synopsys®

DEVSECOPS PRACTICES AND OPEN SOURCE MANAGEMENT IN 2020

A SURVEY OF 1,500 IT PROFESSIONALS

TABLE OF CONTENTS

INTRODUCTION

In August 2020, the Synopsys Cybersecurity Research Center (CyRC) and Censuswide, an international market research consultancy, conducted a survey of 1,500 IT professionals with DevSecOps as part of their role and who work in cyber security, software development, software engineering, and web development.

This survey reports on the tools organizations in the business of building software are employing to **integrate open source management** into their **DevOps practice.**

The group was recruited to take part in an online survey focused on DevSecOps practices and open source use. Participants came from the United States, the United Kingdom, Finland, Germany, China, Singapore, and Japan, with at least 50 respondents from each country. The survey is part of CyRC's ongoing research into cyber security practices and is intended as a complement to Synopsys' annual Open Source Security and Risk Analysis (OSSRA) report.

As the 2020 OSSRA report[1] details, almost 100% of the 1,200+ audited codebases in that report contained open source components or libraries, with open source making up 70% of the codebases themselves. Gartner's report, "Market Guide for Software Composition Analysis,"[2] relates that due to the prevalence of open source in modern software development, corporate interest in software composition analysis (SCA) tools used to manage open source is growing rapidly, with inquiries to the analyst firm on the topic increasing nearly 40% from 2019 to 2020.

While the OSSRA report provides an in-depth snapshot of the current state of open source security, compliance, and code quality risk, this survey reports on the tools organizations in the business of building software are employing to integrate open source management into their DevOps practice. The survey also explores the strategies being used to address open source license compliance, vulnerability management, and the growing issue of legacy open source in commercial code.

Section 1 of this report details the highlights of these survey findings, and Section 2 includes full survey results.

Section 1

SURVEY HIGHLIGHTS

DevOps and the secure SDLC

One of CyRC's areas of interest in conducting this survey was to investigate the prevalence of DevSecOps—the practice of integrating security into every stage of the DevOps pipeline—across industry verticals and throughout organizations around the world. While we expected to find evidence of a DevSecOps trend, our results point to a more mature adoption of DevSecOps among respondents than anticipated.

Thirty-three percent of the respondents noted that their organizations are well on their way to a mature deployment of DevSecOps. An additional 30% reported that they are making measurable strides toward maturity. With a combined 63% of respondents reporting that they are incorporating some measure of DevSecOps activities into their software development pipelines, it's safe to say that adoption of the DevSecOps methodology is an important, rapidly growing trend.

How mature is the adoption of DevSecOps practices within your team?

- ● Mature or deployed widely within our business
- ● Limited to use within specific projects, but expanding
- ● We're still researching how to apply it
- ● We aren't investigating DevSecOps practices at this time
- ● Immature or running pilot programs
- ● Not sure

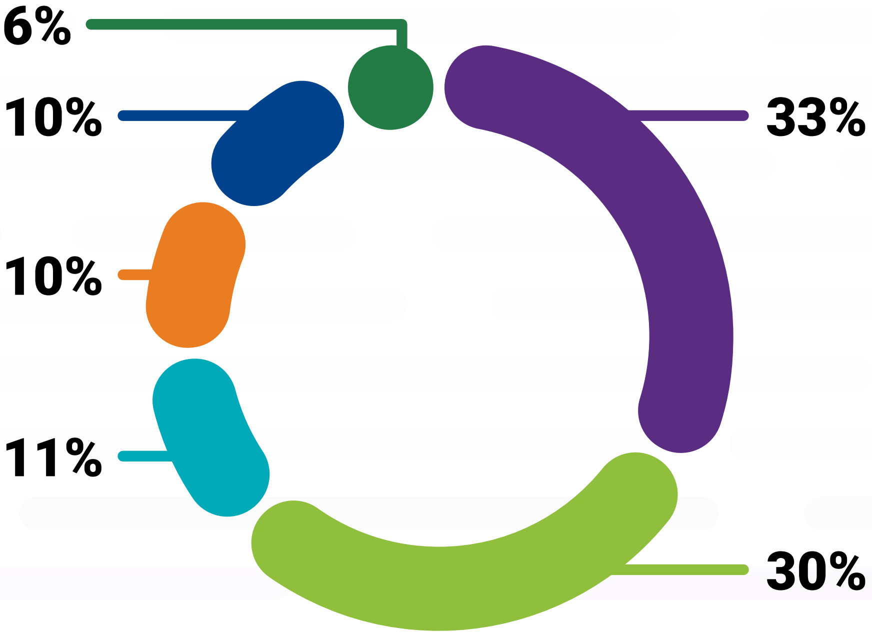


Figure 1

Synopsys' 2020 Building Security in Maturity Model (BSIMM) report,[3] which looks at the current state of application security across a large number of organizations, notes that "the idea of baking security into all phases of a DevOps life cycle is quickly becoming the norm. But organizations are adopting this approach in their own ways and at their own pace."

Tellingly, 42% of respondents to our survey have a dedicated security team. Also known as a software security group (SSG), this team's responsibility may include acquiring, creating, deploying, and managing secure software. Having an SSG is another indicator of maturity in an organization's software security practices, according to the Synopsys BSIMM report as well as other benchmarking tools for software security initiatives.

Who, if anyone, is responsible for application security in your organization?

