KUBERNETES
COMMUNITY DAYS DALIAN

以SBOM 为基础的云原生应用安全治理

董毅@悬镜安全

一瓶“牛奶”——你会喝吗？

KUBERNETES
COMMUNITY DAYS DALIAN

安全的保障——成分清单和监管机构

KUBERNETES
COMMUNITY DAYS DALIAN

- 成分清单用于实现可见性（透明度）
- 监管机构保障成分清单的可信度

软件物料清单

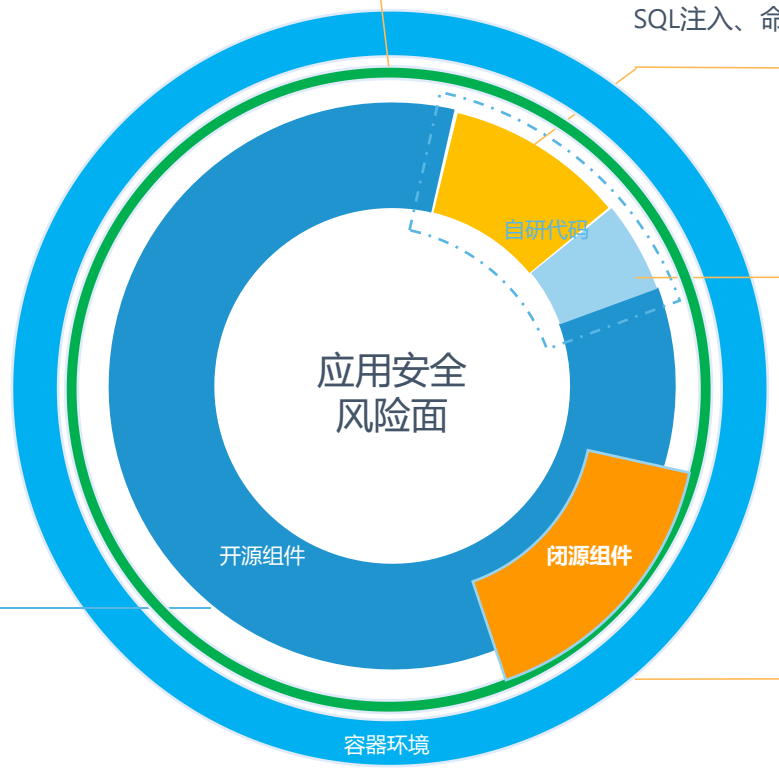
- 软件物料清单（SBOM, Software Bill Of Material）是代码库中所有开放源代码和第三方组件的清单。
- SBOM能够列出管理这些组件的许可证，代码库中使用的组件的版本及其补丁程序状态。

属性	SPDX	CycloneDX	SWID
作者姓名	(2.8) Creator:	metadata/authors/author	<Entity> @role (tag Creator), @name
时间戳	(2.9) Created:	metadata/timestamp	<Meta>
供应商名称	(3.5) PackageSupplier:	Supplier publisher	(softwareCreator/ publisher), @name
组件名称	(3.1) PackageName:	name	<softwareIdentity> @name
版本字符串	(3.3) PackageVersion:	version	<softwareIdentity> @version
组件哈希值	(3.10) PackageChecksum: (3.9) PackageVerificat ionCode:	Hash “alg”	<Payload>/../<File> @[hash-algorithm]:hash
唯一标识符	(2.5)SPDX Document Namespace (3.2) SPDXID:	bom/serialNumber component/bomref	<softwareIdentity> @tagID
关系	(7.1) Relationship: DE- SCRIBES CONTAINS	(Inherent in nestedas- sembly/subassembly and/or dependency graphs)	<Link> @rel, @href

云原生应用安全风险面

- 开发模式：瀑布 > 敏捷 > DevOps
- 应用架构：大型系统 > SOA > 微服务
- 代码实现：闭源 > 开源 > 混源
- 服务器：物理机 > 虚拟化 > 容器化

聚焦到应用系统风险源头



API安全性

失效的用户认证、安全性、错误配置、注入等

Web通用漏洞

SQL注入、命令执行、XXE、XSS等OWASP TOP10

业务逻辑漏洞

水平/垂直越权、短信轰炸、批量注册、验证码绕过等

合规需求、安全配置

未能满足安全合规、未建立安全基线、敏感数据泄漏

第三方组件

开源组件/闭源组件

CNNVD、CNVD、CVE等
开源许可风险

容器环境镜像风险

软件漏洞、恶意程序、敏感信息泄漏、不安全配置、仓库漏洞、不可信镜像

软件物料清单的描述

软件物料清单（SBOM, Software Bill Of Material）是云原生时代应用风险治理的基础设施。

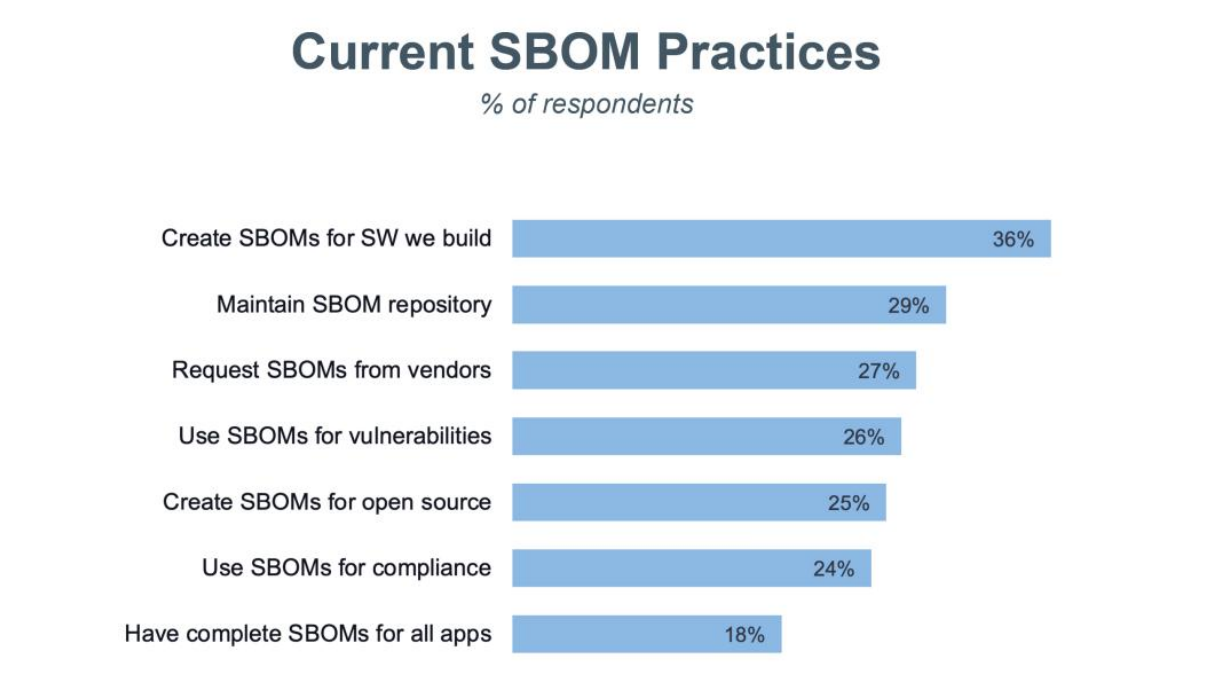
特点：

- 是治理第三方组件风险（开源+闭源）的必备工具；
- 可深度融合于DevOps应用生产模式；
- 可与多种DevSecOps工具链联动强化效能（SCA、RASP、漏洞情报）；
- 在云原生应用的开发端及运营端均发挥作用。

