



**MEND**

Guide to  
**Open Source  
Software  
Security**

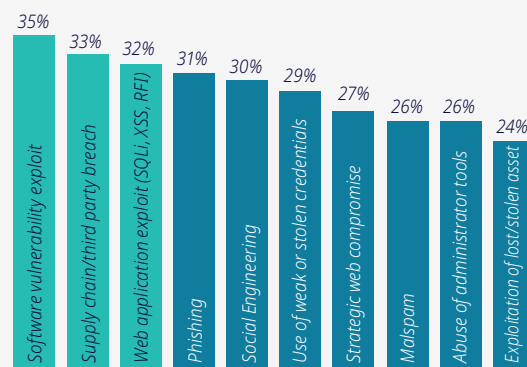
# The Current State of Application Security

As more and more enterprises pursue digital transformation projects, application development teams are increasingly relying on modern software development techniques such as Agile, DevOps, containers, and CI/CD pipelines to produce software at a faster pace. Along with the increased speed has come increased reliance on open source software. 99 percent of codebases now contain some amount of open source, and the total amount of open source in the codebase is now estimated to be 70 percent, up from 36 percent in 2015.

The changes mentioned above have had the beneficial effect of accelerating the pace of application development. Unfortunately, security has suffered.

Cyber attackers have shifted their techniques in order to leverage the changes listed above. They have increasingly focused their attacks at the application layer. Applications are now the most common way in, with software vulnerabilities at number one, supply chain attacks number two, and web application exploits at number three.

## How was the external attack carried out?



**Application attacks and supply chain attacks are the #1 and #2 ways that attackers penetrate enterprises.**

Source: Forrester Analytics Business Technographics Security Survey, 2021  
Base: 530 Security decision-makers with network, data center, app security, or security ops responsibilities who experienced an external attack when their company was breached

As a result, open source security, and in particular the application of software composition analysis (SCA), is now the number-one application security priority for many enterprises. Gartner analysts have recently stated the following:



**“Managing open-source software is the easiest and most impactful thing you can do to improve your application security program.”**

*Gartner®, “Managing Open-Source Software Risks in DevSecOps Environments”, Gartner Security and Risk Management Summit, June 2022.*



**“SCA is now considered a foundational element of application security testing to identify known vulnerabilities and potential supply chain risks in open-source packages and other artifacts. There is hardly any reason for not using SCA.”**

*Gartner, “Hype Cycle for Application Security, 2022, 11 July 2022*

Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings or other designation. Gartner research publications consist of the opinions of Gartner’s research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose. Gartner and Hype Cycle are registered trademarks and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally and are used herein with permission. All rights reserved.

## Desired State: Application Security Confidence

---

The desired outcome is for organizations to be confident in the security of their applications. Confidence is driven by a number of factors, including the following:

- Comprehensive rollout of open source security tools across all applications, ensuring continuous visibility to risks in 100 percent of applications
- Consistent usage by in-house developers across all applications and all languages
- The ability to respond to zero-day open source vulnerabilities (e.g. Log4j) within minutes or hours

An acceptable level of application risk measured using factors such as:

- The number of open source vulnerabilities in production
- The severity of each open source vulnerability (CVSS)
- The reachability of each open source vulnerability within the application
- Whether each open source vulnerability is exposed to attackers via the network
- The business value of the application, i.e. what could be lost if the application were to be compromised

Operational metrics are also very useful to track, such as:

- Mean time to remediation (MTTR)
- Percentage of applications that are routinely (e.g. at least once every two weeks) scanned for vulnerabilities
- The percentage of developers who routinely use a security tool
- The amount of labor required to operate the security tool, prioritize results, and remediate vulnerabilities

## What Is Holding Open Source Security Back?

---

Some enterprises do not understand the importance of open source security or having an SCA tool. They feel that a static scan of the application using a static application security test (SAST) tool is enough. This lack of understanding is caused by the fact that SCA is a relatively new type of application security tool. And this is why analysts at Gartner are doing all they can to make enterprises more aware of how important SCA is.

