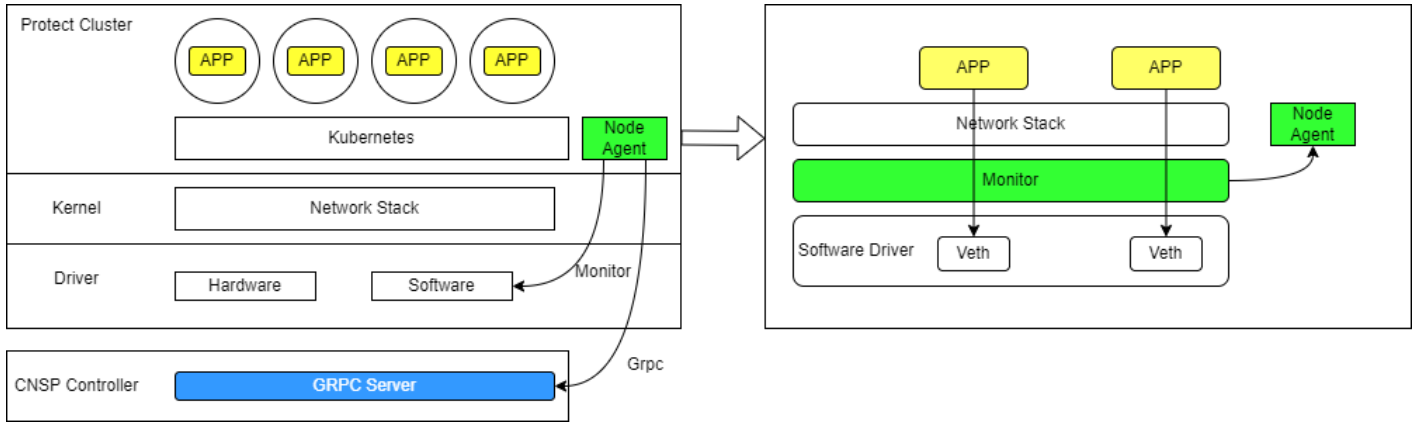
- 安全能力生命周期由Kubernetes控制器管理并可动态跟随POD迁移
- 可根据实际场景精准防护集群资产
- 引流方案对CNI类型依赖较少

SideCar方案缺点

- 安全检测引擎冗余
- 可能会与Service Mesh冲突

集群网络流量审计-NodeAgent

Linux Virtual Ethernet Devices（Veth）是Linux内核自带的虚拟网络接口，一般用于网络命名空间对外收发网络流量。NodeAgent方案在主机驱动层镜像Veth接口网络流量，然后送入NodeAgent流量检测引擎进行审计。



工作流程：

1. 识别集群容器Veth—Peer网络接口
2. 镜像主机侧Veth虚拟网络接口流量
3. 从定向网络流量到NodeAgent对应的Veth接口
4. 检测NodeAgent Veth接口网络流量

NodeAgent方案优点

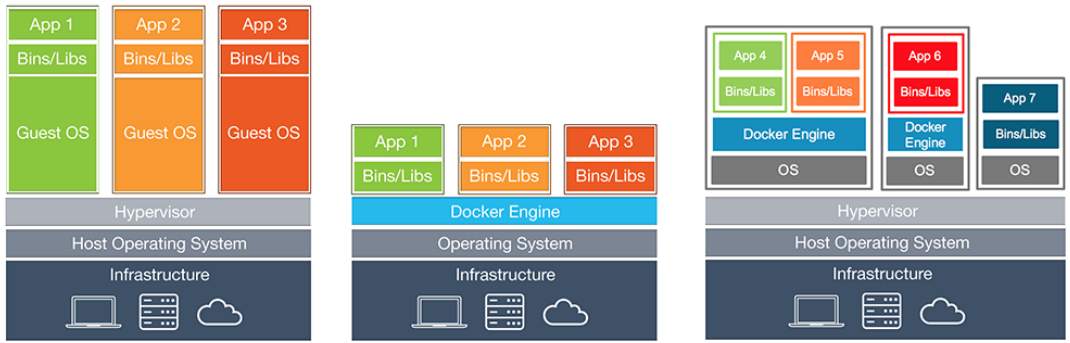
- 容器网络CNI插件无差别兼容
- 主机、容器网络流量可同时检测
- 安全检测引擎不冗余
- 基于驱动层引流性能影响较小