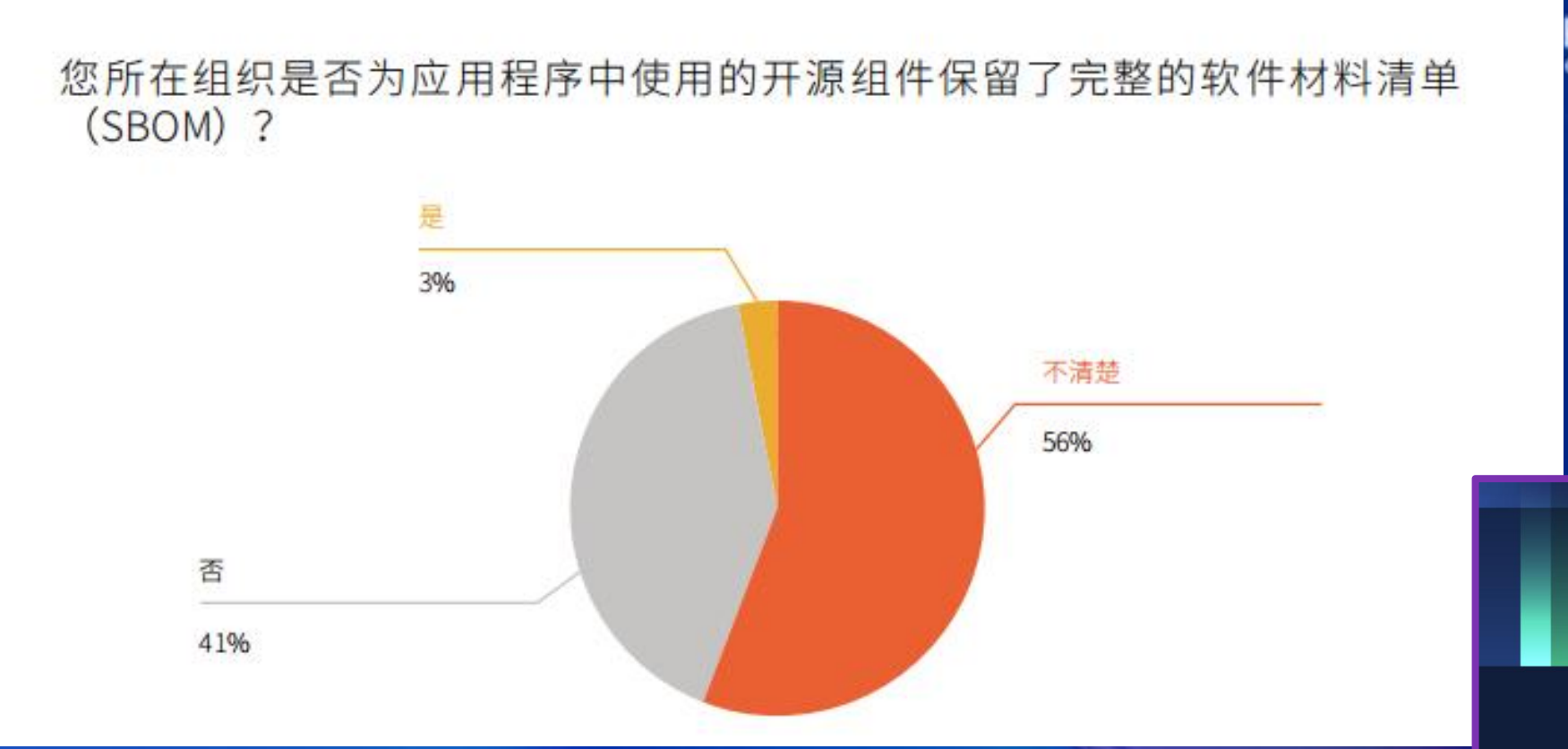
实践现状

SBOM的应用现状

- 根据《Anchore 2022 软件供应链安全报告》，尽管 SBOM 在提供对云原生应用可见性方面发挥着基础性作用，但只有三分之一的组织遵循 SBOM 最佳实践。



SBOM的应用现状


您所在组织是否为应用程序中使用的开源组件保留了完整的软件材料清单（SBOM）？
是
3%
不清楚
56%
否
41%


悬镜 | FreeBuf咨询
2022
DevSecOps
行业洞察报告

云原生基于“责任自负”的开源世界

KUBERNETES
COMMUNITY DAYS DALIAN

GitHub

O. Limitation of Liability

Short version: *We will not be liable for damages or losses arising from your use or inability to use the service or otherwise arising under this agreement. Please read this section carefully; it limits our obligations to you.*

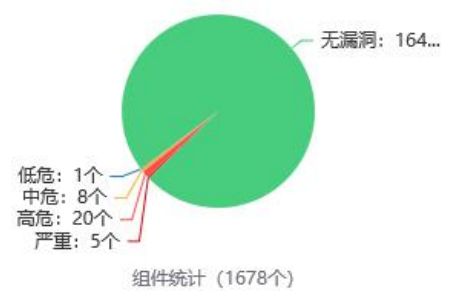
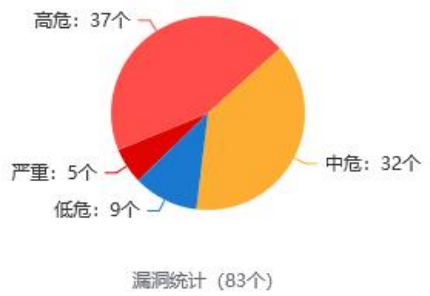
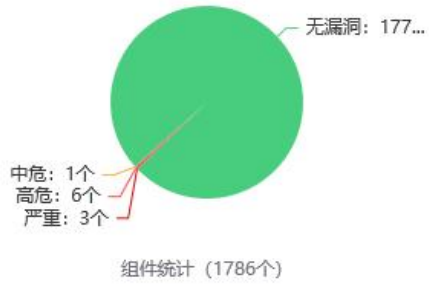
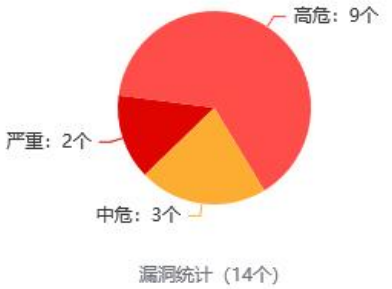
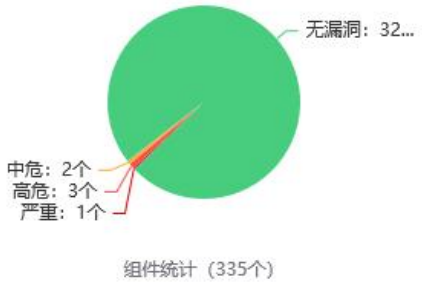
You understand and agree that we will not be liable to you or any third party for any loss of profits, use, goodwill, or data, or for any incidental, indirect, special, consequential or exemplary damages, however arising, that result from