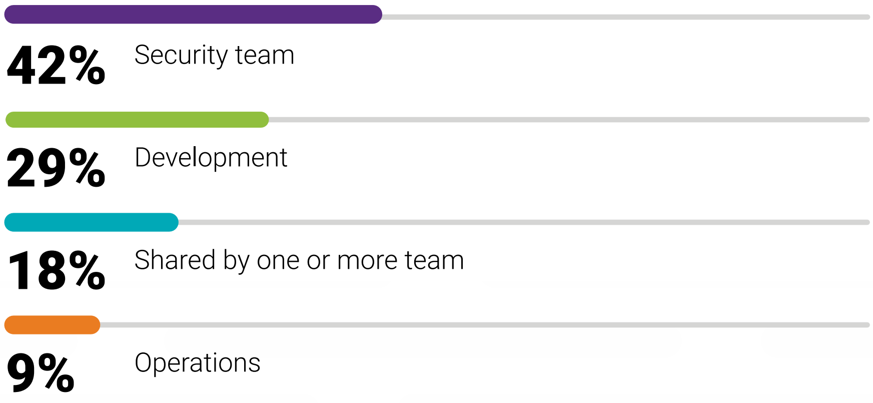


Figure 2

42% of respondents have a **dedicated security team** whose responsibility may include **acquiring, creating, deploying, and managing secure software.**

DevSecOps tools

Figure 3 indicates that many organizations in the business of building software need to increase their investments in software composition analysis (SCA) and interactive application security testing (IAST), which enable security automation at the development phase. Successful DevSecOps strategies entail a full security toolset—including dynamic application security testing (DAST), IAST, static analysis security testing (SAST), and SCA tools—to fully address code quality and security flaws early in the software development life cycle (SDLC).

SCA tools are employed by 38% of the respondent organizations. SCA products analyze applications throughout the entire SDLC to detect open source software. These tools typically produce an inventory, or Bill of Materials (BOM), of all open source in the codebase, the versions of that open source, the download locations for each project and all dependencies, the libraries the code calls to, and the libraries those dependencies themselves link to.

SCA tools also advise of known open source vulnerabilities found in the code, available security patches, and the license(s) used to distribute the respective open source packages. Comprehensive SCA solutions also monitor the BOM to provide customers with early notification of new vulnerabilities, and even deliver upgrade/patch guidance.

Of note is that the tool with the highest adoption rate is still only utilized by 45% of respondents, indicating that there is no universally adopted application security tool.

Which, if any, of the following security tools does your team currently use?

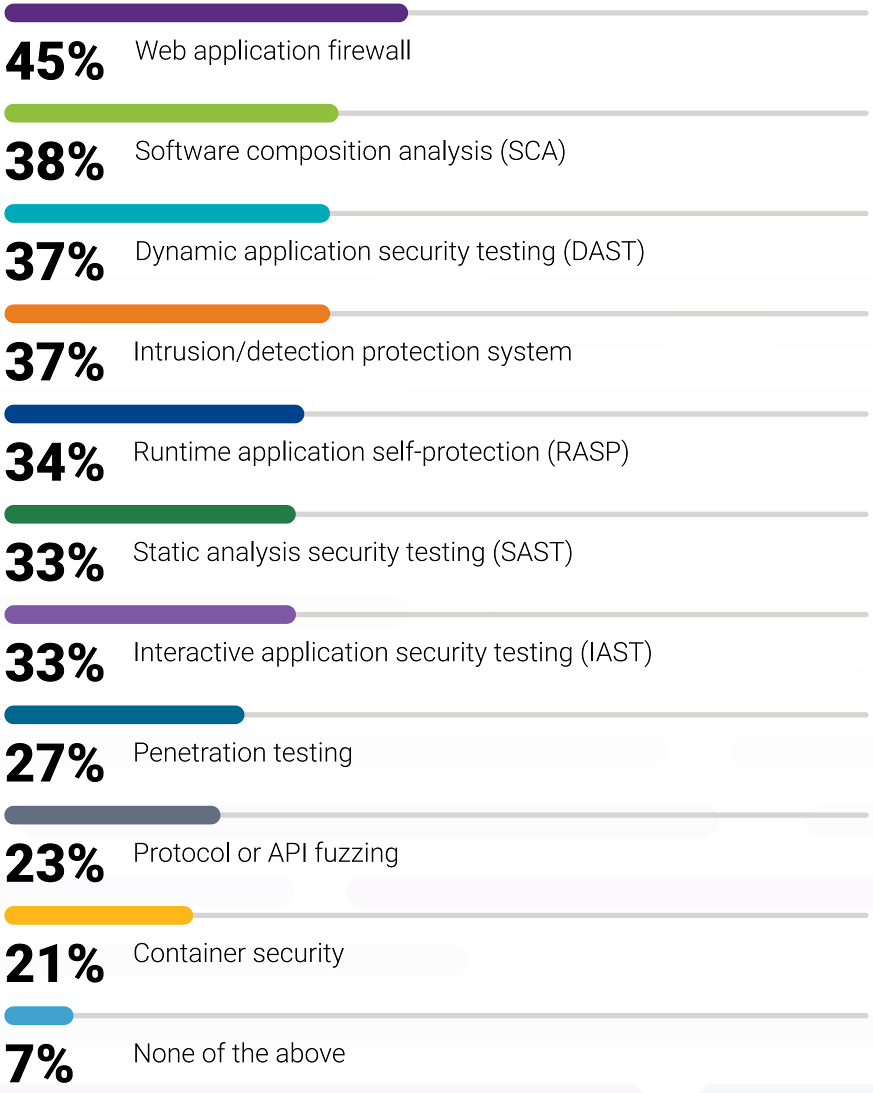


Figure 3

Open source selection and governance

The survey results indicate that most respondents' organizations are at a relatively high level of maturity when it comes to how they select open source and how they ensure that their policies on open source use are being followed. An overwhelming percentage—72%—have a published policy on open source use. The majority of those policies define acceptable open source licenses; 55% prescribe patching/updating requirements, and close to half define open source components that are acceptable for use.

An interesting question arises when correlating these results and the results shown in Figure 3. While only 38% of respondent companies use an SCA tool, 72% have a published policy for open source use.

Without an automated tool, are 34% of the respondents' companies employing manual processes to manage open source? Are they depending on a developer honor system that policies are being followed? The survey results indicate that—as Gartner's 2020 report "Market Guide for Software Composition Analysis" also relates—corporate adoption of SCA tools is still at a relatively early stage. On the other hand, Gartner also relates that interest in SCA is growing rapidly, with inquiries to the analyst firm on the topic increasing nearly 40% from 2019 to 2020.

Does your organization have a published policy for open source use?

