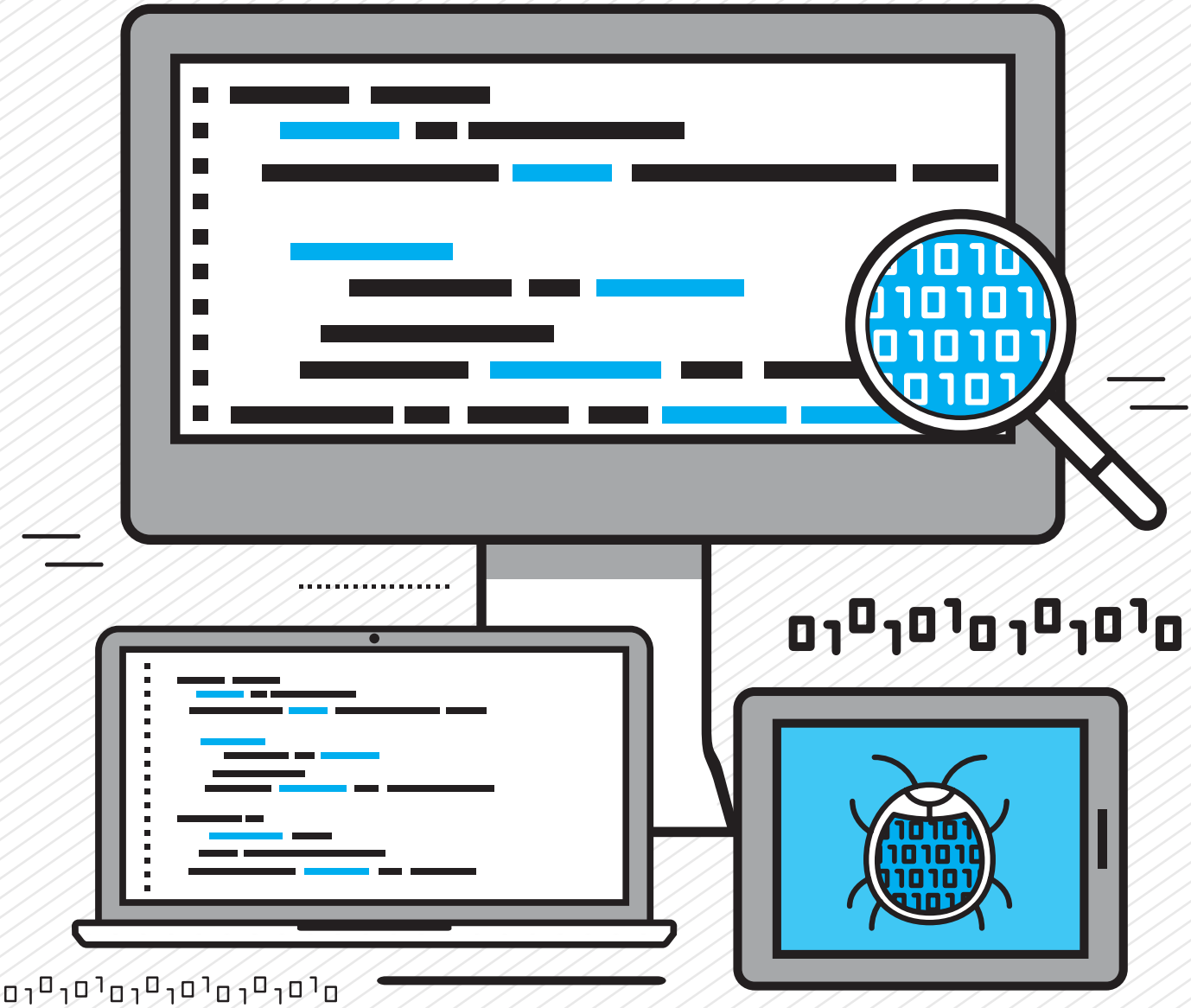
2017 OPEN SOURCE *SECURITY & RISK* ANALYSIS



BLACKDUCK

BLACK DUCK'S SECOND Open Source Security and Risk Analysis (OSSRA) provides an in-depth look at the state of open source security, compliance, and code-quality risk in commercial software. Each year, Black Duck's On-Demand audit services group conducts open source audits on thousands of applications for its customers – primarily in conjunction with merger and acquisition transactions. This analysis was done by Black Duck's Center for Open Source Research and Innovation (COSRI) and examines findings from the anonymized data of more than 1,000 commercial applications audited in 2016.

