
GOTC 2023
全球开源技术峰会
THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE
OPEN SOURCE, INTO THE FUTURE
「聚焦开源安全」专场
基于网络弹性法案的企业级开源供应链解决方案
王永雷 新思科技 2022年05月28日

Agenda

- 欧盟网络弹性法案解读
- 开源供应链面临的严峻的挑战
- 开源供应链的解决方案

欧盟网络弹性法案解读

欧盟网络弹性法案–Cyber Resilience Act

通俗称之为数字产品安全基线

2022年9月，针对当前硬件和软件产品越来越容易受到网络攻击的问题，欧盟公布了《网络安全弹性法案》（以下简称为“法案”）。欧盟网络弹性法案强化网络安全规则，以确保更安全的硬件和软件产品。

1.欧盟范围有史以来**第一个**此类立法，要求关联企业必须在整个欧盟范围内遵守统一的网络安全规则

2.法案主要针对的商品主体是**IOT和智能设备**

3. 违反规定的企业将面临最高 **1500**万欧元 （约1.05亿元人民币）或 全球营收**2.5%**的罚款

欧盟网络弹性法案– Cyber Resilience Act

- 对制造商的安全要求在提升

1. 将安全纳入规划、设计、开发、生产、交付和维护阶段的考虑范围内
2. 提供至少需包括产品的顶层依赖关系的软件材料清单SBOM
3. 制造商必须积极上报开发出的漏洞和事故，公开披露漏洞信息，制定漏洞披露策略；
4. 一旦售出，制造商必须确保在预期产品生命周期或五年期间（以较短者为准），能有效处理好漏洞
5. 确保软件供应链安全
6. 至少在五年内提供安全更新