Other enterprises are wrestling with the inadequacies of first-generation SCA tools purchased when they first came to market a few years ago. The metrics from these early tools do not look great. Security teams still don't have a complete inventory of open source software running in production. CISOs are not confident in their team's ability to quickly identify and resolve all major security issues, such as Log4Shell in December 2021. Developers admit that they often push code to production without running a security scan, or without addressing the alerts generated by the security scan.

These program failures are usually driven by one or more of the following weaknesses in first-generation SCA tools:

### **Cumbersome integrations, slow roll-outs**

Most first-generation SCA tools take a long time to scale because of inherent weaknesses in the integration approach. The two most common integration points are the developers' individual development tools (IDEs) and the CI/CD pipeline. In the case of the IDE integration, the security team is completely at the mercy of individual developers to add the security software to their workflow, and there is no centralized policy enforcement or governance. In the case of the pipeline integration, every DevOps team needs to separately add the SCA product to each pipeline, which can take years to complete and is difficult to upgrade and maintain.

As a result of cumbersome integrations, the SCA tools are slow to scale, which means security teams don't have comprehensive awareness of open source risks.

### **Too many false positives**

First-generation SCA tools commonly alert on open source packages that are present in the application even if they are not actually reachable through the application, or not used in a way that exposes the vulnerability. Our research indicates that approximately 70 percent of all reported vulnerabilities fall into this category.

Consequently, application developers learn to tune out the security tool. They either don't run the scan, or they don't address the alerts generated by the security scan.

## Too little remediation automation

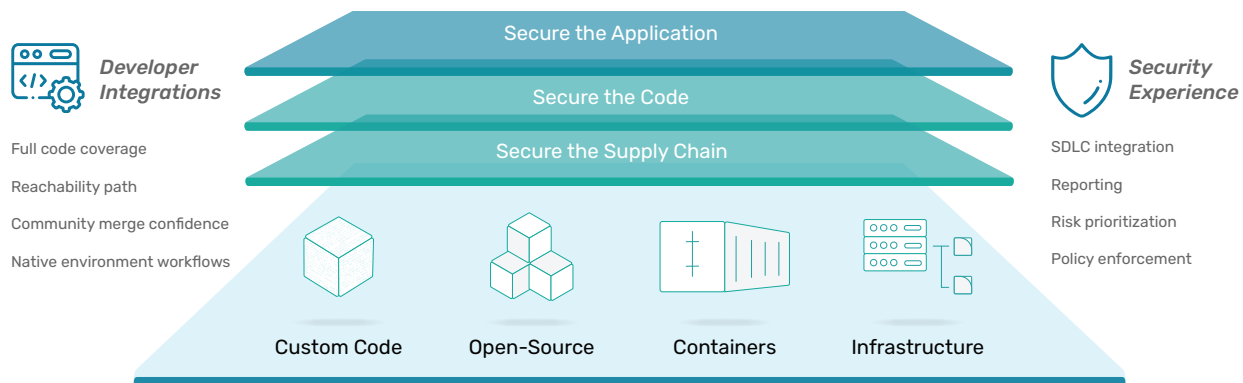
Traditional security tools put most of the responsibility on the developer to figure out how to remediate all the security issues. Some of the better SCA tools contain just-in-time security education for the developer, which is better than nothing, but it still puts too much burden on the developer to conduct his own research and figure out how to fix the vulnerability. This is especially true when deadlines are looming and development is occurring right up to the time when the code needs to be completed.

As a result of the time-consuming nature of remediation, application developers can either choose to slow their pace of application development (a bad outcome) or avoid using the security tool (another bad outcome). Some research indicates that 70 percent of development teams always or frequently skip security steps due to time pressures when completing projects.



## How Mend Solves the Problem

The Mend Application Security Platform is designed to meet the needs of modern application development teams at large enterprises. Simple to use and nearly invisible to developers, Mend lets you reduce your application security risk without affecting development deadlines.