- the use, disclosure, or display of your User-Generated Content;
- your use or inability to use the Service;
- any modification, price change, suspension or discontinuance of the Service;
- the Service generally or the software or systems that make the Service available;
- unauthorized access to or
- statements or conduct of
- any other user interaction
- any other matter relating

Apache License 2.0

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

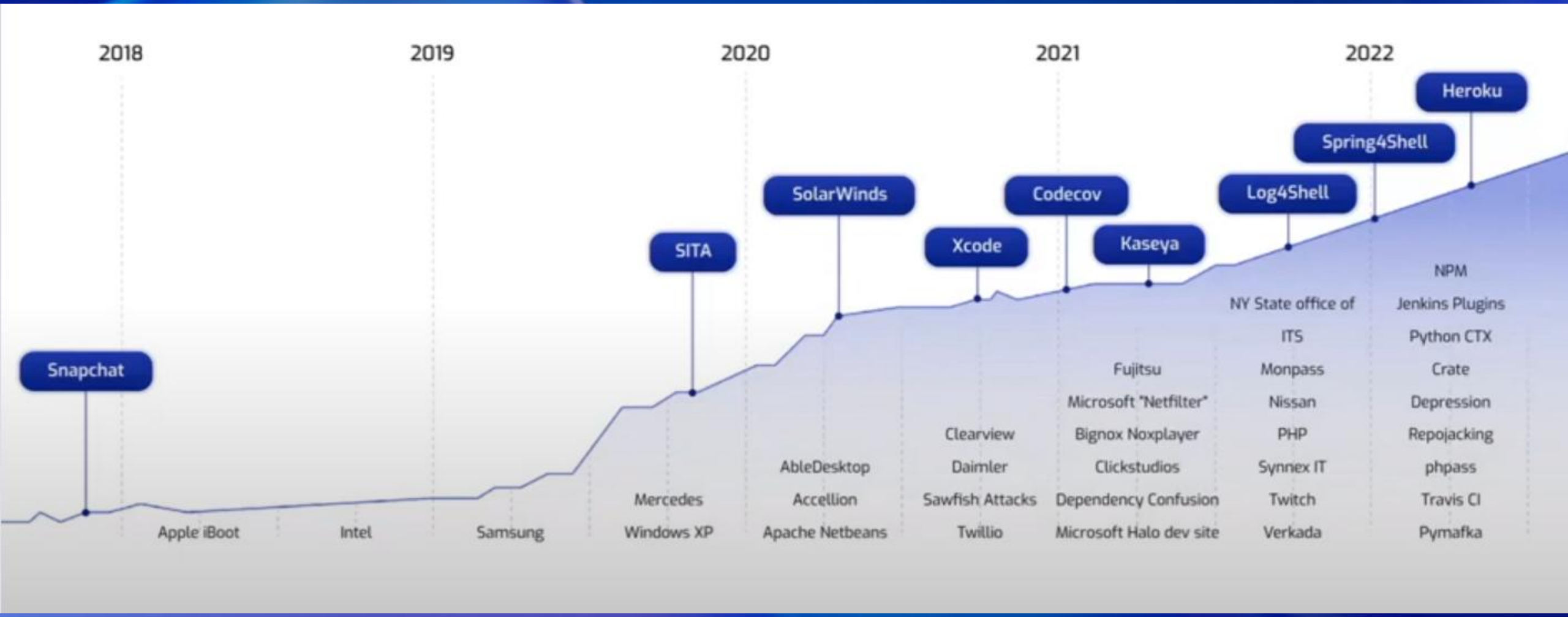
云原生开源应用漏洞

	OpenSCA扫描结果	
h********r-main		
p********s-main		
c**********a-main		

云原生时代下的软件供应链攻击

“到2025年，全球45%的组织会受到软件供应链攻击，比2021年增长三倍”——Gartner

KUBERNETES
COMMUNITY DAYS DALIAN

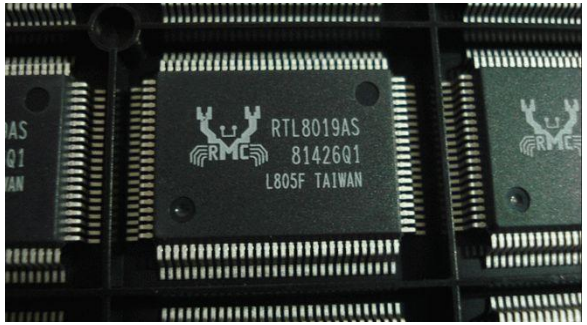


云原生时代下的软件供应链攻击

软件供应链安全事件频发，“核弹级”第三方组件漏洞的影响面和危害大

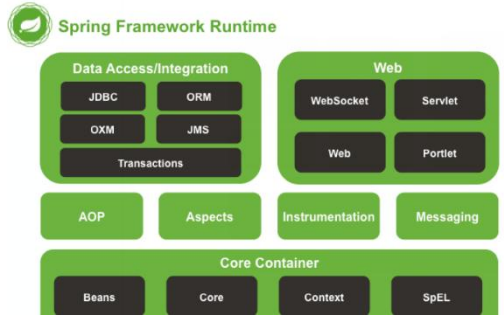
2020 年12月，美国企业和政府网络突遭“太阳风暴”攻击。黑客利用太阳风公司（SolarWinds）的网管软件漏洞，攻陷了多个美国联邦机构及财富 500 强企业网络。2020 年 12 月 13 日，美国政府确认国务院、五角大楼、国土安全部、商务部、财政部、国家核安全委员会等多个政府部门遭入侵。该事件波及全球多个国家和地区的 18000 多个用户，被认为是“史上最严重”的供应链攻击。

“太阳风暴”攻击



2021年12月，Apache开源组件Log4j被发现两个相关漏洞，分别为任意代码执行漏洞和拒绝服务攻击漏洞，攻击者可以通过构造特殊的请求进行任意代码执行，以达到控制服务器、影响服务器执行的目的。该漏洞已影响超6万个开源软件，涉及相关版本软件包32万余个,被认为是“2021年最重要的安全威胁之一”

Apache Log4j2 漏洞



Realtek 的WiFi SDK漏洞

2021年8月，中国台湾芯片厂商Realtek 发布安全公告称在其软件开发套件和WiFi模块中发现了4个安全漏洞。、攻击者可利用该漏洞绕过身份验证，并以最高权限运行恶意代码，有效接管设备。本次暴出漏洞的芯片至少有65家供应商在使用，生产出的设备数量超过十万台。

Spring 框架漏洞

2022年3 月 30 日，国家信息安全漏洞共享平台（CNVD）收录 Spring 框架远程命令执行漏洞（CNVD-2022-23942）。攻击者利用该漏洞，可在未授权的情况下远程执行命令，该漏洞被称为“核弹级”漏洞。使用 JDK9 及以上版本皆有可能受到影响。