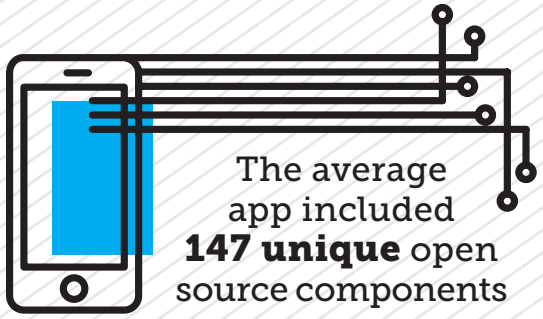
This COSRI analysis includes insights and recommendations intended to help organizations and their security, risk, legal, and development teams better understand the open source security and license risk landscape as they strive to improve their application risk management processes.

Black Duck On-Demand audits found open source components in **96%** OF applications scanned

1

The use of open source software is an essential part of application development

96% OF applications scanned in this analysis utilized open source



The average app included **147 unique** open source components

2

Organizations are not effectively dealing with open source security threats

67% OF analyzed applications using open source had vulnerabilities in the components used

On average, vulnerabilities identified in these applications have been publicly **known for over four years**

3

Financial Services and Financial Technology (FinTech) had the highest number of vulnerabilities per application

Financial industries contained 52 vulnerabilities per application, and **60%** of those applications **contained high-risk vulnerabilities**

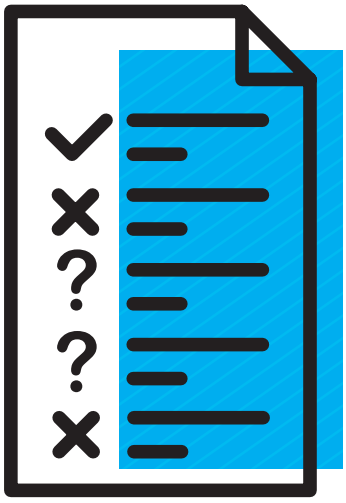
Retail and E-commerce had the highest proportion of applications with high-risk vulnerabilities, with **83% of audited applications** containing **high-risk vulnerabilities**

4

Open source usage spans every industry vertical

Risky versions of components such as Apache Tomcat and OpenSSL were commonly found across industries

5



License conflicts are widespread.

OVER **85%** OF of the analyzed applications contained components with **licenses out of compliance**

53% of applications scanned had **"unknown" licenses**, meaning no one has permission from the creator(s) of the software to use, modify, or share the software.

6

Know your code: *High-risk vulnerabilities were identified in even the most commonly used open source components*

On average, apps contained **27 vulnerable** open source components

7

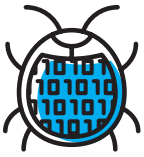
Commonly used infrastructure components, contained high-risk vulnerabilities

Even versions of **Linux Kernel, PHP, MS .NET Framework, and Ruby on Rails** were found to have vulnerabilities

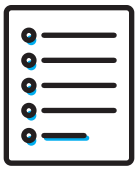
OPEN SOURCE VULNERABILITIES ARE ATTRACTIVE TARGETS FOR ATTACKERS

Open source is neither more nor less secure than custom code. However, there are certain characteristics of open source that make vulnerabilities in popular components very attractive to security attackers. Specifically:

OPEN SOURCE IS UBIQUITOUS in commercial and internal applications, providing attackers with a target-rich environment when vulnerabilities are disclosed.



