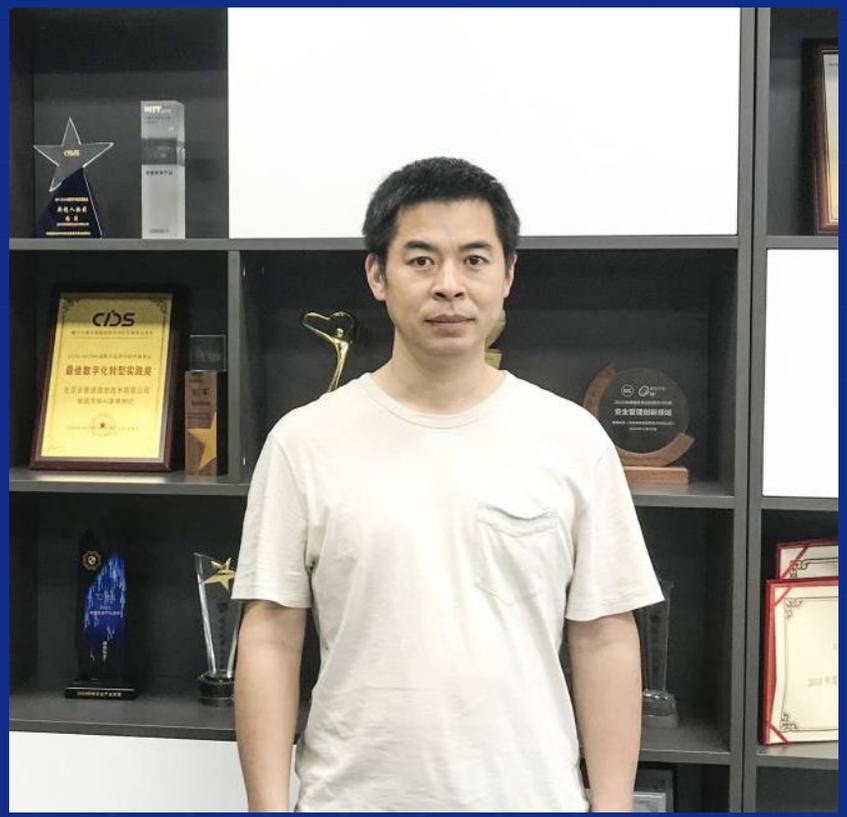
云原生产业大会 2021 CLOUD NATIVE INDUSTRY CONFERENCE
2021/05/26 × 中国 / 北京
原生蓄力 云领未来

开源软件的现状与治理

主讲人 | 霍光

CAICT 中国信通院

悬镜安全



主讲人 | 霍光

悬镜安全

华北区

技术负责人

目录

CONTENTS

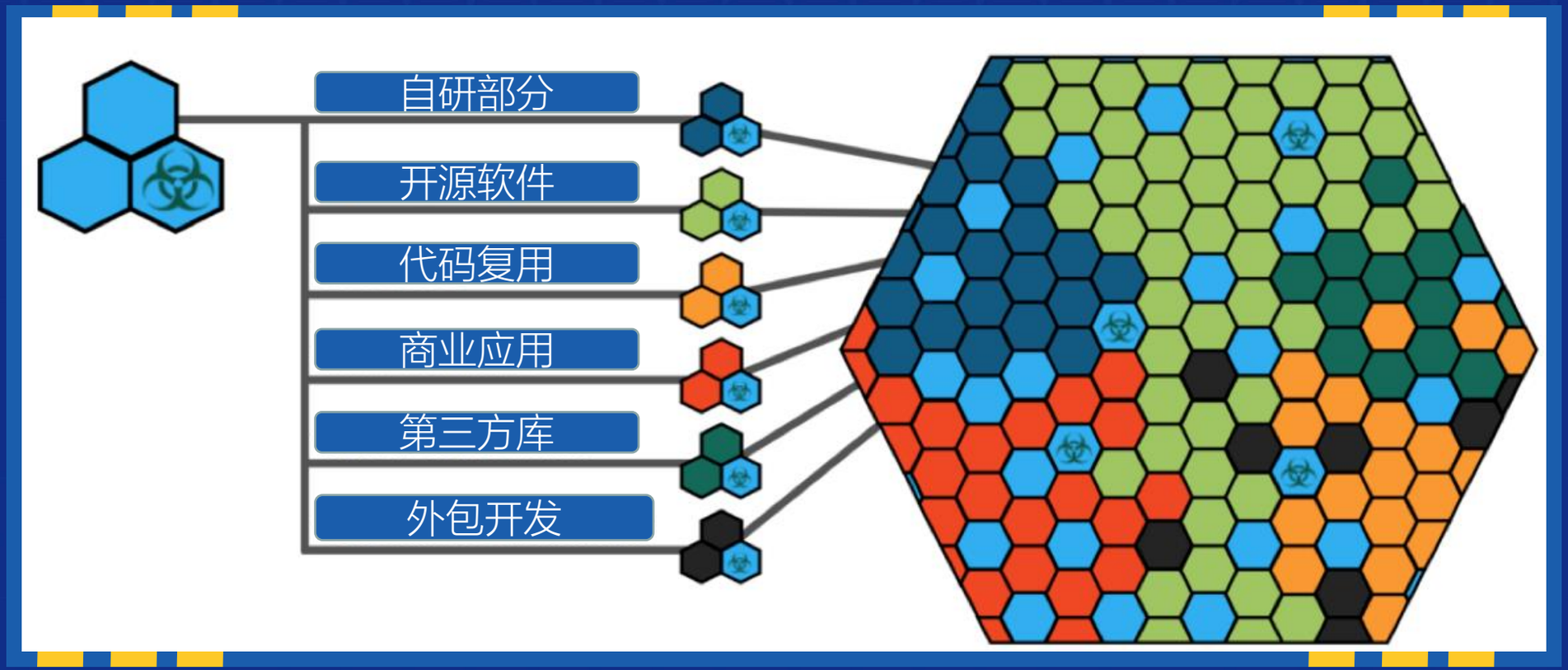
PART 01

开源软件的现状

当前软件编程的特点

- 组装式而不是完全自己编写
- 大量使用开源组件
- 低代码编程

开源软件的使用现状

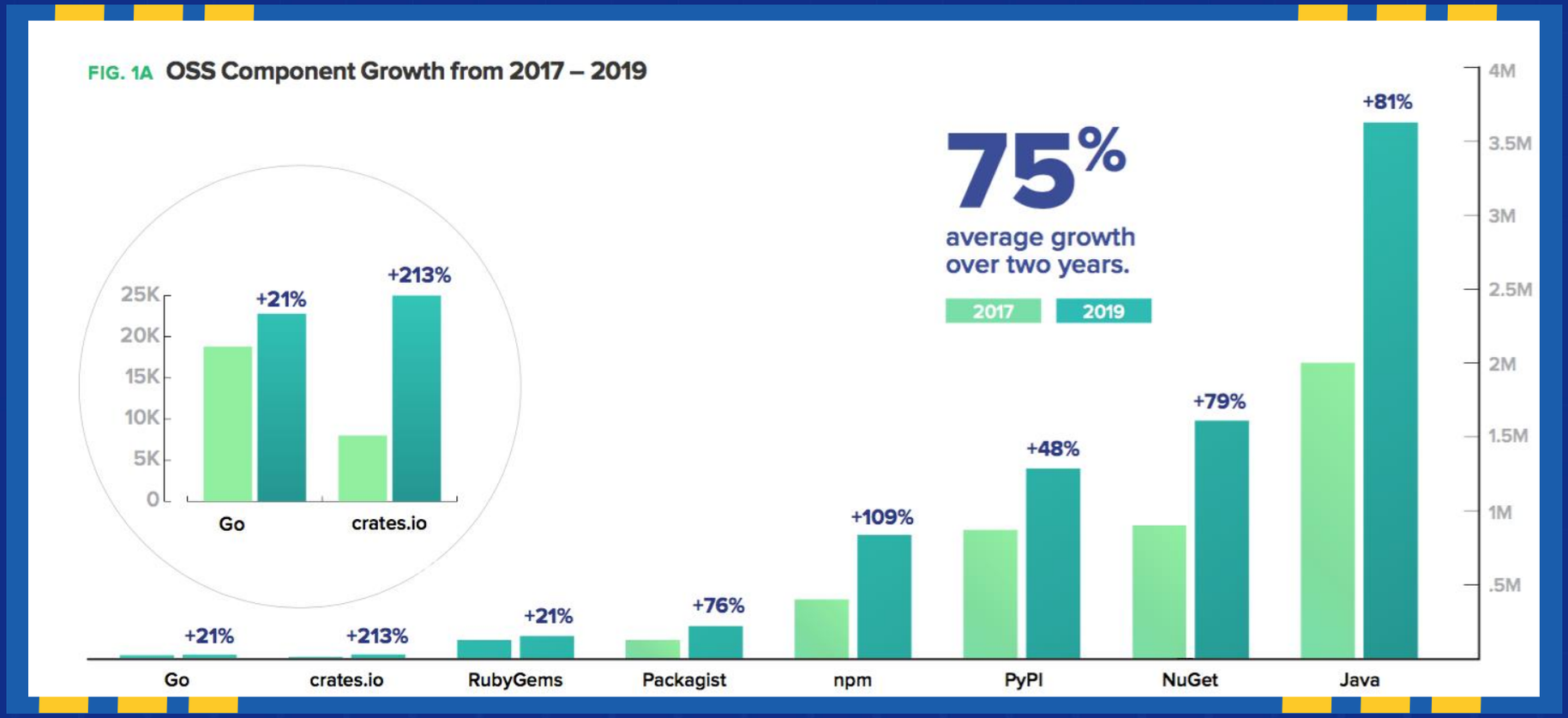


开源软件的定义

开源软件明确定义由1998 年 OSI 给出, 包括十大特性,即自由再发布、源代码公开、允许派生作品、作者源代码完整性、不能歧视任何个人或团体、不能歧视任何领域、许可证的发布、许可证不能只针对某个产品、许可证不能约束其他软件、许可证必须独立于技术。

开源软件的数量

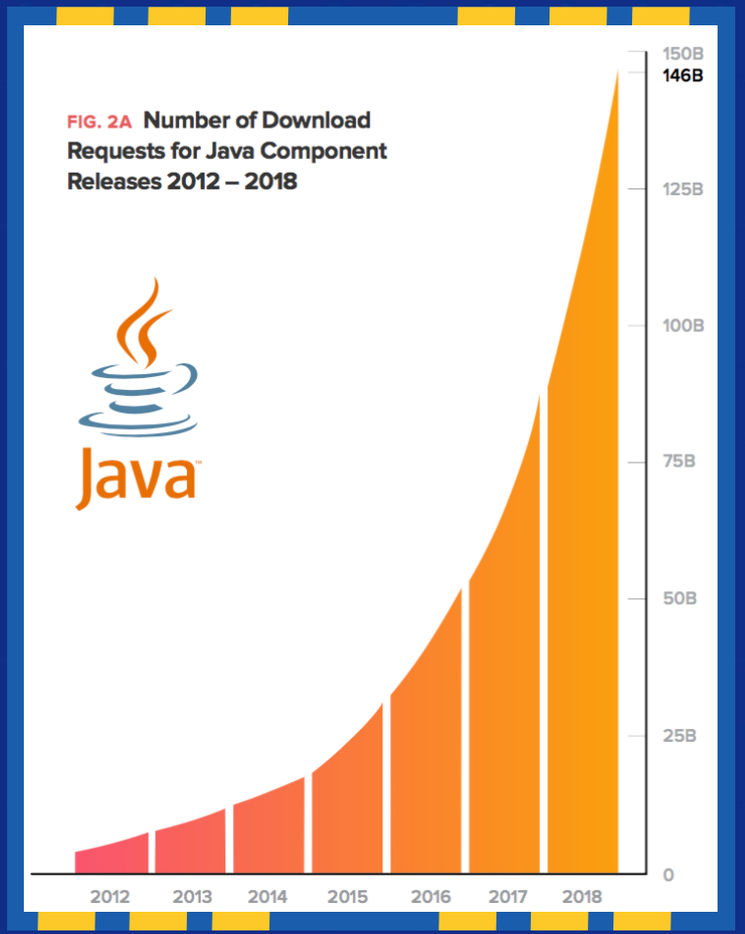
- 开源软件数量巨大
- 3.7 million JAVA 组件
- 800,000 Javascript 包（npm）
- 1.2 million Python
- 1.6 million .NET
- 2.2 million 容器



开源软件的下载数量

自动化的软件开发
极大的刺激了开源软件的下载

- 2018年，java软件的下载量为146 billion
- 2018年，JavaScript 下载量，每周10billion
- 2019年， .NET下载量16 billion



PART 02

开源软件的风险

开源软件的安全态势

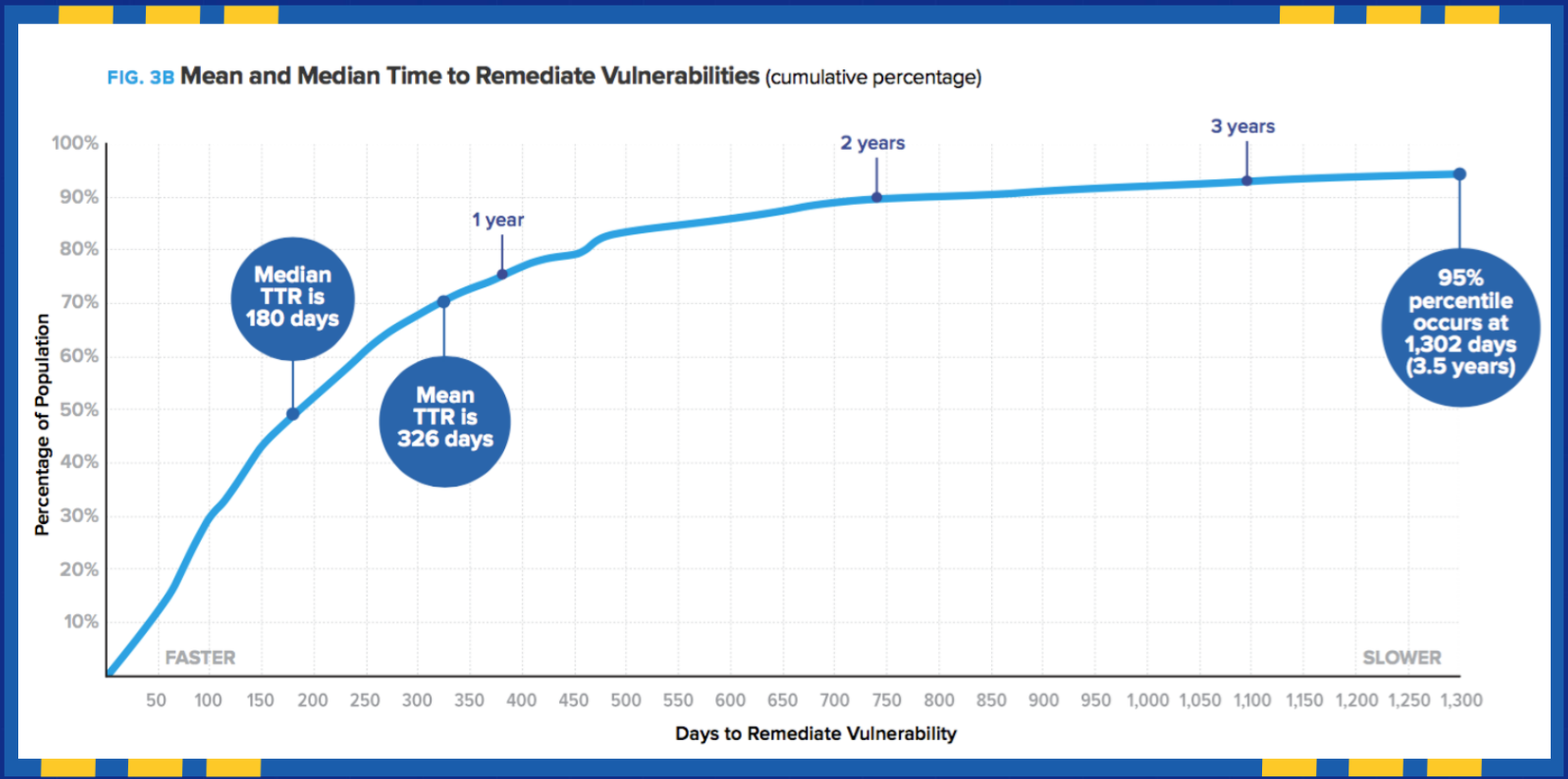
开源软件的特点，所有权和使用权分离，而且，所有权拥有者并不对使用者出现的风险事故直接负责，没有任何"维保"服务。

黑客们如今意识到，**利用开源软件入侵组织机构变得愈发容易。**而这种趋势很可能在2021年进一步加速。目前，**几乎每周都会曝出开源程序包存在恶意代码的消息。**因此，组织机构必须了解他们正在使用且亟需保护的开源组件，并尽可能利用现有解决方案删除易受攻击的程序包（包括开发者意外遗留下漏洞代码的程序包）。

开源软件的漏洞修复时间

- 47%的组件在发布新版本后，存在漏洞。
- 中等修复时间180天，平均修复时间326天
- Top 5%修复时间21 天

只有25%的组织把漏洞通知给用户，10%的漏洞被汇报给CVE。-OWASP 组件依赖检测 创始人



有漏洞组件的下载情况

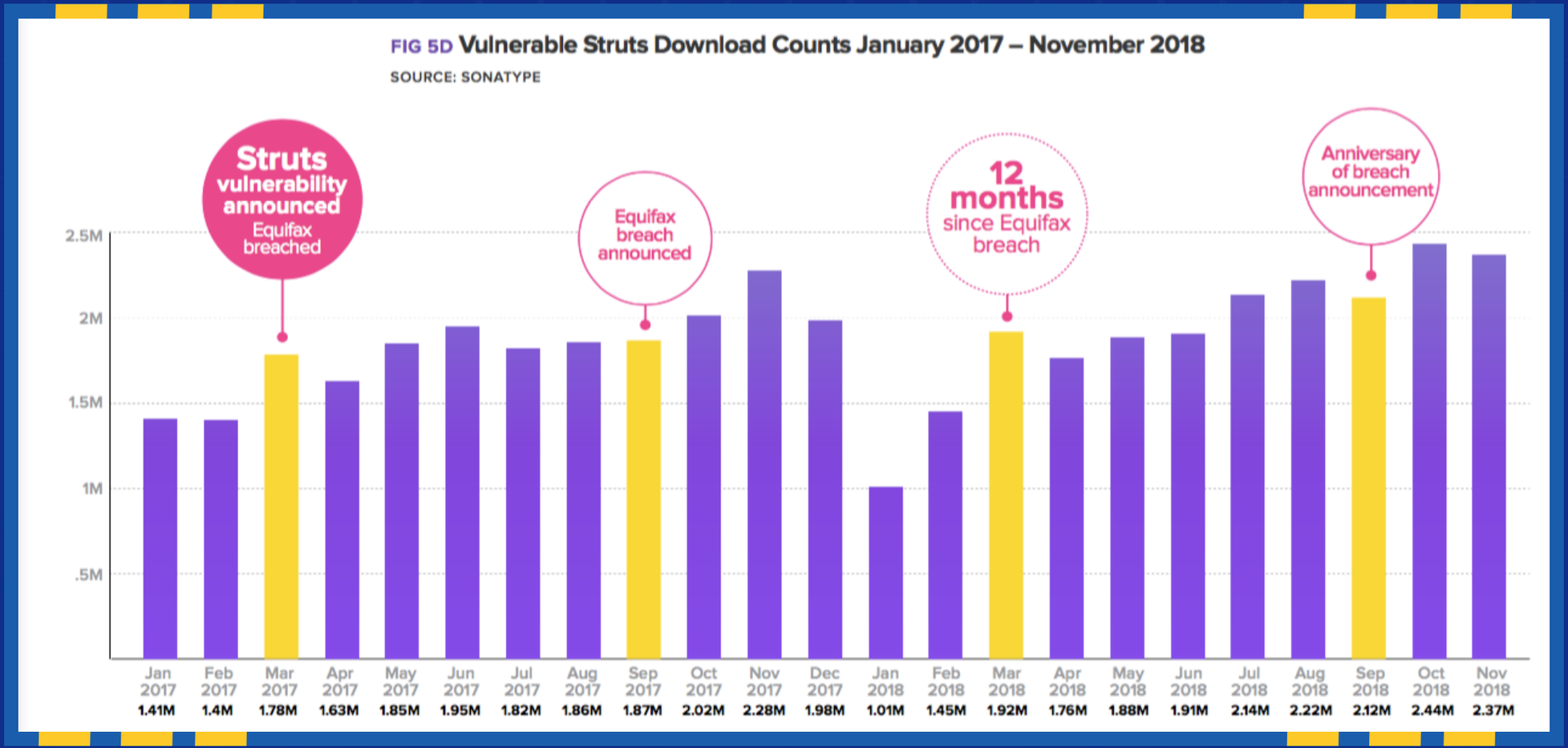


FIG 5D Vulnerable Struts Download Counts January 2017 – November 2018
SOURCE: SONATYPE
Struts vulnerability announced Equifax breached
Equifax breach announced
12 months since Equifax breach
Anniversary of breach announcement
2.5M
2M
1.5M
1M
.5M
Jan 2017 1.41M
Feb 2017 1.4M
Mar 2017 1.78M
Apr 2017 1.63M
May 2017 1.85M
Jun 2017 1.95M
Jul 2017 1.82M
Aug 2017 1.86M
Sep 2017 1.87M
Oct 2017 2.02M
Nov 2017 2.28M
Dec 2017 1.98M
Jan 2018 1.01M
Feb 2018 1.45M
Mar 2018 1.92M
Apr 2018 1.76M
May 2018 1.88M
Jun 2018 1.91M
Jul 2018 2.14M
Aug 2018 2.22M
Sep 2018 2.12M
Oct 2018 2.44M
Nov 2018 2.37M

开源软件的风险-漏洞

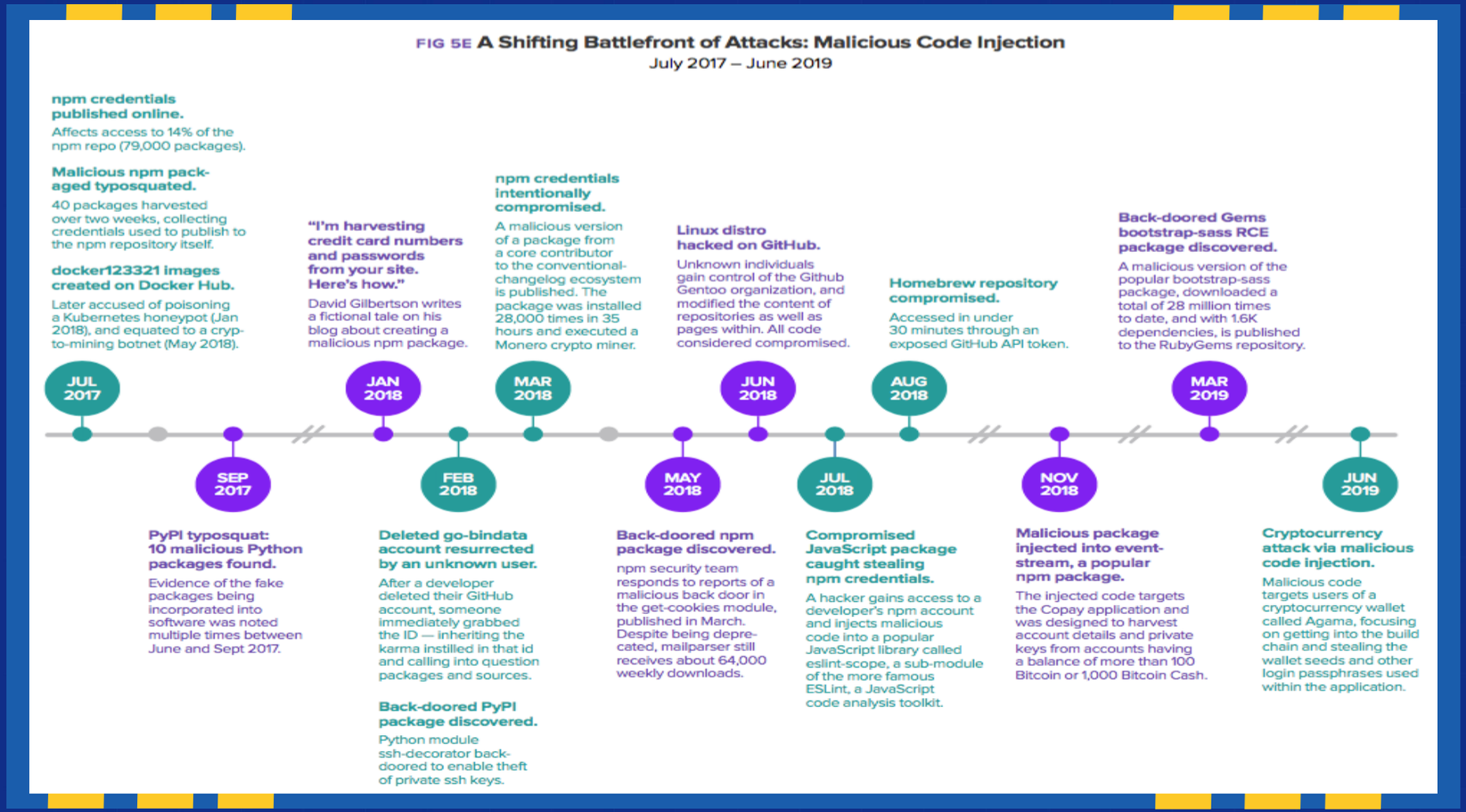


FIG 5E A Shifting Battlefront of Attacks: Malicious Code Injection
July 2017 – June 2019
npm credentials published online.
Affects access to 14% of the npm repo (79,000 packages).
Malicious npm packaged typosquated.
40 packages harvested over two weeks, collecting credentials used to publish to the npm repository itself.
docker123321 images created on Docker Hub.
Later accused of poisoning a Kubernetes honeypot (Jan 2018), and equated to a cryp-to-mining botnet (May 2018).
"I'm harvesting credit card numbers and passwords from your site. Here's how."
David Gilbertson writes a fictional tale on his blog about creating a malicious npm package.
npm credentials intentionally compromised.
A malicious version of a package from a core contributor to the conventional-changelog ecosystem is published. The package was installed 28,000 times in 35 hours and executed a Monero crypto miner.
Linux distro hacked on GitHub.
Unknown individuals gain control of the Github Gentoo organization, and modified the content of repositories as well as pages within. All code considered compromised.
Homebrew repository compromised.
Accessed in under 30 minutes through an exposed GitHub API token.
Back-doored Gems bootstrap-sass RCE package discovered.
A malicious version of the popular bootstrap-sass package, downloaded a total of 28 million times to date, and with 1.6K dependencies, is published to the RubyGems repository.
JUL 2017
SEP 2017
JAN 2018
FEB 2018
MAR 2018
MAY 2018
JUN 2018
JUL 2018
AUG 2018
NOV 2018
MAR 2019
JUN 2019
PyPI typosquat: 10 malicious Python packages found.
Evidence of the fake packages being incorporated into software was noted multiple times between June and Sept 2017.
Deleted go-bindata account resurrected by an unknown user.
After a developer deleted their GitHub account, someone immediately grabbed the ID — inheriting the karma instilled in that id and calling into question packages and sources.
Back-doored PyPI package discovered.
Python module ssh-decorator back-doored to enable theft of private ssh keys.
Back-doored npm package discovered.
npm security team responds to reports of a malicious back door in the get-cookies module, published in March. Despite being depre-cated, mailparser still receives about 64,000 weekly downloads.
Compromised JavaScript package caught stealing npm credentials.
A hacker gains access to a developer's npm account and injects malicious code into a popular JavaScript library called eslint-scope, a sub-module of the more famous ESLint, a JavaScript code analysis toolkit.
Malicious package injected into event-stream, a popular npm package.
The injected code targets the Copay application and was designed to harvest account details and private keys from accounts having a balance of more than 100 Bitcoin or 1,000 Bitcoin Cash.
Cryptocurrency attack via malicious code injection.
Malicious code targets users of a cryptocurrency wallet called Agama, focusing on getting into the build chain and stealing the wallet seeds and other login passphrases used within the application.

开源软件的风险-开源许可证

开源许可证风险

尽管开源软件拥有"免费"的优势，但它与其它软件一样都要受到许可证的约束。67%的代码库包含某种形式的开源代码许可证冲突，33%的代码库包含没有可识别许可证的开源组件。

信通院的《开源生态白皮书风》介绍，根据美国专利组织 Unified Patents 公布 的一项研究结果表明,2012 年到 2018 年底在美国地区法院共产生了 约 260 个开源项目/平台的专利诉讼案例

2019年11月6日，数字天堂（北京）网络技术有限公司诉柚子（北京）科技有限公司、柚子（北京）移动技术有限公司侵犯计算机软件著作权纠纷一案终审判决，判令柚子公司停止侵权并赔偿71万元。该案被称为中国第一个涉及GPL协议的诉讼案件。虽然该案在开源社区中有不同的讨论和解读，但进一步说明了开源软件的知识产权风险，是不可忽视的重要风险之一。

开源许可证介绍

大约有200种开源软件许可协议，每个都定义了不同的术语和条件，有自己的限制和许可描述。

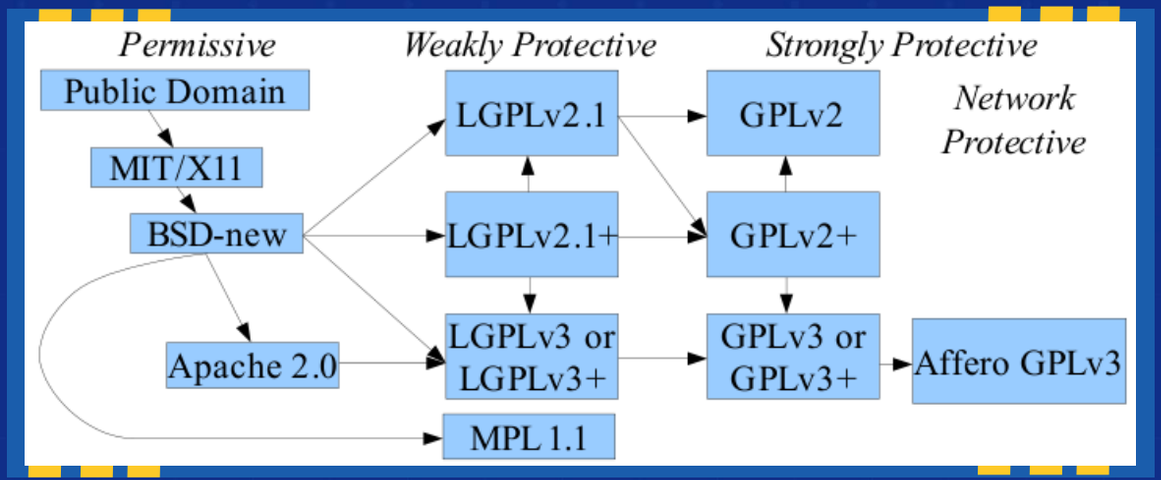
分成两大类型：

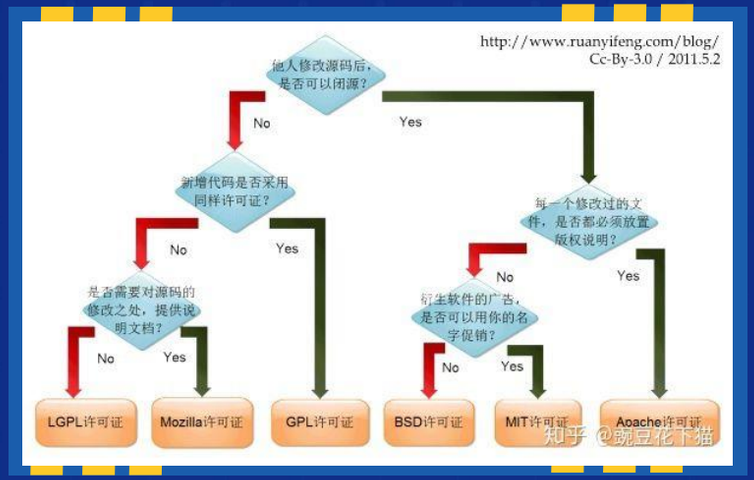
- 宽松许可 permissive

  可以自由使用，修改，分发软件，最小限制。但也有一些细节的条款区别，如专利，版权，声明条款等。

- 著佐权 copyleft

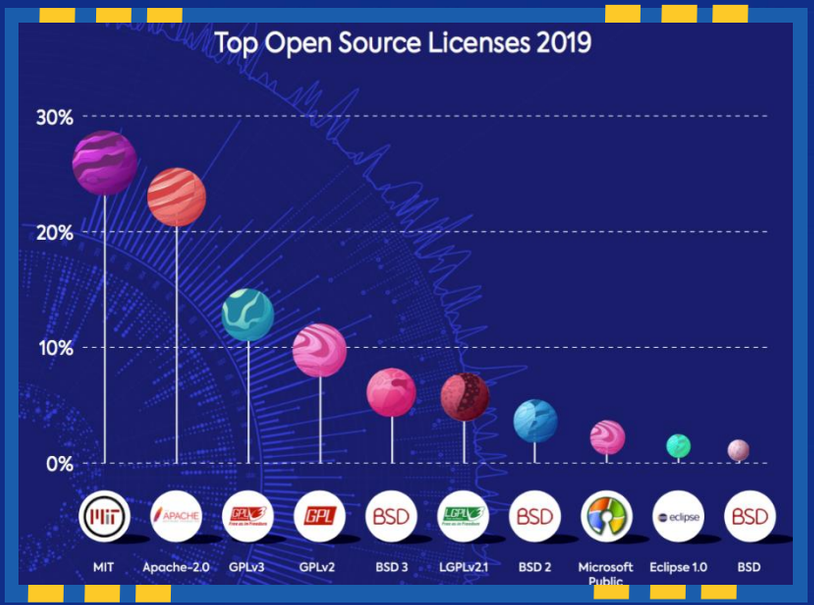
  允许用户在条件允许的情况下，使用，修改，分享软件。并对其他人开放。协议彼此不兼容，除非有明确兼容的说明。



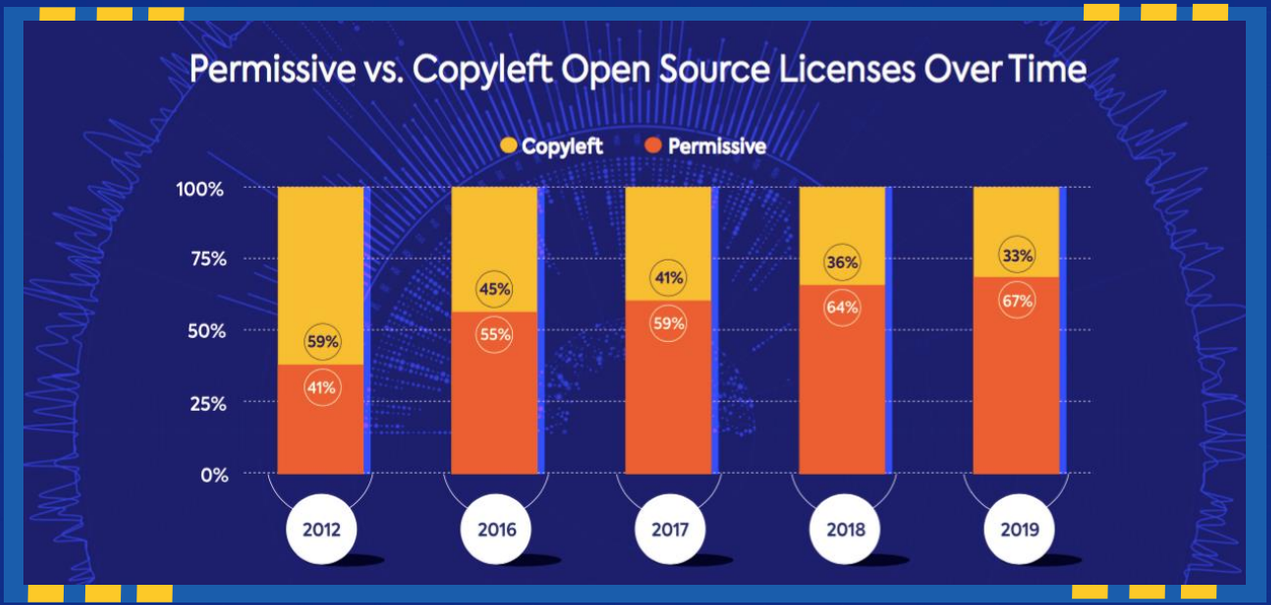


开源许可证的发展趋势

- 排名前十位的开源许可证



- 宽松和著佐权许可证变化趋势



开源软件的风险-软件组成清单SBOM

软件组成清单（SBOM）能够跟踪开源组件的使用情况，能够很快识别出有漏洞的组件，这样在出现问题的时候能够快速修复。最小化软件供应链的风险。

也可对当前的开源软件资产做一个梳理。

开源软件的风险总结

- 合规性风险
  - 开源许可证违规导致的法律诉讼
  - 风险：产品召回，自由代码被迫开源等
- 安全性风险
  - 引入开源软件的同时，被动引入安全漏洞
  - 多数开源许可证没有承诺对安全性风险负责
  - 风险：安全漏洞爆发造成巨大损失
- 企业内部的代码检测需求
  - 项目开源之前，查找自有代码中的开源风险
  - 针对闭源商业代码，也要规避合规性风险
- 政府部分监管政策造成的开源治理需求
  - 国家项目对自主知识产权的要求
  - 国家对某些重要领域的风险防控要求
- 国际合作中甲方提出的代码扫描需求
  - 国内产品出口到欧美，甲方需要检测报告免责
- 企业兼并过程中的代码审计需求
  - 邀请第三方进行审计，对收购标底进行软件资产评估

用户使用开源组件的困境

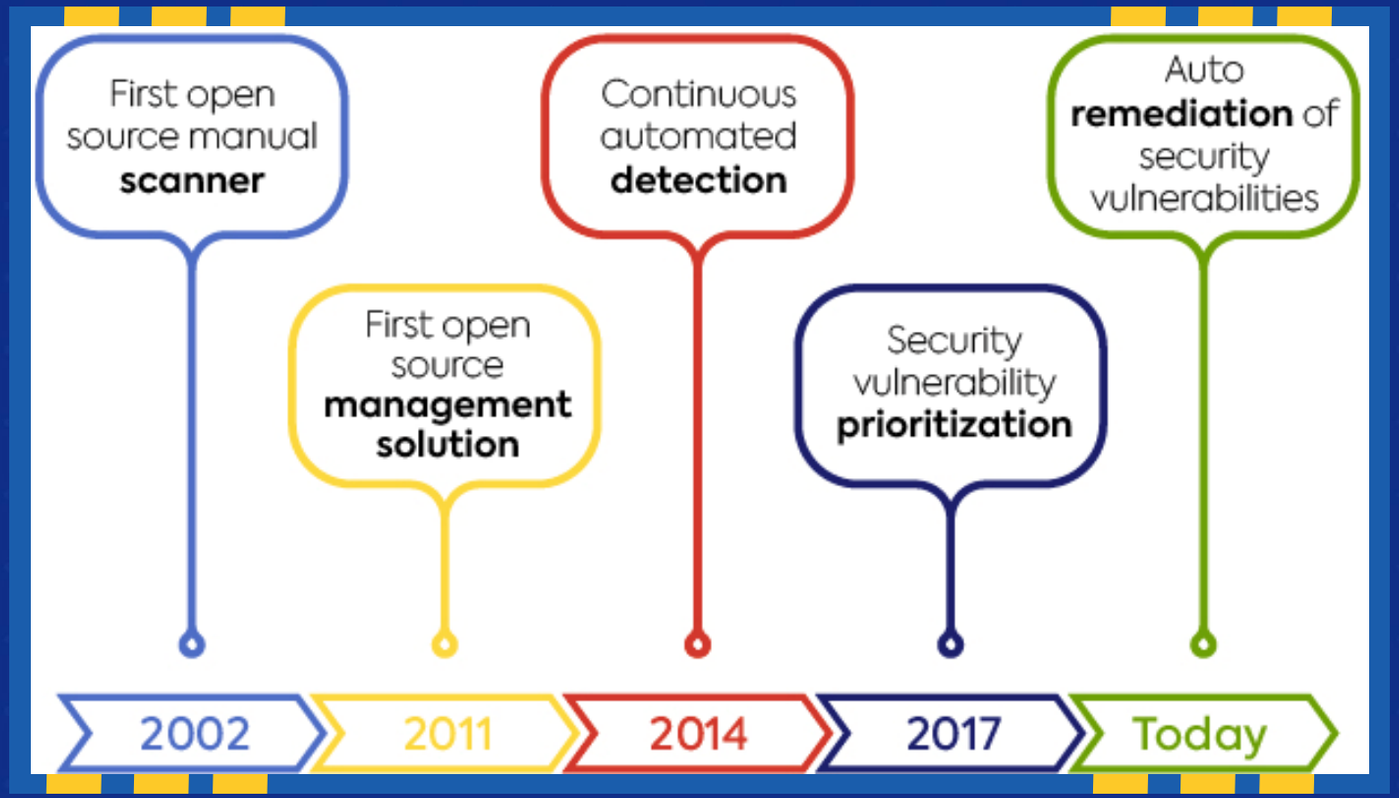
- 用户不清楚系统中使用了多少开源软件，开源组件
- 目前使用的开源软件中，存在多少已知安全漏洞和知识产权风险
- 企业在开源软件或组件出现漏洞时，无法快速定位到漏洞组件的影响范围，并及时止损，禁止漏洞组件下载
- 针对开源软件安全漏洞，企业缺少评估和修复能力

SCA 定义

软件成分分析(SCA，Software Composition Analysis)专门用于分析开发人员使用的各种源码、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，并识别已知的安全漏洞或者潜在的许可证授权问题，把这些风险排查在应用系统投产之前，也适用于应用系统运行中的诊断分析。

软件成分分析软件的发展历史

2002年，第一个手动的开源组件扫描器就诞生了，但由于误报率和人工操作，不符合敏捷开发的需求。早期主要给一些知识产权的律师事务所使用。后来随着软件供应链攻击的增加，逐渐认识到识别开源组件的重要性。



开源软件的分析原理

（1） 工具扫描开源软件源代码，通过进行源代码片段式比对来识别组件并识别许可证类型。此类工具扫描后得出的组件信息需要人工确认。

（2）工具不扫描开源软件源代码，对文件级别提取哈希值，进行文件级哈希值比对，若全部文件哈希值全部匹配成功则开源组件被识别出来。此类工具扫描后得出的组件信息是自动确认的。

（3）工具不扫描开源软件源代码，通过扫描包配置文件读取信息，进行组件识别从而识别组件并识别许可证类型。此类工具扫描后得出的组件信息是自动确认的。

（4）工具不扫描开源软件源代码，对开源项目的文件目录和结构进行解析，分析开源组件路径和开源组件依赖。此类工具扫描后得出的组件信息是自动确认的。

（5）工具不扫描开源软件源代码，通过编译开源项目并对编译后的开源项目进行依赖分析，这种方式可以识别用在开源项目中的开源组件信息。此类工具扫描后得出的组件信息是自动确认的。

开源软件的治理要求-美国

随着软件在生产和生活中的重要性日益提高，软件漏洞导致后果越来越严重，对软件本身的安全也日渐纳入政府和主管机构的日程中，除了要检测软件产品的安全性，也要检测软件编程中所使用组件的安全。

- 新的PCI 安全软件开发标准

2019年1月，支付卡行业安全标准委员会，引入了重要的软件开发安全标准

- PCI 安全软件标准
- PCI安全软件生命周期标准

新标准要求组织管理他们使用的开源软件，强调，用于支付系统的任何应用系统，必须在设计之初就考虑安全，特别是引入的开源软件，或第三方代码库，组织必须负责保证代码及其供应商：

- 通过SDLC，360度监控开源组件
- 出现漏洞，必须有及时的策略和流程尽快修复

为有效的成规模利用开源组件，PCI标准建议组织生成一个软件的使用清单（SBOM），以便他们能够跟踪嵌入在他们软件应用系统里面的每一个开源组件的来源和过程。

开源软件的治理要求-美国

国家电信和信息管理（NTIA）软件使用清单立法提案

隶属国家商务部的NTIA，正在考虑要求公司列出他们软件来源，保护美国的软件供应链。推进软件组件透明策略，　需要公司 和供应商说清软件的来源和安全情况。这些策略来源于政府机构对软件代码中的恶意代码注入攻击的担心。政府机构和私人公司都在一起合作，制作出软件使用清单的要求。

众议院商业委员会

2018年11月，美国众议院商业委员会发布了信息安全策略报告，报告中详细强调了在现代应用开发过程中，使用开源软件使用清单（SBOM）来最小化供应链风险的重要性。可以提示组织清楚购买软件中的软件漏洞情况，并能够在出现漏洞时及时修复。

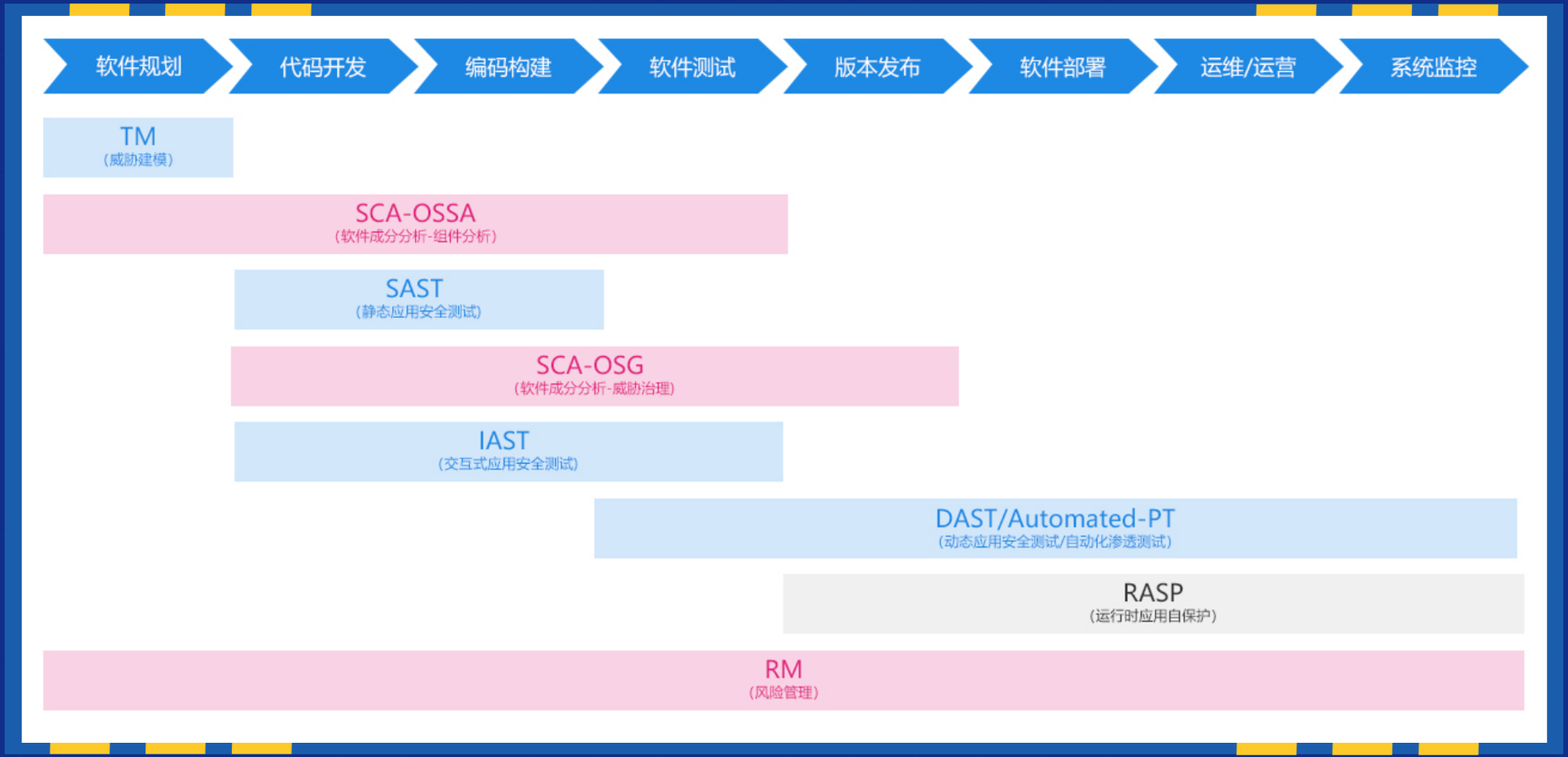
美国食物和药品管理委员会（FDA）

2018年10月，FDA发布医疗设备的信息安全管理指南，要求提供一个商业软件，开源软件，其他未知的，软件和硬件组件清单，保证设备使用者，有效的管理资产，明晰漏洞对资产的潜在影响，保证设备基本功能部署对策。如果一个医疗设备在有已知漏洞的情况下，继续销售，需要付相应的法律责任，包括不能提供适时的报警，对存在漏洞的合适的监控，已知漏洞的解决。

PART 03

SCA产品功能的描述

开源软件的风险-漏洞


软件规划
代码开发
编码构建
软件测试
版本发布
软件部署
运维/运营
系统监控
TM
(威胁建模)
SCA-OSSA
(软件成分分析-组件分析)
SAST
(静态应用安全测试)
SCA-OSG
(软件成分分析-威胁治理)
IAST
(交互式应用安全测试)
DAST/Automated-PT
(动态应用安全测试/自动化渗透测试)
RASP
(运行时应用自保护)
RM
(风险管理)

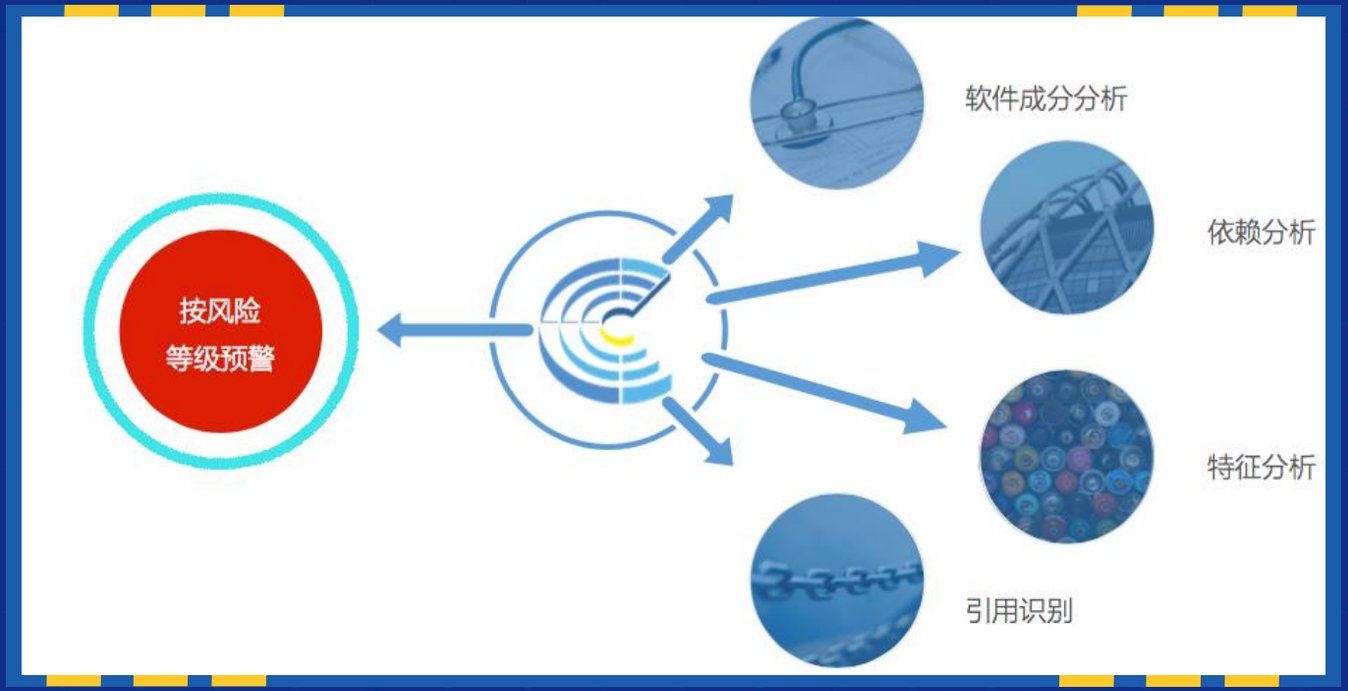
和现有流程无缝集成

在不改变组织现有开发、测试流程的前提下,与源代码管理系统(Git、SVN等)、缺陷管理系统(如Jira、Bugzilla、禅道等)、 持续集成工具(如Jenkins、TFS)无缝对接,将代码第三方库检测融入企业的DevOps研发流程,实现了软件成分分析、成分中的 已知漏洞分析及评估、许可证分析、漏洞范围影响分析等功能,帮助组织快速构建代码安全保障体系。

语言	包管理器	框架		IDE插件
PHP	Maven	Android	JSF	Eclipse Intellij
C#.Net	Gradle	Angular	JQuery	Visual Studio ...
JavaScript	NuGet	ASP.NET	Node.js	代码库
Java	NPM	Apache Commons	React	Git SVN
HTML/HTML5	Bower	Drupal	Sonic	...
XML	Composer	Hibernate	Spring Boot	缺陷/项目管理工具
Python	PIP	JSP	Struts	Jira Bugzilla
...	...	...		...

按风险等级排序

与传统SCA产品相比,源鉴OSS依靠独有的IAST技术,拥有运行时分析能力,通过开源软件成分分析、依赖分析、特征分析、 引用识别等多种技术组合能够精确识别软件中的开源组件信息,以准确识别应用程序是否实际使用了易受攻击的组件,进一步确认 漏洞的真实有效性,使开发人员避免面对数量巨大的误报和无法利用的漏洞,帮助他们区分优先级,将有限的修复精力集中在真正 重要的漏洞上。



可视化呈现

平台支持开发阶段、CI/CD阶段及测试阶段全流程对开源组件的检测,梳理并管控开源组件信息,并从企业、部门、项目及任 务等多角度分析组件的影响范围和依赖关系,通过可视化拓扑图,多维度的展示详细的BOM清单的调用关系,助力使用者快速制 定解决方案。

漏洞

根据每一个新发现的漏洞信息,快速追溯 到漏洞所影响的组件及项目 。

组件

可视化展示组件与组件、任务、项目、部门的依赖关系,清晰的展示组件安全动态 及影响范围。

任务

对每次任务检测出来的组件及漏洞进行预警、管理、依赖关系分析、漏洞处理 。

项目

站在项目的角度,梳理并管控项目内部组件的使用情况及依赖关系,从而确定风险 及影响范围,及时处理风险问题。

部门

站在部门的角度,梳理并管控部门内部组件的使用情况及依赖关系,从而确定风险及影响范围,及时处理风险问题 。

企业

站在企业管理的角度,梳理并管控企业内部组件的使用情况及依赖关系,从而确定风险及影响范围,及时处理风险问题。

漏洞库实时更新

SCA的漏洞信息兼容OWASP TOP10、国家信息安全漏洞库(CNNVD)、国家信息安全漏洞共享平台(CNVD)及CWE 标准,监控众多开源软件漏洞情报来源,通过清洗、匹配、关联等一系列自动化数据分析处理后，让用户及时获取影响其安全的最新开源软件漏洞和许可证风险情报。

自动化修复

支持C、Java、PHP、Python、Go等多种主流开发语言的源代码检测,将源代码缺陷检测和源代码合规检测融入到企业开发测试流程中,识别风险代码片段、风险函数及函数依赖关系,自动化帮助企业修复发现的安全问题，降低软件安全问题的修复成本,提升软件安全质量。

风险代码获取

依靠悬镜启发式爬虫技术实时爬取开源风险代码并及时更新风险代码库

生成风险函数向量

根据风险代码库数据,分析风险函数特征,生成风险函数向量

自动修复

根据风险函数向量,对源码进行函数 及依赖关系分析,确定影响范围及修复方案，自动修复

云原生 产业大会
2021 CLOUD NATIVE INDUSTRY CONFERENCE
2021/05/26 × 中国 / 北京
原生蓄力 云领未来

THANKS!

CAICT 中国信通院