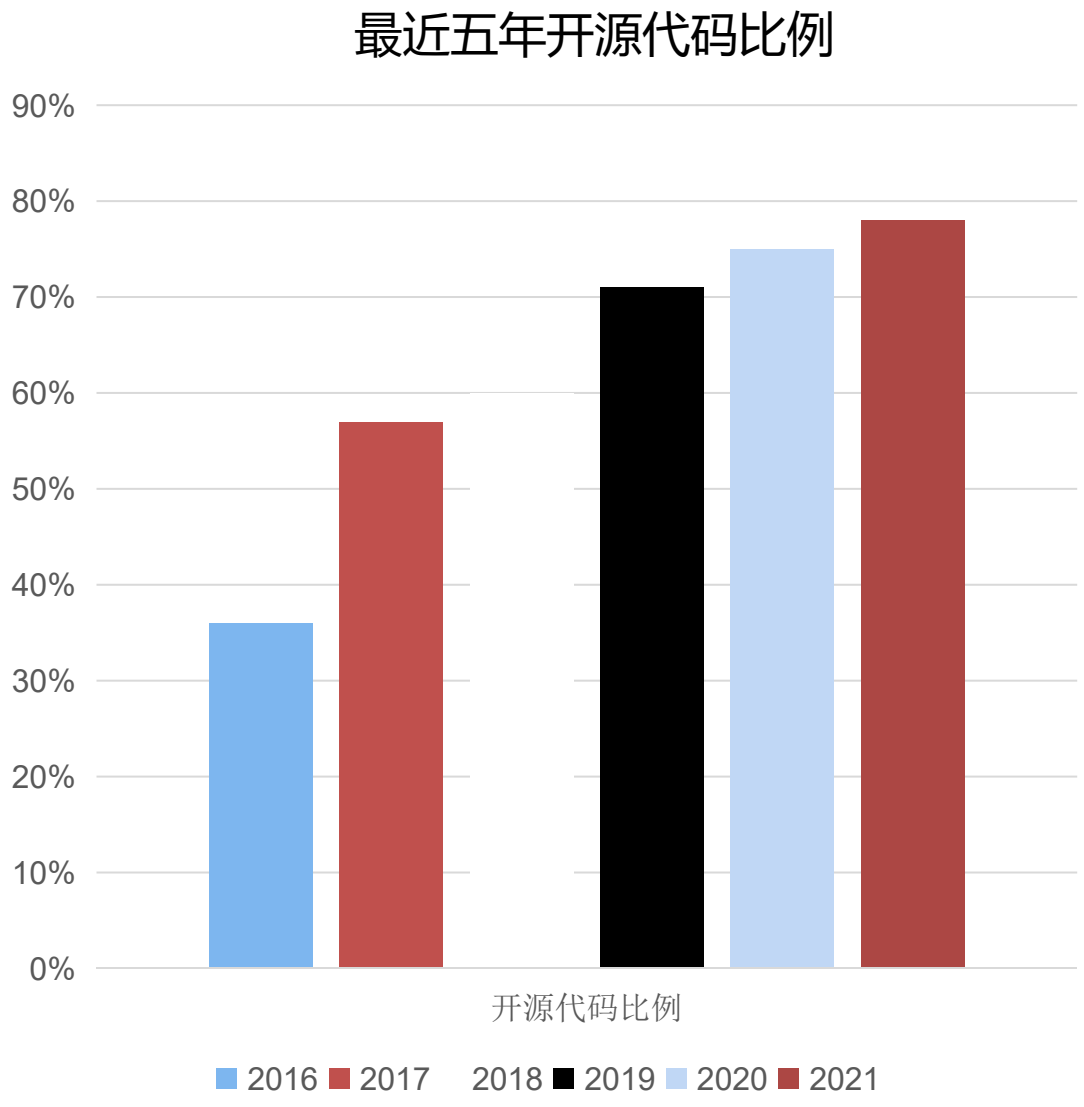
• 2022开源软件安全和风险分析报告(OSSRA)解读

- 王永雷-Leo Wang

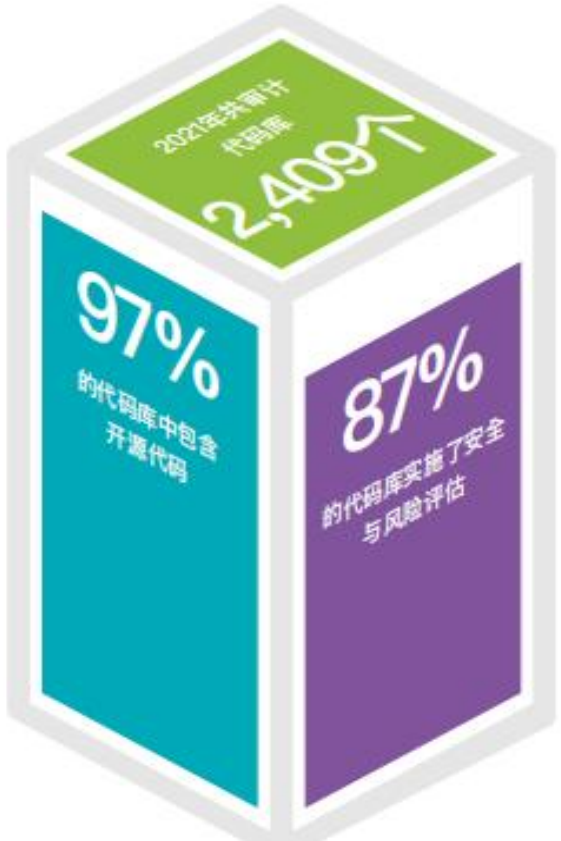
软件继续吞噬世界，开源吞噬软件



开源变好了么？

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

Log4J漏洞启示录-开源依赖管中窥豹

CVE-2021-44228

2022 开源安全和风险分析报告 | ©2022 Synopsys, Inc.