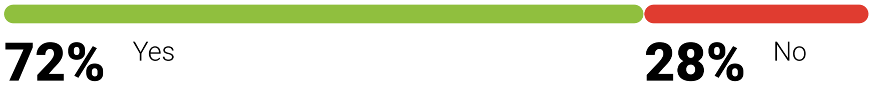


Figure 4

Which, if any, of the following requirements are true for your policy on open source use? (Select all that apply)

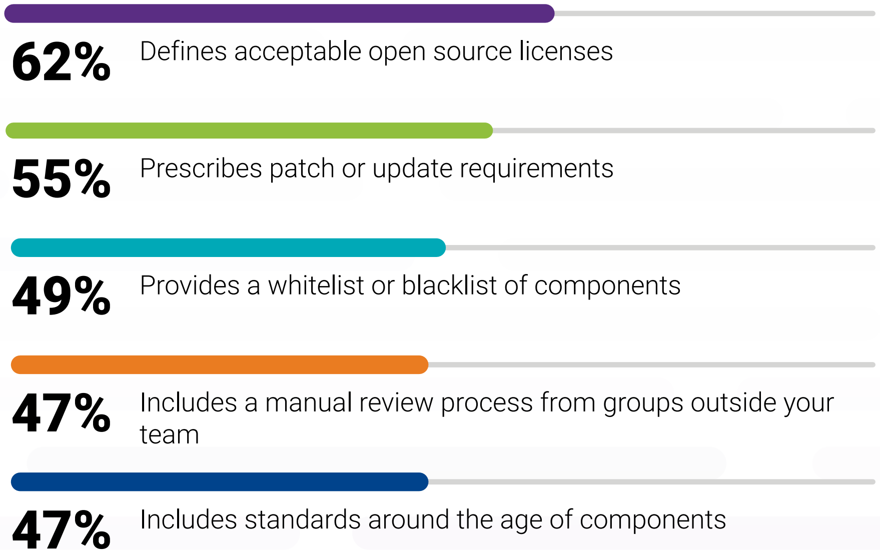


Figure 5

A large majority (64%) of the respondents' organizations also have a specific individual or board of governors charged with open source oversight and whose responsibilities include developing open source governance processes, setting use policies, and defining acceptable open source components for organizational use (see Section 2—survey question, *Which, if any, of the following are true for the governance board/ individual?*).

Security and a component's vulnerability to exploit was top-of-mind to 50% of respondents, and the #1 selection criterion when vetting a new open source component. Completeness of a component's implementation, frequency of releases/ patches, license restrictions, and community viability were also cited as important considerations in the decision process.

Forty-four percent of respondents noted that their team's familiarity or involvement with the component's development or with its community (28%) were important factors in their decision. Both are welcome percentages to see, as one of the factors leading to open source's successful adoption is the volunteer communities improving and updating code.

Do you have an open source governance board or specific individual charged with open source governance?

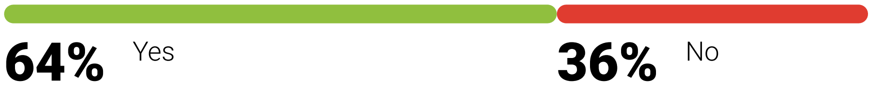


Figure 6

What criteria is used in the vetting process for a new open source component? (Select all that apply)

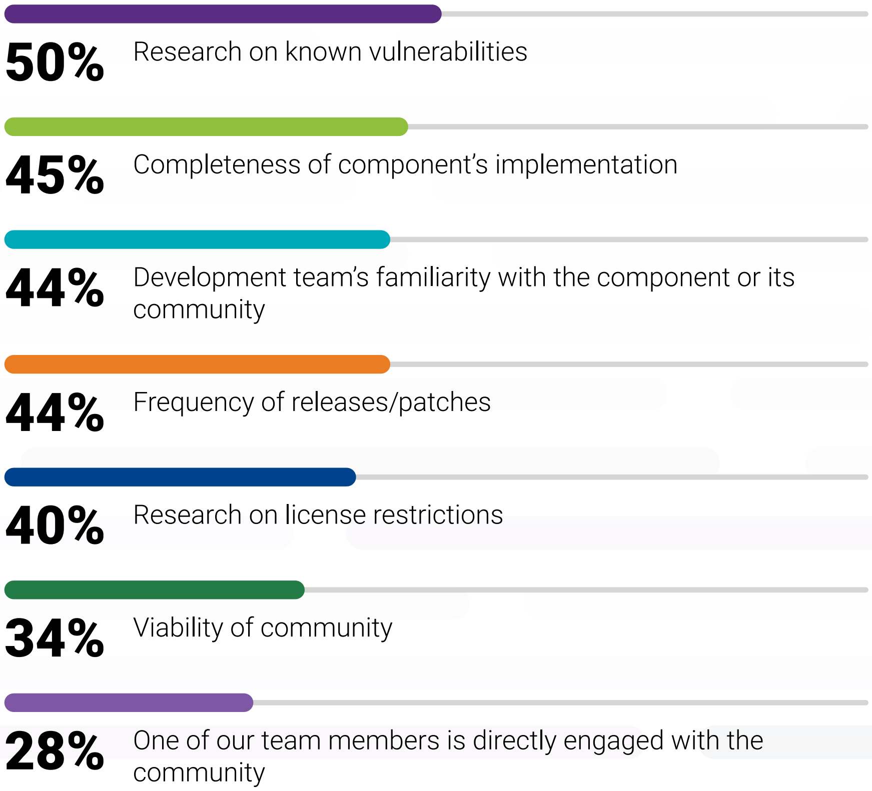


Figure 7

Open source security and patching

Somewhat troubling were the results in the respondents' answers to the question of patching. Over half—51%—say it takes 2 to 3 weeks for their organization to apply an open source patch, with 24% noting that it can take up to a month, even when the patch addresses a critical issue.

Parsing the results by country, it appears that the United States is being most heavily impacted by unpatched open source components. Over half of respondent organizations in the U.S. have had their software delivery schedule affected in the past year because of addressing a critical open source patch, compared to 40% globally.

Based on the survey results, many organizations—especially in the U.S.—would be well-advised to explore accelerating their time-to-patch schedules. When a vulnerability is disclosed, you're in a race with attackers. For example, in March 2017, a critical vulnerability in the Apache Struts open source framework was publicly disclosed. Security researchers observed a high number of exploitation attempts almost immediately after disclosure. In fact, on the same day as the disclosure, information about how to exploit the Apache Struts flaw was posted to several websites popular with hackers.