1 软件安全不是单点的，是贯穿整个研发流程体系的

2 SBOM是重要的抓手

3 开放搞安全

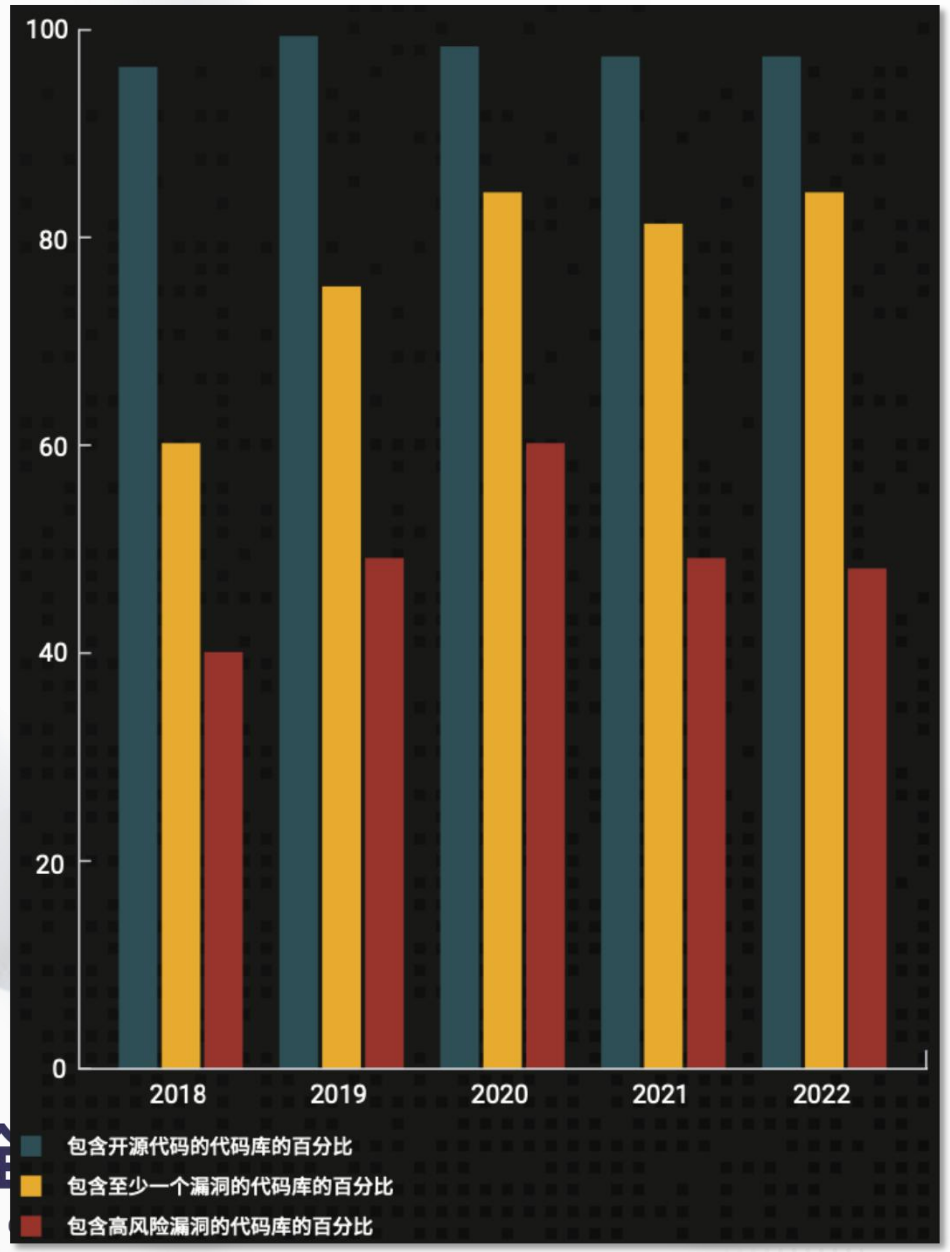
4 持续的漏洞监控

5 供应链安全是难点和挑战

6 召回和更新机制，系统化的安全体系能力

开源供应链面临的严峻的挑战

开源漏洞这5年–开源供应链不容忽视



- 84%的代码库包含至少1个已知开源漏洞（同比增加4%）
- 48%的代码库包含高风险漏洞（同比仅减少2%）

*高风险漏洞是指已被主动利用、已有POC(证明漏洞存在)记录、或已被归类为远程代码执行的漏洞

标准风险管理服务不涵盖软件风险

风险管理解决方案	解决的主要风险
业务与集成	• 美国证券交易委员会(SEC)义务 • 合规性
供应商	• 财务稳定 • 企业责任
制造供应链	• 物流 • 韧性
网络及IT	• 业务连续性 • 预防数据丢失
软件供应链	• 云服务 • 预装软件

标准供应链风险管理：

- 不专注在软件
- 没有紧跟不断变化的软件漏洞态势

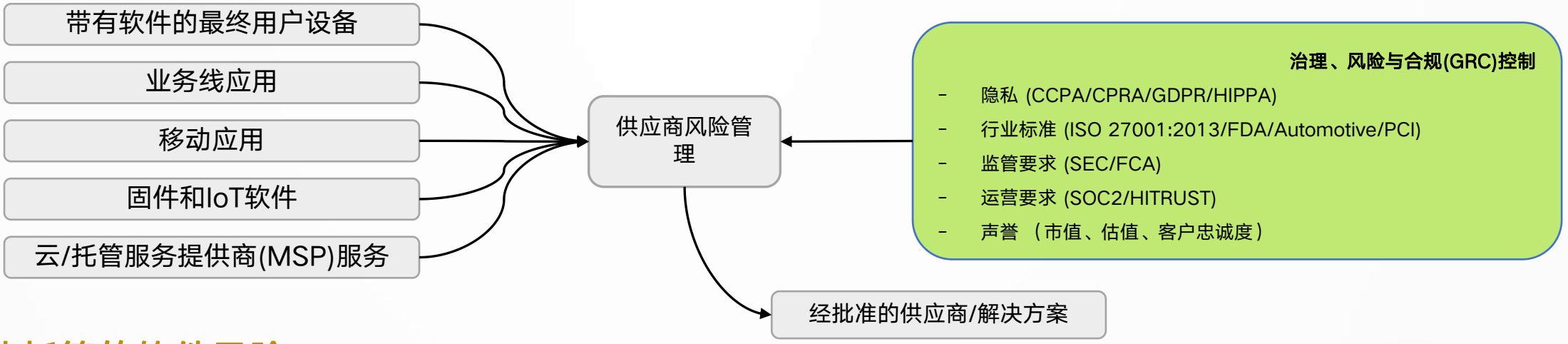
软件供应链风险管理(SSCRM)紧跟软件发展动态

- 每天都会披露新的软件漏洞
  - 2021年，每日52个新的CVE漏洞披露 [1]
- 大多数软件是由第三方开发，超出您的控制范围，并且标准风险管理服务对此也不知情
  - 每个应用程序 含有508 个库组件

