OWASP
Open Web Application Security Project

用OWASP DependencyTrack管理软件供应链安全

马伟

关于我

马伟

- ThoughtWorks 资深安全咨询师
- 专注应用安全
- OWASP DependencyCheck Gradle插件开源贡献者

OWASP
Open Web Application Security Project

内容大纲

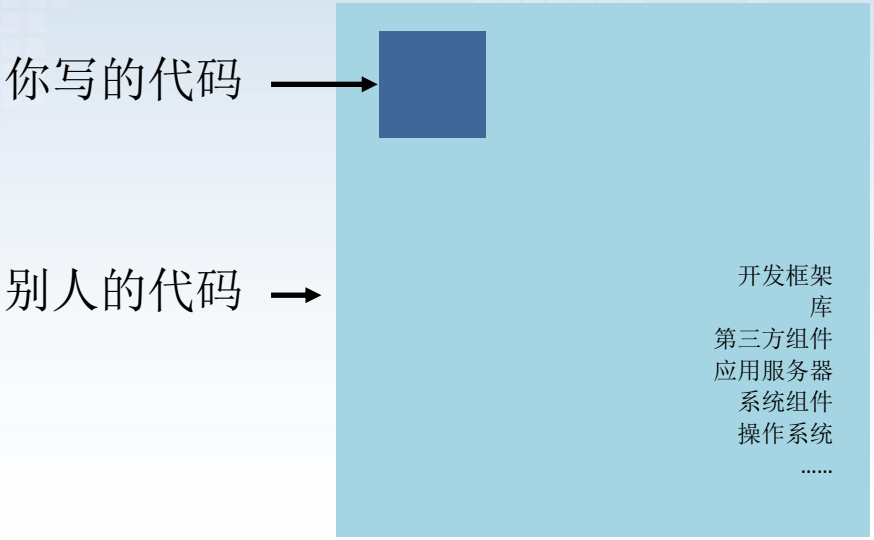
- 软件供应链安全问题
- 流行的开源依赖安全检查工具
- OWASP DependencyTrack特点解析
- OWASP DependencyTrack实践经验分享
- 总结

1.软件供应链安全问题

软件供应链

约80%的代码不是开发团队写的

而是来自应用程序使用的依赖、第三方组件



软件供应链中可能存在的问题

依赖中可能含有已知安全漏洞

案例：Apache Tomcat CVE-2020-1938任意文件读取

攻击者可以利用这个漏洞读取或包含 Tomcat 上所有 webapp 目录下的任意文件

攻击途径：远程网络	攻击复杂度：低
认证：不需要认证	机密性：完全地
完整性：完全地	可用性：完全地

漏洞评分： 10.0

软件供应链中可能存在的问题

依赖的软件许可协议可能存在法律风险

案例：Oracle索赔谷歌88 亿美元的大事件

OpenJDK这个GPL项目的著作权属于Oracle，而当时谷歌工程师直接从OpenJDK复制了9行代码到谷歌的Android项目中。重点在于，Android项目没有按GPL兼容的方式授权，于是触犯了Oracle的著作权，并被要求赔偿88亿美元。

2.流行的开源依赖安全检查工具

流行的应用程序的依赖检查工具

前端应用： NPM Audit 命令足矣

后端应用（尤其Java/.NET）：

- OWASP DependencyCheck
- OWASP DependencyTrack
- Sonatype OSS Index 工具集合
- Github Dependabot

OWASP DependencyCheck

- 识别应用程序依赖是否含有已知安全漏洞
- 多种构建工具支持，如CLI、Maven、Gradle
- 基于NVD漏洞数据库
- 提供HTML报告

Sonatype OSS Index 工具集合

继OWASP DependencyCheck之后，入选最新一期（第23期）ThoughtWorks技术雷达

Scan your projects for open source vulnerabilities, and build security into your development toolchain with native tools and integrations. The following scan tools all utilize the OSS Index public REST API.

Java / JVM

- Maven plugin
- Gradle plugin
- Maven Enforcer rules

JavaScript

- AuditJS scans npm projects
- VS Code plugin

Go

- Nancy scans Golang projects

C/C++

- Cheque scans C/C++ projects

.NET

- Audit.NET scans NuGet projects
- DevAudit is a cross-platform security auditing tool

Python

- ossaudit scans Python projects
- Jake scans Python and Conda projects

PHP

- Bach scans Composer projects

Ruby

- Chelsea scans Ruby projects

Rust

- Cargo Pants scans Cargo projects

R

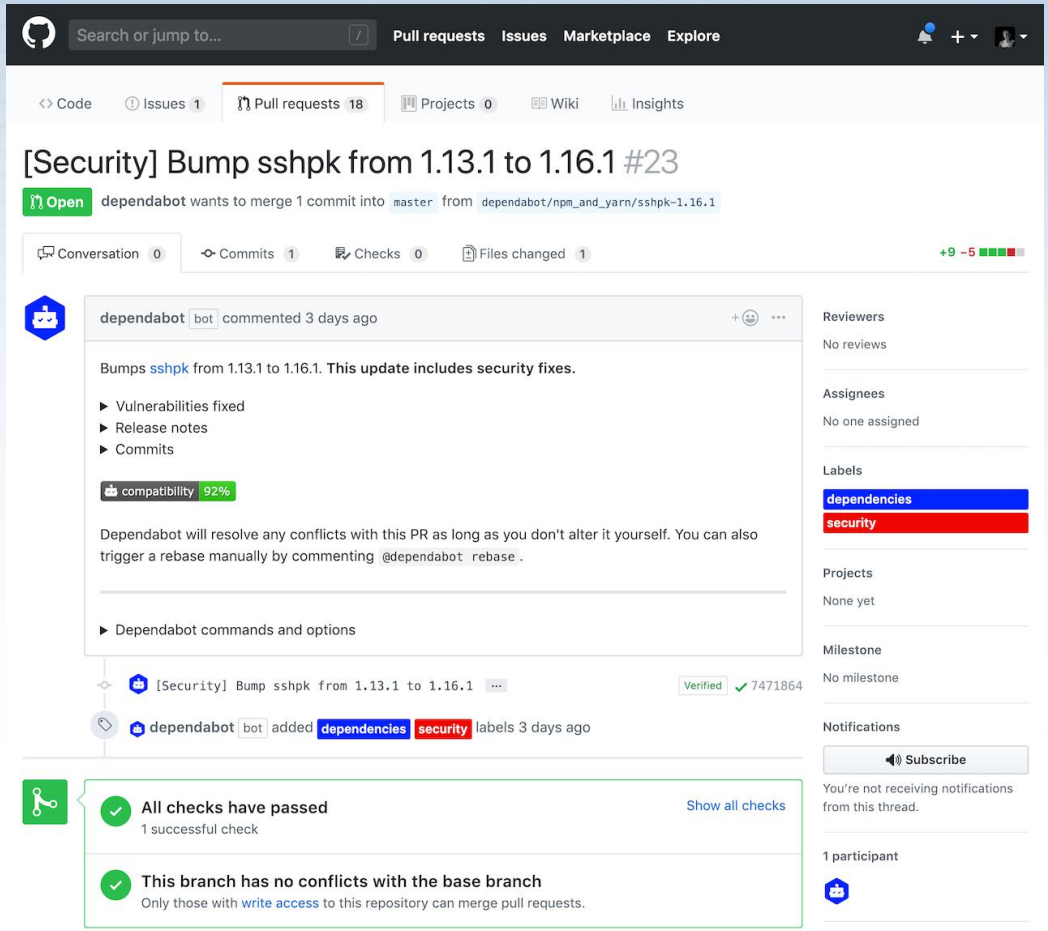
- oysteR scans R projects

Other

- Sonatype DepShield continuously monitors GitHub projects for vulnerabilities
- Ahab scans apt and yum operating systems
- OWASP Dependency-Check is an SCA utility for scanning project dependencies
- OWASP Dependency-Track is a component analysis platform