Thousands of organizations were attacked, and even though many applied the patch to their systems immediately, the attacks kept coming. All it took to create the conditions for a breach was for one department at one firm to miss patching a version of Struts containing the vulnerability—and that breach happened at a company called Equifax.

When a critical open source vulnerability is identified by your organization, on average, how long does it take for your teams to provide a fix?

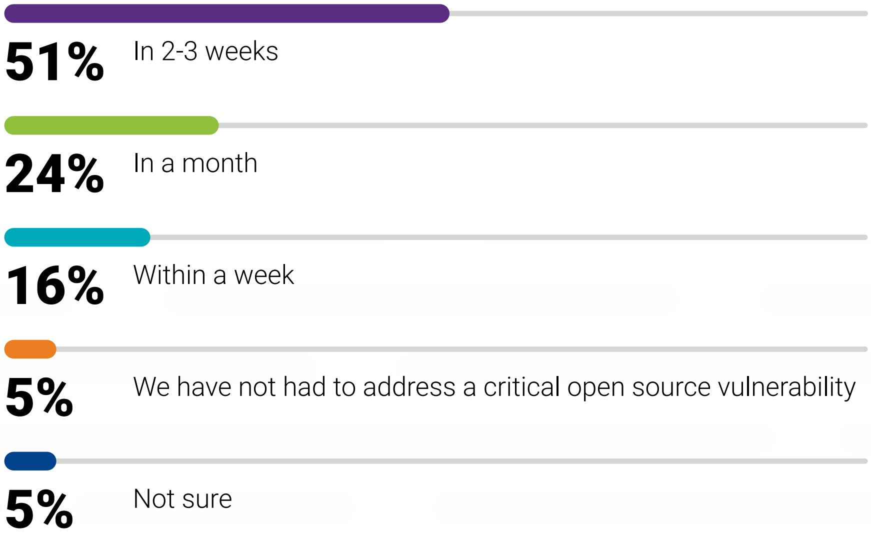


Figure 8

Has addressing a critical open source patch impacted your software delivery schedule within the past year?

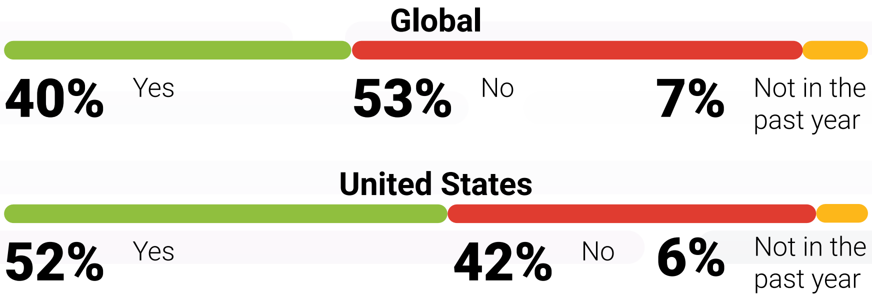


Figure 9

The media comes under regular—sometimes deserved—criticism from the open source community for exaggerating security incidents and the risk of open source use. However, most coverage notes that the risk comes from the *unmanaged* use of open source, with reported incidents usually involving unpatched or outdated components, or the lack of an up-to-date software inventory, one of the primary causes behind the Equifax data breach.[4]

Our survey results demonstrate that media coverage of open source issues definitely affects how organizations manage their open source use. Forty-six percent of respondents noted that media coverage had prompted their organization to apply more stringent controls on open source usage.

Has media coverage of an open source issue ever caused your organization to do any of the following? (Select all that apply)

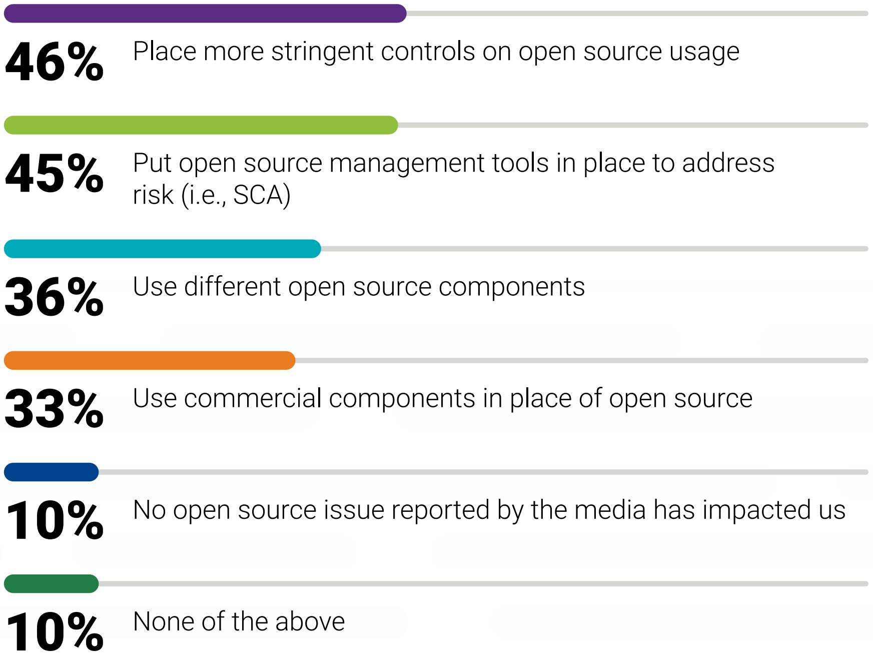


Figure 10

46% of respondents noted that coverage of **open source issues definitely has an effect** on how their organizations **manage open source.**

Open source project sustainability

As shown in the highlighted section from Figure 5, 47% of respondents' organizations define standards around the age of open source components, an acknowledgement of a growing problem in the open source community—project sustainability.

There's no guarantee that the people behind any given open source project will continue maintaining the code. In fact, of the 1,200+ codebases examined for the 2020 OSSRA report, 88% contained open source components that had no development activity in the last two years.[5]

All software ages. As it ages, it loses support. With open source, the number of developers working to ensure updates—including feature improvements, as well as security and stability updates—decreases over time. The component becomes more likely to break without the support needed to provide fixes. At some point, as that open source component ages and the number of people who handle bug reports and code reviews diminishes, the component becomes increasingly likely to open a codebase to exploit.