软件下载投毒、SDK/恶意代码污染、基础开源组件漏洞、商业许可证限制

Equifax信息泄露事件

2017.02.14
一名安全研究员发现了Struts漏洞，并通过其安全邮件列表向Apache报告了该漏洞

2017.03.07
Apache Struts项目管理委员会(PMC)公开披露了Struts漏洞，且该漏洞被评为10分

2017.03.08
国土安全部的计算机安全应急小组(U.S.-CERT)向Equifax发出了一份关于需要修补Apache Struts漏洞的通知

2017.03.09
Equifax的GTVM小组对漏洞进行了通告，并强调将其升级到指定的Struts 2版本

2017.03.14
Equifax的应急威胁小组发布了一个Snort特征规则，应对此攻击

2017.03.10
攻击者利用该漏洞并执行了“whoami”命令去发现Equifax其他潜在受影响的服务器

2017.07.29
Equifax的对抗小组将67个新的SSL证书上传到数据中心的SSL Visibility（SSLV）设备上，恢复了入侵检测与防御系统对流量的分析和识别

2017.08.02
Equifax联系了外部律师，并聘请网络安全公司Mandiant完成对数据泄露的全面调查分析并确定入侵的范围，同时向联邦调查局通报了这一事件

2017.08.11
Mandiant安全公司首先确认了攻击者对用户PII数据的访问

2017.09.07
Equifax宣布公司发生了一起“网络安全事件”，影响到大约1.43亿美国消费者，包括姓名、社会安全号码、出生日期、地址和驾驶执照、209000个信用卡号码和182000份用户信用报告申诉文件

2017.09.01
Equifax召开了董事会会议，讨论调查、受侵害的PII范围、以及对外通知的计划

2017.08.17
Equifax已经确定有大量的用户数据被侵害，包括首席执行官、首席信息官、首席法务官、首席财务官、被入侵系统ACIS的业务负责人、Mandiant安全公司代表、外部律师一起讨论了调查结果

KUBERNETES
COMMUNITY DAYS DALIAN

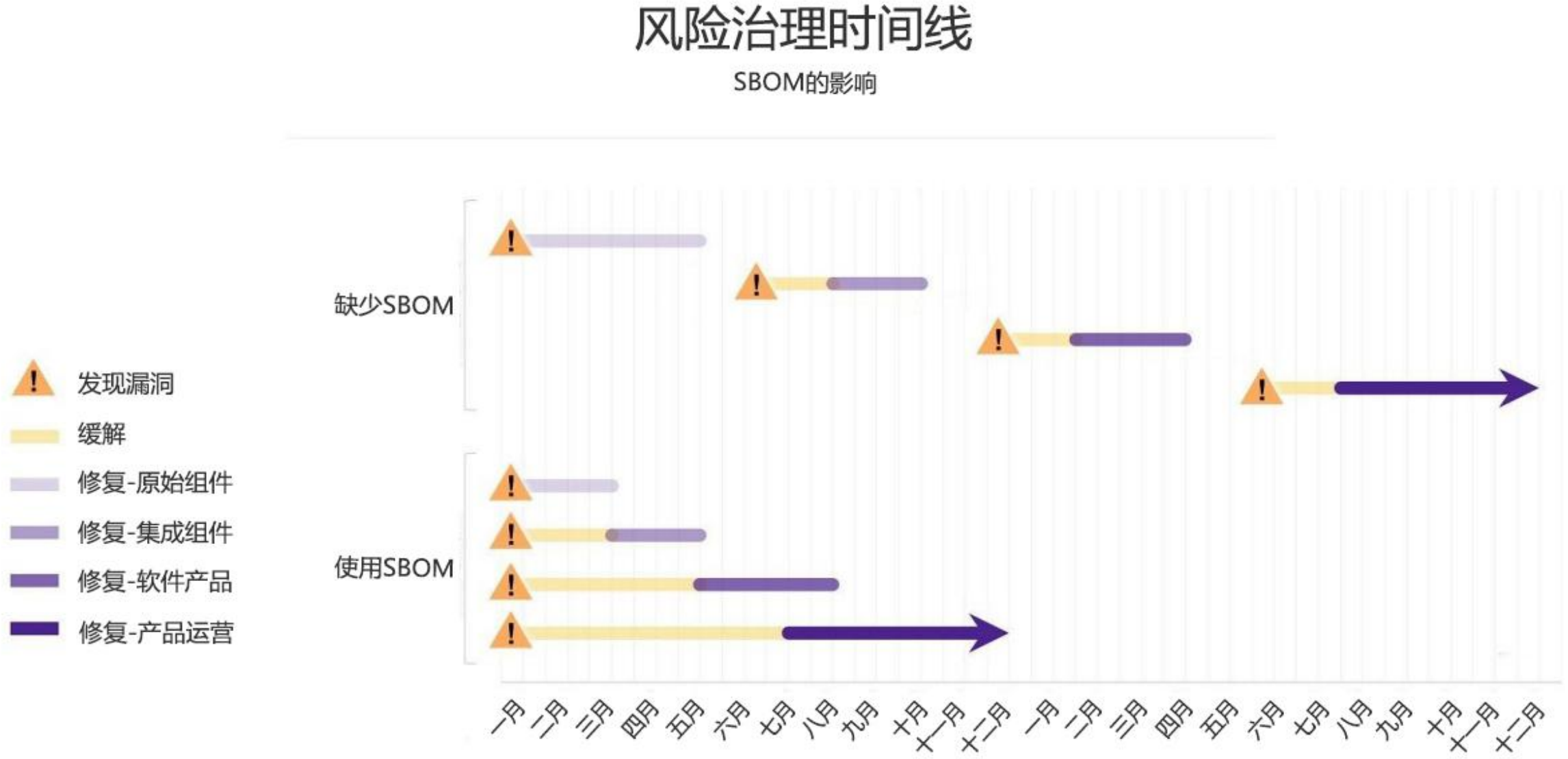
SBOM概述

SBOM的作用

实施 SBOM 有助于揭示整个软件供应链中的漏洞与弱点，提高软件供应链的透明度，减轻软件供应链攻击的威胁，驱动云原生应用的安全。

通过使用 SBOM 可以帮助企业进行漏洞管理、应急响应、资产管理、许可证和授权管理、知识产权管理、合规性管理、基线建立和配置管理等。

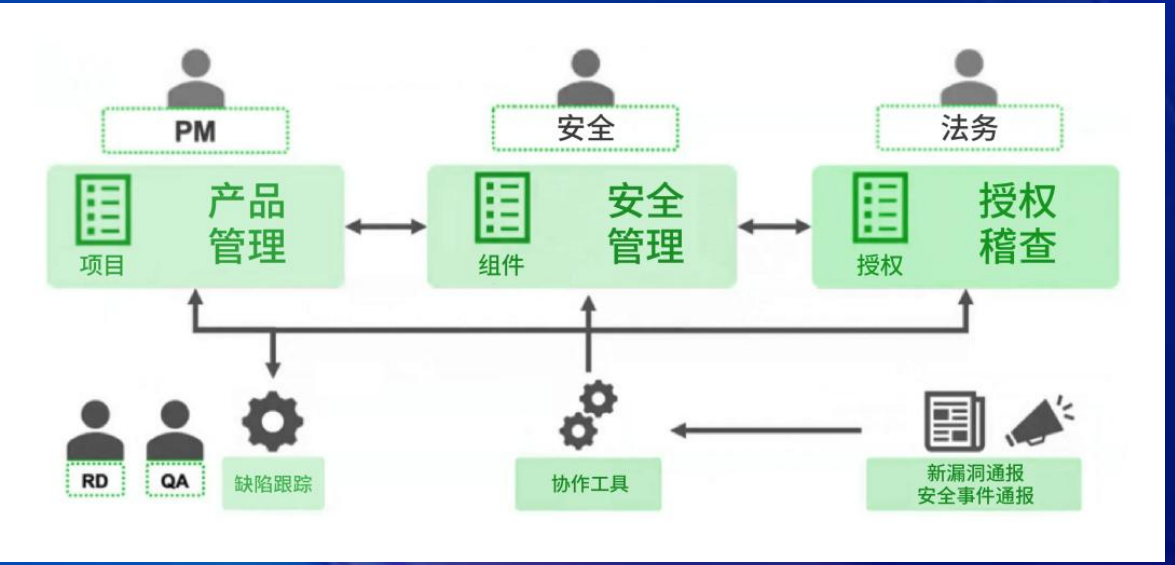
作用示例


风险治理时间线
SBOM的影响
缺少SBOM