VULNERABILITIES – and often exploits – are publicly disclosed through sources like the National Vulnerability Database (NVD), mailing lists, and project home pages.



UNLIKE COMMERCIAL SOFTWARE, where updates are automatically "pushed" to users, open source has a "pull" support model – users are responsible for keeping track of vulnerabilities as well as fixes and updates for the open source they use.

OPEN SOURCE can enter code bases through a variety of ways. If an organization is not aware of all the open source in use, it cannot defend against common attacks targeting known vulnerabilities in those components.

TRADITIONAL TESTING TOOLS MISS OPEN SOURCE

Many of the audited organizations have internal security programs and deploy security testing tools such as static analysis and dynamic analysis. While those tools are useful at identifying common coding errors that may result in security issues, they have proven ineffective at identifying vulnerabilities that enter code through open source components. For example, over 4% of the tested applications included the Poodle vulnerability; over 4% included Freak; and over 3.5% included Drown. Even Heartbleed, perhaps the most well-known vulnerability of all, was found in over 1.5% of the code bases, more than two years after it was publicly disclosed.

Known open source vulnerabilities were found in more than

67%

OF

applications using

open source components

OPEN SOURCE IS AT THE CORE OF COMMERCIAL APPLICATION DEVELOPMENT

THE USE OF OPEN SOURCE for application development continues to grow. A recent Forrester Research report called attention to open source's pre-eminence in application development, with custom code comprising only 10 to 20% of applications. Open source is used in all industries by organizations of all sizes. The reasons are straightforward – open source lowers development costs, speeds time to market, and accelerates innovation.

Black Duck On-Demand audits found open source components in 96% of the applications scanned. On average, open source comprised 36% of the code base in these applications. This is a lower percentage than cited by Forrester, a reflection of the mature application codebases that are typically the focus of Black Duck audits.

The functionality of certain open source components is so important that those components are used in a significant share of applications. By far the most popular is jQuery, a component that makes it easier to use JavaScript on websites. It was present in 58% of audited applications. The ubiquity of such components speaks to their utility, but also provides opportunities for far-reaching attacks from those seeking to exploit security vulnerabilities.

Top 10 Most Common Components	Percent Apps with Component
jQuery	57.9%
jQuery UI - jQuery/jQuery-UI on GitHub	36.2%
Bootstrap	35.8%
JUnit	30.9%
Apache Log4j	26.1%
Apache Commons Lang	25.9%
Commons IO	25.5%
Modernizr	23.3%
SAX	21.8%
Json.NET	19.9%

COMPANIES ARE NOT TAKING THE NECESSARY STEPS TO PROTECT APPLICATIONS

With 3,623 new open source component vulnerabilities reported in 2016 – almost 10 per day on average and a 10% increase from last year – the need for effective open source security and management is more important than ever. The need for greater visibility into and control of the open source in use is clear. Detection and remediation of security vulnerabilities should be a high priority.

However, the Black Duck On-Demand audits revealed that many organizations are unprepared for an attack. Known open source vulnerabilities were found in 67% of the applications using open source components. Furthermore, these vulnerabilities (and in many cases associated exploits) have, on average, been publicly disclosed for 1,527 days, giving would-be hackers a ripe target. This is only a small improvement from the last OSSRA report that showed 67% of application scanned had vulnerabilities with an average vulnerability age exceeding five years. In short, companies have a lot of work to do to close the vulnerability management gap, and progress remains unacceptably slow.

