MEND

The Devil in The Details:

The Importance of SBOMs in Protecting the Software Supply Chain

When security researchers discovered in December 2020 that attackers had trojanized software updates in a SolarWinds application, it was a rude introduction into software supply chain attacks for more than 18,000 businesses and governmental agencies.

While SolarWinds is the largest and best known software supply chain attack thus far, it's unfortunately not a unique occurrence. The reality is that attacks against the software supply chain are one of the most pervasive threats that companies face, with attackers launching almost 7,000 software supply chain attacks in just the past year, as bad actors look for ways to steal data, corrupt targeted systems and gain access to other parts of the network through lateral movement.

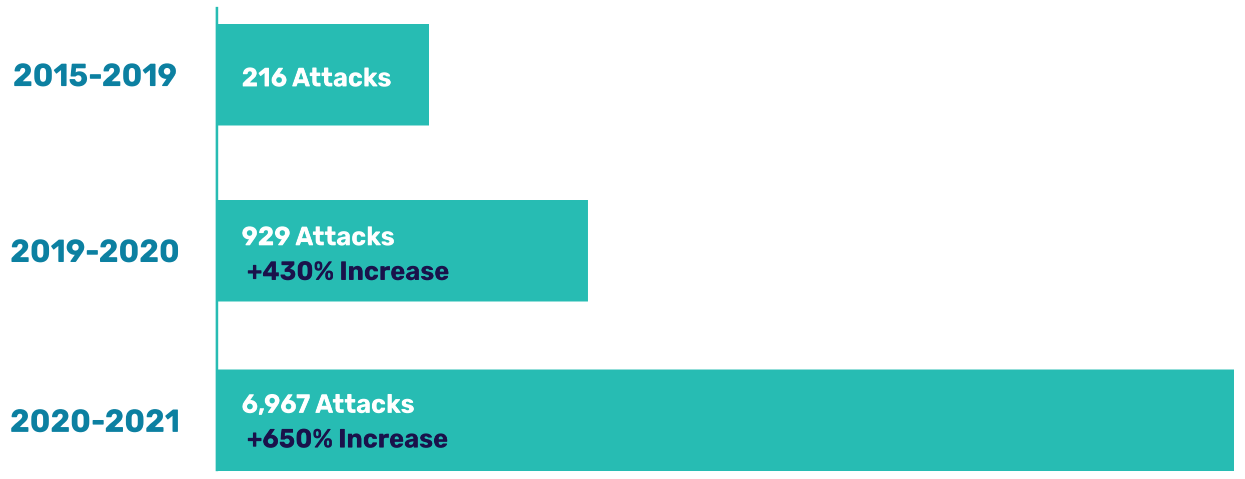


Figure 1: Increase in Software Supply Attack Frequency

Software supply chain attacks occur when a bad actor infiltrates a software vendor's network, employing malicious code to compromise the software before the vendor sends it to a customer. The compromised software then compromises the customer's data or system. Newly acquired software may be compromised from the outset, or a compromise can occur through a patch or hotfix. Such attacks affect all users of the compromised software and can have widespread consequences for government, critical infrastructure and private sector software customers.

According to the European Union Agency for Cybersecurity (ENISA), almost 60 percent of supply chain attacks are aimed at gaining access to data – including personal data and intellectual property – and around 16 percent of attacks are an attempt to gain access to people. It should also be noted that in 66 percent of incidents, attackers focused on suppliers' code in order to further compromise targeted customers.

The software supply chain is made up of everything that goes into or affects code from development all the way until it's deployed into production. It includes code, binaries and other components as well as and includes information about who wrote it, when it was contributed, how it's been reviewed for security issues, known vulnerabilities, supported versions, and license information – basically everything that touches it at any point.