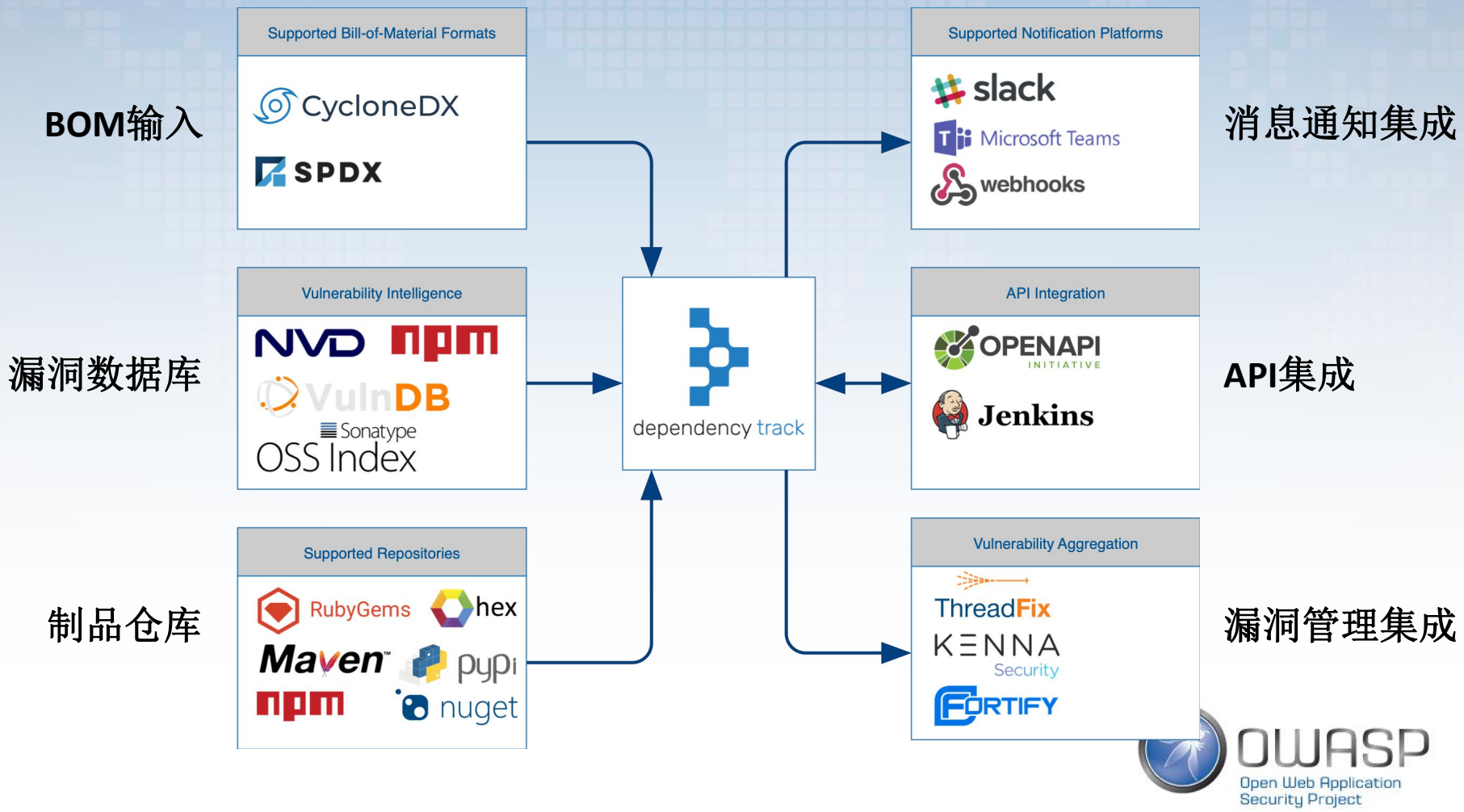
Github Dependabot

自动创建Pull Request，一键升级有问题的依赖，提升开发人员体验



3.OWASP DEPENDENCYTRACK特点解析

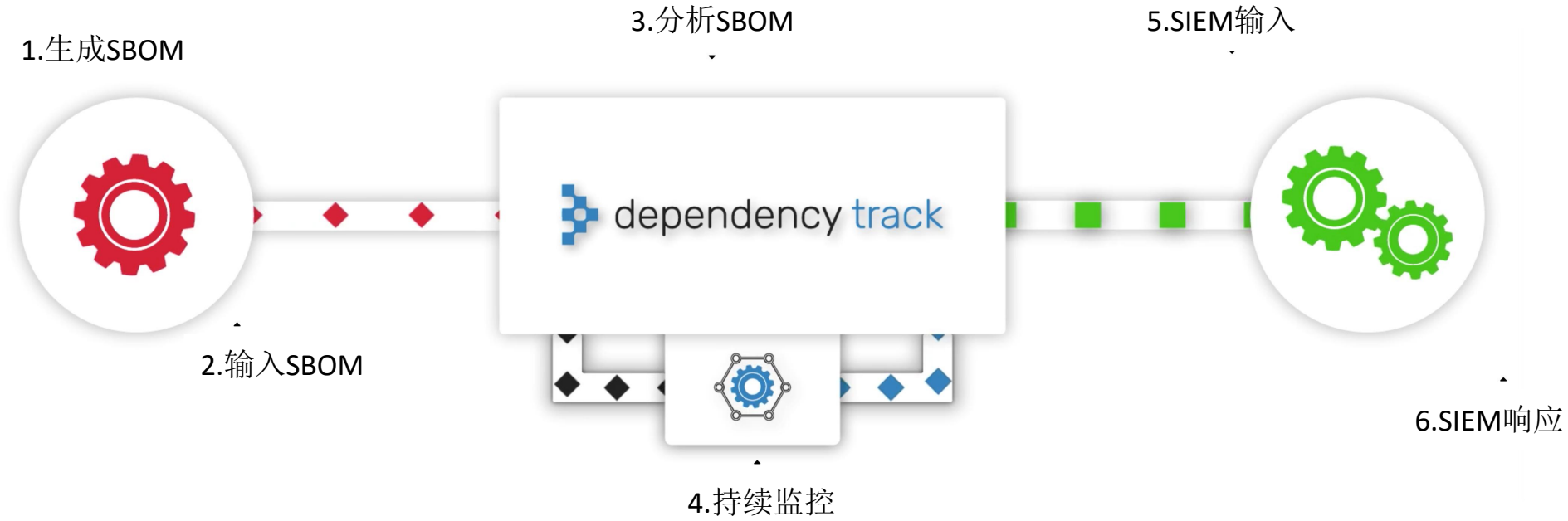
DependencyTrack生态系统



DependencyTrack 特点解析

- 自动、持续监测依赖安全性
- 多种类的漏洞数据库
- 丰富的可视化功能
- 持续集成友好
- 可通过多种渠道发送告警
- AD/LDAP集成
- 丰富的API
- DependencyCheck和DependencyTrack的区别

3.1 自动、持续监测依赖安全性



3.2 多种类的漏洞数据库

- National Vulnerability Database
- NPM Public Advisories
- Sonatype OSS Index
- VulnDB