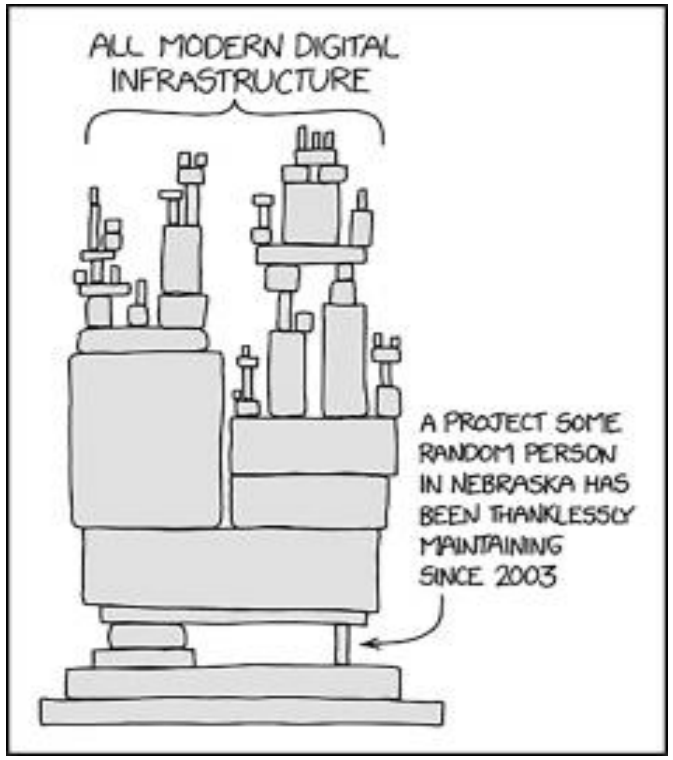
15%

>7000

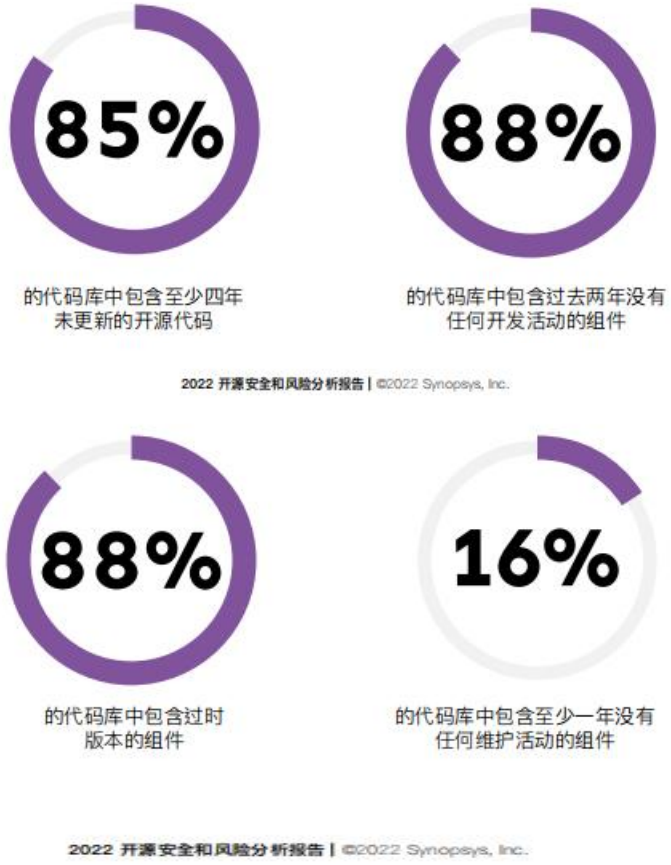
Log4j事件还揭示了企业对开源软件的固有**信任**问题：大多数开发团队在使用开源软件时都没有像对待商业或私有软件那样进行安全审查和社区健康度的考察

开源运维风险

- 80/20



Randall Monroe's XKCD comic illustrates the open source dilemma: overreliance on a small number of volunteer project maintainers



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。

开源合规风险

开源合规风险

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。

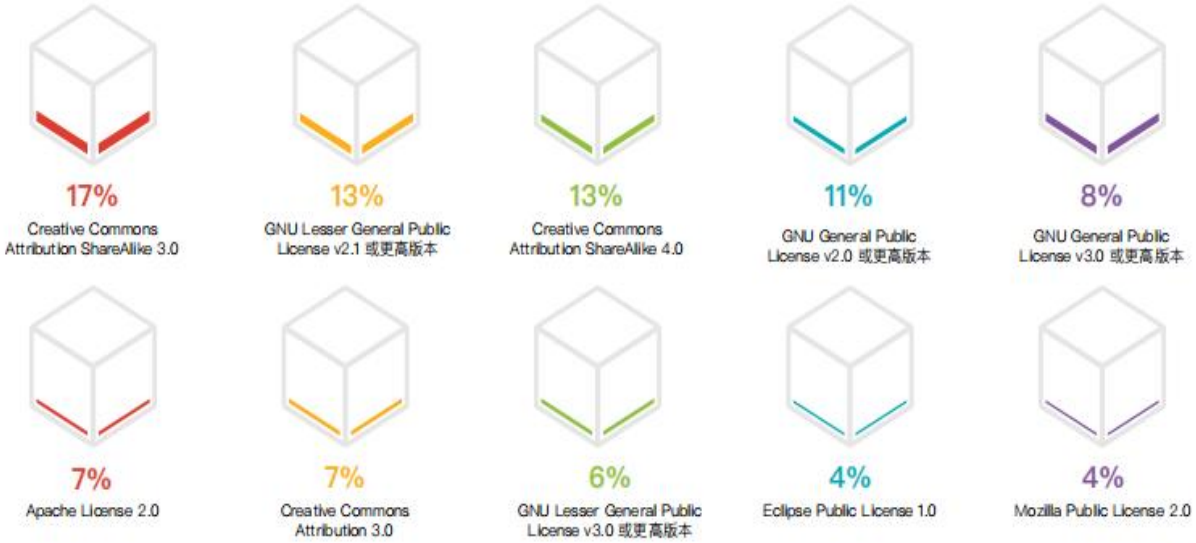
常用的node.js平台就是一个很好的例子。0.64.0及以下版本的node.js中通常包含名为react-native的组件，该组件采用了在Stack Overflow上发布的在Creative Commons Attribution ShareAlike3.0授权许可下的代码。从而导致了一个潜在的许可证冲突，因为react-native组件不可避免地需要满足Creative Commons Attribution ShareAlike 3.0中规定的许可证要求。该问题在Synopsys CyRC 研究人员Gary Armstrong和Rich Kosinski的一篇文章中进行了更详细的探讨。[7]

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



我们的目光一直在GPL上，其他的许可证

Share Alike

1)  署名（Attribution 简写为by）：必须提到原作者。

2)  非商业用途（Noncommercial 简写为nc）：不得用于盈利性目的。

3)  禁止演绎（No Derivative Works 简写为nd）：不得修改原作品。

4)  相同方式共享（Share Alike 简写为sa）：如果允许修改原作品，那么必须使用相同的许可证发布。