The Mend Application Security Platform is designed for both developers and security professionals. Developers like the fact that Mend is so seamlessly integrated into their native development workflow that they never need to leave their familiar development environment. Remediation suggestions are completely automated, with all the information that developers need to make educated decisions. At the same time, security professionals appreciate the scalability, reporting, and centralized policy enforcement capabilities within the Mend platform.

The following are some highlights of the Mend platform.

### Automated dependency updates

Ensuring your dependencies are kept up to date is one of the easiest ways to keep software secure and reduce technical debt. Research shows that more than 90 percent of new vulnerabilities are fixed with security patches before the vulnerability is publicly disclosed. So by maintaining up-to-date dependencies, you avoid most vulnerabilities completely and avoid fire drills.

Mend automates the process of keeping dependencies up to date. Here is how it works:

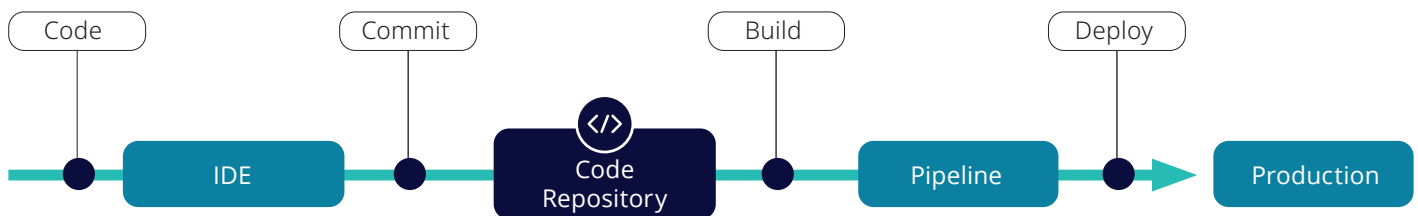
- Mend scans your repos to detect dependencies. We support over 65 different package managers.
- Mend checks to see if newer versions of your dependencies exist.
- Mend raises pull requests for available updates and shows crowdsourced merge confidence data. This is information gleaned from hundreds of thousands of Renovate users worldwide. Developers can then use this data to identify whether an update can be safely merged or whether it contains potential risk to the application. We call this feature "Merge Confidence". Nothing like it is available from any other SCA vendor, because no other vendor has anything comparable to the huge number of users that Renovate has.

Package	Change	Age	Adoption	Passing	Confidence
org.webjars:jquery (source)	3.5.1 -> 3.6.0	1y	40%	100%	very high

### Advanced repo integrations

Although Mend [supports integrations](#) across multiple points of the software development lifecycle (SDLC), the integration point that has proven to be a game-changer for many organizations is integration with the code repository. This integration point provides an ideal combination of capabilities:

- Developers obtain test results and automated remediation suggestions quickly, automatically, and early in the SDLC when it is easiest to fix the problems.
- Scans are performed against feature branches, not the entire application. This is exactly what you want. Developers don't want to know about vulnerabilities introduced by other people in other parts of the application. They just want to know about the change that they just made – did that change introduce a new security problem or maybe fix an older problem?
- Security professionals get fast and easy scalability across the organization, centralized policy enforcement, and centralized reporting. To deploy to your entire organization, all you need to do is add the Mend application to your source control, and instantly all projects are enabled for scanning by Mend. Literally 10,000 or more repos can be enabled with one command.