3.2 多种类的漏洞数据库

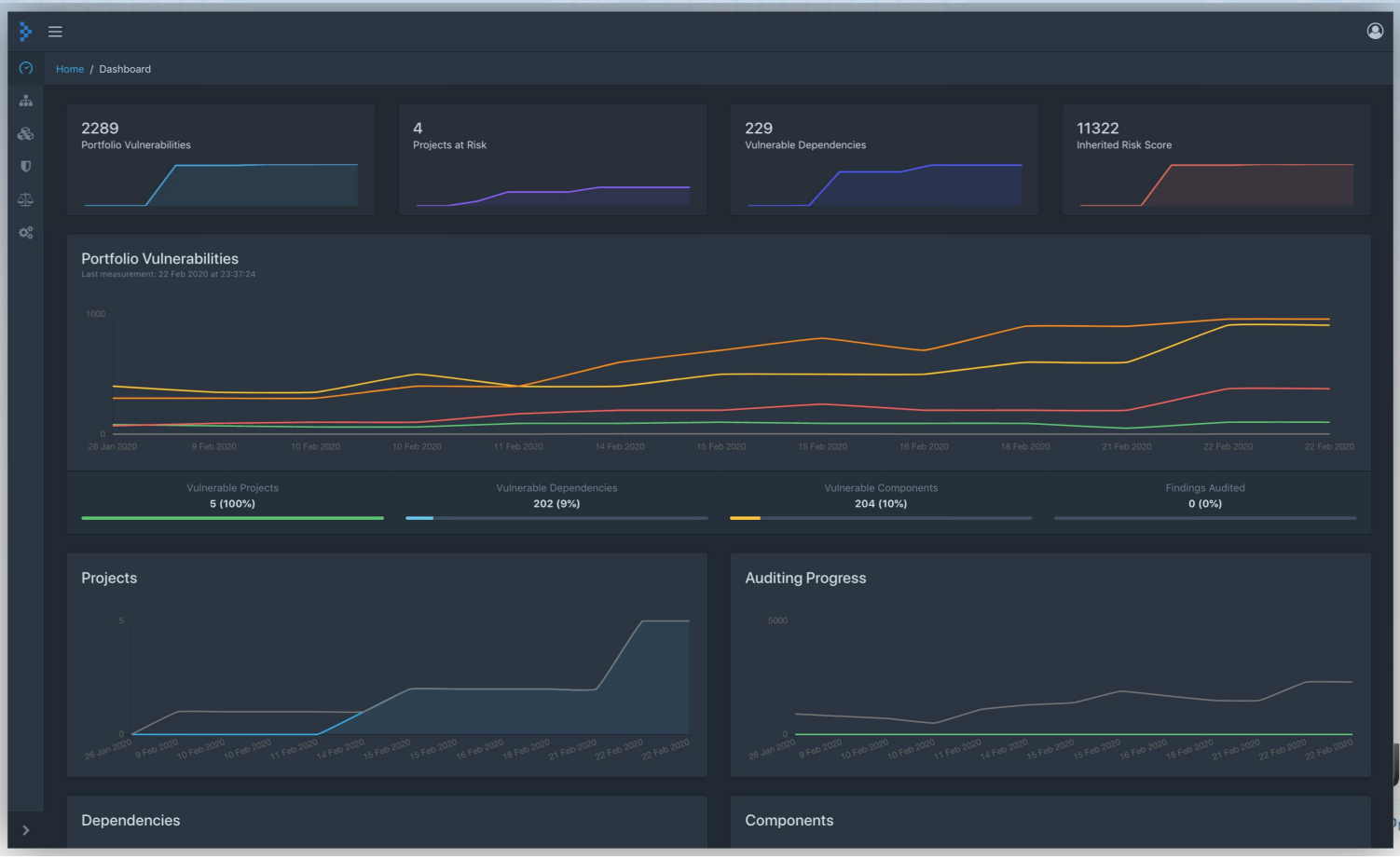
支持多种构建工具、制品库

Ecosystem	Repository	Resolution Order
gem	RubyGems	1
hex	Hex	1
maven	Maven Central	1
	Atlassian Public	2
	JBoss Releases	3
	Clojars	4
	Google Android	5
npm	NPM	1
nuget	NuGet	1
pypi	PyPi	1