使用SBOM
发现漏洞
缓解
修复-原始组件
修复-集成组件
修复-软件产品
修复-产品运营
一月
二月
三月
四月
五月
六月
七月
八月
九月
十月
十一月
十二月
一月
二月
三月
四月
五月
六月
七月
八月
九月
十月
十一月
十二月

SBOM使用场景

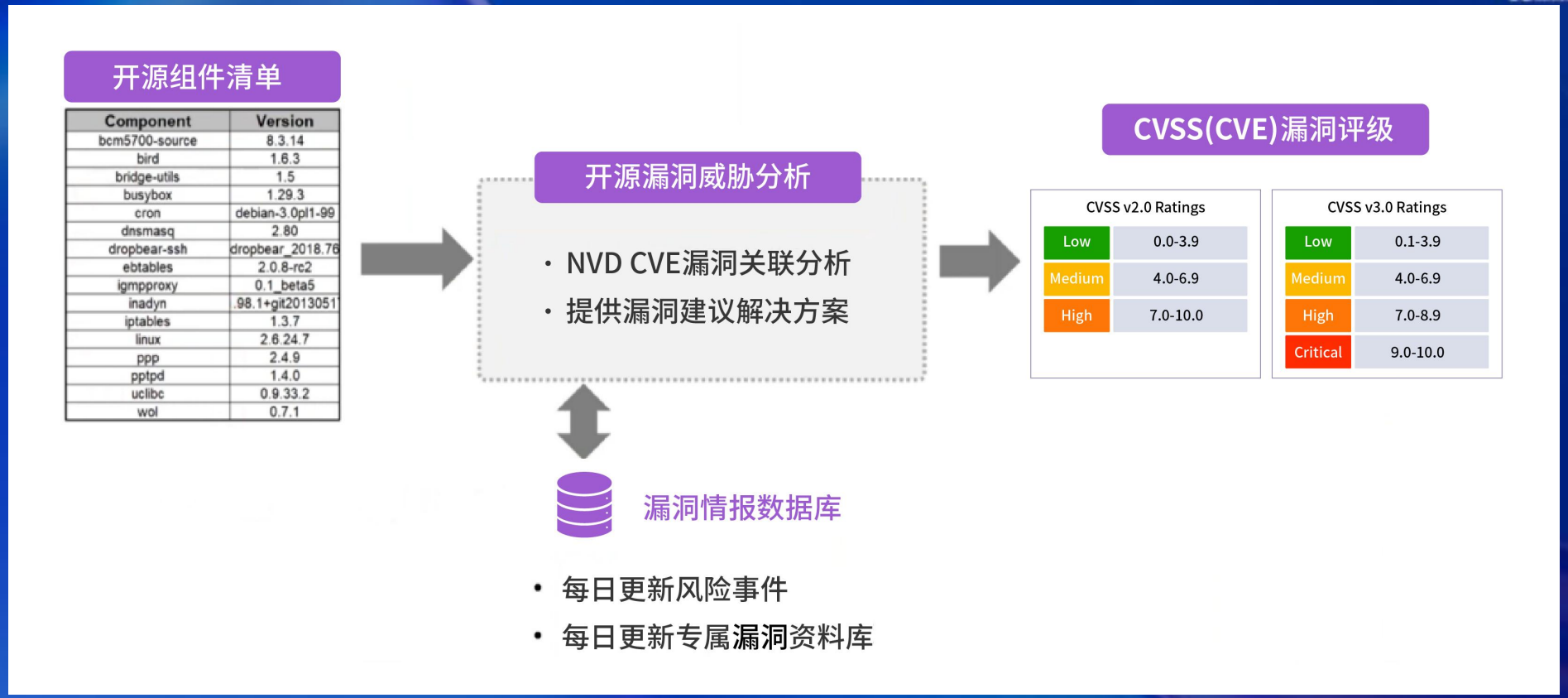
1）从广义的分类上看，SBOM有三种不同的使用主体：
〇 软件生产商使用SBOM来协助构建和维护他们提供的软件；
〇 软件采购商使用SBOM来进行采购前参考、协商折扣和制定采购策略；
〇 软件运营商使用SBOM为漏洞管理和资产管理提供信息，管理许可和合规性，并快速识别软件和组件依赖关系以及供应链风险。

2）从企业角色类型来看，对SBOM有不同的使用需求：
〇 开发团队：用于管理软件资产，在开发早期即可评估安全风险，筛选适合的组件/软件，并持续更新SBOM；
〇 安全团队：通过提交的SBOM分析软件风险，并通过统一管理进行持续监控，及时响应安全事件；
〇 法务团队：核查软件授权问题，避免后续公司业务自身权益受到损害。



实践要点

实践要点——与漏洞情报关联



实践要点——拥抱自动化

KUBERNETES
COMMUNITY DAYS DALIAN

Automation Support

Support for automation, including automatic generation and machine-readability, allows the ability to scale across the software ecosystem, particularly across organizational boundaries. Taking advantage of SBOM data will require tooling, which necessitates predictable implementation and data formats. For example, some agencies may want to integrate this capability into their existing vulnerability management practices; others might desire real-time auditing of compliance against security policies. Automation will be key for both, which in turn requires common, machine-readable data formats.