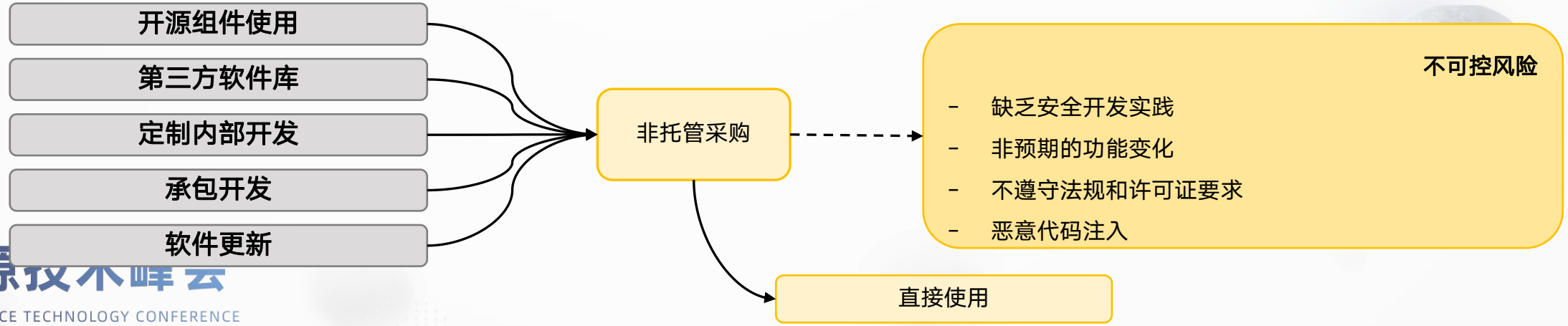
[1] US-CERT National Vulnerability Database