HIGH SEVERITY VULNERABILITIES ARE IN COMPONENTS ORGANIZATIONS KNOW THEY USE

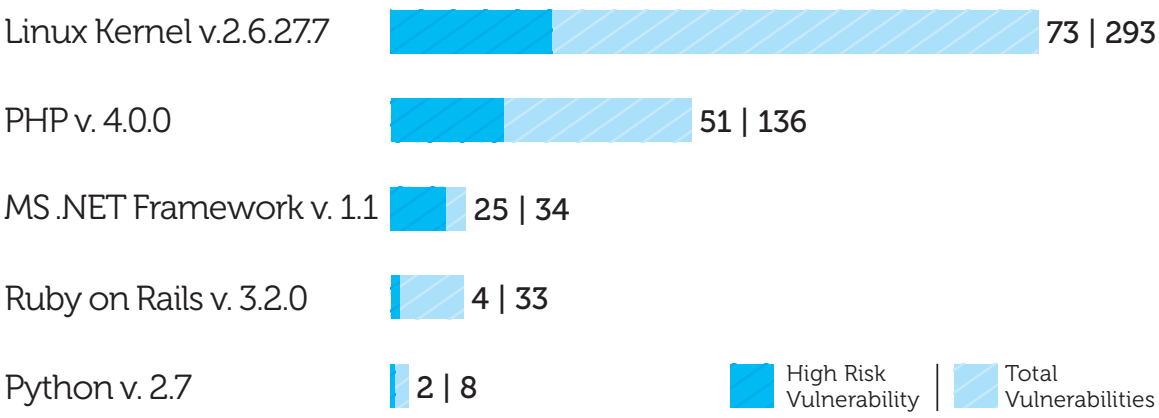
With an average of 27 vulnerable components found in each application (up from 22.5 in 2015), security practitioners may wonder how many of those components are truly putting their applications and organizations at risk. The audits showed that 52.6% of the vulnerabilities found in applications were ranked as "high" severity by the National Institute of Standards and Technology (NIST).

NIST Method to calculate CVSS (Common Vulnerability Scoring System) score. A vulnerability is rated "High" based among other things on exploitability: 1) it is network exploitable, or internet accessible, 2) an attacker with relatively few skills could execute the exploit, and 3) authentication is not required to exploit the issue

Additionally, these high-risk vulnerabilities were found in some of the most used versions of the most prevalent components, including Apache Commons Collections and Spring Framework. It is wise for organizations to take a close look at the open source in their applications and check for the components the audits identified as both common and highest-risk.

Top 10 Most Common Higher-Risk Components	Percent of Apps Analyzed	High-Risk Vulns Found per Component
Apache Commons FileUpload	13.8%	3
Apache Commons Collections	11.8%	2
Apache Tomcat	10.1%	11
Spring Framework	9.9%	2
OpenSSL	8.3%	27
Apache Geronimo	4.6%	4
zlib	4.2%	4
Apache Struts	3.9%	20
PNG reference library: libpng - libpng-stable	3.3%	9
libxml2	2.2%	7

Additionally, some of the most commonly used infrastructure components (those components that are core to a particular operating system or programming language) can themselves be a source of significant security risks:

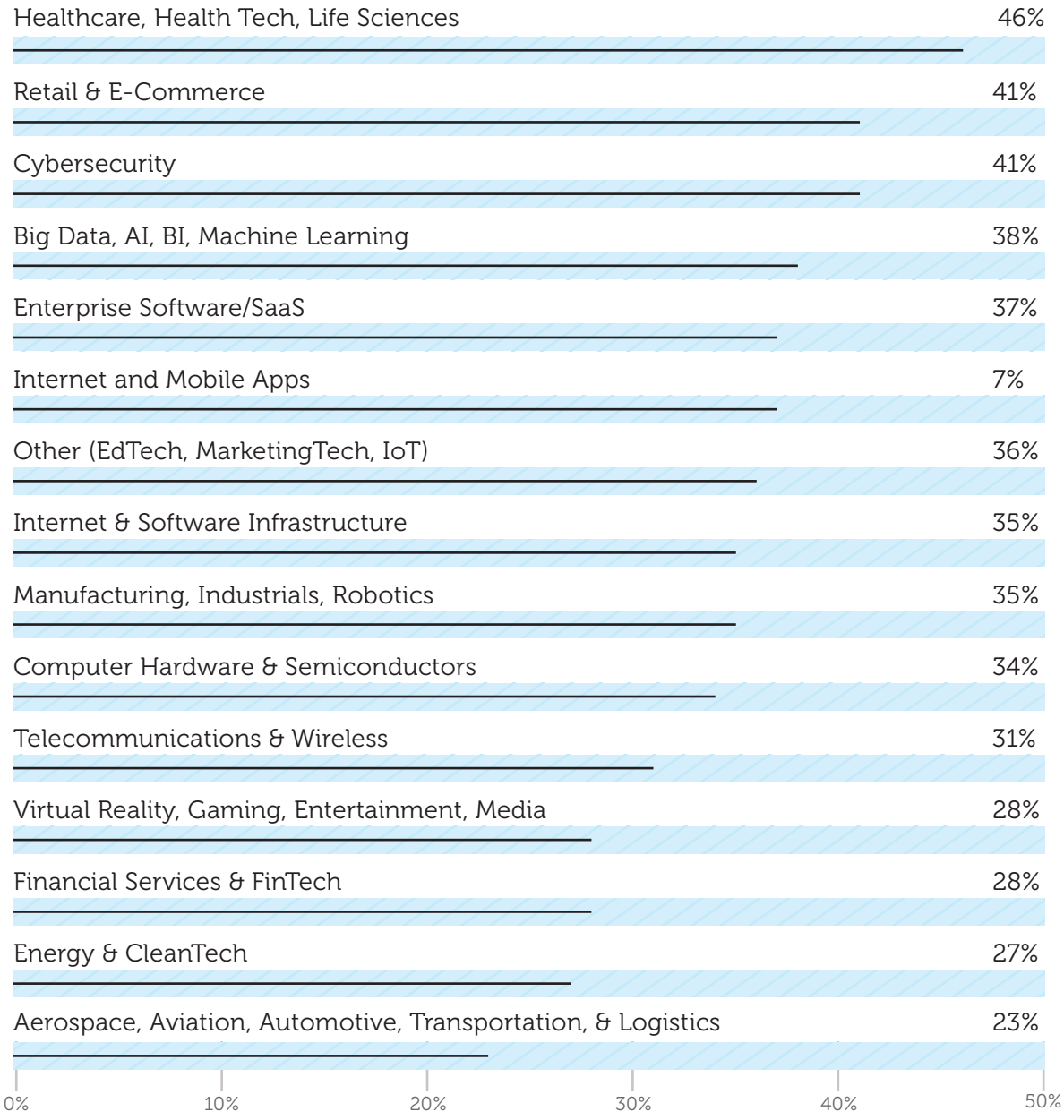


It is important for organizations to remember that these infrastructure and language components also need to be monitored as part of a sound open source security practice.

OPEN SOURCE SPANS INDUSTRY VERTICALS

Open source use is pervasive across every industry vertical. The audits showed that open source components made up between 23% to 46 % of organizations' commercial applications, depending on the industry.

PERCENT OF OPEN SOURCE BY INDUSTRY



Consistent with the cross-industry findings on open source use, highly vulnerable components affect applications in individual industries. Apache Tomcat and OpenSSL, for example, were among the most common high-risk components found across multiple industry verticals. Once again, this suggests that security threats are widespread. An application exploit effective against one industry is likely to be effective in others.

Industry	Most Common Higher-Risk Component	Percent of Industry Apps	High-Risk Vulns per Component
Aerospace, Aviation, Automotive, Transportation, & Logistics	Apache Commons FileUpload	10%	2
Big Data, AI, BI, Machine Learning	Apache Tomcat	24%	6
Computer Hardware & Semiconductors	Sun Java Platform Standard Edition SDK (J2SDK) (JDK)	6%	28
Cybersecurity	OpenSSL	13%	19
EdTech	Symfony	9%	9
Energy & CleanTech	Sun Java Platform Standard Edition (JRE) (J2RE)	8%	220
Enterprise Software/SaaS	OpenSSL	8%	13
Financial Services & FinTech	Apache Struts	22%	17
Healthcare, HealthTech, Life Sciences	Apache Tomcat	11%	6
Internet & Software Infrastructure	Apache Tomcat	26%	6
Internet and Mobile Apps	libxml2	15%	1
Internet of Things	OpenSSL	20%	3
Manufacturing, Industrials, Robotics	OpenSSL	20%	8
MarketingTech	Symfony	11%	1
Retail & E-Commerce	Apache Tomcat	33%	5
Telecommunications & Wireless	OpenSSL	22%	11
Virtual Reality, Gaming, Entertainment, Media	Apache Tomcat	10%	5