NodeAgent方案缺点

- 涉及容器接口容器状态维护，引流方案较复杂
- 不适用于网络流量代理场景

容器安全

云原生工作负载主要由容器技术实现，业务应用在容器中运行时会直接与主机操作系统进行交互，但容器技术相比虚拟化技术缺乏Hypervisor作为操作系统和工作负载之间的一层安全屏障，所以容器逃逸后对操作系统和PaaS架构会造成较大威胁。另外云原生引入微服务架构后整体业务复杂度提升，导致传统安全检测方案不再适用。安天智甲容器安全检测系统（CNSP）针对容器工作负载安全问题提供3个解决方案。



1 进程行为审计

Trace容器中进程行为，包括系统调用、Fork、Exec等，并使用分析引擎给出风险等级

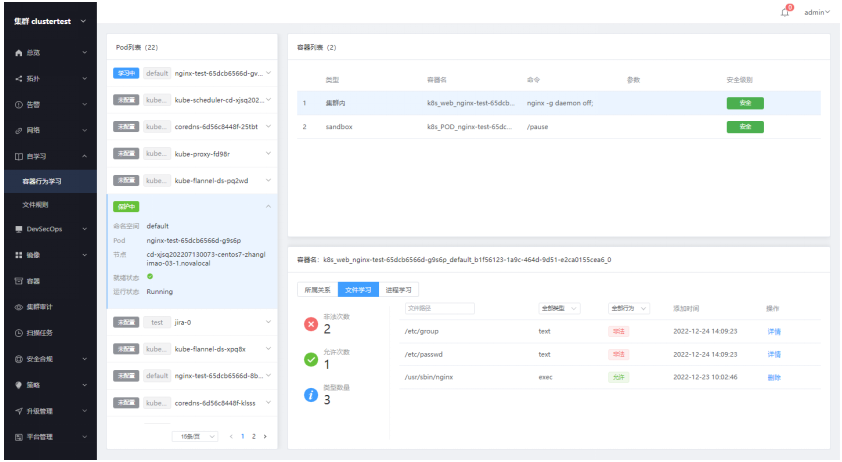
2 进程行为学习

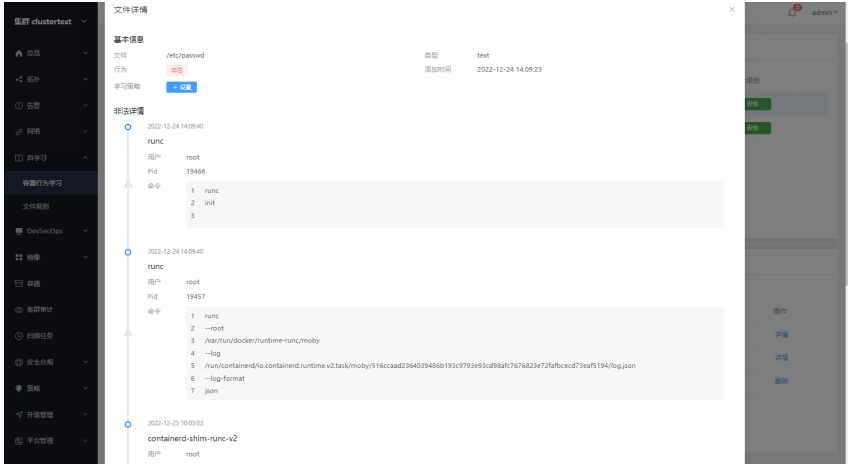
启用学习模式学习进程正常运行过程中的行为，形成进程行为基线，启用保护模式阻断或告警基线以外行为