Without policies in place to identify and manage the risks that legacy open source can create, organizations open themselves up to the possibility of issues in their software.

Which, if any, of the following requirements are true for your policy on open source use?

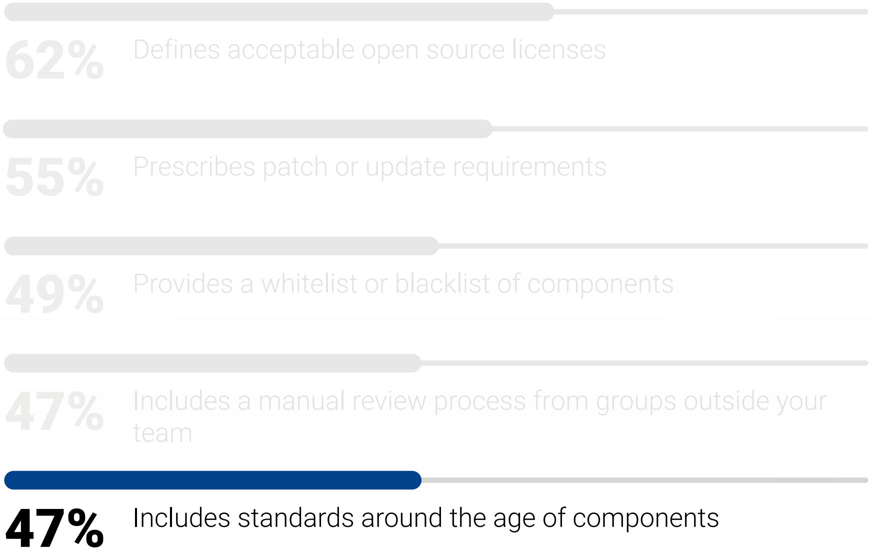


Figure 5

Conclusion: Developing security in depth for the SDLC

Organizations are producing and deploying software applications faster than ever before. Ensuring that developers are on board with security practices is even more critical to improving their efficiency. The Forrester report, "The State of Application Security, 2020" notes, "To meet developer needs, security pros must integrate application security testing tools into the CI/CD pipeline and enable scans to run automatically on check-in, build, and integration while also enabling autoremediation to make mitigating security flaws quick and painless."

> "Firms **must move faster** at pushing prerelease testing **earlier in the SDLC.**"
>
> —Forrester report, "The State of Application Security, 2020"[6]

Taking the steps toward a more secure SDLC

If the findings in this report mirror your organization's current software development environment and you're looking toward a more secure SDLC, an objective analysis of the maturity level of your software security efforts can be an invaluable first step.

A benchmarking tool such as the Building Security in Maturity Model (BSIMM) report is designed to help CISOs and security leaders objectively measure existing security practices and identify areas for improvement within their organization. The BSIMM gives an objective, data-driven view into your current software security initiative and includes key areas we've reviewed in this report, such as patching third-party components used in software development, test automation, and the impact of DevSecOps practices.

Armed with the results of an independent assessment of your current practices, you can develop a strategy to improve your security practices in a structured and cost-efficient manner. If you're looking for a software security roadmap, the Synopsys Managed Services team has developed a series of Maturity Action Plans (MAPs) that can help chart a path to goals ranging from developing an overall DevSecOps culture to implementing more-targeted activities such as cloud security.

Avoiding application security testing pain

As the responses to the DevSecOps tools question indicate, there is no shortage of application security tools and techniques. But for many teams, each tool represents a pain point within their development workflow—and that can slow development efforts.

To ease some of that pain, vendors have focused on integrating their tools within CI/CD pipelines. While this can help with tool deployment, it doesn't really address the pain felt by development teams. With over 50% of U.S. respondents and 40% of respondents worldwide indicating that addressing an open source vulnerability impacted their delivery schedules, it's clear that unpatched vulnerabilities are a major source of developer pain. But so too are tools that slow down or generate additional work for development teams.

Contextual information is key to addressing security pain. For example, if a developer can see in their IDE that a feature won't pass a security policy check, they can adjust the implementation to meet the policy. Empowering developers in this way is one objective of the Synopsys Code Sight™ IDE plugin. Code Sight provides both SAST and SCA information to developers, so they can find and fix open source and proprietary security defects as they code. And it does so in manner familiar to anyone who has used an IDE syntax checker.

Context is also an attribute of runtime validation technologies. Using Seeker®, Synopsys' IAST tool, to test an application provides actionable value to a security-minded development team. Armed with data flows and call graphs targeting the vulnerability, development teams can quickly remediate an issue—and Seeker automatically confirms that the issue has been resolved.

Know your risks

It's important to remember that security practices employed by an open source development team are likely different than those of an internal team creating custom code. Open source security updates probably won't be tied to the same sprint cycle or release interval used by a proprietary coding team. Default settings for a given open source component may not align to an organization's security policies.

Identifying risks associated with open source requires an accurate understanding of all the open source used in an application, including the open source embedded within commercial libraries. This is the role that SCA tools like Synopsys Black Duck® provide—starting with a comprehensive software Bill of Materials (BOM). An accurate, up-to-date software BOM of open source components allows you to pinpoint vulnerable components quickly and prioritize remediation efforts appropriately.

As noted in the survey results, many organizations—especially those in the United States—should accelerate their time-to-patch schedules. Over half say that it currently takes two to three weeks to apply a patch. A production-grade SCA solution is the key to reducing patching timelines from weeks to days by providing continuous monitoring for new vulnerabilities and giving guidance on mitigation.