OWASP
Open Web Application Security Project

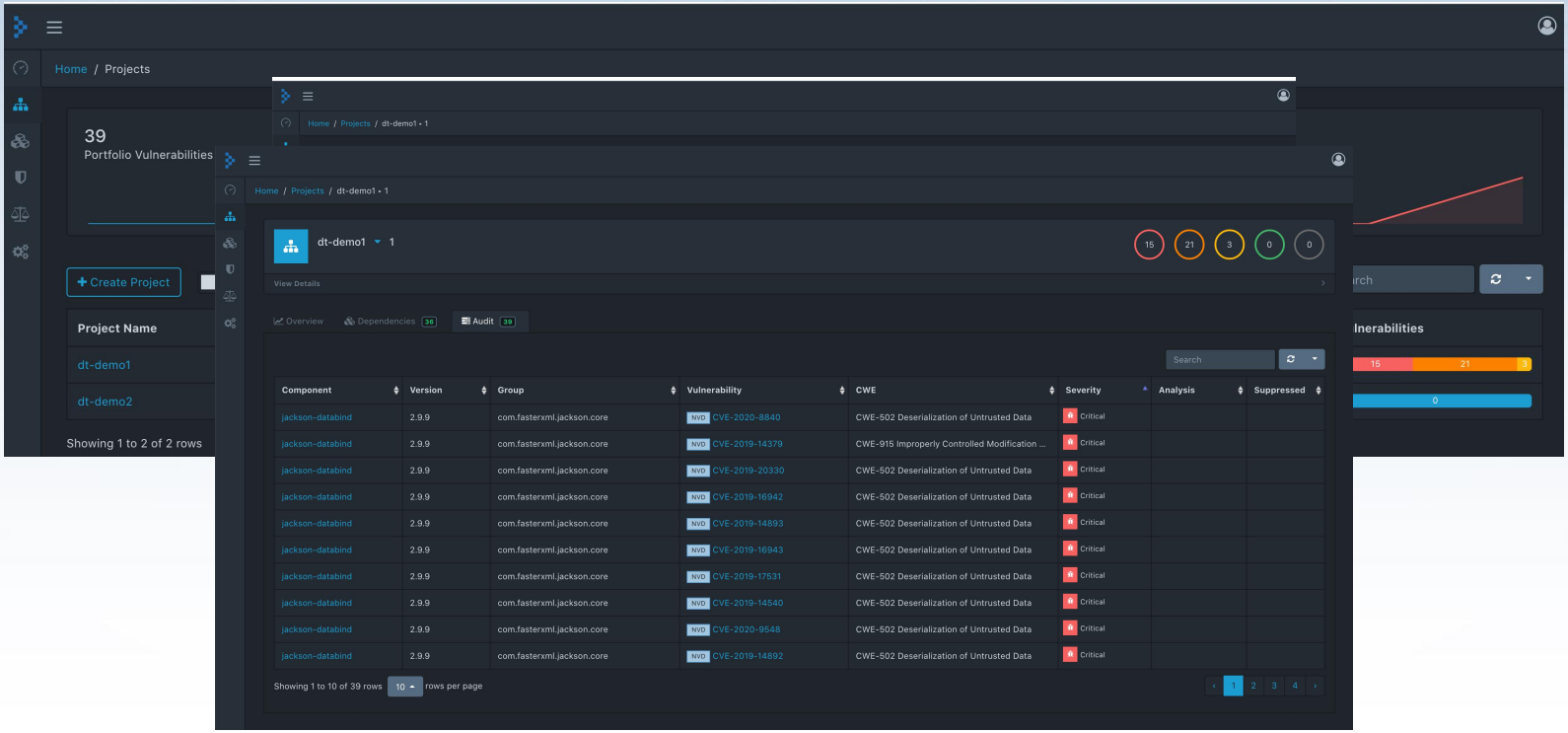
3.3 丰富的可视化功能

通过仪表板，可以直观的追踪依赖安全态势



3.3 丰富的可视化功能

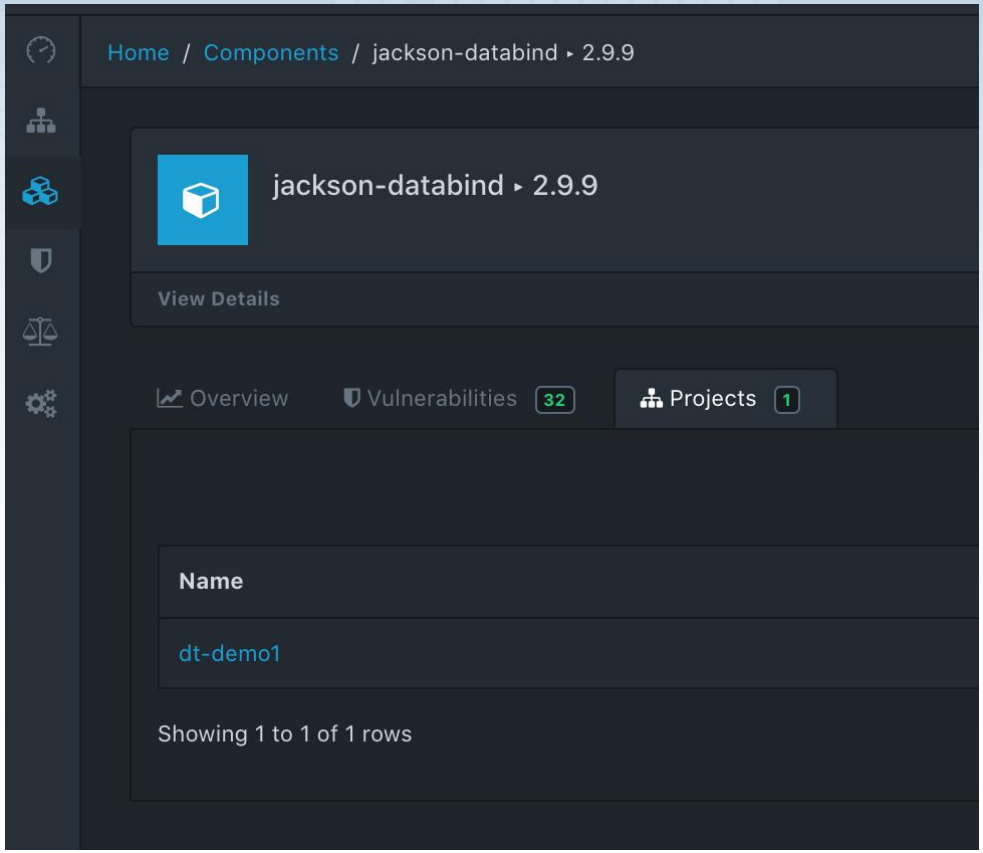
可从Project视角深入了解、审计当前依赖安全状况



OWASP
Open Web Application Security Project

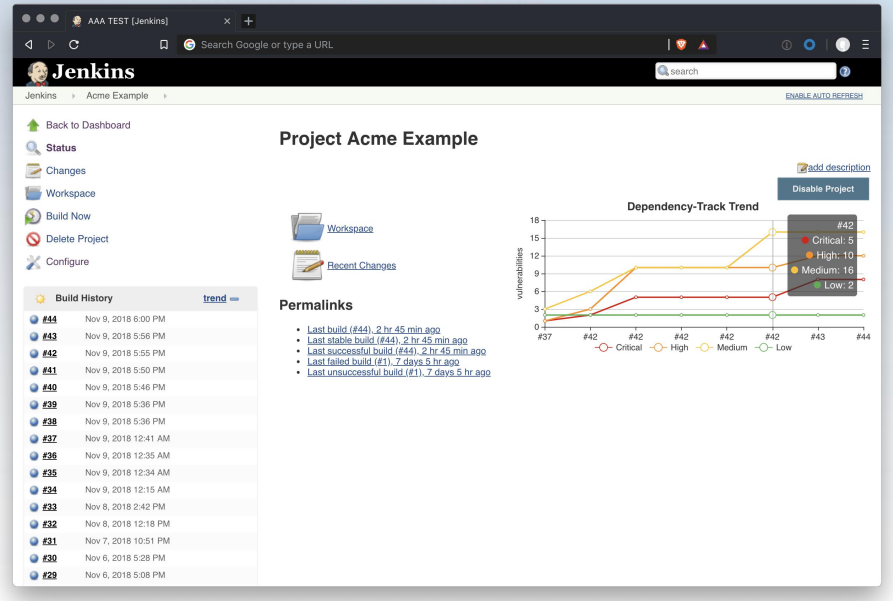
3.3 丰富的可视化功能

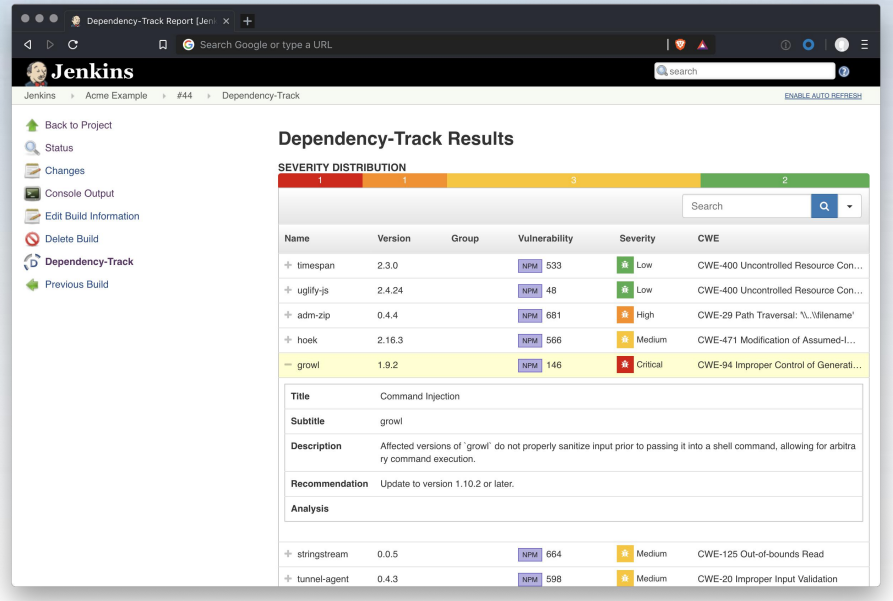
最好用功能：方便排查某含有已知安全漏洞的第三方组件被哪些项目或开发团队使用了



3.4 持续集成友好

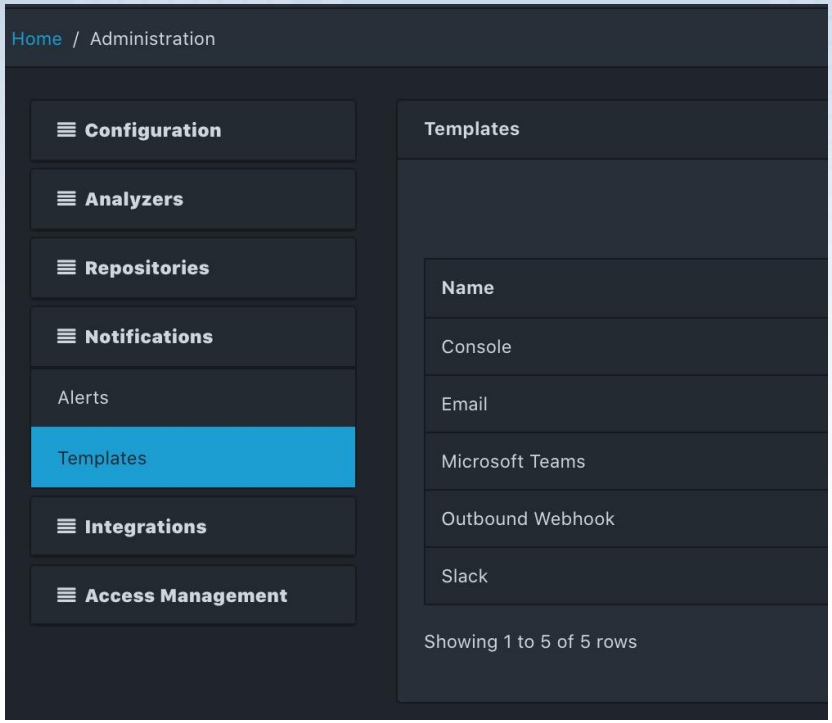
通过Jenkins插件，可以方便的将SBOM信息提交给DependencyTrack,以及查看扫描结果