——《The Minimum Elements for an SBOM》，NTIA

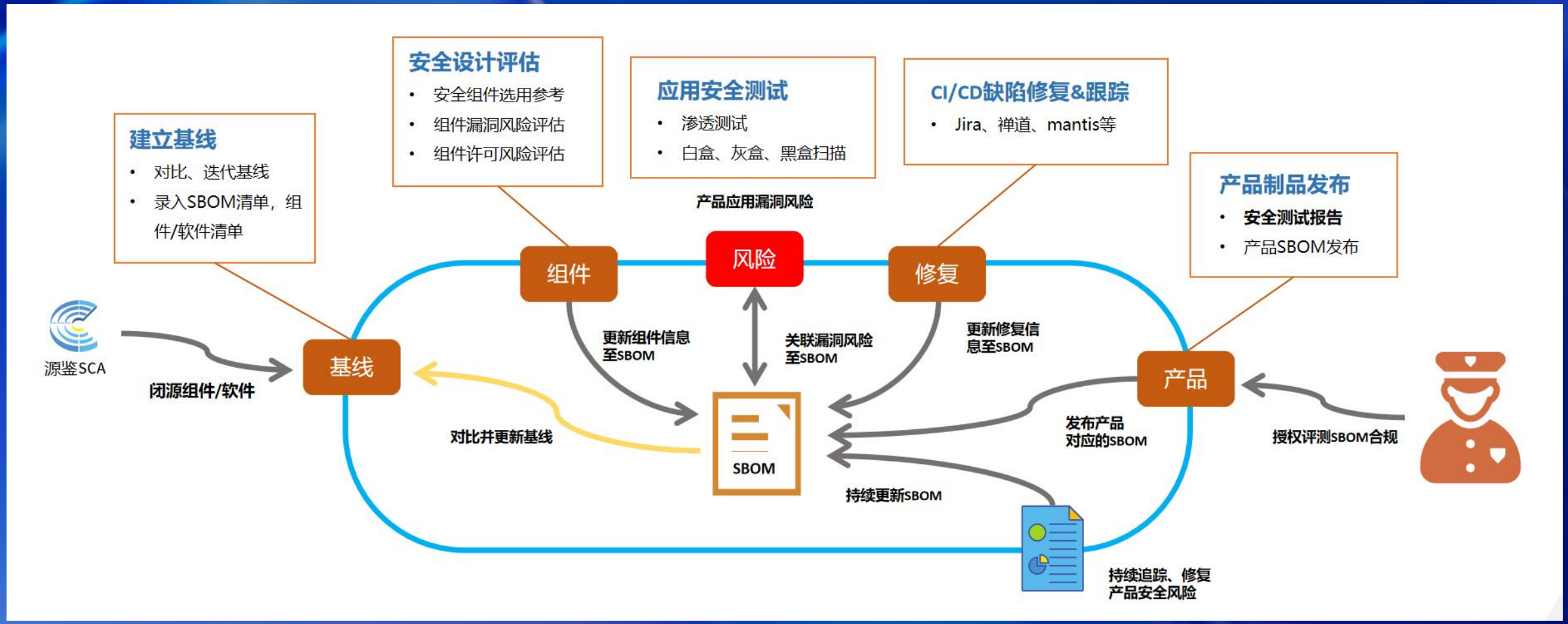
实践要点——使用标准化格式

属性	SPDX	CycloneDX	SWID
作者姓名	(2.8) Creator:	metadata/authors/author	<Entity> @role (tagCreator), @name
时间戳	(2.9) Created:	metadata/timestamp	<Meta>
供应商名称	(3.5) PackageSupplier:	Supplier publisher	<Entity> @role (softwareCreator/ publisher), @name
组件名称	(3.1) PackageName:	name	<softwareIdentity> @name
版本字符串	(3.3) PackageVersion:	version	<softwareIdentity> @ version
组件哈希值	(3.10) PackageChecksum: (3.9) PackageVerificat ionCode:	Hash “alg”	<Payload>/../<File> @ [hash-algorithm]:hash
唯一标识符	(2.5) SPDX Document Namespace (3.2) SPDXID:	bom/serialNumber component/bomref	<softwareIdentity>@tagID
关系	(7.1) Relationship: DESCRIBES CONTAINS	(Inherent in nestedassembly/ subassembly and/or dependency graphs)	<Link> @rel, @href

美国国家电信和信息管理局（NTIA）发布的《构建软件组件透明度：建立通用软件物料清单（SBOM）》第二版中提出：**SBOM 是一个包含软件组件列表和层次依赖信息且机器可读的规范性清单。**

实践要点——固化到流程和体系

围绕SBOM建立管理流程



轻量方案

落地方案

KUBERNETES
COMMUNITY DAYS DALIAN

源头检测

开发测试：将SCA工具对接到DevOps流程里，对编译构建环节卡点，保障软件构建时所依赖组件的安全性，确保不引入存在漏洞的组件；使用基于插桩技术的IAST工具，在功能测试的同时，检测是否存在高危漏洞风险，并展示漏洞触发数据流，便于修复指导。

出厂免疫

积极防御：针对今后随时可能爆发的未知0DAY漏洞，推荐使用RASP应用自防御能力，针对该类漏洞的攻击利用方式精准有效的防护。它可以通过应用的函数行为分析、上下文情境感知及热补丁技术有效阻断绝大部分RCE类未知漏洞攻击。