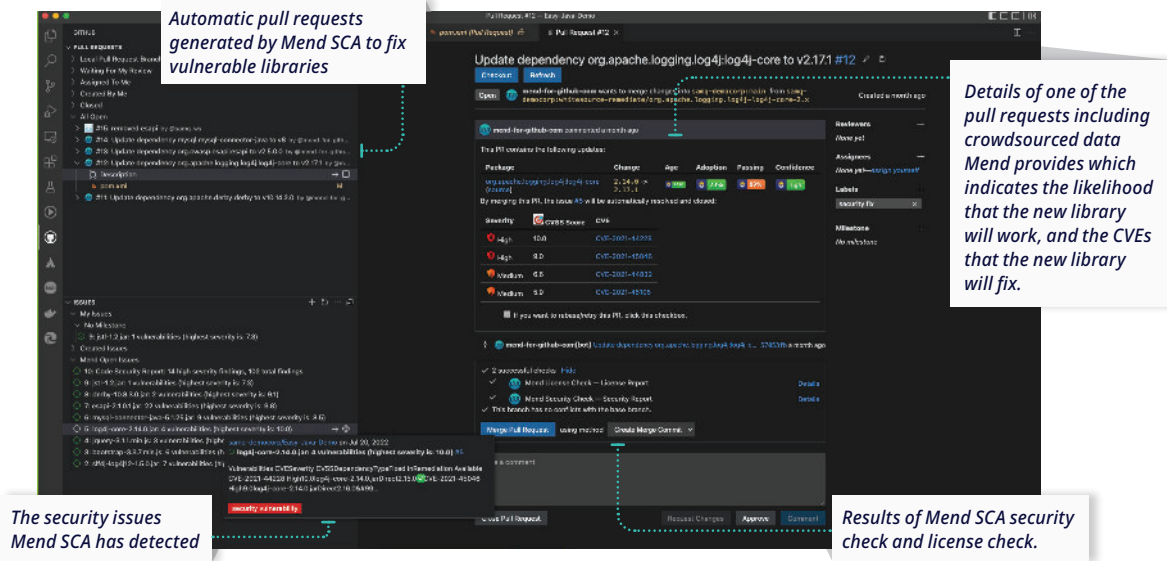
#### Full AppSec in one place:

- Shift left
- In-context feedback loop
- Scan on commit
- Remediation suggestions
- Pull request
- Centralized deployment
- Centralized policy enforcement

*The repository is the optimal place to integrate SCA*

## No context switching

Mend provides a complete find-and-fix workflow within the developer's normal workflow – their IDE and code repository. Developers never need to click out of their normal workflow, never need to switch context, and never need to log into and learn how to use a security tool. This significantly improves the developer experience and, therefore, the likelihood that developers will consistently use the tool to minimize security risks.



## Patented reachability path analysis

Our pioneering and patented reachability path analysis dramatically reduces the number of false positives that Mend produces – typically down to zero. Other SCA vendors make marketing claims that sound similar to our technology, but when you look deeply into what their products offer, you find that they don't compare to Mend's capabilities.

In the screenshot below you can see that Mend SCA scanned an application and identified thirteen vulnerable libraries. However, nine of those libraries were assigned green shields to indicate that they are not reachable vulnerabilities, so they do not need to be remediated.

Library	Product	Severity	Total Alerts	Dependency	Library Type	Creation Date
<input type="checkbox"/> log4j-1.2.13.jar	GH_easybuggy	<span style="color: red;">✖</span> High: 6 Low: 1 details	7	Transitive	Java	23-04-2022
<input type="checkbox"/> commons-io-2.2.jar	GH_easybuggy	<span style="color: red;">✖</span> Medium: 1 details	1	Transitive	Java	23-04-2022
<input type="checkbox"/> mysql-connector-java-5.1.25.jar	GH_easybuggy	<span style="color: orange;">⚠</span> High: 1 Medium: 6 Low: 2 details	9	Direct	Java	23-04-2022
<input type="checkbox"/> esapi-2.1.0.1.jar	GH_easybuggy	<span style="color: orange;">⚠</span> High: 1 Medium: 1 details	2	Direct	Java	26-04-2022
<input type="checkbox"/> commons-fileupload-1.3.1.jar	GH_easybuggy	<span style="color: green;">✔</span> High: 3 details	3	Transitive	Java	23-04-2022
<input type="checkbox"/> bsh-core-2.0b4.jar	GH_easybuggy	<span style="color: green;">✔</span> High: 1 details	1	Transitive	Java	23-04-2022
<input type="checkbox"/> antisamy-1.5.3.jar	GH_easybuggy	<span style="color: green;">✔</span> Medium: 5 details	5	Transitive	Java	23-04-2022
<input type="checkbox"/> commons-beanutils-core-1.8.3....	GH_easybuggy	<span style="color: green;">✔</span> High: 2 details	2	Transitive	Java	23-04-2022
<input type="checkbox"/> derby-10.8.3.0.jar	GH_easybuggy	<span style="color: green;">✔</span> High: 1 Medium: 1 details	2	Direct	Java	23-04-2022
<input type="checkbox"/> commons-httpclient-3.1.jar	GH_easybuggy	<span style="color: green;">✔</span> Medium: 1 details	1	Transitive	Java	23-04-2022
<input type="checkbox"/> xercesImpl-2.8.0.jar	GH_easybuggy	<span style="color: green;">✔</span> Medium: 4 details	4	Transitive	Java	23-04-2022
<input type="checkbox"/> jstl-1.2.jar	GH_easybuggy	<span style="color: green;">✔</span> High: 1 details	1	Direct	Java	23-04-2022
<input type="checkbox"/> nekohtml-1.9.16.jar	GH_easybuggy	<span style="color: green;">✔</span> High: 1 details	1	Transitive	Java	23-04-2022