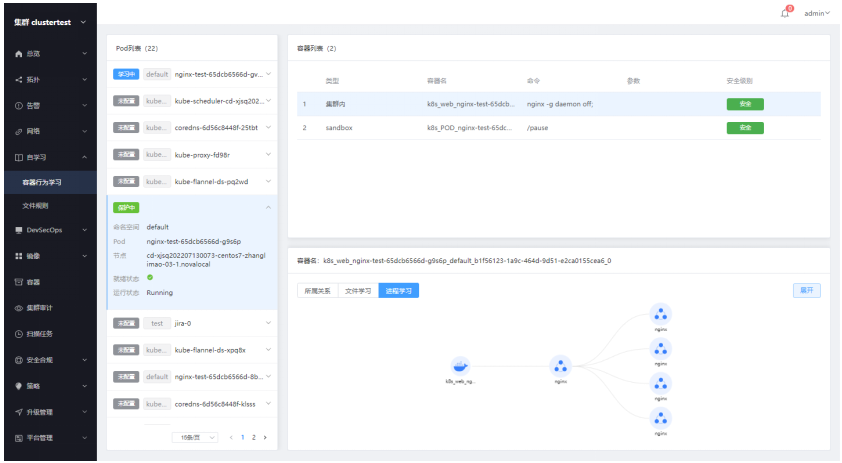
3 容器网络隔离

服务不中断方式代理容器网络流量，使用黑白名单、认证等方式限制外来或对外发送风险网络流量

容器安全







学习 01

调整进程基线策略

02 保护

调谐 03

阻断进程异常行为跟踪异常调用链

容器安全-行为学习

CNSP容器行为学习功能基于零信任理念持续学习容器进程行为，形成容器行为基线，基线以外行为会触发告警事件并回溯调用链可视化攻击线路，同时也可以动态添加信任行为完善基线策略。