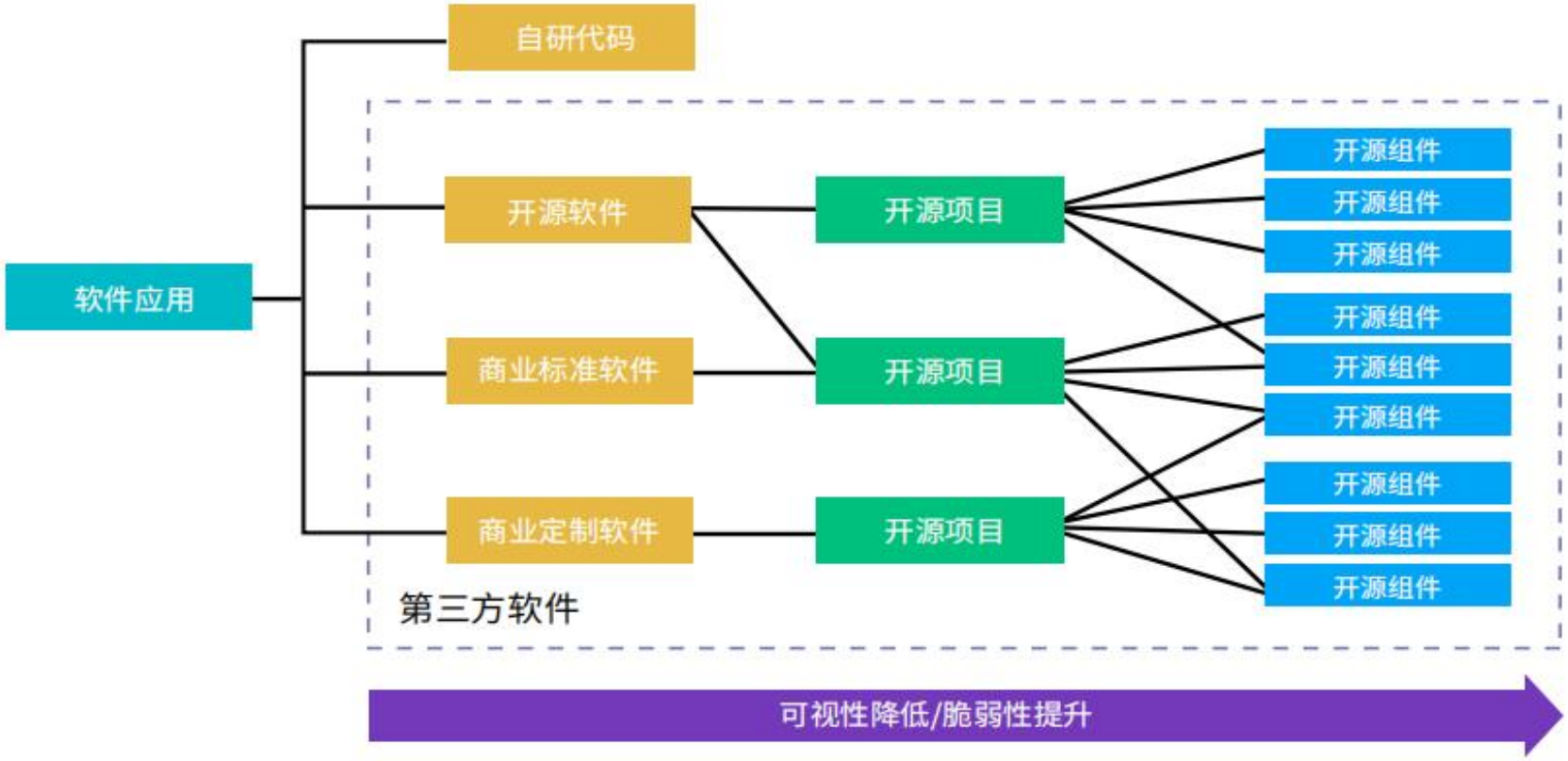
持续运营

安全运营：常态化使用和运营安全可信的制品库，通过SCA和SBOM持续为每个应用程序构建详细的软件物料清单，全面洞察每个应用软件的组件情况。RASP配合开源漏洞情报，第一时间发现并处理开源漏洞风险。

SCA——获取SBOM

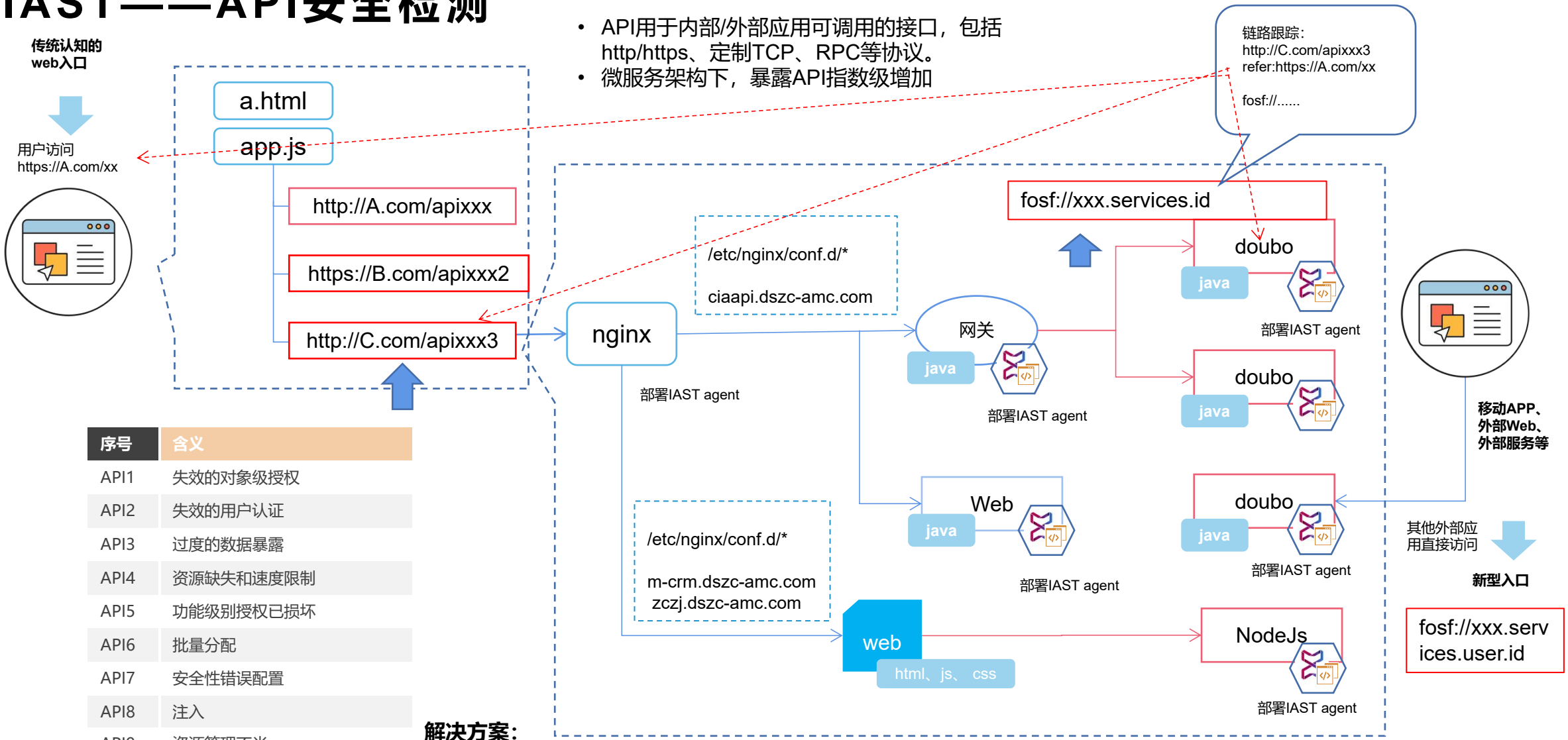
软件成分分析 SCA 技术是通过对二进制软件的组成部分进行识别、分析和追踪的技术。

SCA 可以生成完整的 SBOM，SBOM 作为制品成分清单，同时建立软件构成图谱，为后续分析提供基础，即分析开发人员所使用的各种源码、模块、框架和库，以识别和清点开源软件（OSS）的组件及其构成和依赖关系，并精准识别系统中存在的已知安全漏洞或者潜在的许可证授权问题。