NOT ALL INDUSTRY RISK IS CREATED EQUAL

Though vulnerable components were found in applications in every industry, there was a wide range in the overall vulnerability risk found in the applications in specific industries. Notably, the code audits of applications from industries the public entrusts daily with its most sensitive personal, financial, and intellectual property information were found to be at highest risk. Clearly these organizations had failed to take the appropriate steps toward detection, remediation and monitoring of open source components and vulnerabilities in their applications.

The Retail & E-commerce industry had the highest proportion – 83% - of applications containing high-severity vulnerabilities. The Financial Services and FinTech industry had the highest (52.5) average vulnerabilities per application, and 60% of those applications contained high-risk vulnerabilities. Ironically, the audits also revealed that cybersecurity applications had a disturbingly high incidence of high-risk vulnerabilities at 59%.

Industry	Vulns Per Application	Percent Apps with High-Risk Vulns
Retail & E-Commerce	51.8	83%
Internet & Software Infrastructure	33.0	70%
Financial Services & FinTech	52.5	60%
Big Data, AI, BI, Machine Learning	21.0	59%
Cybersecurity	39.0	59%
Manufacturing, Industrials, Robotics	34.9	59%
Enterprise Software/SaaS	7.9	55%
Healthcare, Health Tech, Life Sciences	11.8	47%
Telecommunications & Wireless	26.7	44%
Energy & CleanTech	40.0	42%
Internet and Mobile Apps	12.5	40%
Computer Hardware & Semiconductors	17.2	33%
Virtual Reality, Gaming, Entertainment, Media	24.5	33%
Other Tech (EdTech, Marketing Tech, IOT)	3.9	31%
Aerospace, Aviation, Automotive, Transportation, & Logistics	1.1	30%

LICENSE-COMPLIANCE RISK MAY BE HIGHER THAN YOU THINK

Today, security risk is top of mind because of the high-profile vulnerabilities such as Heartbleed, Poodle, Freak, and Shellshock, and the sometimes-costly associated exploits. However, it is also important to recognize the importance of open source license compliance in reducing risk.

85% OF applications analyzed contained components with license conflicts

Audited applications contained 147 open source components on average, a daunting number of license obligations to keep track of without automated processes in place. Indeed, 85% of audited applications contained components with license conflicts, the most common of which were GPL license violations. 75% of applications contained components under the GPL family of licenses, but only 41% of those applications complied with GPL obligations.

WHAT YOU DON'T KNOW CAN HURT YOU

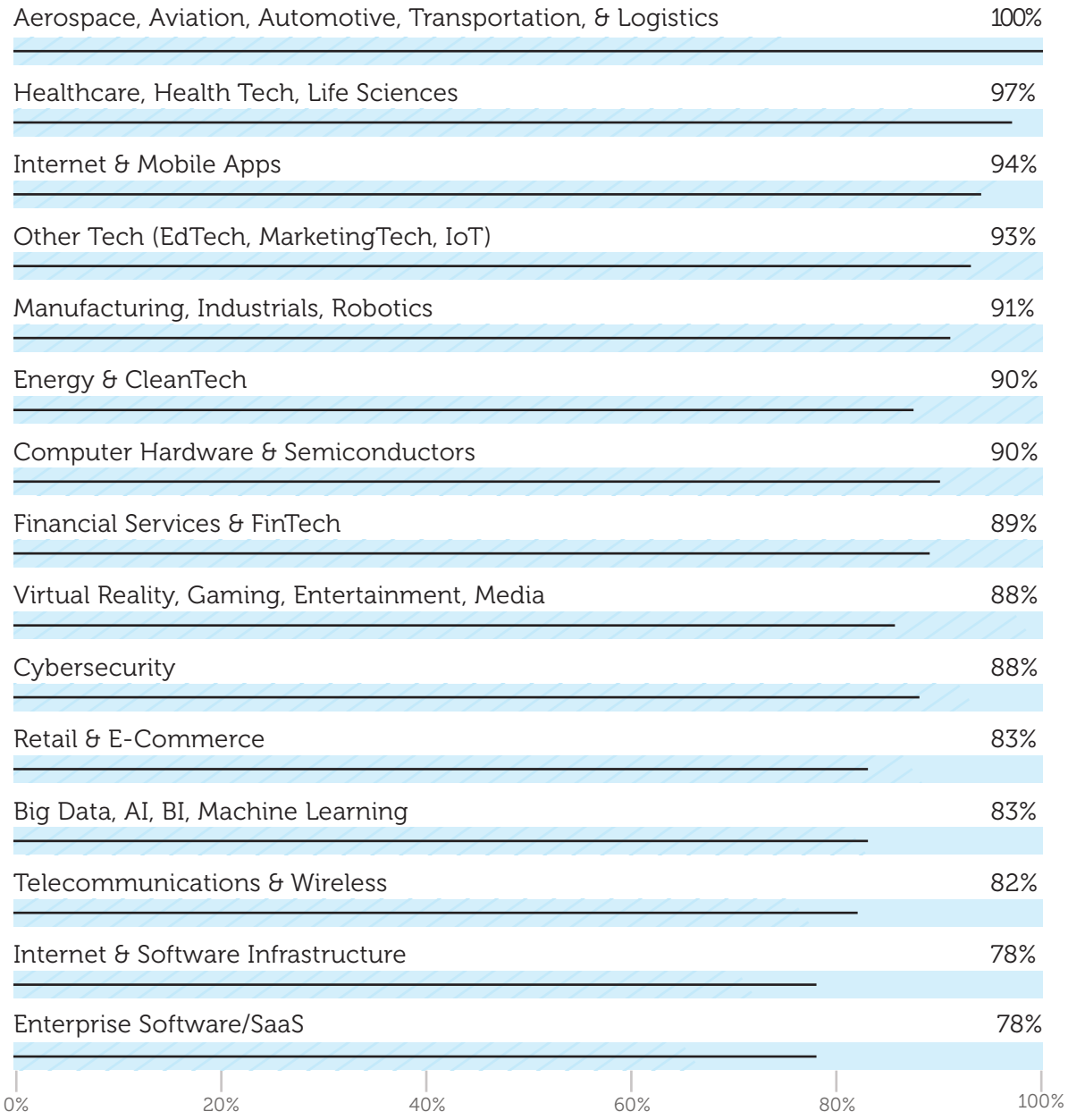
Most open source components are governed by one of about 2,500 known open source licenses, and the license obligations can be tracked and managed if the components themselves are identified. However, components with no identifiable license terms are problematic. Software that does not have a license generally means no one has permission from the creator(s) of the software to use, modify, or share the software. Creative work (which includes code), is under exclusive copyright by default. Unless a license specifies otherwise, nobody else can use, copy, distribute, or modify that work without being at risk of litigation. Lack of clear statements of rights and obligations leaves teams at greater risk of violation of "hidden" terms. As one open source-savvy attorney put it, "At least with the GPL, you know what you are dealing with."

The audits designate a license as "unknown" when the component is made publicly available but with no clear grant of license or terms of use. Not all teams publishing free software assign a license to their project, and GitHub, the most popular source of open source on the Internet, introduced the ability to attach a license to a project only three years ago. In addition, example code commonly available from Stack Overflow and other developer forums can also be a common source of software components' "unknown" licenses. While only about one percent of components analyzed had no clear license, these components were found in 53% of applications scanned.