学习模式

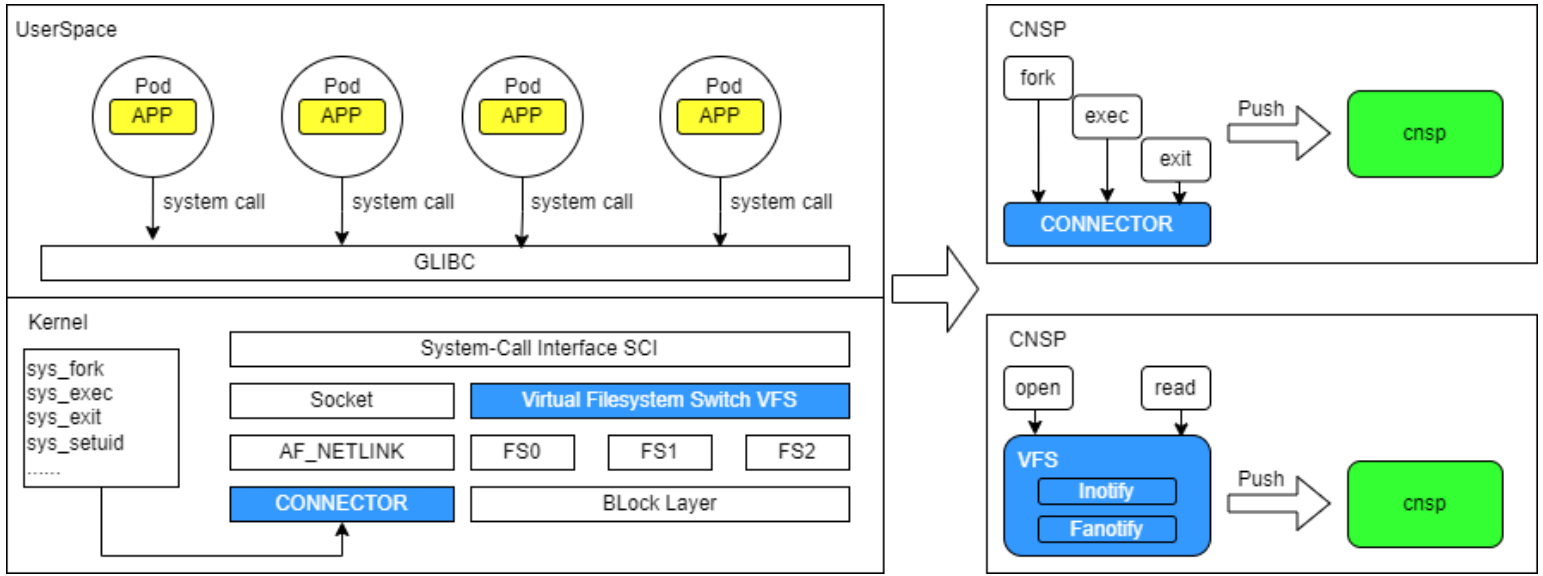
1. 监听容器进程行为
2. 学习进程拓扑关系
3. 学习容器行为基线

保护模式

1. 监听容器进程行为
2. 对比容器行为基线
3. 阻断非法行为并告警

监控模式

1. 监听容器进程行为
2. 对比容器行为基线
3. 放行非法行为并告警



工作原理

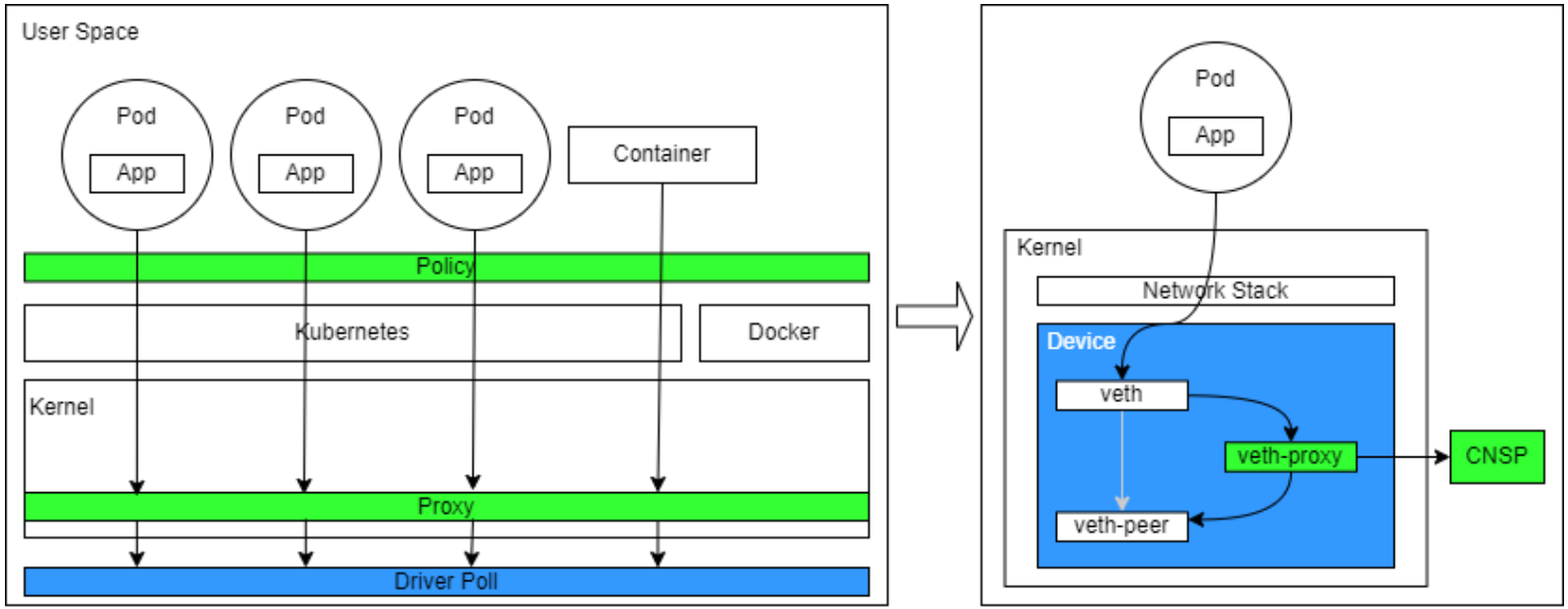
- Socket Connector监控进程行为
- Inotify监控容器文件系统行为
- Fanotify控制容器文件系统操作权限
- 根据以上事件动态完善进程行为基线

性能&资源

- 进程、文件行为非轮询模式，而是基于事件驱动方式进行学习或保护，性能较好
- 节点4核CPU 8G内存，50个POD容器全保护，CPU消耗<2%，内存消耗<2%

容器安全-微隔离

CNSP微隔离方案采用虚拟网卡代理技术实现，并提供认证、授权等方式管理资产网络访问行为。此方案与网络协议栈耦合性较低，在保证兼容性的前提下也可提供高性能转发能力，而且开启与停止CNSP微隔离功能时不会导致集群服务业务中断或重连。