List of vulnerable libraries present in an application, along with indication as to whether each library is potentially risky (red or yellow shields) or not (green shields).

Across all our customers, our research indicates that between 60 percent and 80 percent of all vulnerable libraries are not reachable and therefore can be ignored. This results in considerable time-savings, which results in increased developer adoption. (Developers utilize security tools that don't waste their time with false positives.)



**“There are times when we receive alerts about seemingly important libraries, but then Mend Prioritize will show us that our application isn't actually using the vulnerable method.”**

*Dragan Pleskonjic, IGT*

## Why false positives are so prevalent with open source, and how Mend eliminates them

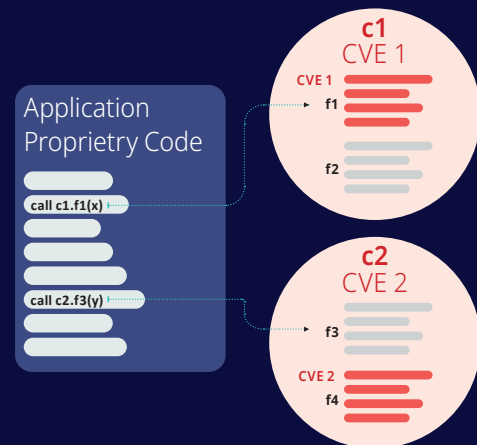
The typical open source software library contains many functions, but an application will typically only call a subset of those functions. For example, in the diagram below, the application calls two dependencies, c1 and c2. Each dependency contains two functions, only one of which is vulnerable.

The typical SCA tool will alert that both c1 and c2 dependencies are vulnerable and are used by your application. Therefore, you would ask your developers to fix both vulnerabilities. But as you can see, half of this work is unnecessary. Your application is not utilizing the vulnerable function in c2, so that library can't hurt you. It is not a risk.

The effect of these false positive alerts will be that —

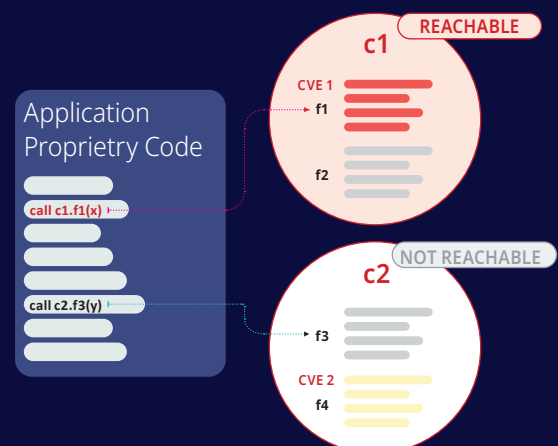
- Your developers' precious time will be wasted
- The software release will be delayed, which can negatively impact business results
- If developers know that their SCA tool is generating false positive alerts, they will become discouraged and may stop using the SCA tool entirely.





Through patented technologies, Mend SCA is able to solve this problem. Mend SCA can tell whether the way you are using each dependency exposes you to the vulnerability. As shown by the diagram below, if the vulnerable function is used by your application, the vulnerability is deemed “reachable”. If not, it is deemed “not reachable”.