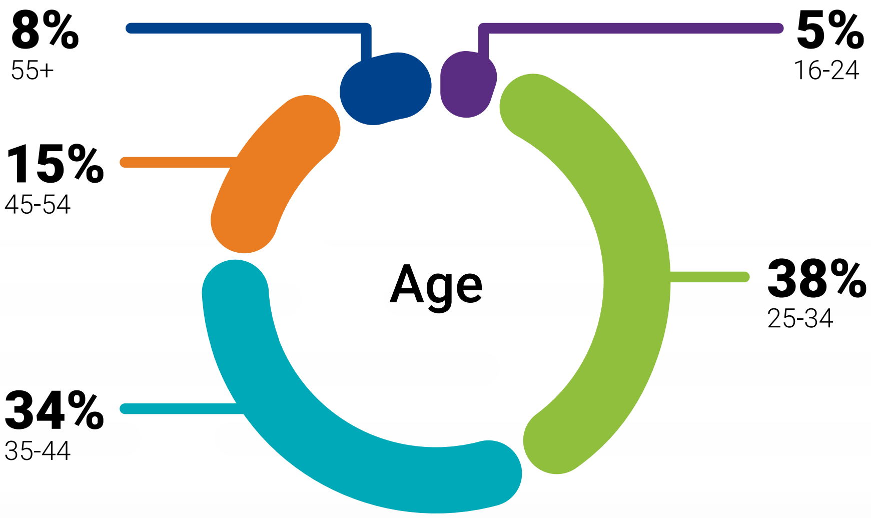
References

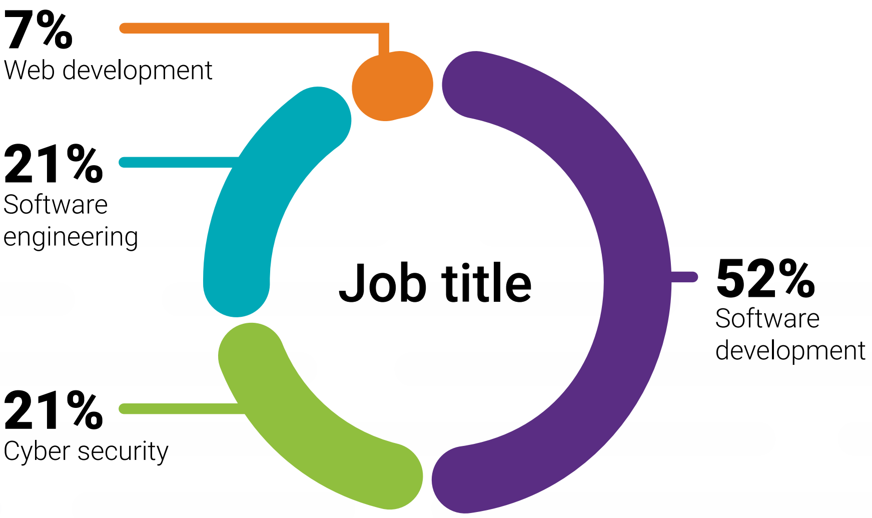
1. Synopsys Cybersecurity Research Center, "2020 Open Source Security and Risk Analysis Report," Synopsys, May 2020.
2. Dale Gardner, "Market Guide for Software Composition Analysis," Gartner, August 2020.
3. Sammy Migues, John Steven, and Mike Ware, "Building Security in Maturity Model," Synopsys, 2020.
4. Permanent Subcommittee on Investigations, How Equifax Neglected Cybersecurity and Suffered a Devastating Data Breach, Committee on Homeland Security and Governmental Affairs, U.S. Senate, accessed September 12, 2020.
5. Synopsys Cybersecurity Research Center, "2020 Open Source Security and Risk Analysis Report," Synopsys, May 2020.
6. Sandy Carielli, "The State Of Application Security, 2020," Forrester, May 2020.