INDUSTRIES MANAGE LICENSE RISK DIFFERENTLY

As with security risks, we found license conflicts in applications in every industry. However, the prevalence of those conflicts varied widely. There was no correlation between security and license risk levels. Industries such as Internet & Software Infrastructure and Retail & E-Commerce showed a low rate of license compliance issues versus security risks, while nearly 100% of audited applications in the Aerospace, Transportation, & Logistics industry contained license challenges.

PERCENT OF APPS WITH LICENSE CONFLICTS BY INDUSTRY



OPEN SOURCE DOMINATES application development for good reasons. Open source decreases development costs while accelerating time to market and increasing feature functionality. Organizations in every market and across every industry vertical build applications using open source as their foundations. For nearly every aspect of software development, the question is no longer "Should we consider open source?" but rather "Why wouldn't we?".

OVER HALF of vulnerabilities found in apps were ranked as **"HIGH" SEVERITY**

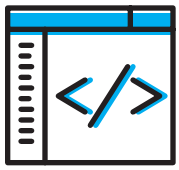
However, open source risks accompany the benefits, particularly when organizations do not sufficiently track and manage the open source in use. As this COSRI analysis shows, known vulnerabilities in open source are particularly attractive to attackers. These vulnerabilities (and often their exploits) are publicly disclosed, and users are often completely unaware of their use of the components themselves, much less the vulnerabilities, or the potential updates they might use to mitigate their risks. Likewise, this lack of visibility exposes organizations to potential litigation and IP loss risks from license compliance violations.

A PATH TO PROGRESS

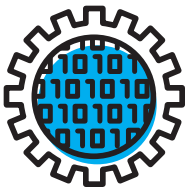
The results of this COSRI analysis clearly demonstrate that organizations have a long way to go in managing their open source. To make progress in defending against open source security threats and compliance risks, organizations must adopt open source management practices that:

FULLY INVENTORY OPEN SOURCE SOFTWARE: Organizations cannot defend against threats that they do not know exist. A full and accurate inventory (bill of materials) of the open source used in their applications is essential.

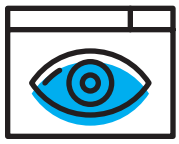
MAP OPEN SOURCE TO KNOWN SECURITY VULNERABILITIES: Public sources, such as the National Vulnerability Database provide information on publicly disclosed vulnerabilities in open source software. Organizations need to reference these sources to identify which of the open source components they use are vulnerable.



IDENTIFY LICENSE AND QUALITY RISKS: Failure to comply with open source licenses can put organizations at significant risk of litigation and compromise of IP. Likewise, use of out-of-date or poor quality components degrades the quality of applications that use them. These risks also need to be tracked and managed.



ENFORCE OPEN SOURCE RISK POLICIES: Many organizations lack even basic documentation and enforcement of open source policies that would help them mitigate risks. Manual policy reviews are a minimum requirement, but as software development becomes more automated so too must management of open source policies.



MONITOR FOR NEW SECURITY THREATS: With more than 3,500 new open source vulnerabilities discovered every year, the job of tracking vulnerabilities does not end when applications leave development. Organizations need to continuously monitor for new threats as long as their applications remain in service.

As open source use continues to increase, effective management of security and license compliance risk is increasingly important. By integrating processes and automated solutions such as Black Duck's into their SDLC, organizations can maximize the benefits of open source while effectively managing their risks.

ABOUT BLACK DUCK SOFTWARE

Organizations worldwide use Black Duck Software's industry-leading products to automate the processes of securing and managing open source software, eliminating the pain related to security vulnerabilities, license compliance and operational risk. Black Duck is headquartered in Burlington, MA, and has offices in San Jose, CA, Vancouver, London, Belfast, Frankfurt, Hong Kong, Tokyo, Seoul and Beijing. For more information, visit www.blackducksoftware.com.

CONTACT

To learn more, please contact: sales@blackducksoftware.com or +1 781.891.5100
Additional information is available at: www.blackducksoftware.com

BLACKDUCK

blackducksoftware.com