IAST——API安全检测

- API用于内部/外部应用可调用的接口，包括http/https、定制TCP、RPC等协议。
- 微服务架构下，暴露API指数级增加

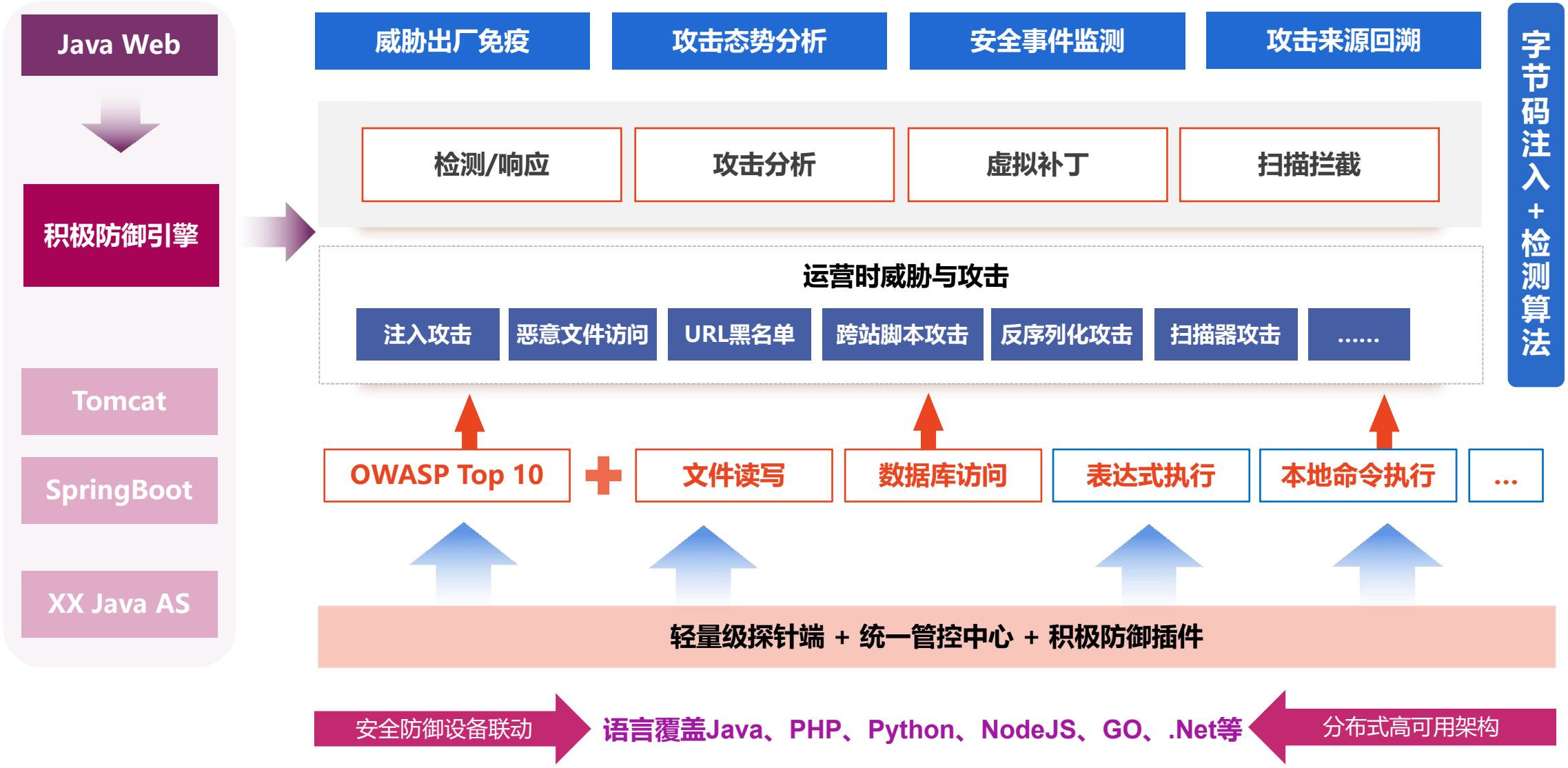


序号	含义
API1	失效的对象级授权
API2	失效的用户认证
API3	过度的数据暴露
API4	资源缺失和速度限制
API5	功能级别授权已损坏
API6	批量分配
API7	安全性错误配置
API8	注入
API9	资源管理不当
API10	日志和监控不足

解决方案：

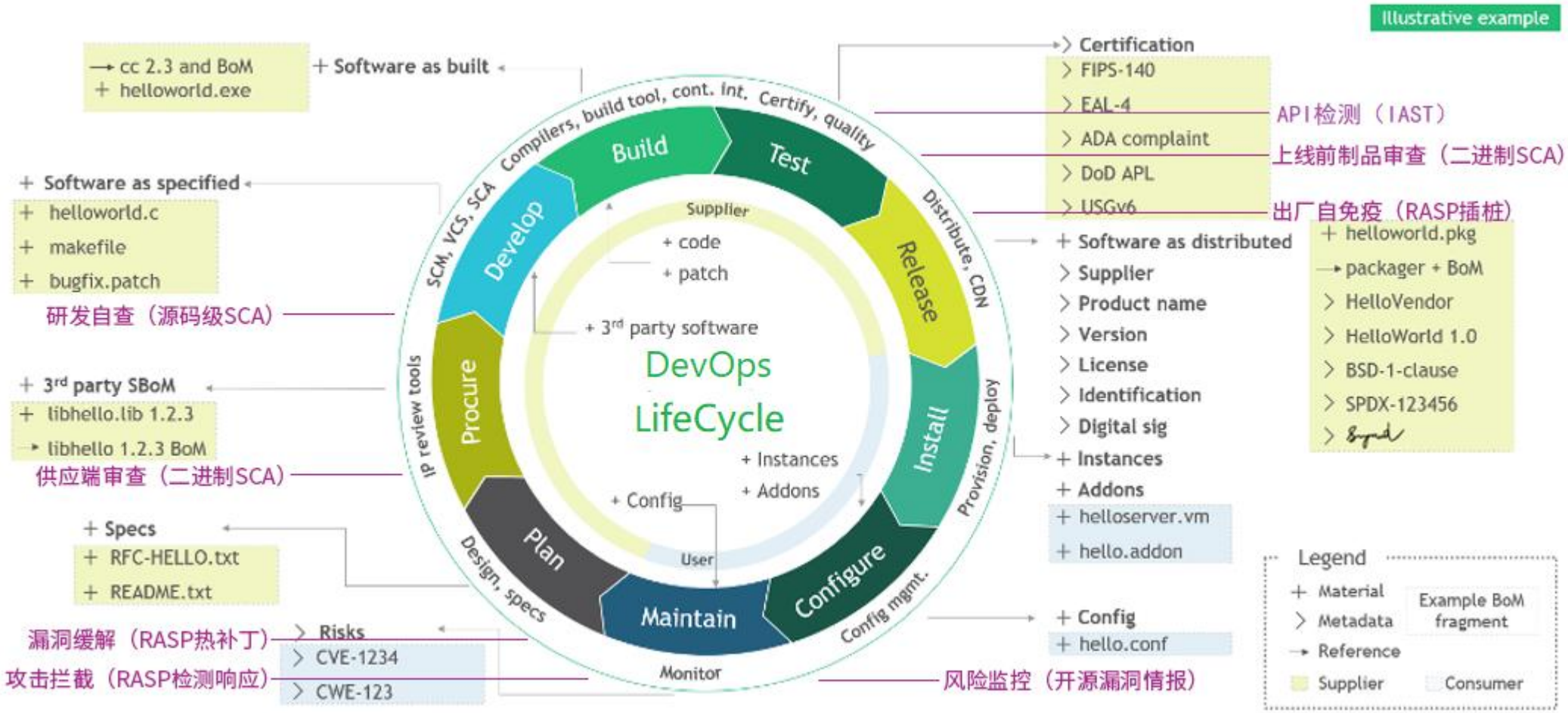
- API资产梳理（暴露面、风险分析）
- API链路调用威胁阻断
- OWASP API安全 TOP 10（权限控制、注入等）

RASP——应用出厂免疫


Java Web
积极防御引擎
Tomcat
SpringBoot
XX Java AS
威胁出厂免疫
攻击态势分析
安全事件监测
攻击来源回溯
检测/响应
攻击分析
虚拟补丁
扫描拦截
运营时威胁与攻击
注入攻击
恶意文件访问
URL黑名单
跨站脚本攻击
反序列化攻击
扫描器攻击
.....
字节码注入+检测算法
OWASP Top 10
文件读写
数据库访问
表达式执行
本地命令执行
...
轻量级探针端 + 统一管控中心 + 积极防御插件
安全防御设备联动
语言覆盖Java、PHP、Python、NodeJS、GO、.Net等
分布式高可用架构

基于SBOM构建云原生应用风险治理流程

SCA+IAST+RASP+漏洞情报



快速获取SBOM——OpenSCA