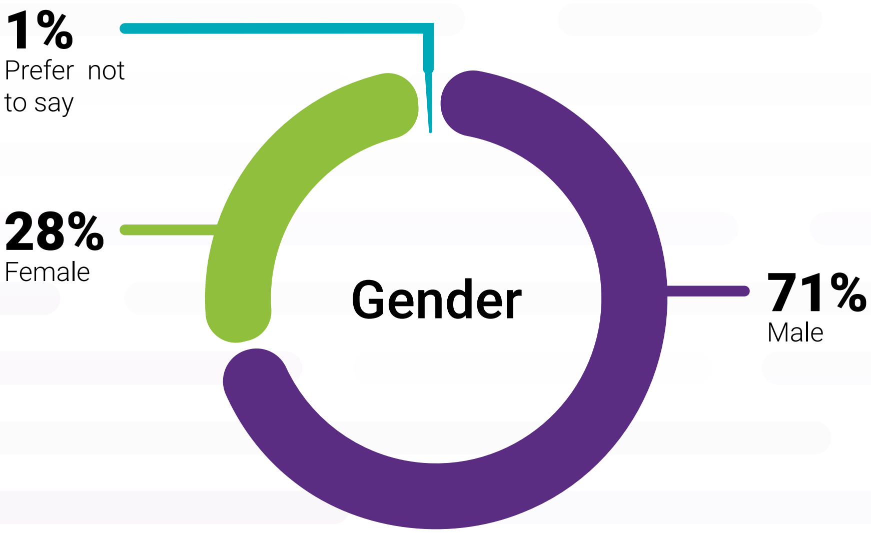
Section 2

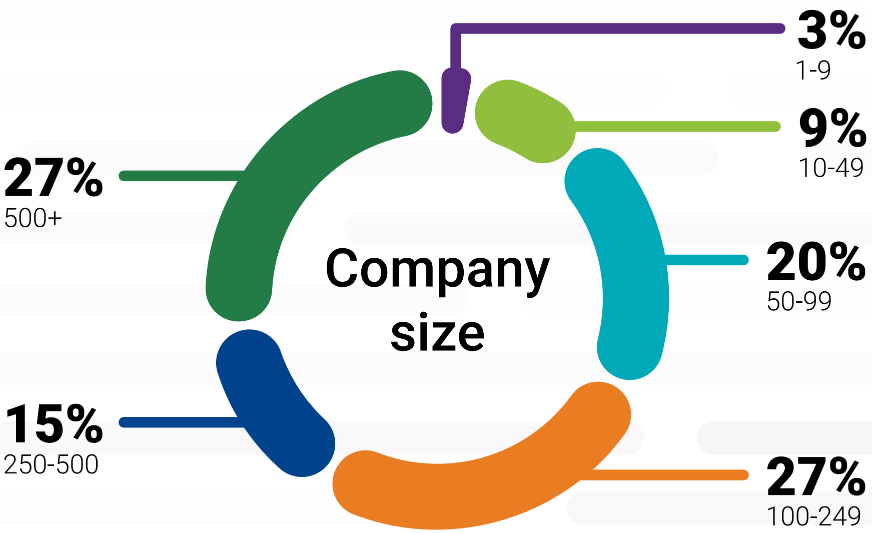
FULL SURVEY RESULTS

Respondent demographics

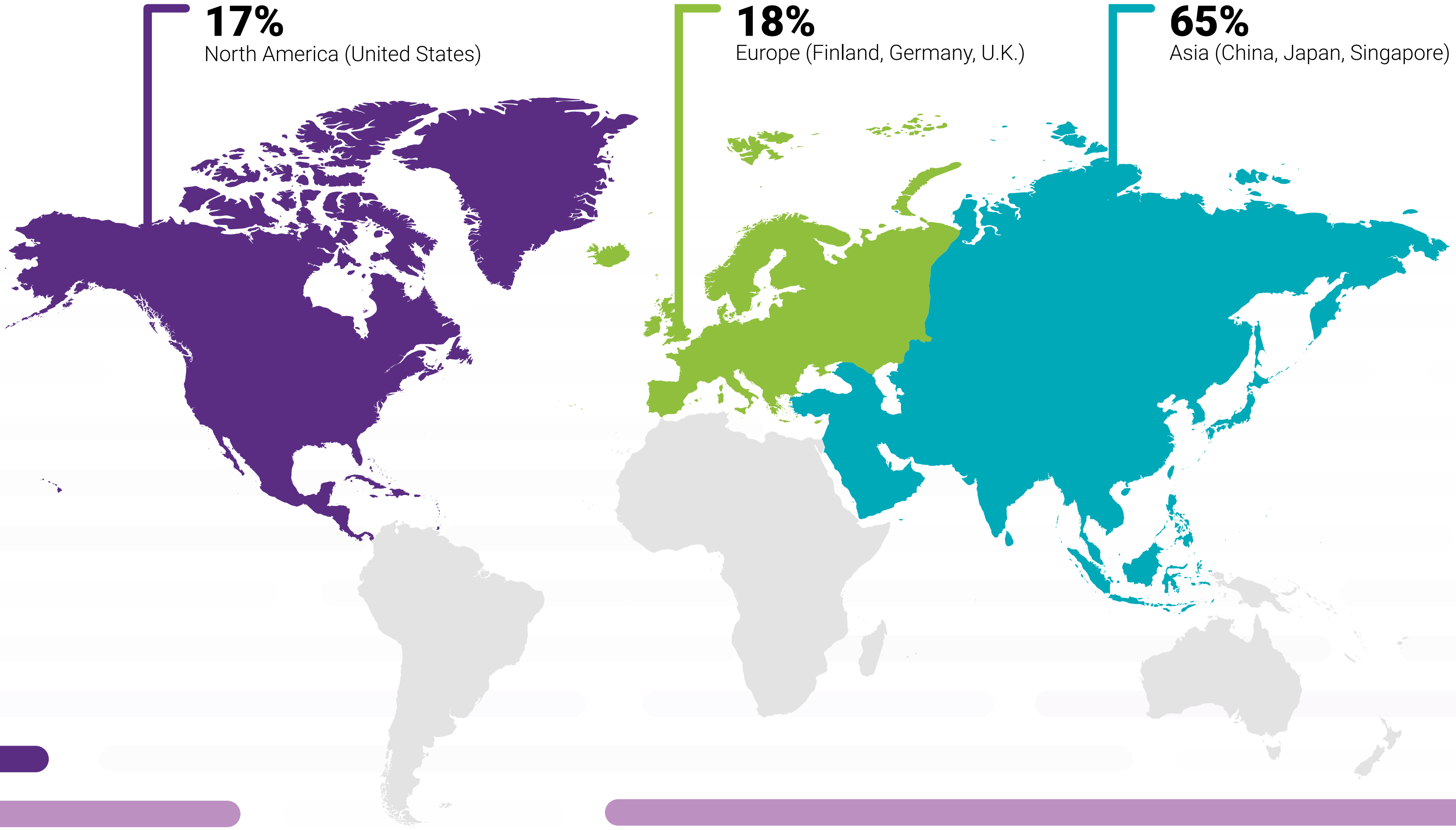








Respondents geographically


17%
North America (United States)
18%
Europe (Finland, Germany, U.K.)
65%
Asia (China, Japan, Singapore)

Questions

General

What types of applications does your team usually create or manage? (Select all that apply)

Web services	51%
Mobile applications	48%
Software libraries provided to third parties	44%
Packaged commercial software	42%
Firmware for embedded systems such as IoT, medical, or automotive	38%
Other (please describe)	4%

How mature is the adoption of DevSecOps practices within your team?

Mature or deployed widely within our business	33%
Limited to use within specific projects, but expanding	30%
We're still researching how to apply it	11%
We aren't investigating DevSecOps practices at this time	10%
Immature or running pilot programs	10%
Not sure	6%

When selecting a new component or application, where does your organization usually look? (Please select best match)

Mixture of both commercial and open source vendors	34%
Open source code repository (e.g., GitHub)	22%
Commercial software vendors	20%
Vendor-supported open source component (e.g., Red Hat)	19%
We never select a new component or application	5%
Other, please specify	0.60%

In your opinion, where does open source risk rank in your organization compared to other AppSec risks (e.g., proprietary code defects)?

Equal to	51%
Higher	29%
Lower	14%
Not sure	5%
Not applicable	2%

Security and patching

Who, if anyone, is responsible for application security in your organization?

Security team	42%
Development	29%
Shared by one or more team	18%
Operations	9%
No one	1%
Other, please specify	0.20%

How is your team primarily informed of vulnerabilities or weaknesses in the applications or services you create or manage?

Alerts from security teams	27%
Notification from security tools	25%
Periodic audits	14%
Customer support	10%
Diligence review during procurement	9%
Our team is not informed in any specific way about vulnerabilities or weaknesses in our applications or services	6%
External security researchers or bug bounties	5%
Media coverage	4%
Other, please specify	0.27%

Which, if any, of the following security tools does your team currently use? (Select all that apply)

Web application firewall	45%
Software composition analysis (SCA)	38%
Dynamic application security testing (DAST)	37%
Intrusion/detection protection system	37%
Runtime application self-protection (RASP)	34%
Static analysis security testing (SAST)	33%
Interactive application security testing (IAST)	33%
Penetration testing	27%
Protocol or API fuzzing	23%
Container security	21%
None of the above	7%

What, if anything, is your team's primary source for open source component security information?

Software composition analysis (SCA) tool	33%
Corporate software asset management tools	16%
Internet (forums, mailing lists, etc.)	13%
NVD (or country-specific equivalent)	11%
Package management tool	10%
Threat intelligence feeds	8%
News media	4%
We don't have a primary source of information for open source component security information	4%
Other, please specify	0.44%

When an open source component's patch is issued, on average, how quickly does your organization apply the patch?

Within a week, please specify in days	13%
In 2–3 weeks	53%
In a month	23%
Longer than a month	0.80%
Not sure	7%
We have no open source component patching policy	3%

Has an unpatched open source component resulted in a security incident within your organization within the past year?

Yes	29%
No/Unknown	65%
Prefer not to say	6%

Has addressing a critical open source patch impacted your software delivery schedule within the past year?

Yes	40%
No	53%
We haven’t had to address a critical open source patch within the past year	7%

Has media coverage of an open source issue ever caused your organization to do any of the following? (Select all that apply)

Place more stringent controls on open source usage	46%
Put open source management tools in place to address risk (i.e., SCA)	45%
Use different open source components	36%
Use commercial components in place of open source	33%
No open source issue reported by the media has impacted us	10%
None of the above	10%

When a critical open source vulnerability is identified by your organization, on average, how long does it take for your teams to provide a fix?

Within a week	16%
In 2–3 weeks	51%
In a month	24%
Longer than a month, please specify in months	0.29%
We have not had to address a critical open source vulnerability	5%
Not sure	5%

In your opinion, is the patching process for open source components faster or slower than applying patches of the commercial software you use?

Faster	37%
Slower	24%
About the same	33%
Not sure	6%

In your opinion, where in the software development life cycle (SDLC) is open source vulnerability management best suited?

Development	42%
Testing	42%
Deployment	12%
Not sure	3%
Other, please specify	0.51%

Selection, use, and governance

How are open source components selected within your organization? (Select all that apply)

Developers can select open source components based on approved license types and their meeting security policies	51%
Developers must use preapproved components but can request that new components be added to approval lists	47%
Developers have the freedom to select any component providing it is currently patched and up to date	37%
Developers can select any open source component without restriction	27%
Not sure	4%

What criteria is used in the vetting process for a new open source component? (Select all that apply)

Research on known vulnerabilities	50%
Completeness of component's implementation	45%
Development team's familiarity with the component or its community	44%
Frequency of releases/patches	44%
Research on license restrictions	40%
Viability of community	34%
One of our team members is directly engaged with the community	28%
Our team doesn't usually vet open source components	4%
Other (please specify)	0.51%

Which team manages the approval process for a new open source component?

Note: The following are from respondents who answered "Yes" to "Developers must use preapproved components but can request that new components be added to approval lists."

Security team	45%
Development team	28%
Operations team	15%
Legal/Compliance team	11%
No particular team	0.77%

What is the primary method used by your team to track open source usage in your apps?

SCA tool	39%
Package managers	25%
Individual developer tracks (through spreadsheet or other method)	20%
N/A – We don't use open source components	9%
We don't track open source usage in our apps	8%
Other method, please specify	0.53%

Does your organization have a published policy for open source use?

Yes	72%
No	28%

Which, if any, of the following requirements are true for your policy on open source use? (Select all that apply)

Defines acceptable open source licenses	62%
Prescribes patch or update requirements	55%
Provides a whitelist or blacklist of components	49%
Includes a manual review process from groups outside your team	47%
Includes standards around the age of components	47%
None of the above	0.41%

Do you have an open source governance board or specific individual charged with open source governance?

Yes	64%
No	36%

Which, if any, of the following are true for the governance board/individual? (Select all that apply)

Develops open source management processes	60%
Sets policies and handles exceptions concerning open source usage	58%
Can whitelist specific open source components	49%
Provides training for the organization	46%
Can blacklist specific open source components	45%
None of the above	0.34%

Open source project contributions

Does your organization have a published policy for its developers to make open source contributions?

Yes	65%
No	35%

Which, if any, of the following is true for your open source contribution policy? (Select all that apply)

Requires internal review of all potential code contributions	59%
Allows developers to agree to contributor license agreements	55%
Requires registration of supported projects with HR or Legal	50%
Allows team members to support external users of the project	49%
None of the above	0.78%

License compliance

How important is open source license compliance to your organization?

Very important	45%
Somewhat important	48%
Not very important	6%
Not at all important	0.80%

Who has primary responsibility in your organization to verify license compliance?

Security team	38%
Development	28%
Legal	17%
Operations	14%
We don’t verify license compliance	1%
Other, please specify	0.44%

The Synopsys difference

Synopsys helps development teams build secure, high-quality software, minimizing risks while maximizing speed and productivity. Synopsys, a recognized leader in application security, provides static analysis, software composition analysis, and dynamic analysis solutions that enable teams to quickly find and fix vulnerabilities and defects in proprietary code, open source components, and application behavior. With a combination of industry-leading tools, services, and expertise, only Synopsys helps organizations optimize security and quality in DevSecOps and throughout the software development life cycle.

About CyRC

The Synopsys Cybersecurity Research Center (CyRC) works to accelerate access to information around the identification, severity, exploitation, mitigation, and defense against software vulnerabilities. Operating within the greater Synopsys mission of making the software that powers our lives safer and of the highest quality, CyRC helps increase awareness of issues by publishing research supporting strong cyber security practices.

For more information, go to www.synopsys.com/software.

Synopsys, Inc.
185 Berry Street, Suite 6500
San Francisco, CA 94107 USA

Contact us:
U.S. Sales: 800.873.8193
International Sales: +1 415.321.5237
Email: sig-info@synopsys.com