软件供应链决策与风险控制

由供应商风险管理解决方案管理的软件风险



暴露及非托管的软件风险



现代软件开发的流程

设计 开发和测试 部署

开发流水线 制品 版本控制 自研代码 软件包

开发商业软件

需求分析
1. 功能
2. 安全
3. 监管
4. 法律

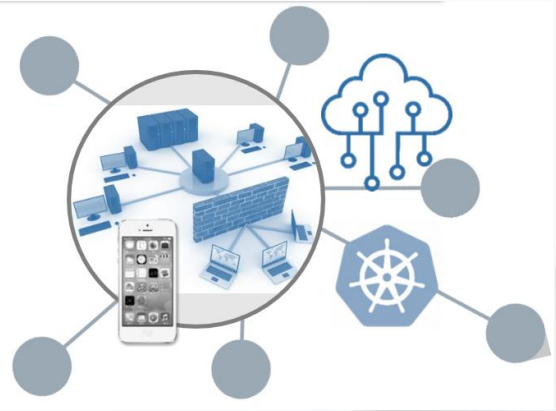
公共仓库 第三方组件

开发开源软件

开源组件 开源组件

开源组件 开源组件 开源组件

开源组件 开源组件 开源组件 开源组件 开源组件



软件开发面临的开源供应链面临的挑战

设计

需求分析
1. 功能
2. 安全
3. 监管
4. 法律

开发和集成

开发商业软件

开发流水线　制品

版本控制　自研代码

开发开源软件

公共仓库

有风险软件包

第三方组件

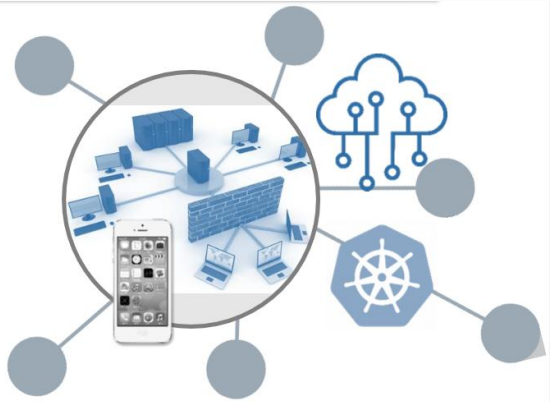
开源组件　开源组件

开源组件　开源组件　开源组件

开源组件　开源组件　开源组件　开源组件　开源组件

植入后门　误植域名　敏感信息　开源许可证/专利　安全漏洞

部署

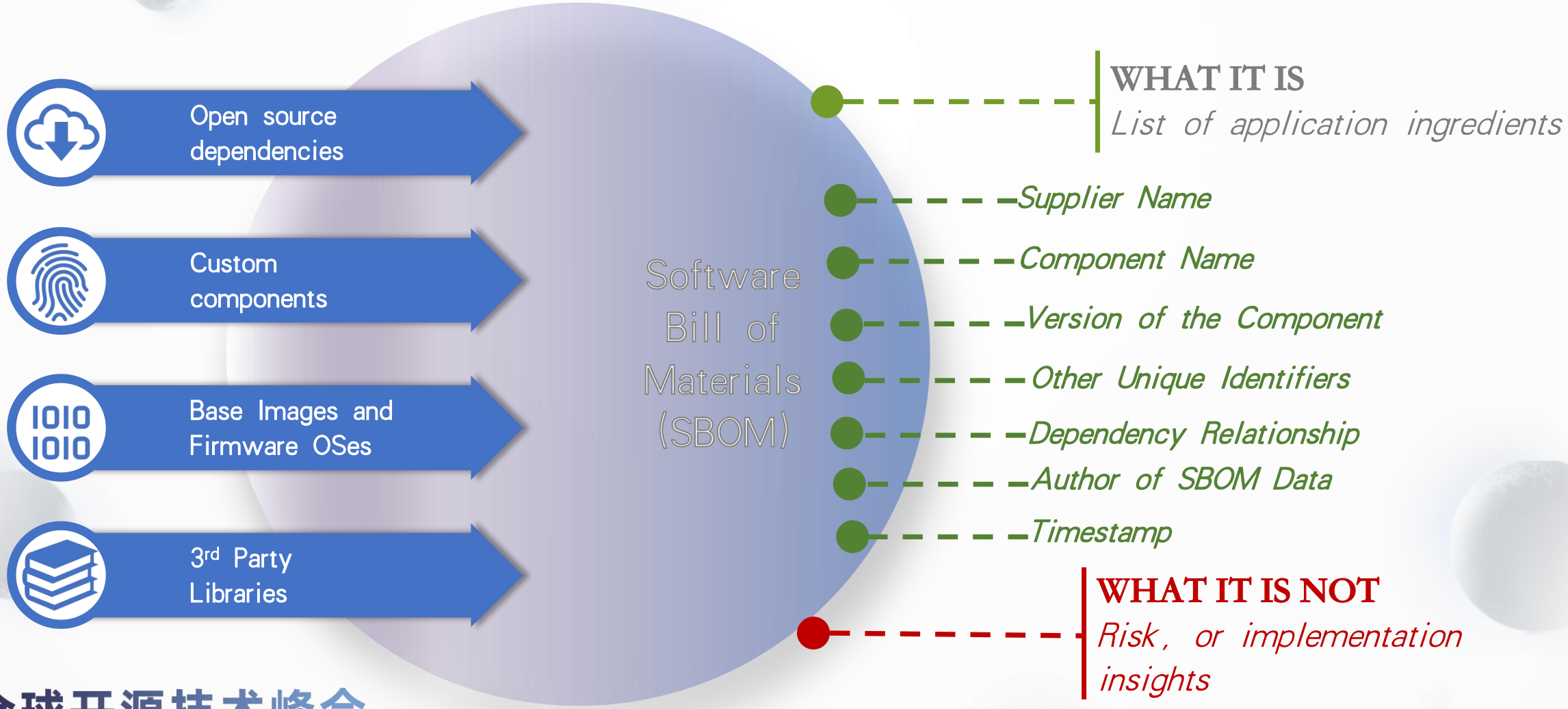


风险
1. 数据泄漏
2. 僵尸网络
3. 合规风险
4. 知识产权

开源供应链的解决方案

供应链治理的基础

SBOM软件供应链治理的基石



企业治理实践 — 研发运维过程自动化治理

DevSecOps

架构师 | 开发人员 | 开发Leader | 构建工程师 | QA/QC/安全 | 运维/应急

Plan → Code → PR/MR → Build → Test → Operate

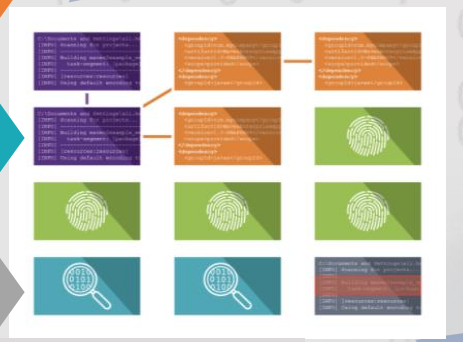
1. 定义项目分发类型(External, Internal)
2. 定义项目重要级别
3. 定义白名单
4. 定义黑名单