Reachability path analysis is extremely helpful if you want to prioritize your remediation efforts and save developers time. For this reason, we call this feature “Mend Prioritize”.

Mend Prioritize results are displayed in the Mend application (see screenshot on previous page). They can also be published via reports and extracted programmatically via our API. Each vulnerability is assigned a different colored shield to indicate whether the vulnerability can be ignored. Any vulnerability that is displayed with a green shield is guaranteed to not pose a threat to the scanned application. No other security vendor can offer this level of assurance.



 <p>This is a reachable vulnerability. Your proprietary code is making calls to the vulnerability.</p>	 <p>The analyzed open source vulnerability could not be established.</p>	 <p>This is not a reachable vulnerability. Your proprietary code is NOT making calls to the vulnerability.</p>	 <p>A new scan is recommended due to updated vulnerability information</p>
---	---	--	---

### Automated remediation for both custom code and open source software

Mend is the only application security vendor that provides automated remediation workflow for both custom code weaknesses and open source software vulnerabilities. Developers can maximize their productivity by leveraging Mend's auto-generated, real-time pull requests that make fixing a vulnerability or software weakness as easy as a single click.

### Merge confidence data

The crowdsourced merge confidence data that Mend SCA displays to help developers decide whether to update their dependencies (see above) is also displayed whenever the product suggests an update to fix a vulnerability. This data helps developers decide whether to accept or reject the recommended remediation.

For example, suppose that Mend SCA identifies that you have an effective vulnerability in version 1.1.7 of a package named `mocha-multi-reporters`. Like any good SCA product, Mend SCA also tells you that the first version of `mocha-multi-reporters` that fixes this vulnerability is version 1.5.0. But Mend's merge confidence data shows that there is low confidence that this upgrade will work properly:

Package	Update	Type	Change	Age	Adoption	Tests	Confidence
<code>mocha-multi-reporters</code>	minor	dependencies	1.1.7 -> 1.5.0	5d	0%	4%	low

Mend SCA also displays other remediation options. Here, you can see that release 1.5.1 looks like a much better bet:

Package	Update	Type	Change	Age	Adoption	Tests	Confidence
<code>mocha-multi-reporters</code>	minor	dependencies	1.1.7 -> 1.5.1	5d	63%	100%	high

### Malicious package identification and automated blocking

While vulnerabilities have been around for a long time, malicious open source software is a relatively new phenomenon. Since approximately 2020, black hats have been coming up with new ways to inject malicious software directly into the supply chain. These attacks are also sometimes known as "supply chain attacks", and they may include techniques such as:

- Typosquatting
- Makefile pollution
- Accidental injections
- Package tampering
- Brandjacking
- Dependency confusion

Forrester Research has [reported](#) that these types of supply chain attacks increased 650 percent in open source projects from 2020 to 2021.

For details on a few malicious packages that Mend discovered, check out this blog:

<https://www.mend.io/resources/blog/new-supply-chain-attack-methods-april/>



To address the serious and rapidly growing problem of supply chain attacks, Mend Supply Chain Defender detects and blocks malicious open source packages before your developer can download them — and before they can pollute your codebase. Mend Supply Chain Defender has already detected and reported thousands of malicious packages.

Mend Supply Chain Defender can be deployed by individual developers via a plugin to their package managers. Alternatively, enterprises using JFrog Artifactory and Mend SCA Enterprise can centrally activate Mend Supply Chain Defender to protect all projects linked to their JFrog Artifactory registries.

## SBOM

The recent Log4j fire drill has pointed out the importance of having a software bill of materials (SBOM) for every application that an enterprise owns. Cybersecurity and Infrastructure Security Agency (CISA) Director Jen Easterly has called for more widespread use of SBOMs, and SBOMs are now mandated by President Biden's Executive Order issued in May 2021.

The Mend SBOM allows you to meet regulatory requirements and provides a path to remediation for vulnerable components found in software. Results are provided in SPDX format, which is a machine-readable inventory of software components, their dependencies, and their hierarchical relationships.