优势

- ✓ 兼容主流CNI网络
- ✓ 高性能转发
- ✓ 多维度认证、授权
- ✓ 启停不影响业务

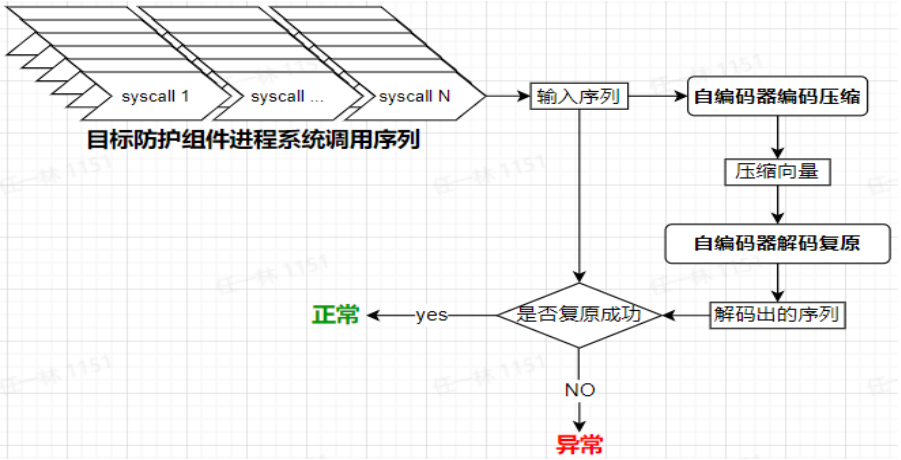
性能

- CPU：4核
- 内存：8G内存
- 容器：50个
- CPU消耗：<2%
- 内存消耗：<2%
- 容器转发性能：1000—3000Mbps/s

漏洞利用入侵捕获

对重点组件进程的系统调用级AI防护，对组件已知Nday漏洞入侵与未知0day漏洞入侵具备灵敏检测能力，捕获恶意进程行为

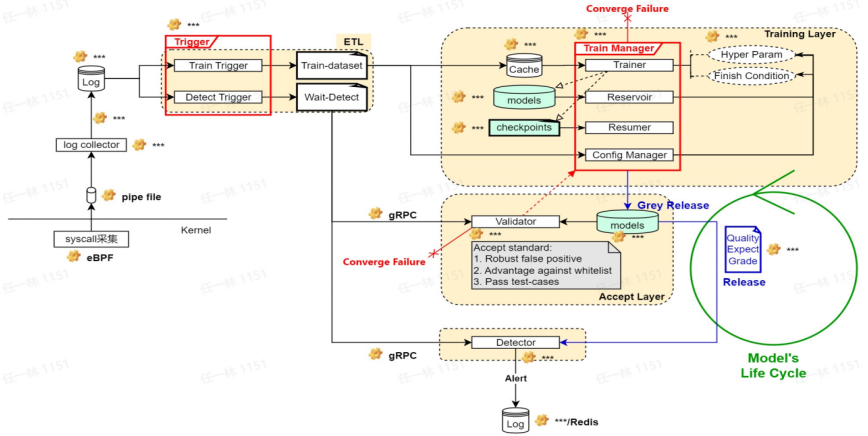
1. 采集保护目标组件进程的系统调用序列
2. 训练深度变分自编码器模型以学习正常组件行为
3. 模型运行：输入待检测系统调用序列
4. 模型判定：根据解码质量好坏判定是否异常



隐蔽利用行为捕获

对于无进程创建、无文件落盘，仅在脆弱组件进程内存空间内执行的隐蔽恶意行为，仍旧具备检测能力

- 学习模式 → 监控验收模式 → 检测模式
- 训练容灾：断点续训
- 质量验收基线：通过自动化的模型验收机制保障交付模型质量
- badcase自动反馈：对用户反馈误报自动加入训练工作流



机器学习

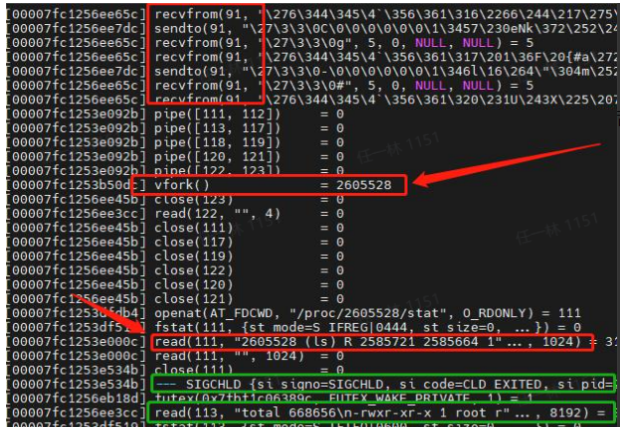
- 传统白名单方案无法兼顾误报量的控制与检出能力的保障
- VAE模型能够通过训练完成自适应，兼顾低误报与有效检出

检测方案	单进程预估日均误报告警数量	不同【异常系统调用个数】下的检出率			
		1	2	3	4
传统白名单	5000	100%	100%	100%	100%
	1590	全部漏检	100%	100%	100%
	227	全部漏检	全部漏检	100%	100%
	趋近于0	全部漏检	全部漏检	全部漏检	100%
VAE深度学习	趋近于0	0.9236	0.9917	100%	100%

单目标防护组件进程无数据积压下的训练CPU占用：0.57%；
一核CPU满载可承担175个对标防护组件进程的学习

模型	对标进程	CPU型号	核心数	单核CPU占用率(%)	内存(MB)	估算无积压训练的最低单核占用率(%)	单核满载无积压训练的可承载量(N个对标进程)
AntiyVAE	htop	Intel Xeon Processor (Cascadelake)	1	27	405	0.570711297	175.2199413

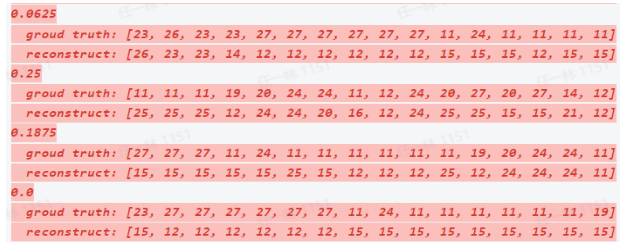
Confluence组件漏洞利用检出实测：CVE-2022-26134



①Socket传入漏洞利用Payload

②触发漏洞执行命令

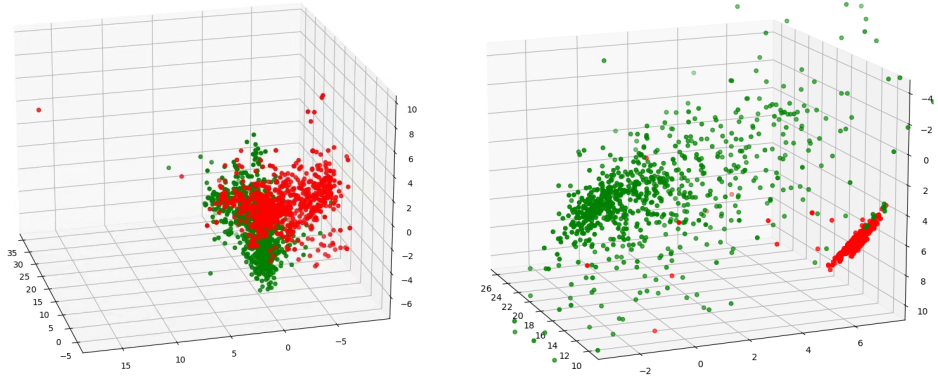
③命令执行成功



Antiy VAE：解码准确率低于0.25，表明异常

AI文件防护算法

- 文本型文件：Webshell/恶意bash脚本等
- 二进制文件：ELF恶意样本检测



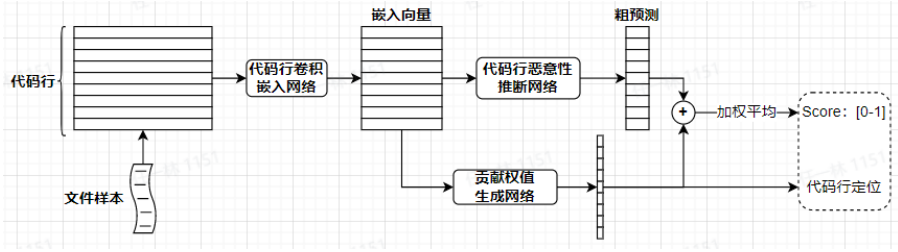
算法	cpu型号	cpu占用	输入文件个数	文件大小	模型推导用时(s)
Webshell检测	Intel Xeon Processor (Cascadelake)	1*100%	10000	40K	97
Webshell检测	Intel Xeon Processor (Cascadelake)	2*100%	10000	40K	63
Webshell检测	Intel Xeon Processor (Cascadelake)	3*100%	10000	40K	48
ELF无监督检测	Intel Xeon Processor (Cascadelake)	2*100%	10000	40K	226

Webshell检测模型

1. 通过卷积网络提取代码行深层特征
2. 通过权值贡献网络定位恶意代码行

检测效果

- 黑样本检出率：0.9758（高于某盾的87.9%）
- 白样本准确率：0.9983（误报率万分之17）



ELF无监督检测模型

1. 提取ELF文件结构特征：节区滑动熵、节区字节分布等
2. 无监督特征重建任务：学习恶意elf样本的特征性质

检测效果

- 黑样本检出率：0.95
- 白样本误报率：趋近于0

感谢大家的关注

安天冬训营 wtc.antiy.cn