依赖分析（扫描效率高，精度高，只能覆盖依赖开源组件，不同阶段依赖的组件不同，支持动态构建过程和非动态构建识别）(Rapid → Full)

特征分析（扫描速度快，精度适中，覆盖自研代码或者是Vendor SDK和外部库）

片段分析（扫描速度慢，需要人工确认，覆盖自研代码）

二进制分析（扫描速度适中，扫描最终打包文件检查，容器，系统依赖文件）



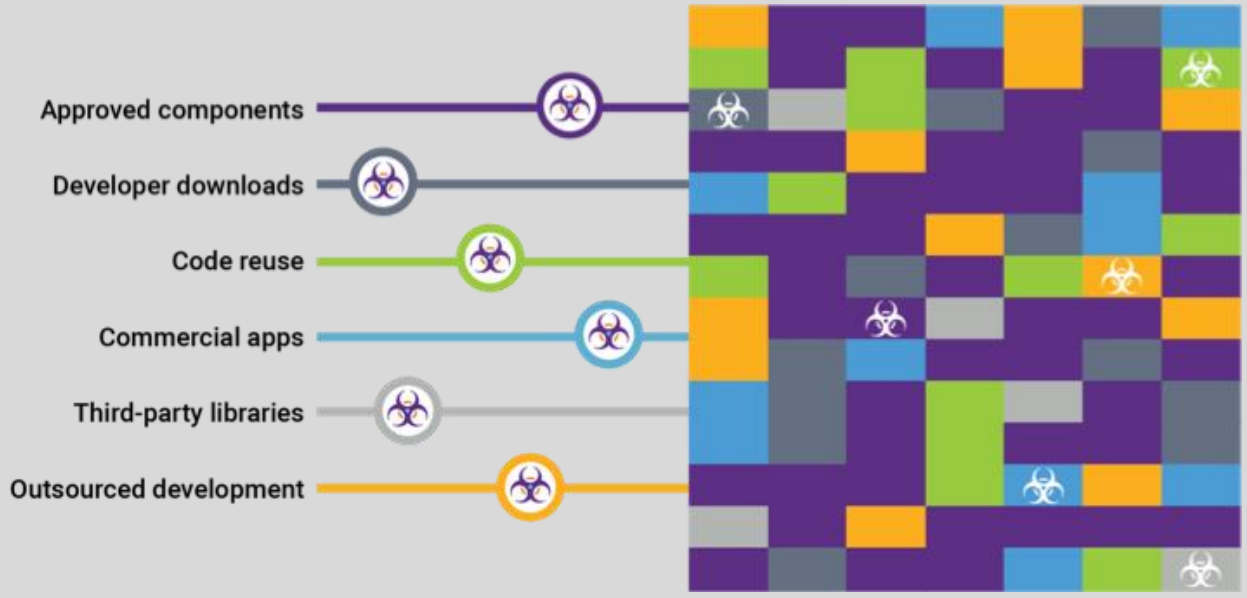
SBOM

Monitoring and alerts

开源供应链治理的「道」与「术」

GOTC

条条小道儿通开源……



11%的Java代码库包括易受攻击的log4j版本说明

- 企业误以为开源软件与商业软件一样经历了完善的安全检测和保护 ---「**理念不对**」
- 企业无法全面识别其应用中的开源组件 ---「**方法不对**」

「道 · 理念」
信任，但要验证

- 盲目信任开源代码的安全性有可能会导致毁灭性后果
- 必须摒弃这种不科学的天然信任，最好以"零信任"的理念对待OSS，与自有代码一视同仁、全面检测

「术 · 方法」
验证，基于SBOM

- SBOM是对抗供应链攻击的首选武器
- SBOM旨在帮助管理开源和第三方代码的使用，提供应用程序组成的可视性
- SBOM列出了应用程序中的所有开源组件，以及这些组件的许可证、版本和补丁状态等，支撑快速锁定风险组件、确定补救措施

GOTC

THANKS

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE