SYNOPSYS®

2022
开源安全和风险分析报告

目录

简介

关于2022年开源安全和风险分析报告与CYRC

欢迎阅读《2022年开源安全和风险分析（OSSRA）报告》。第7版OSSRA提供了我们对商业软件中开源安全、合规性、许可和代码质量风险的当前状态的年度深入研究。Synopsys分享了这些调查结果，以帮助安全、法律、风险和开发团队更好地了解安全和许可证风险状况。本报告中的数据源自Synopsys网络安全研究中心（CyRC），

该中心的任务是发布安全建议和调研报告，以帮助企业更好地开发和使用安全的高质量软件。

今年，CyRC团队共审查了17个行业中超过2,400个商业代码库的匿名审计结果。被审代码库的数量比去年增长了64%，反映了并购（M&A）交易在整个2021年的显著增长。根据摩根士丹利的数据，2021年的并购交易数量创下历史新高，总价值超过4.9万亿美元。[1]审计量的增长还可归因于人们认识到软件是公司知识产权（IP）的关键要素。因此，并购交易中的收购方希望了解他们所收购的软件可能存在哪些风险，特别是与许可、安全和该软件中使用的开源代码质量相关的风险。

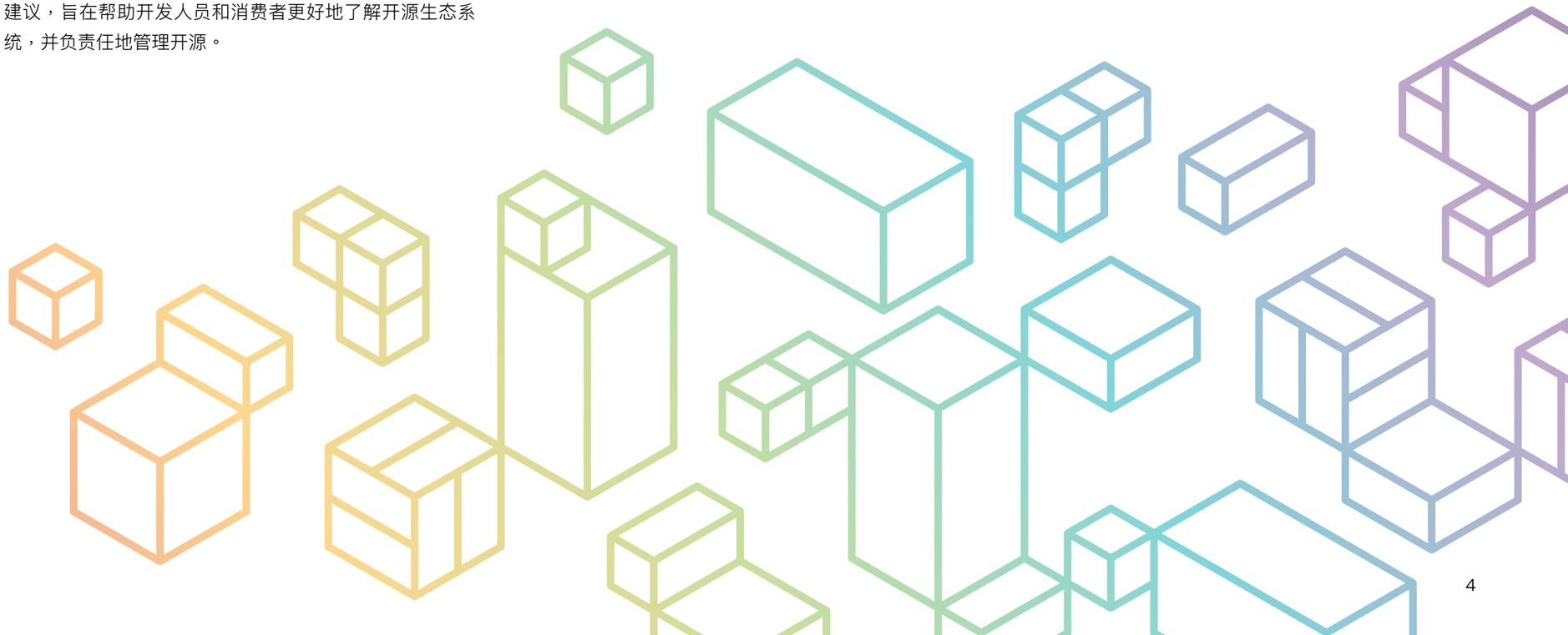
过去20年间，Black Duck®软件组成分析（SCA）解决方案和审计服务被世界各地的开发、安全和法律团队所信赖。我们的SCA解决方案帮助企业有效识别、跟踪开源代码、并跨越多个开发环境自动执行开源策略。

每年，我们的审计服务团队都会为客户审计数千个代码库，主要是为了识别并购交易中的一系列软件风险。Black Duck审计提供全面且最新的软件物料清单（SBOM），涵盖应用程序中的开源代码、第三方代码、Web服务和API。审计服务团队依靠来自 Black Duck KnowledgeBase™ 的数据识别潜在的许可证合规与安全风险。该知识库中存储了超过510万个开源组件的近2亿个版本的信息。这些组件使用了来自超过2.6万个独特来源的数据，且这些数据均由CyRC精心组织和验证。

2021年的审计数据分析由CyRC的Belfast团队负责。除了收集和分析本报告中使用的数据外，该团队还负责Synopsys Black Duck 增强安全解决方案（Black Duck Security Advisories，BDSA），旨在通过这些详细的漏洞通知信息直接向Black Duck的商业客户提供增强的漏洞信息。

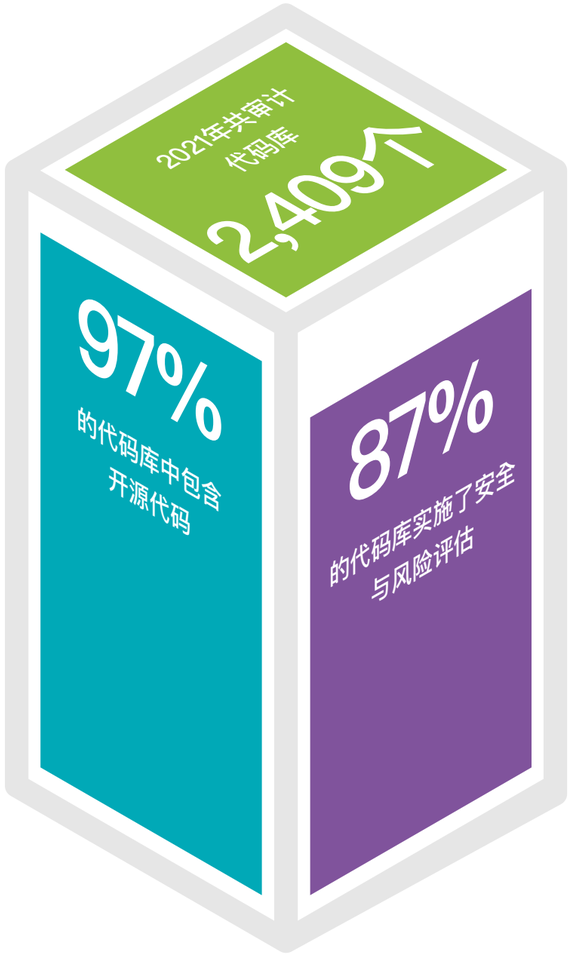
OSSRA 数据都表明，无论您身处哪个行业，为谨慎起见，您都应该假设您所构建和使用的软件中包含开源组件。正如我们的调研结果所强调的那样，开源软件无处不在，您需要对其使用进行妥善管理。开源是我们今天所依赖的每一个应用程序的基础。识别、跟踪和管理开源代码对于有效确保软件安全至关重要。本报告提供了一些关键建议，旨在帮助开发人员和消费者更好地了解开源生态系统，并负责任地管理开源。

开源无处不在，您需要对其使用进行妥善管理



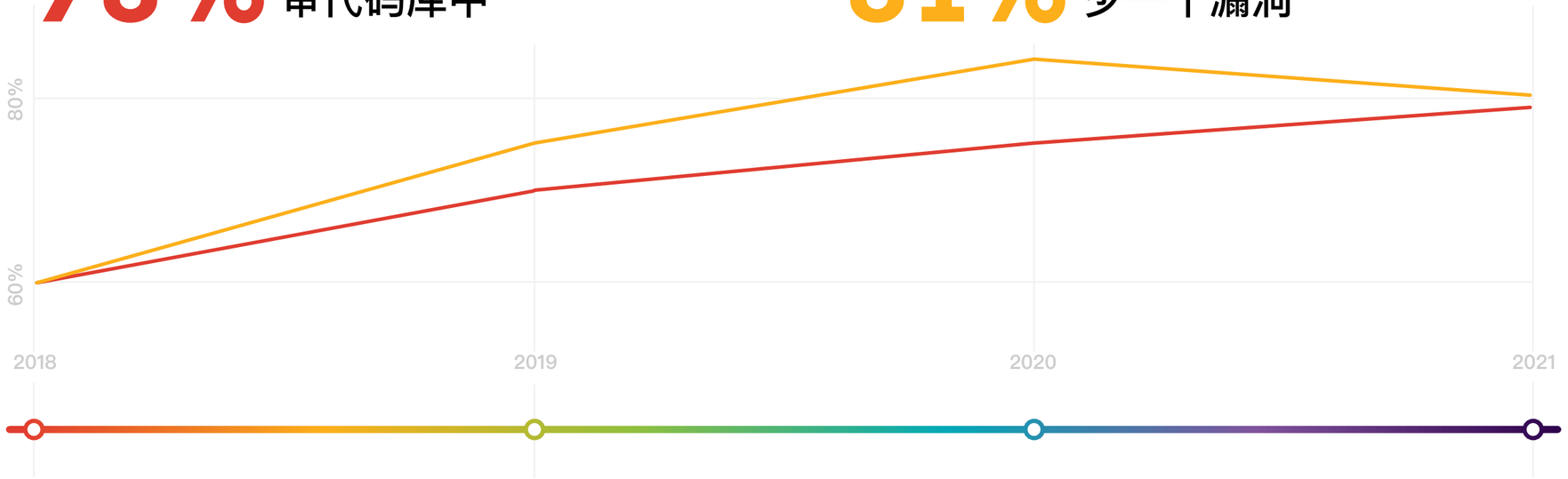
概述

2022回顾



78% 的开源代码存在于被审代码库中

81% 的被审代码库中包含至少一个漏洞

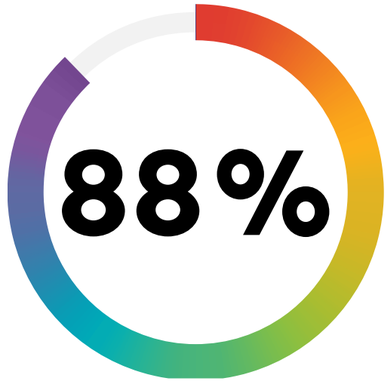


88% 的被审代码库中包含两年内未更新的组件

85% 的被审代码库中包含至少已过时四年的开源代码

的被审代码库中存在许可证冲突

的被审代码库中包含没有许可证或使用定制许可证的开源代码



的被审代码库使用了不是最新版本的组件

概述

术语

代码库

组成应用程序或服务的代码和相关的库。

Black Duck增强安全解决方案（BDSA）

由CyRC安全研究小组识别的开源代码的漏洞。BDSA为Synopsys客户提供关于开源代码漏洞的早期和/或补充通知以及开源软件升级/补丁指导方案。

软件组件

开发人员可以添加到其软件中预先写好的代码。软件组件可以是日历函数等实用程序，也可以是支持整个应用程序的综合软件框架。

依赖项

当某个软件组件被其他软件使用时，也就是说当这些软件依赖于该组件时，该软件组件就变成了依赖项。任何给定的应用程序或服务都可能有许多依赖项，而这些依赖项本身也可能依赖于其他组件。

开源许可证

说明在软件中使用开源组件（或开源组件的代码片段）时的最终用户义务的一组条款和条件，包括如何使用和重新分发该库。开源许可证基本上分为以下两类：

宽松型许可证（Permissive License）

宽松型许可证对开源代码的使用基本不设任何限制。一般来说，此类许可证的主要要求是原始开发者不承担任何责任，并将原始代码的归属权提供给原始开发者。

著佐权许可证（Copyleft license）

此类许可证通常涵盖互惠义务，规定如果衍生作品基于的是在著作权许可证下提供的原始代码，则其发布条件和条款本须与原始代码相同，并且有改动的源代码必须可用或按要求提供。商业实体应对在其软件中使用著佐权许可证开源代码十分谨慎，因为它的使用可能会带来受著作权许可证保护的代码库的权利、所有权和控制权问题。

软件物料清单（SBOM）

代码库中对开源代码依赖的全面清单，通常由软件组成分析工具生成。SBOM列出了所有的开源代码、专有代码、关联许可证、正在使用的版本、组件/依赖项以及依赖项下的子依赖项的下载位置。由于SBOM旨在跨公司和社区共享，因此必须具有一致的格式（即人机可读）和内容，这一点至关重要。美国国家标准与技术研究所的指南目前将三种格式指定为得到批准的标准格式：SPDX、CycloneDX和SWID。

软件组成分析（SCA）

用于自动执行开源软件管理流程的一类应用安全工具。SCA工具可识别代码库中使用的开源代码，提供风险管理和缓解建议，并执行许可证合规验证。

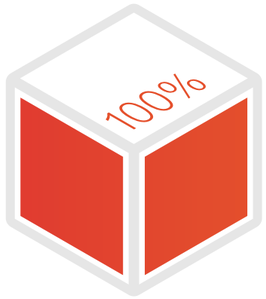
Apache Log4j2漏洞（BDSA-2021-3887和CVE-2021-44228等）

开源组件ApacheLog4J2（通常称为Log4j）广泛用于在Java社区中实施应用程序日志记录。Log4j中已经发现了多个漏洞，包括　程代码执行、拒绝服务和LDAP漏洞。

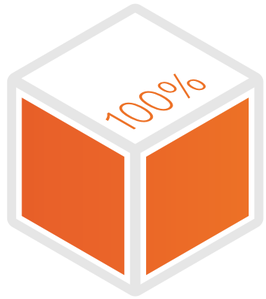
第14028号行政命令

2021年5月，美国总统拜登发布了一项名为“改善国家网络安全状况”的行政命令，指示各联邦政府机构为与联邦政府开展业务的企业制定软件安全指南。该命令中包括一个时间表，里面列出了截至本报告撰写之际不要求强制履行合同义务的各项活动。 然而，尽管不存在硬性要求，但该命令已经促使很多企业重新审查其安全实践，并严格审查其软件安全风险水平。《第14028号行政命令》大力提倡使用软件物料清单，因为这可以促进软件生产者和消费者之间交流软件供应链信息。

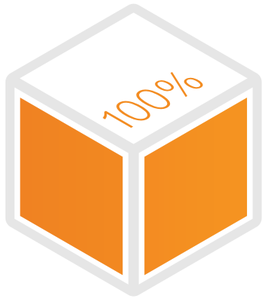
被扫代码库所在各行业使用开源的比例



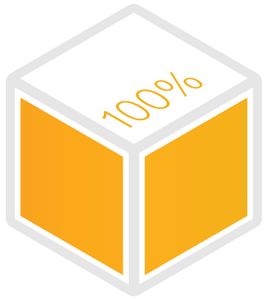
计算机硬件和半导体



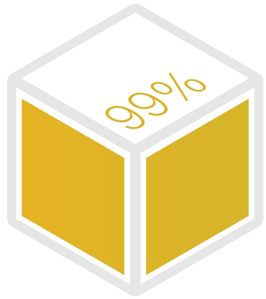
网络安全



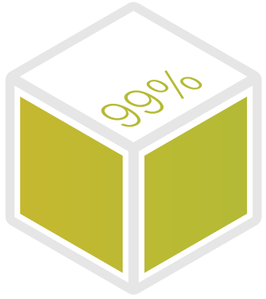
能源与清洁科技



物联网



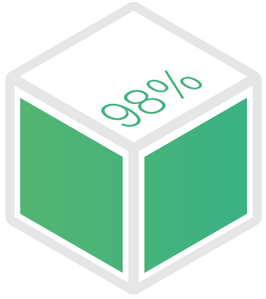
互联网和移动应用



营销科技

零售和电子商务

互联网和软件基础架构

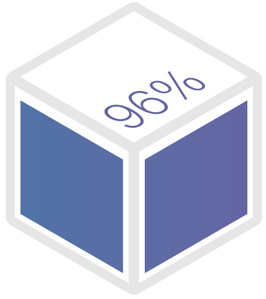


虚拟现实、游戏、娱乐和媒体

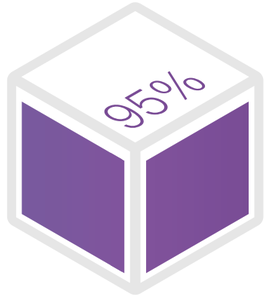
航空航天、汽车、运输和物流

金融服务和金融科技

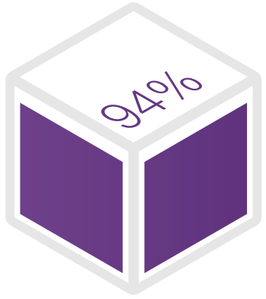
制造业、工业和机器人



企业软件/SaaS

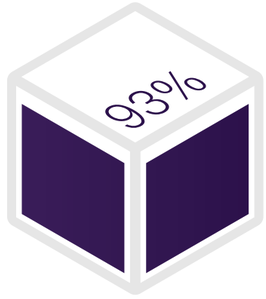


电信和无线



大数据、AI、 BI和机器学习

教育科技



医疗保健、健康科技和生命科学

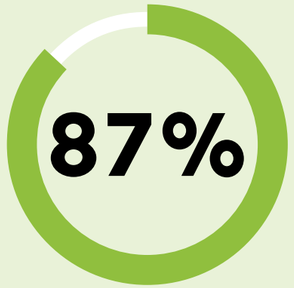
漏洞与安全

开源漏洞与安全

在Black　Duck审计服务团队今年分析的2,409个代码库中，有97%包含开源漏洞。81%包含至少一个公开开源漏洞，比2021 OSSRA 的调查结果仅减少了3%。

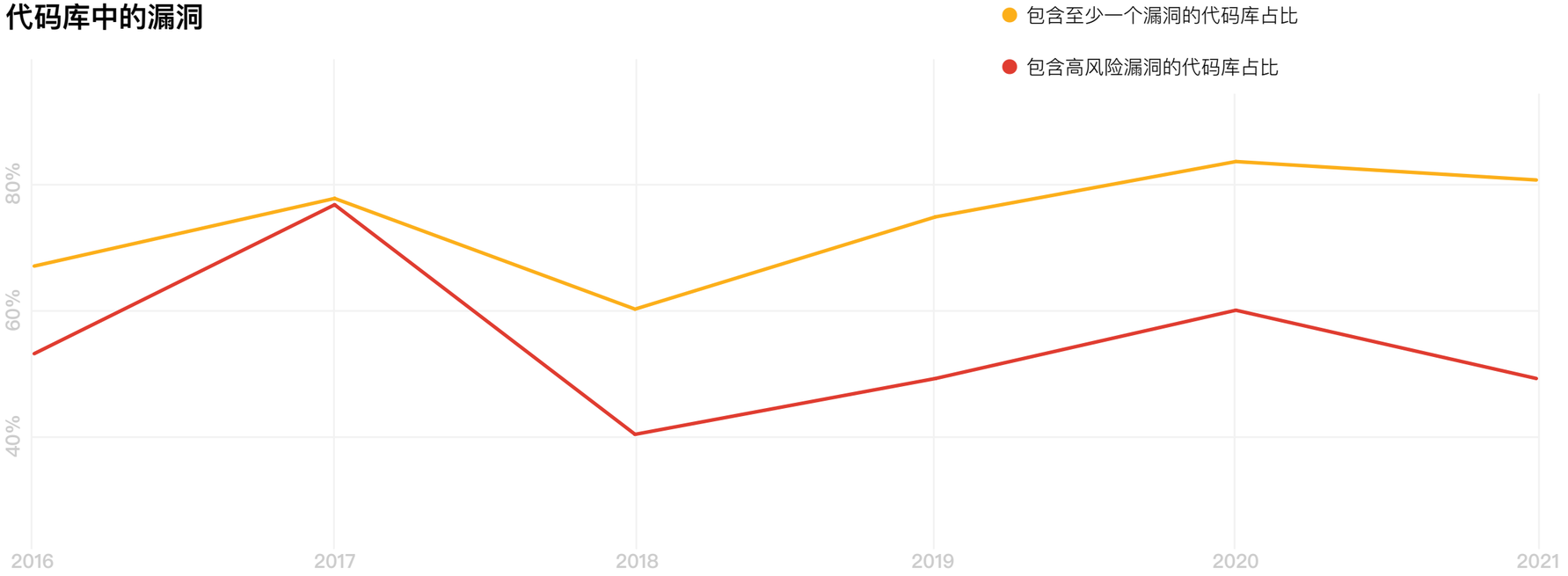
我们发现，包含至少一个高风险开源漏洞的代码库数量大幅减少；今年的被审代码库中只有49%包含至少一个高风险漏洞，比去年减少了11%。“高风险”漏洞表示该漏洞已被主动利用，并且有记录在册的概念验证（POC）或已被归类为　程代码执行（RCE）的漏洞。

所有的Black　Duck审计都会检查开源许可证的合规性，但客户可以自行决定放弃该审计的漏洞/运营风险评估部分。2021年，Black　Duck审计服务团队共进行了2,409次审计。在这些审计中，13%的客户（312家）选择放弃安全和运营风险评估。在2022年的OSSRA报告中，“开源漏洞与安全”以及“开源维护”部分的数据基于包含风险评估的　2,097个代码库，而“许可证”部分的数据则基于全部的2,409个代码库。



的代码库实施了风险评估

代码库中的漏洞



Percentage of Codebases Containing Vulnerable Components

43% jQuery

33% Lodash

23% Bootstrap (Twitter)

9% jackson-databind

8% Spring Framework

5% Netty Project
4% Apache Tomcat
4% OpenSSL
<1% Ruby
<1% PHP

漏洞与安全

2021：开源年

尽管审计中发现的高风险漏洞的减少令人鼓舞，但2021年仍然是充满开源问题的一年，包括供应链攻击[2]、黑客利用Docker镜像的攻击[3]、以及开发人员蓄意破坏自己的开源库从而破坏了数千个对其依赖的应用程序[4]。特别是，2021年底在被广泛采用的Apache Log4j程序中新发现的零日漏洞。这个被称为Log4Shell（CVE–2021–44228）的Log4j漏洞允许攻击者在受影响的服务器上执行任意代码。随着分析的展开，该漏洞的潜在严重性也变得清晰起来。

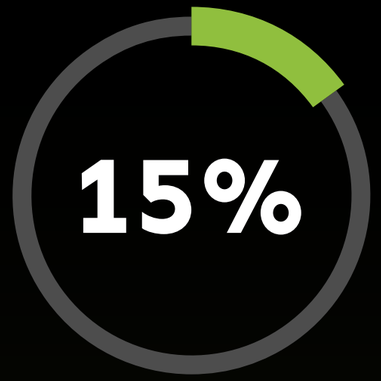
然而，Log4Shell最令人值得关注的地方并不是它的普遍性，而是它激发了人们的意识。随着该漏洞的发现，企业和政府机构被迫重新审视如何使用和保护主要由无偿志愿者而非商业供应商创建和维护的开源软件。该漏洞的发现还暴露了许多企业根本不知道其软件中使用了多少开源代码的问题。

同时，Log4j事件还揭示了企业对开源软件的固有信任问题：大多数开发团队在使用开源软件时都没有像对待商业或私有软件那样进行安全审查。

另外，开源代码的多样性使情况变得更复 。例如，GitHub有数百万个项目，但开发者只有个位数。而对于Kubernetes等广受欢迎的开源项目，则有大量的志愿开发者参与维护代码。当然，其中有一些维护人员受雇于使用Kubernetes的公司，因此其维护工作也符合公司利益。

Log4Shell的发现也有积极的一面，例如提醒企业他们需要采取一些措施来降低使用开源软件带来的相关业务风险。请注意，产生业务风险的并不是开源软件本身，而是对开源软件的不当管理。

降低业务风险的第一步应该是对企业使用的所有软件进行全面盘点，而不管这些软件的来源和获取方式。只有有了这个称为软件物料清单（SBOM）的完整清单，工作团队才能确定哪些资产使用了哪些组件。由软件组成分析工具（SCA）提供的这一级别的信息，能够使安全团队规划前进的道路，并制定计划来解决像Log4Shell等新披露的安全漏洞所带来的风险。



的被审计Java代码库包含了受漏洞影响的Log4J组件

各行业的漏洞情况

今年，我们发现在本报告涵盖的17个行业中，有4个行业 — 计算机硬件和半导体、网络安全、能源与清洁科技、物联网 — 的代码库中100%包含开源代码。其余的垂直行业有93%到99%的代码库中包含开源代码。即使比例最低的行业 — 医疗保健、健康科技和生命科学 — 也仍然有高达93%代码库包含开源软件。很明显，开源确实无处不在。美国政府也没有忽视这一事实。一份白宫在其2022年1月发布的简报中将软件描述为“无处不在，遍及各个经济领域，是美国人每天使用的产品和服务的基础。大多数的主要软件包中都存在开源软件……它带来了独特的价值，但也带来了独特的挑战。”[5]

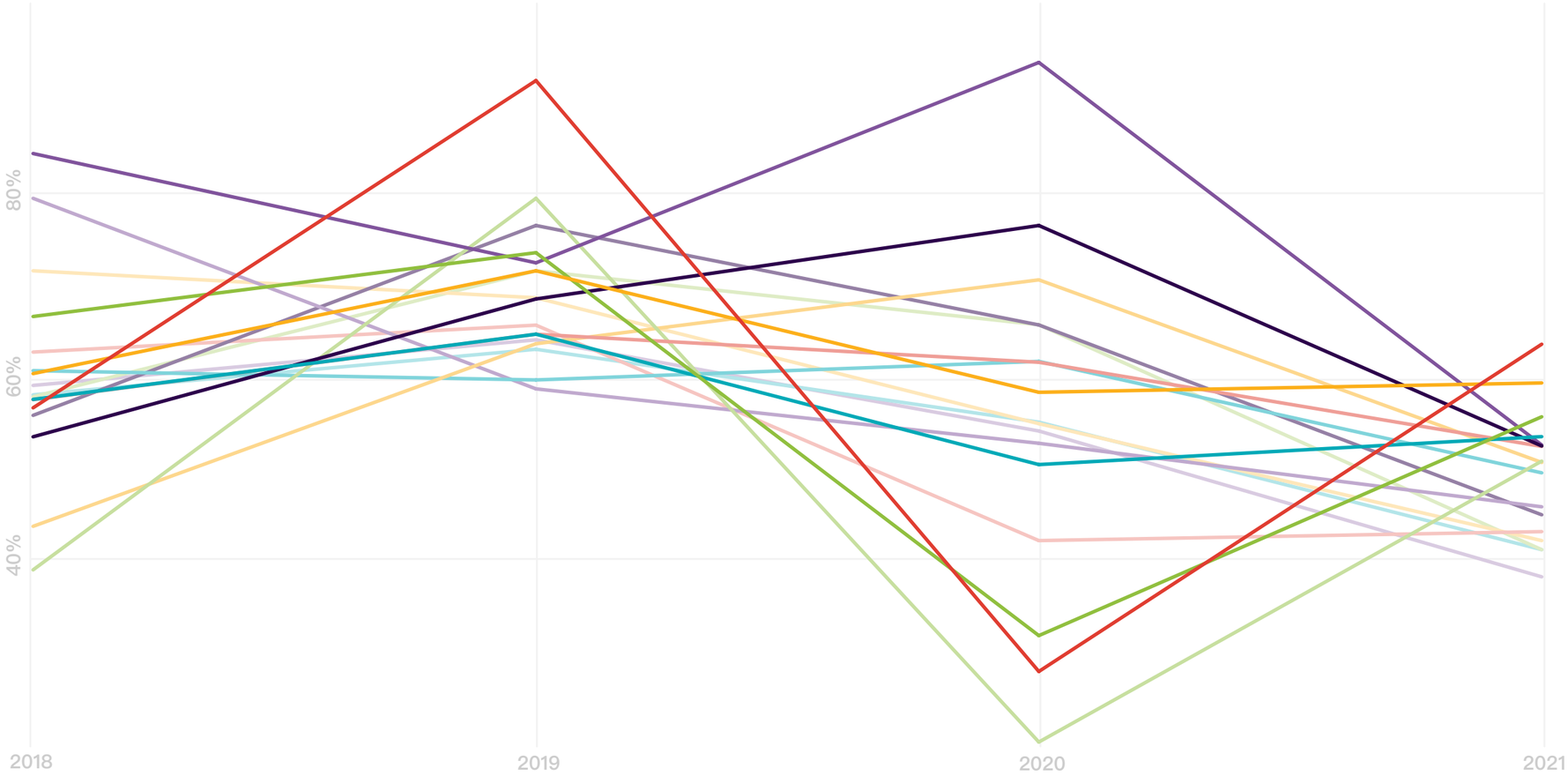
更深一层地，代码库中的开源代码数量也很高。例如，物联网领域100%的代码库包含开源代码，而该领域的被审源代码中高达92%都是开源代码组成。令人不安的是，64%的物联网代码库中还存在漏洞。

类似的，在航空航天、汽车、运输和物流行业，97%的代码库中包含开源代码，60%的代码由开源代码组成。当我们查看开源漏洞时，意外地发现该行业60%的代码库中存在开源漏洞。

我们在互联网和移动应用行业发现了更多的相同情况；99%的代码库中包含开源代码，80%的代码由开源代码组成，56%的代码库中存在开源漏洞。

同样的故事在所有行业上演。我们扫描的几乎所有代码库中都包含开源代码。开源代码在这些代码库中占比很高，而且很大一部分代码库容易被利用和攻击。

包含开源漏洞的代码库占比，按行业划分



2021

- 64% 物联网
- 60% 航空航天、汽车、运输和物流
- 56% 互联网和移动应用
- 54% 教育科技
- 53% 营销科技
- 53% 能源与清洁科技
- 53% 金融服务和金融科技
- 51% 零售和电子商务
- 51% 制造业、工业和机器人
- 50% 企业软件/SaaS
- 46% 虚拟现实、游戏、娱乐和媒体
- 45% 医疗保健、健康科技和生命科学
- 43% 计算机硬件和半导体
- 42% 大数据、AI、 BI和机器学习
- 41% 互联网和软件基础架构
- 41% 电信和无线
- 38% 网络安全

漏洞与安全

行政命令与供应链安全

于今年安全漏洞的增加，拜登总统颁布了《第14028号行政命令》，概述了与联邦政府开展业务的企业应如何保护其软件安全。虽然该行政命令旨在帮助加强美国的网络安全状况，但却推动了全国上下的各行各业和各个组织开展安全实践分析。

具体到软件供应链背景下的开源安全，首先必须承认开源软件就像商业软件一样，是由许多组件构成的，而这些组件本身可能也利用了大量的子组件或“依赖项”。

实际上大多数软件都是这种情况，无论其是用于移动应用、物联网固件、业务逻辑功能还是任何其他用途。每个元素都具有软件正常工作所需的依赖项。任何给定应用中所使用的依赖项都属于该软件供应链中的供应商。其中一些供应商可能是商业实体，例如提供定制SDK的供应商，但正如我们在开源使用中看到的那样，大多数依赖项都是开源的，而这些开源软件正是软件供应链中风险最大的地方。

将这种风险降到最低的唯一方法是使用全面且详尽的SBOM来跟踪依赖项及其相关风险，以便需要的时候及时通知并优先采取行动。

TOP 10 漏洞

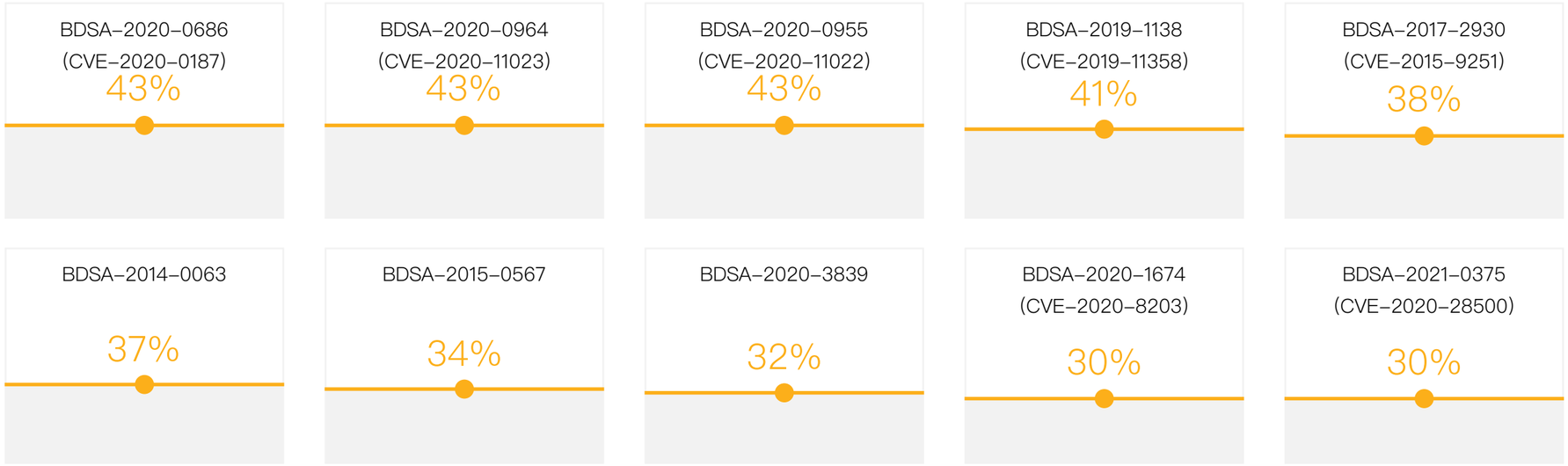
去年审计中发现的几个漏洞今年再次被发现，着实令人担忧。去年，我们在37%的代码库中发现了CVE-2020-11023和CVE-2020-11022。今年，两者的比例均上升至43%。我们在各个版本的jQuery中都发现了这两个被国家漏洞数据库（NVD）定为“中等严重级别”的CVE[6]。我们的审计显示，jQuery是存在漏洞最多的组件。而43%的被审代码库中包含jQuery 组件。

当某个漏洞的占比保持不变或逐年增加时，可以得出这样一个结论：某些DevSecOps团队难以控制开源风险。

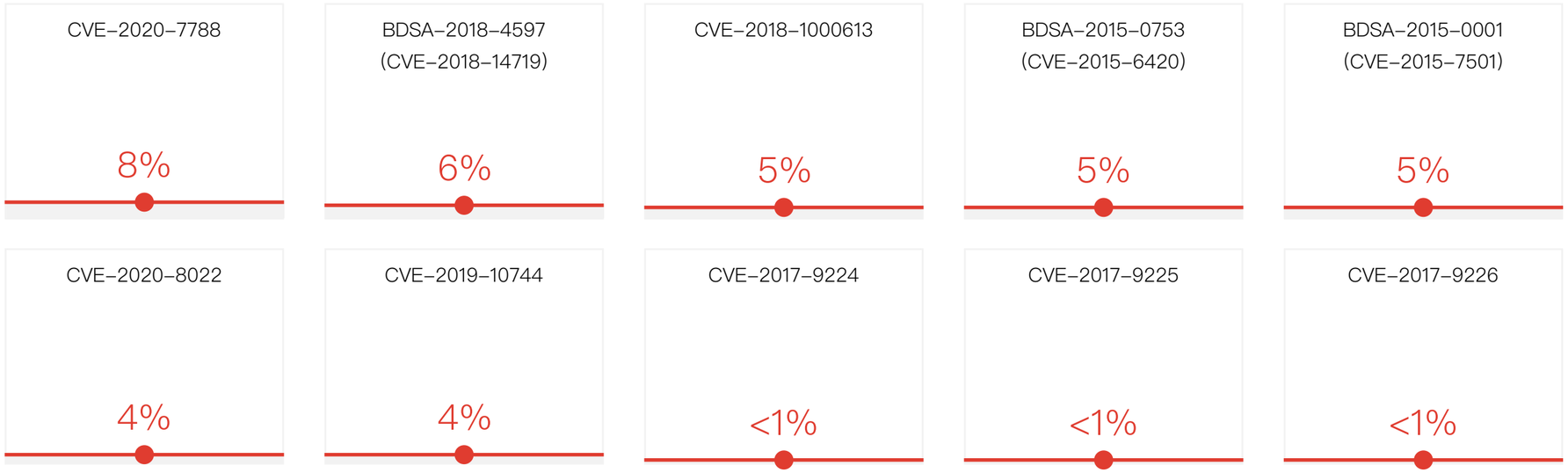
另一方面，我们看到在今年的被审代码库中，高风险CVE/BDSA漏洞的发现数量有了明显的改善。去年，排名第一的漏洞存在于29%的代码库中。而今年，排名最高的高风险漏洞CVE-2020-7788也仅仅出现在8%的代码库中。并且，所有高风险漏洞的重复出现比例均显著降低。

及时识别、确定优先级，并对高风险漏洞进行控制，才能够帮助团队避免对其组织构成最大威胁的风险。

存在 TOP 10 CVEs/BDSAs 的代码库占比



存在 TOP 10 高危漏洞 CVEs/BDSAs 的代码库占比



许可

许可

开源许可

Black Duck审计服务团队发现，2021年有53%的被审代码库包含的开源代码存在许可证冲突，比2020年的65%大幅减少。总体来说，许可证冲突在2020至2021年间减少了。

但具体到某个许可证，我们在2021年看到了一个和Creative Commons ShareAlike 3.0许可证有关的增长的例子。2021年，我们在17%的被审代码库中发现了与该许可证相关的某种形式的冲突，而这一比例在2020年是15%。

Creative Commons ShareAlike 3.0许可证冲突的数字，揭示出开源许可证方面一个经常被忽视的问题。商业和开源软件的开发人员均可将代码片段、函数、方法和可运行的部分代码引入其软件，因为整个软件都依赖于这些代码，所以它们通常被称为依赖项。因此，软件（包括开源项目）通常包含了更多的条款和条件，而不仅仅只有约束项目本身的许可证。

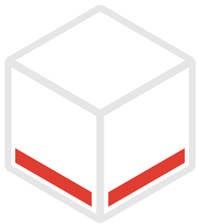
常用的node.js平台就是一个很好的例子。0.64.0及以下版本的node.js中通常包含名为react-native的组件，该组件采用了在Stack Overflow上发布的在Creative Commons Attribution ShareAlike3.0授权许可下的代码。从而导致了一个潜在的许可证冲突，因为react-native组件不可避免地需要满足Creative Commons Attribution ShareAlike 3.0中规定的许可证要求。该问题在Synopsys CyRC 研究人员Gary Armstrong和Rich Kosinski的一篇文章中进行了更详细的探讨。[7]

正如本报告在“简介”中所述，并购交易中的收购方对其收购的软件所带来的潜在风险变得更加敏感，特别是与许可、安全和软件中使用的开源代码质量相关的风险。另外，我们2021年的审计数据表明，被收购方对其软件中存在的可能破坏交易的潜在许可证冲突也变得越来越敏感，这促使他们在开始并购之前采取积极的措施来避免可能的许可证问题。

的被审代码库存在许可证冲突

的被审代码库中无许可证或使用定制许可证的开源代码

存在 TOP 10 许可证冲突的代码库占比



17%

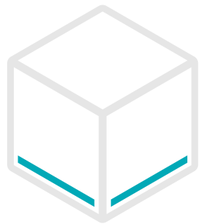
Creative Commons Attribution ShareAlike 3.0

13%

GNU Lesser General Public License v2.1 或更高版本

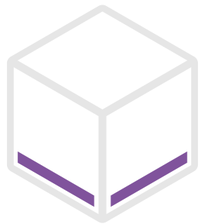
13%

Creative Commons Attribution ShareAlike 4.0



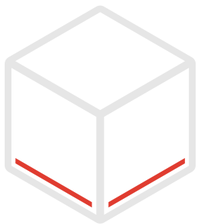
11%

GNU General Public License v2.0 或更高版本



8%

GNU General Public License v3.0 或更高版本



7%

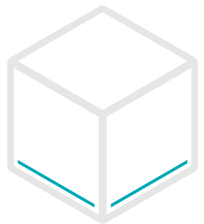
Apache License 2.0

7%

Creative Commons Attribution 3.0

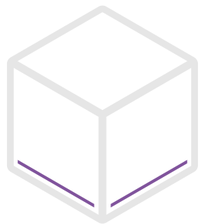
6%

GNU Lesser General Public License v3.0 或更高版本



4%

Eclipse Public License 1.0



4%

Mozilla Public License 2.0

许可

按行业划分，存在开源许可证冲突的代码库比例最高的行业（93%）是计算机硬件和半导体行业。其次是物联网行业，比例为83%。医疗保健、健康科技和生命科学行业中存在开源代码许可证冲突的代码库的比例最低，仅为41%。

正如本报告前面所讨论的那样，大多数包含开源代码的被审代码库，通常主要是由开源代码组成，并且包含了大量的开源漏洞。

了解许可证风险

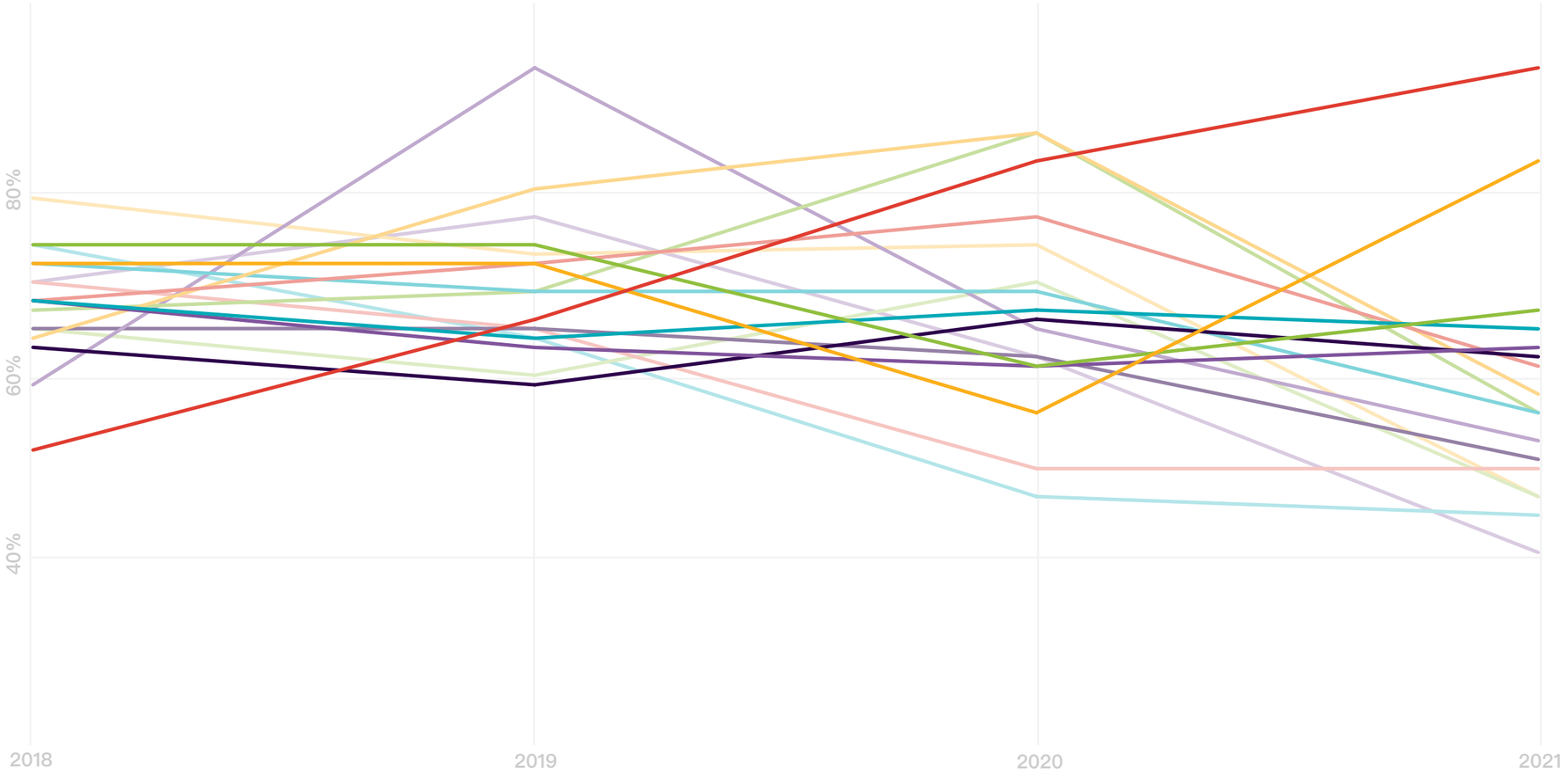
在美国和许多其他地方，创新作品（包括软件）默认受专有版权的保护。如果未经创作者/作者以授权许可证的形式明确允许，没有人可以合法的使用、复制、分发或修改该软件。即使最宽松的开源许可证也会规定用户在使用该软件时需要承担的义务。

当代码库中包含的开源代码许可证与该代码库的总体许可证可能存在冲突时，就会出现潜在的许可证风险。例如，GNU通用公共许可证（GPL）通常监管商业软件中开放源代码的使用。但商业软件供应商可能会忽视GPL许可证的要求，并制造出与该许可证的冲突。

定制化的开源代码许可证可能会对被许可方提出非预期的要求，因此经常需要对可能的知识产权IP问题或其他影响进行法律评估。例如，JSON许可证是源于宽松型MIT许可证，但是添加了“该款软件严禁用于恶意用途，仅限用于善意用途”的注释[8]。该含糊不清的声明导致其含义有待进一步界定，这往往给企业并购带来了特别的隐患，也就是收购方不愿承担的此类模糊不清的法律风险。

包含无许可证或使用定制许可证的开源组件的代码库还存在额外的风险。而30%的被审代码库中都包含无许可证或使用定制许可证的开源代码。

存在许可证冲突的代码库占比，按行业划分



2021:

- **93%** 计算机硬件和半导体
- **83%** 物联网
- **67%** 互联网和软件基础架构
- **65%** 网络安全
- **63%** 电信和无线
- **62%** 虚拟现实、游戏、娱乐和媒体
- **61%** 航空航天、汽车、运输和物流
- **58%** 制造业、工业和机器人
- **56%** 能源与清洁科技
- **56%** 大数据、AI、 BI和机器学习
- **53%** 互联网和移动应用
- **51%** 金融服务和金融科技
- **50%** 教育科技
- **47%** 营销科技
- **47%** 企业软件/SaaS
- **45%** 零售和电子商务
- **41%** 医疗保健、健康科技和生命科学

开源代码的维护

开源代码的维护

由开源代码的开发人员进行维护

在Black Duck审计团队审计并进行风险评估的2,000多个代码库中，居然有88%的代码库含有在过去两年中没有发生任何开发活动的开源组件，也就是说：在过去的24个月中该开源组件没有功能升级、代码优化和安全问题修复。这可能是因为项目参与者对他们的工作感到满意，认为不需要增加新功能或实施改进。但更有可能是意味着该项目不再被维护。

Linux基金会和哈佛大学创新科学实验室近期联合开展的“自由和开放源码软件普查II–应用库”（Census II）发现，[9]最受欢迎的开源代码几乎都是由少数贡献者开发和维护的。通过对前50个非npm项目进行研究，他们发现23%的项目是仅由一名开发人员贡献了超过80%的代码，完美诠释了“80–20”规则。[10]

94%的项目都是由不到10名开发人员贡献了超过90%的代码。正如“Census II”研究总结的：“这些调查结果与人们通常认为的数千或数百万开发人员一起负责开发和维护[免费和开源软件]的想法正好相反。”

诸如Kubernetes等受众非常广泛并成为行业标准的开源项目，是有大量的志愿开发者在编写代码的，甚至包括依赖Kubernetes开展业务的企业员工，因为这些企业可以从支持和鼓励员工参与Kubernetes项目中获益。

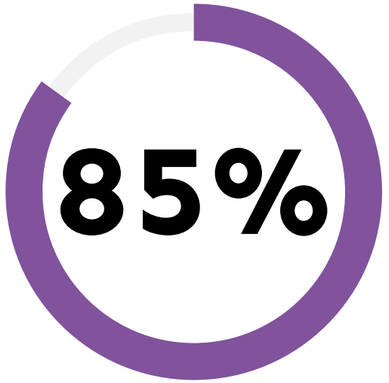
这种模式使Kubernetes 生态系统具有弹性。对于诸如此类的覆盖面极为广泛的项目，即便是有核心团队成员离场，通常也可以轻松应对，基本上不对整个项目造成干扰。

然而较小的项目就不尽如此了。GitHub有数百万个只有个位数开发人员的项目。这意味着可能一名开发人员的离场便意味着该项目失去了确切了解代码编写方式和其原因的唯一人员。顺便说一句，这些较小项目通常是执行基本任务的 — 例如执行日志数据保存任务的Log4j —因此是应用程序最常见的依赖项之一。

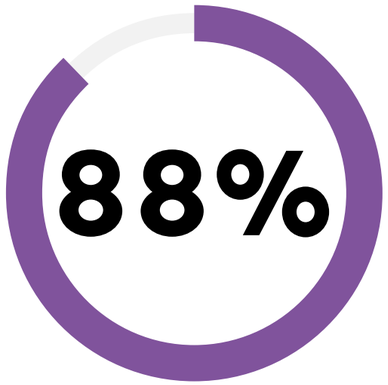
贵组织是否支持开源项目？

如果贵组织的软件依赖于开源项目的安全和 定性，则应该通过鼓励开发人员参与贡献、金钱援助或其他方式支持该项目，将其作为一种标准做法。响应这一号召的企业逐年递增。

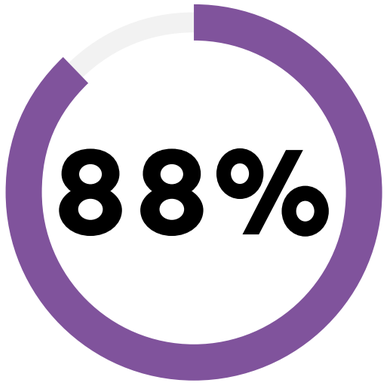
由Linux基金会赞助的《2020年FOSS贡献者报告》[11]显示，近一半的受访者都是受雇于企业才参与开源项目的。CyRC的一份调查报告[12]显示，大多数（65%）从事软件开发业务的企业都制定了相关政策，允许其开发人员为开源项目做出贡献。开源社区希望这一趋势能够继续下去。



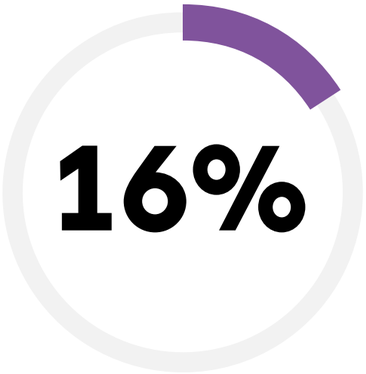
的代码库中包含至少四年未更新的开源代码



的代码库中包含过去两年没有任何开发活动的组件



的代码库中包含过时版本的组件



的代码库中包含至少一年没有任何维护活动的组件

开源代码的维护

由开源代码的消费者进行维护

在Black　Duck审计团队审计并包含风险评估的2,000多个代码库中，88%包含过时版本的开源组件。也就是说，未安装最新的升级包或补丁。

他们不对软件进行及时更新是有原因的。DevSecOps团队可能认为引入新版本会带来意料之外的后果，由此产生的风险会超过由此带来的益处。嵌入式软件仅存在从外部源码引入的漏洞风险，因此风险最低。

其不对软件及时更新还可能是因为时间或资源问题。对很多团队来说，构建和测试新代码已经耗掉了全部精力，因此，对现有软件进行更新就成了低优先级的事项，关键问题除外。

但是，这些占比达88%　　包含过时版本开源组件的代码库中，有很大一部分大概率是因为DevSecOps团队并不知道该开源组件的新版本已经可用，甚至根本不知道该组件的存在。正如早期的OSSRA报告所指出的那样，开源软件不同于商业软件 — 没有好与坏之分，只是性质不同 — 需要采用不同的技术进行管理。

例如，商业软件和开源软件的采购和补丁处理方式不同。购买商业软件通常需要采购部门的参与，并且需要引入在供应商风险管理计划中的审查标准。开源软件可能只是由开发人员自行决定是否下载和使用。其使用可能存在一些组织方面的限制　—　例如，只允许使用宽松型许可证的代码　—　但在许多情况下，这种指导实际上可能是不存在的。

除非开发人员能够对其引入到代码中的开源代码保留一份准确且时新的清单，否则，当他们转去其他项目或离职时，这些信息便可能丢失。开源组件从而可能被遗忘和忽略，直到其崩溃或变成高风险易受攻击的对象，才会被匆匆更新，就像Log4Shell一样。

此外，商业软件供应商会主动向用户推送补丁和升级包。开源软件则很少会这样，需要用户自己去了解组件的安全和　定性并及时安装新版本。

如果贵组织使用软件的话（现在没有哪个企业不使用任何软件），该软件无疑会包含许多开源组件。数据清楚地表明：组织需要准确、全面地清点软件中的开源组件，并制定适当的流程和策略来监控这些开源组件的漏洞、升级和整体运行状况。

23%的项目是仅由一名开发人员贡献了超过80%的代码。**94%**的项目都是由不到10名开发人员贡献了超过90%的代码。

“这些调查结果与人们通常认为的数千或数百万开发人员一起负责开发和维护[免费和开源软件]的想法正好相反。”

— Linux基金会“自由和开放源码软件普查II – 应用库”

结语

结语

开源软件：“女巫酿”的配方

尽管在审计中的高风险漏洞出现有所减少令人鼓舞，但2021年仍是开源漏洞和漏洞利用肆虐的一年（可以说现在每年都是如此）。例如，新加到最受关注漏洞之列的Log4Shell漏洞在2021年引起了巨大轰动。

从下载量来看，Log4j是目前最受欢迎的开源组件之一，也是其他7,000多个开源项目的依赖项。我们认为Log4Shell漏洞的危险程度可能使其足以在NVD发布的CVSS严重等级获得最高分—10分（最高10分）。在一系列的报告中可以看到，2021年我们不断收到有关Log4Shell试图扫描和攻击脆弱系统的警告。

我们的系统脆弱吗？我们的客户会受到攻击吗？我们会被追责吗？

据联邦贸易委员会（FTC），Log4j漏洞正在被广泛利用，它对包括企业软件和Web应用在内的“数百万消费品构成严重风险”。事实上，FTC认为该漏洞非常危险，所以发布了一份声明，称其“打算利用其全部法律权限来追究那些未能采取合理措施保护消费者数据免受Log4j或类似已知漏洞影响的企业。”[13]

得益于DevSecOps团队艰苦卓绝的努力—其中许多人不惜牺牲节假日的休息时间—很多企业均已基本控制住了Log4Shell带来的威胁。但在这场喧嚣中，他们却忽略了一个事实：企业之所以要马不停蹄的执行补救措施，是由于其不知道Log4j在其系统应用中的存在位置，或者说，企业根本不知道Log4j是否存在。于是大量的IT团队中参与识别该问题的人数成倍增加，他们都希望尽快找到“我们是否容易受到Log4Shell攻击？我们供应商的软件是否易受攻击？使用我们软件的客户是否易受攻击？”等问题的答案。

蝾螈之目青蛙趾，蝙幅之毛犬之齿，蝮舌如叉蚯蚓刺，蜥蜴之足枭之翅，炼为毒蛊鬼神惊，扰乱人世无安宁。

（威廉莎士比亚 · 麦克白，第四幕）

OSSRA报告的主旨之一是强调未对开源的使用实施管理可能带来的风险。正如我们之前所说的那样，切记开源本身并没有问题，这一事实与缺乏开源管理所导致的问题并不冲突。

在早期阶段，开源社区不得不承受来自商业软件世界的贬低评论，通常是暗示开源使用的危险。此类言论包括将开源软件 称为“万灵药（Snake Oil）”（Ken Olsen，1987年，曾任Digital Equipment Corporation公司CEO），还有将其称为“一种在知识产权意义上附着在其所接触的一切事物上的癌症”（Steve Ballmer，2001年，曾任Microsoft首席执行官）。

这些CEO的批评既不准确，也禁不住长期考验。事实上，开源现已成为商业软件的基础，正如今年的OSSRA报告所指出的那样，97%的商业代码中都包含开源组件。

然而，尽管使用开源代码已成为一种普遍现象，但认为开源软件在某种程度上具有内在危险的误解却依然存在。就在去年，美国国家安全局负责网络安全的副顾问Anne Neuberger 在讨论 Log4Shell 漏洞时还将开源描述为“女巫酿(Witches’Brew)”。[14]

开源：一盅佳酿

“女巫酿”一词可以追溯到16世纪，虽然有多种解释，但所有解释都围绕着“未知和潜在危险成分的混合物”这一含义展开，这似乎也是Neuberger女士对开源的看法。

对任何安全程序而言，核心原则都是了解您所构建或使用的代码中包含哪些内容。如果没有这些信息，就是身处黑暗之中。有效的开源管理始于对开源组成的识别。

软件物料清单

软件物料清单（SBOM）的概念来自制造业，传统BOM详细列出产品组成的物资明细。当发现缺陷零件时，制造商可以准确地知道哪个产品受到了影响，以便安排修理或更换。同样，为确保代码始终保持高质量、合规与安全，您必须始终对开源组件进行盘点，维护准确时新的SBOM。与制造业一样，开源组件的SBOM助您快速定位存在风险的组件，并合理确定修复工作的优先级。全面的SBOM应列出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。

2022年的审查显示，97%的商业代码中包含开源代码，因此对应用程序中的开源组件的可见性是有效DevSecOps或AppSec工作的强制和最低要求。如果没有这些信息，业务风险就会增加。可通过全面了解为企业提供动力的开源组件来进行预防。

结语

相关读物

NTIA关于软件组件透明度的多利益相关方流程（NTIA Multistakeholder Process on Software Component Transparency）

关于改善国家网络安全状况的行政令（Executive Order on Improving the Nation's Cybersecurity）

免费和开源软件第二次统计 — 应用库（Census II of Free and Open Source Software—Application Libraries）

Log4Shell：通过设置限制来打造可信开源软件的案例（Log4Shell: A Case for Trusting Open Source—With Guardrails）

参考资料

1. Ian Forsyth，“全球并购势头像火车一样继续前行”（Global M&A momentum to keep going like a train），《the Press and Journal》，2022年2月25日。
2. Eran Orzel，“2021年软件供应链安全报告”（2021 Software Supply Chain Security Report），《Argon Security》，2021年。
3. CVEdetails.com，“Docker 安全漏洞”（Docker security vulnerabilities），2021年。
4. Dominick Reuter，“一名开发人员捣毁了他们自己的开源库，破坏了数千个应用程序，公然叫板这家超大规模企业”（A developer sabotaged their own open-source libraries, breaking thousands of apps, in apparent protest of mega-corporations），《Business Insider》，2022年1月10日。
5. Jen Psaki，“白宫软件安全会议解读”(Readout of White House Meeting on Software Security)，whitehouse.gov，2022年1月13日。
6. 国家漏洞数据库，“CVE-2020-11023详细信息”（CVE-2020-11023 Detail），2022年2月7日。
7. Gary Armstrong和Rich Kosinski，“使用Node.js的许可证和安全风险”（The license and security risks of using Node.js），synopsys.com，2019年8月13日。
8. “JSON许可证”（The JSON License），json.org，2002年。
9. Linux基金会和哈佛大学创新科学实验室，“免费和开源软件第二次普查-应用库”（Census II of Free and Open Source Software—Application Libraries），linuxfoundation.org，2022年1月。
10. Vincent Tabora，“软件工程中的帕累托原则 – 应用80/20规则”（The Pareto Principle In Software Engineering —Applying the 80/20 Rule），medium.com/0xcode，2021年8月30日。
11. Linux基金会和哈佛大学创新科学实验室，“2020年FOSS贡献者报告”（Report on the 2020 FOSS Contributor Survey），linuxfoundation.org，2020年12月10日。
12. Synopsys网络安全研究中心，“2020年DevSecOps实践和开源管理报告”（DevSecOps Practices and Open Source Management in 2020），synopsys.com，2020年。
13. 联邦贸易委员会，“FTC警告各公司修复Log4j安全漏洞”（FTC warns companies to remediate Log4j security vulnerability），ftc.gov，2022年1月4日。
14. Jack Gillum和Jennifer Jacobs，“官方称，一些联邦系统受到了软件缺陷的影响”（Some Federal Systems Affected by Software Flaw, Official Says），彭博社，2021年12月16日。

Synopsys 的与众不同之处

Synopsys软件与质量部门提供集成解决方案，可以改变您构建和交付软件的方式，在应对业务风险的同时加速创新。我们业界领先的软件安全产品和服务是市场上最全面的产品组合，并可与第三方和开源工具互操作，从而帮助企业利用现有投资来构建最能满足组织需求的安全程序。只有Synopsys能够满足您在构建可信软件时的一切需求。

关于 CyRC

Synopsys网络安全研究中心（CyRC）致力于加快获取有关软件漏洞的身份确认、严重性、漏洞利用、缓解和防御信息。CyRC旨在助力Synopsys履行更　大的使命：让软件帮助我们的生活变得更安全、更有质量。CyRC通过发布支持强有力网络安全实践的调研报告来帮助相关人员提高对问题的认识。

有关详细信息，请访问 www.synopsys.com/software.

Synopsys, Inc.
690 E Middlefield Road
Mountain View, CA 94043 USA

美国销售：800.873.8193
国际销售：+1 415.321.5237
电子邮件：sig-info@synopsys.com