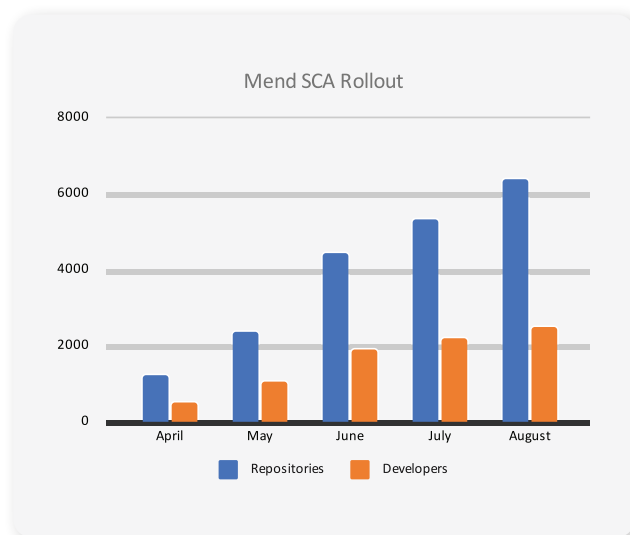
## OSS Licensing

Confidence in your code isn't limited to security concerns, you must also be sure that you are using the code properly and without introducing legal risk. Mend's platform automatically identifies the license used by each component and alerts you to any problematic or undesirable licenses on each scan, as soon as it's added. You can setup Mend to indicate which licenses are allowed, which ones are not, or you can have the product initiate a review process for more complex cases. All steps are performed automatically with the highest precision in the industry on each scan, so you know exactly what licenses are being introduced as soon as possible.

## Real-world Results

Within a few months of implementing the Mend Application Security Platform, our customers report the following results:

- The product has been deployed to all software projects throughout the company via the repo integration.
- The security team has higher confidence that they have visibility to all open source security risks across the organizations.
- Developers have higher confidence in their security tools. They routinely use the product, keep their dependencies up to date based on Mend's recommendations, and promptly act on the remediation recommendations provided by Mend.
- MTTR is lower than before Mend was deployed.
- Developers are spending 80 percent less time remediating open source vulnerabilities than before Mend was deployed.
- Overall level of open source security risk is much lower, often by 90 percent.



*Mend's integration with repositories allows fast, easy rollout such as shown in this actual customer example in 2022.*

## Conclusion

---

SCA is now considered a foundational element of application security. The importance of SCA has been driven by the ubiquitous use of open source software combined with the increasing number of attacks against open source software.

But security teams need to think carefully before adding yet another type of security test on top of the burden already being shouldered by developers. Many enterprise security teams have already tried – and been burned by – first-generation SCA tools that suffered from:

- Cumbersome integrations with the software development tool chain, which made the tools slow to scale
- Too many false positives, which increased the burden on developers, wasted their time, and led to resistance to use the tool
- Too little remediation automation, which led to developers deciding not to remediate security issues

Mend SCA has several unique advantages over other SCA products that solve the problems mentioned above. Mend's customers — ranging from Microsoft to Siemens and many more — have found great success and have even been willing to [tell their stories](#) in the hope that others may follow in their footsteps.



**Working with Mend has been the right decision. When we have the right set of recommendations, we feel more secure. Mend has been able to scale to our needs. It's been able to scale to the ecosystems that we want to cover. Overall it's been a great decision."**

*Poonam Gupta, the Director of Microsoft's 1ES team*

## About Mend

Mend, formerly known as WhiteSource, effortlessly secures what developers create. Mend uniquely removes the burden of application security, allowing development teams to deliver quality, secure code, faster. With a proven track record of successfully meeting complex and large-scale application security needs, the world's most demanding software developers rely on Mend. The company has more than 1,000 customers, including 25 percent of the Fortune 100, and manages Renovate, the open-source automated dependency update project.

For more information, visit [www.mend.io](http://www.mend.io), the Mend blog, and Mend on LinkedIn and Twitter.