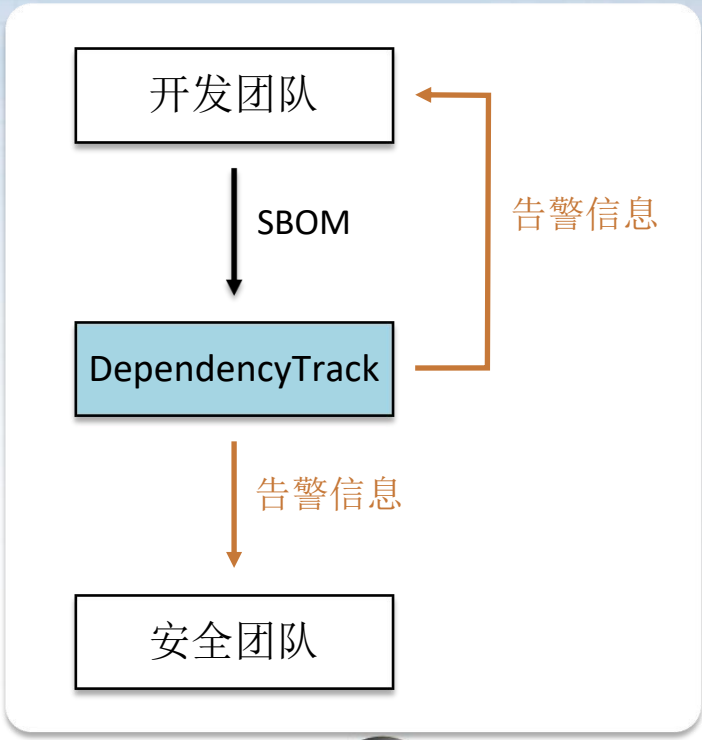




3.5 可通过多种渠道发送告警

- 支持Email、Microsoft Teams、Slack等提醒



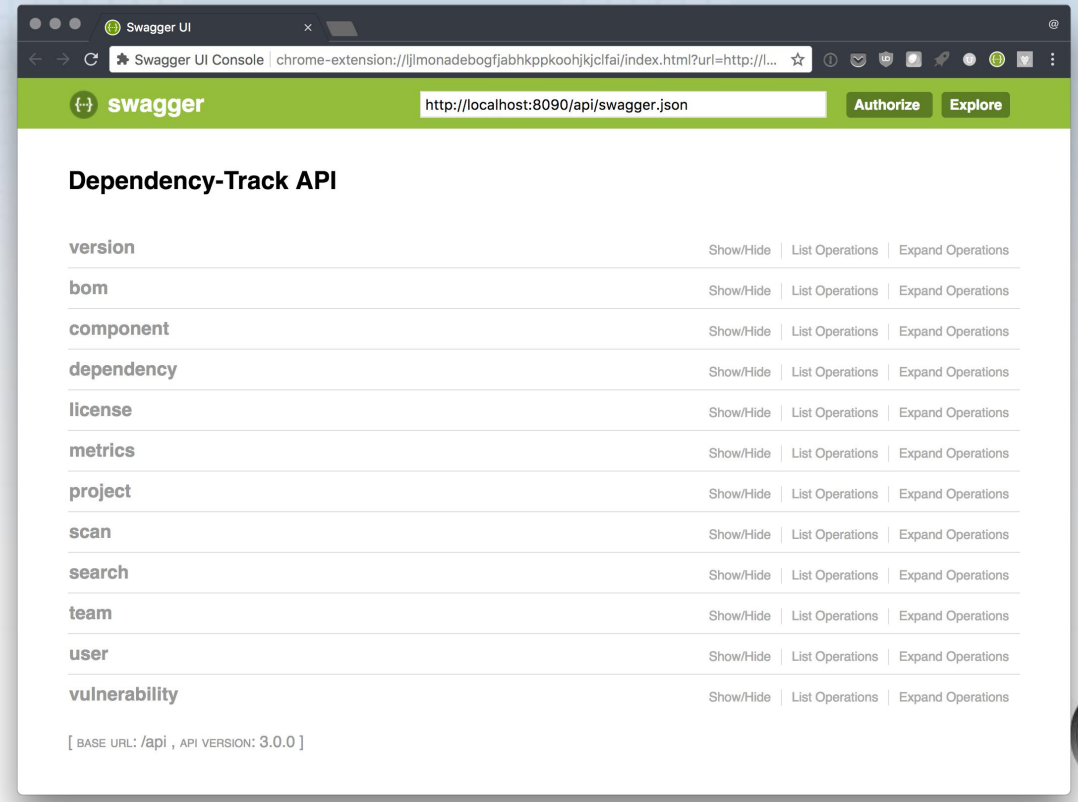


3.6 AD/LDAP集成

- 支持多种类的LDAP服务集成例如
  - Microsoft Active Directory
  - ApacheDS
  - Fedora 389 Directory
  - NetIQ/Novell eDirectory

3.7 丰富的API

- DependencyTrack从设计上遵循API First原则，因此开发团队、安全团队可以调用API以实现自己的定制化需求



OWASP
Open Web Application Security Project

3.8 DependencyTrack和DependencyCheck的区别？

	DependencyCheck	DependencyTrack
一句话描述	依赖安全检测工具	软件安全供应链管理平台
目标用户	开发团队	安全团队 / 开发团队
漏洞数据库	只有NVD漏洞数据库	集成多种漏洞数据库，NVD是其中之一，还有OSSIndex、NPM等
持续集成	支持，但需要一定程度的自定义配置以提升扫描效率	支持，且配置简单
最佳使用阶段	开发过程中使用	开发过程中，以及安全运营阶段皆可使用
License安全检测	不支持	支持
趋势追踪	不支持	支持
漏洞提醒	不支持	支持
依赖分析、排查	仅支持单个团队做自我分析、排查	可以方便的集中分析、排查所有团队的依赖

4.OWASP DEPENDENCYTRACK实践经验分享

OWASP DependencyTrack实践经验分享

- 用DependencyTrack检查依赖安全问题
- 配置消息通知
- 实用小提示
- 推动开发团队思维模式转变

4.1 用DependencyTrack检查依赖安全问题

- 基本步骤：
  1. 生成SBOM
  2. 提交SBOM到DependencyTrack
  3. 在DependencyTrack中查看扫描结果

步骤1：生成SBOM

DependencyTrack支持并推荐使用CycloneDX BOMs

```
buildscript {
  repositories {
    mavenCentral()
    maven {
      url "https://jitpack.io"
    }
  }
  dependencies {
    classpath "com.cyclonedx:cyclonedx-gradle-plugin:1.2.0"
  }
}

apply plugin: "org.cyclonedx.bom"
```

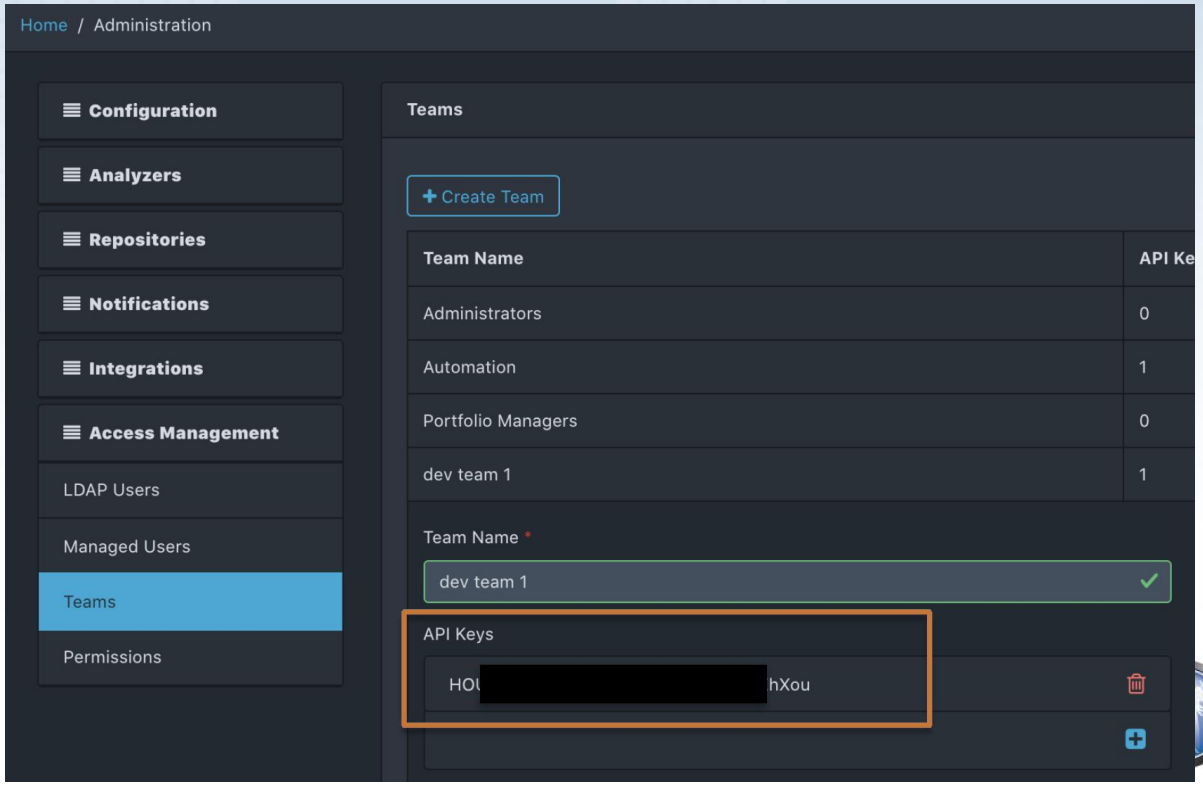
以Gradle构建工具为例，引入CycloneDX插件来生成SBOM

注意，该插件需自定义其所在Maven仓库地址，否则无法安装

步骤2： 提交SBOM到DependencyTrack

推荐使用Jenkins Plugin，配置简单，示例步骤如下

1.在DependencyTrack中，为开发团队创建账号并生成API Key

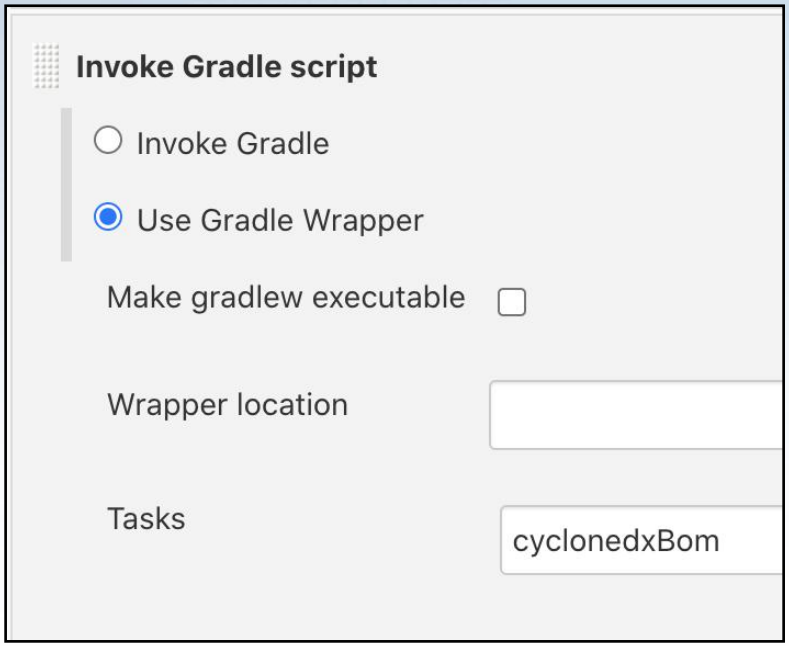


步骤2： 提交SBOM到DependencyTrack

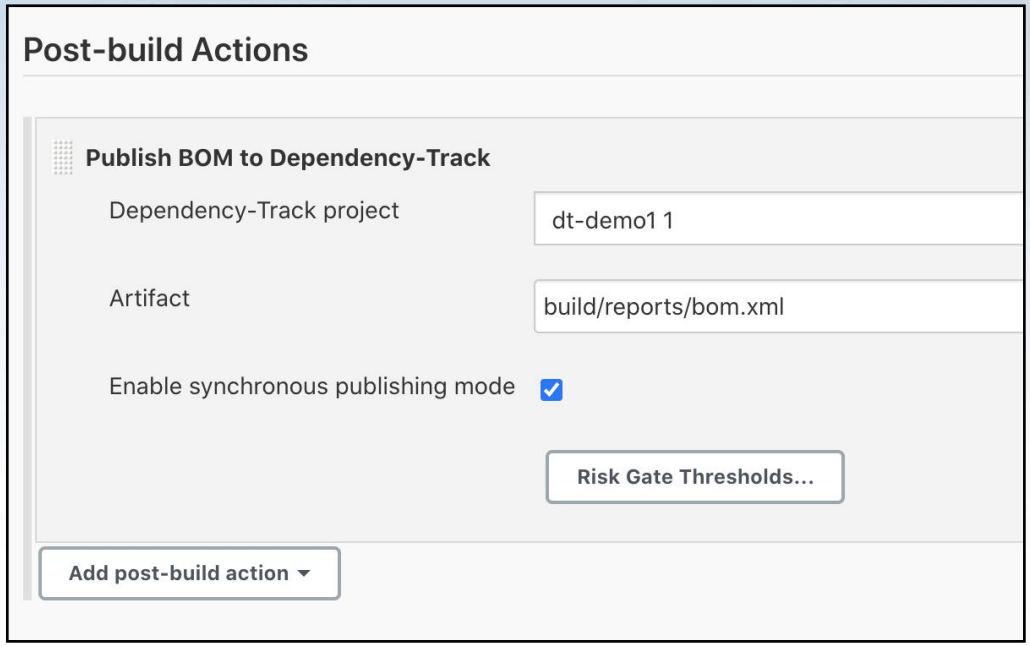
2.作为开发团队，配置Jenkins, 设置DependencyTrack的服务器地址及API Key

步骤2： 提交SBOM到DependencyTrack

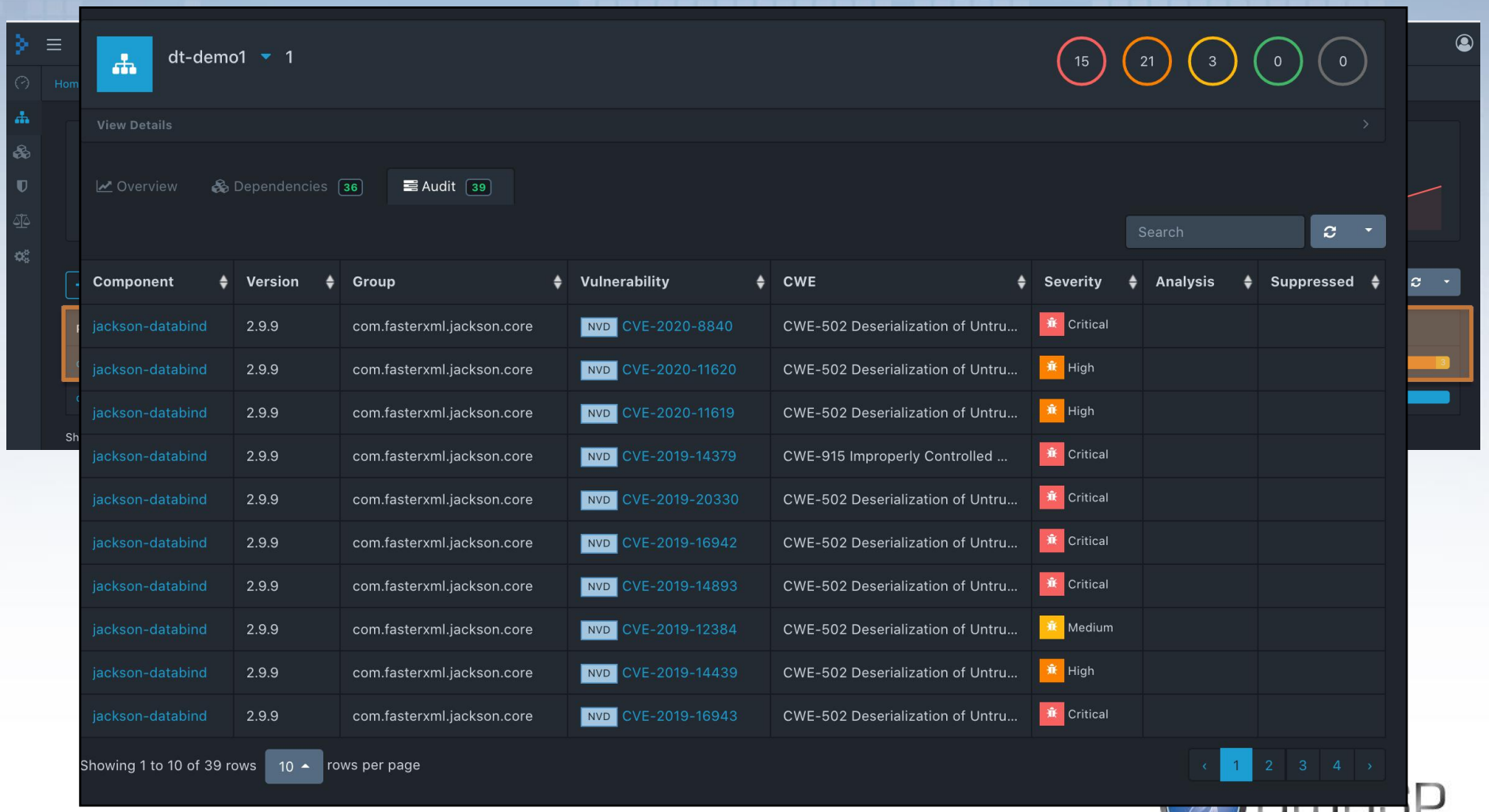
3.配置Job运行cyclonedxBom命令



4.配置Job，设置要上传的bom.xml文件的路径

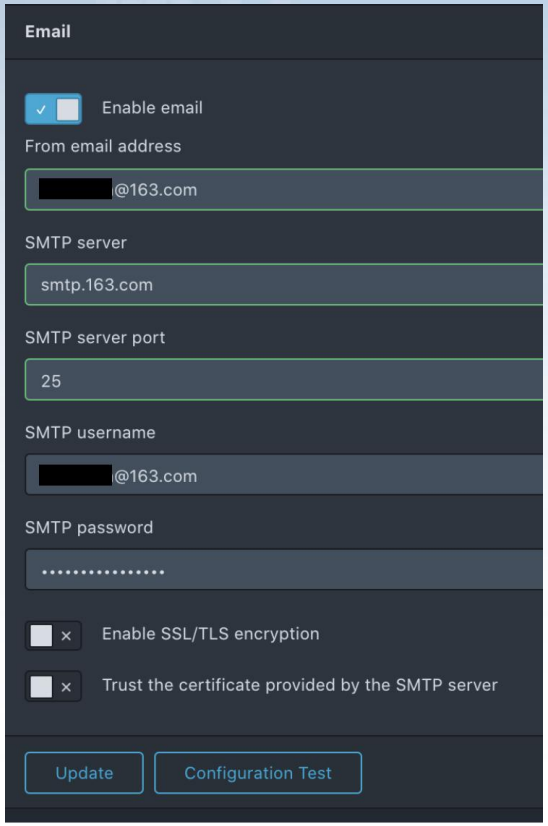


步骤3：在DependencyTrack中查看扫描结果

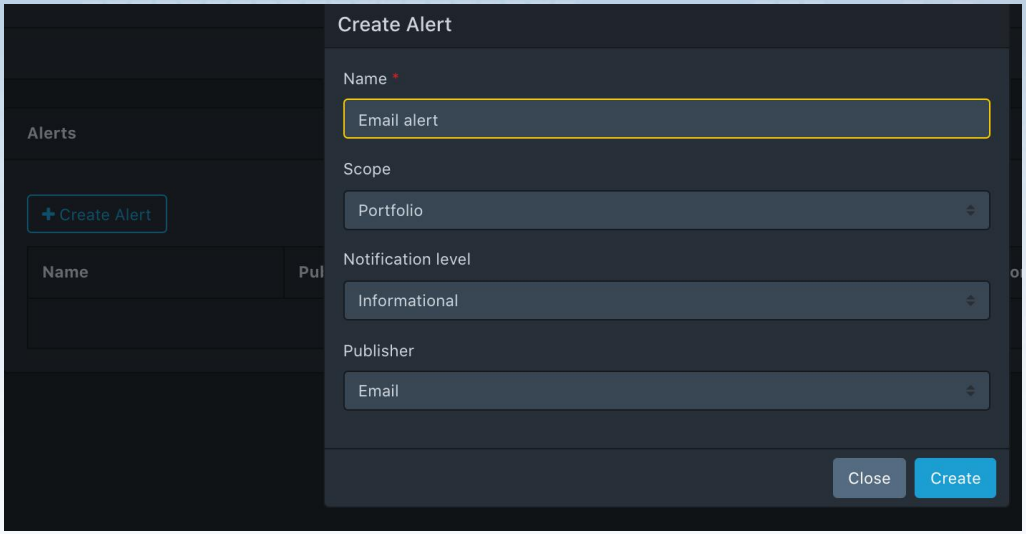


4.2 配置消息通知（邮件通知为例）

1.配置SMTP服务



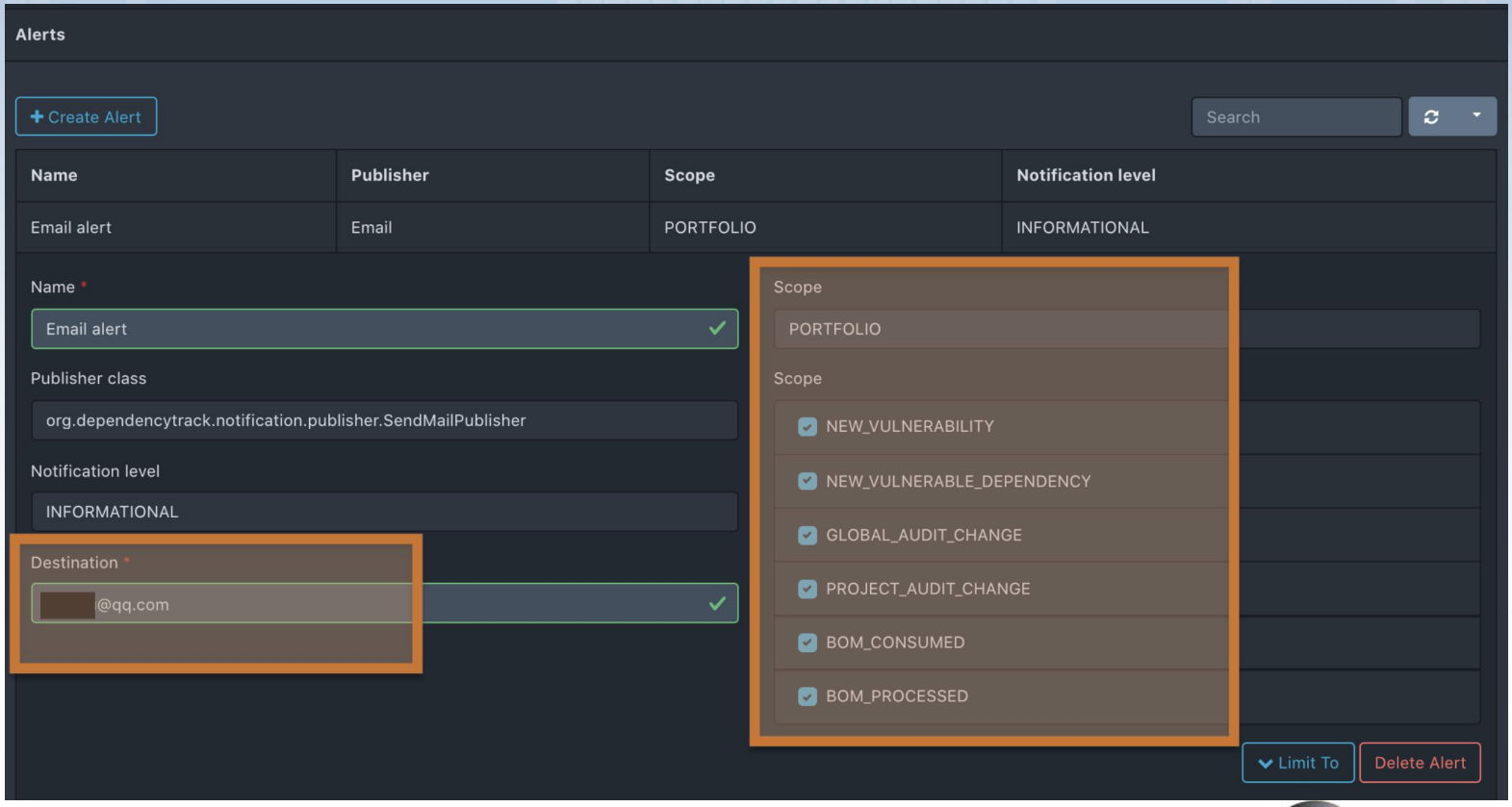
2.在Notifications > Alerts 菜单下新建一个Alert



OWASP
Open Web Application Security Project

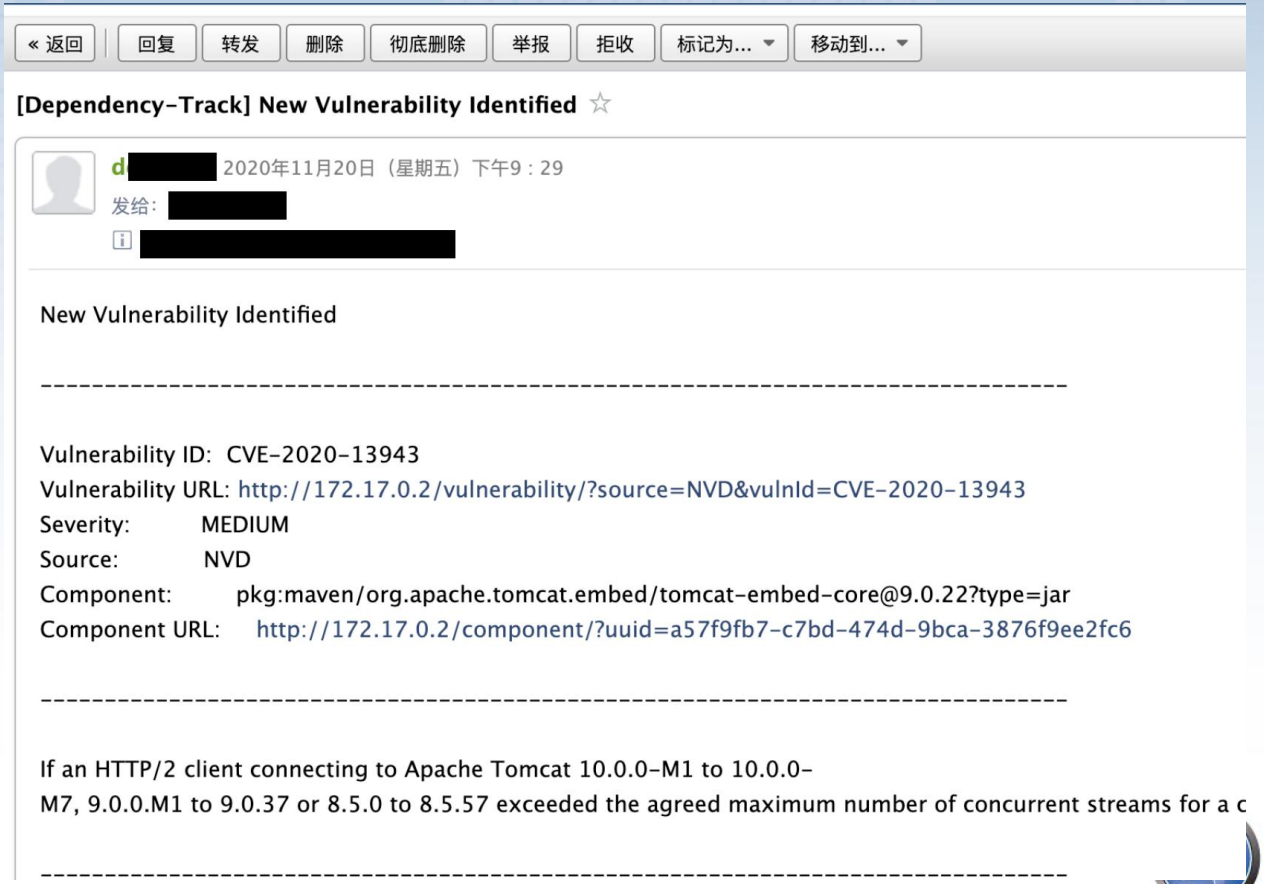
4.2 配置消息通知（邮件通知为例）

3.配置Alert，填入需要接受邮件提醒的邮箱地址，并选择相应的Scope



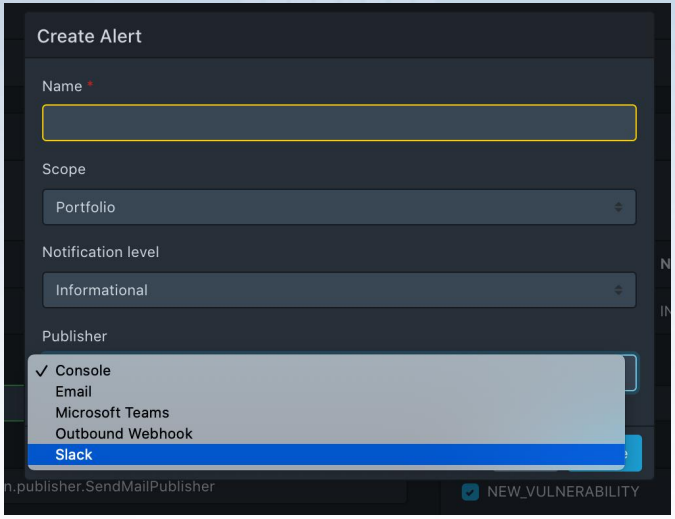
4.2 配置消息通知（邮件通知为例）

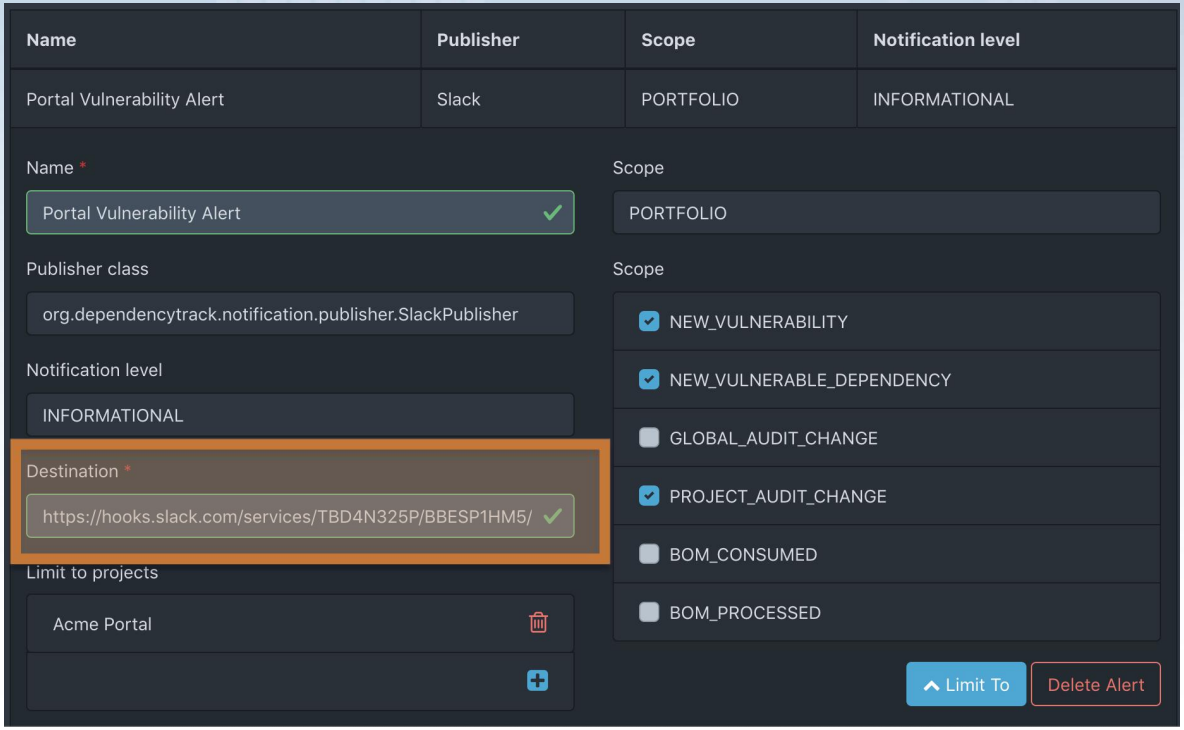
只要DependencyTrack发现有安全问题，则刚才配置的邮箱就能接受到提醒



4.2 配置消息通知（其他类型通知）

在创建Alert时选择对应类型，随后在Destination中填入对应信息即可





4.3 实用小提示

- 开启Sonatype OSS Index Analyzer以提高检测准确性
- DependencyCheck 报告分析支持功能已经在3.7版本里彻底移除，不要再费力寻找上传DependencyCheck报告的地方了
- 如果你不用Jenkins，那么很可能会遇到上传SBOM的API接口行为和API文档描述不符的情况

4.4 推动开发团队思维转变

常见误区1：	常见误区2：
依赖很少变化，不用频繁检测和升级	上线前做一次扫描就够了
但其实：	**但其实：**
虽然依赖很少变化，但依赖的漏洞随时可能出现，因此也需要在第一时间得到告警信息并进行处理	技术债还起来会很痛苦，尽可能保持依赖总是处于最新版本，反而维护成本更低

5.总结

总结

- 软件供应链安全问题
  - 含有安全漏洞
  - 软件协议安全
- 流行的开源依赖安全检查工具
  - NPM Audit
  - OWASP DependencyCheck & DependencyTrack
  - Sonatype OSS Index 工具集合
  - Github Dependabot
- OWASP DependencyTrack特点解析
- OWASP DependencyTrack实践经验分享

分享到此结束
谢 谢

扫描二维码，关注网安加学院查看回放

OWASP
Open Web Application Security Project