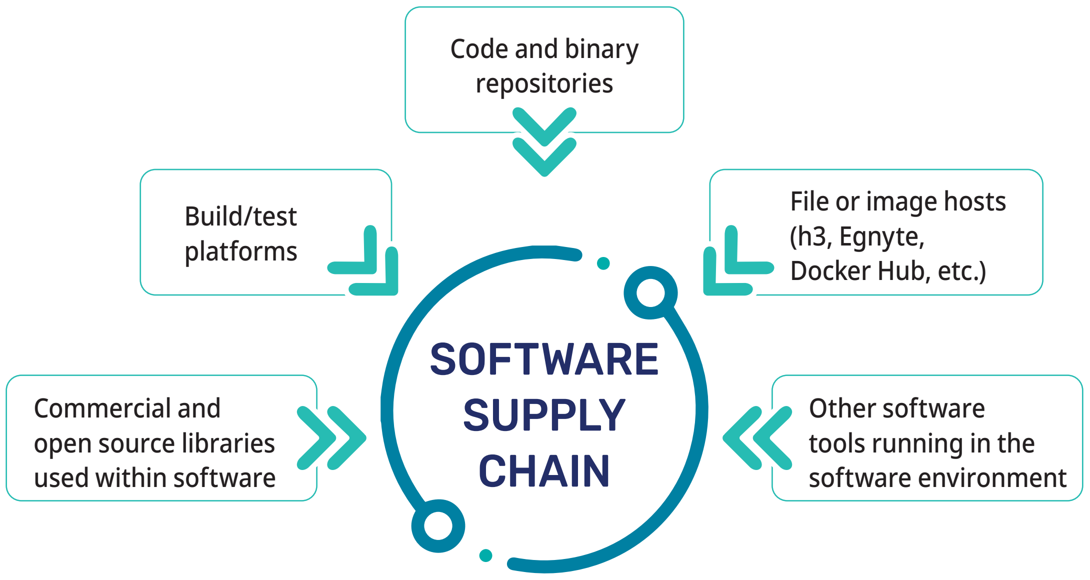

Code and binary repositories
Build/test platforms
File or image hosts (h3, Egnyte, Docker Hub, etc.)
SOFTWARE SUPPLY CHAIN
Commercial and open source libraries used within software
Other software tools running in the software environment

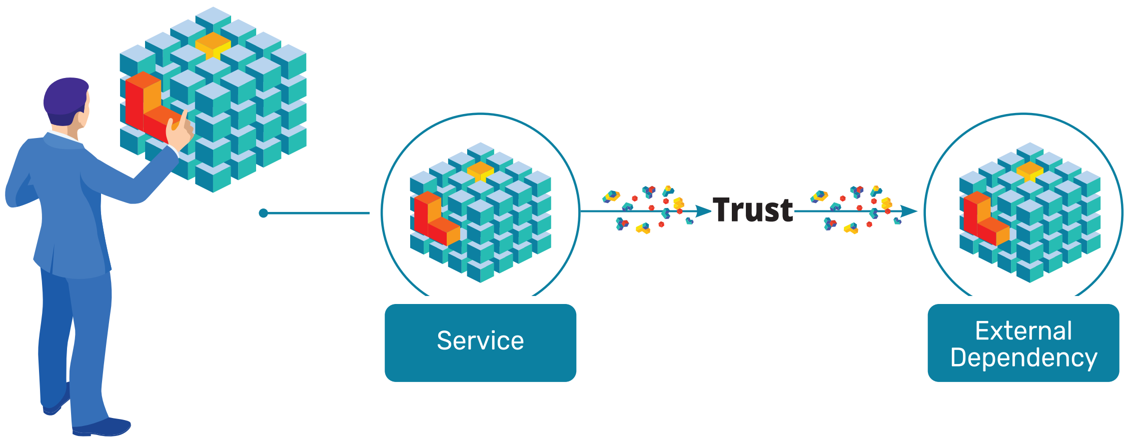
Figure 2: Components of the Software Supply Chain

As shown in Figure 2, the software supply chain consists of every non-organic service or piece of software that's used to build software. Importantly, this includes components and packages that are open source – a massive undertaking given the use of open source code bases has increased from 36 percent in 2015 to 75 percent in 2020.

Open source is extremely attractive to attackers, who implant malware directly into open source projects to infiltrate the software supply chain. In the past, software supply chain exploits were used against publicly disclosed open source vulnerabilities that were left unpatched. But instead of waiting for disclosures to pursue an exploit, attackers now are choosing to inject new vulnerabilities into open source projects that feed the global supply chain – and then exploit those vulnerabilities before they are discovered.

The growth in software supply chain attacks has – not surprisingly – resulted in a firestorm of activity on the part of regulators and governmental organizations to determine how such attacks can be prevented.

A key facet for all of the regulatory agencies involved with securing the software supply chain is the demand for a software bill of materials (SBOM), a formal record that contains the details and supply chain relationships of various components that are used to build software.



Timeline Of Regulatory Efforts To Address Software Supply Chain Attacks

1 **February 2021**

President Biden issues an Executive Order (EO) laying out changes to secure all supply chains, including software, calling for the secretaries of Commerce and Homeland Security to report on the security and integrity of critical information and communications technology software supply chains.

2 **February 2021**

The CIO for the Department of Defense rolls out a new reference for Zero Trust Architecture (ZTA), including a section centered on protecting applications and software supply chains.

3 **April 2021**

The National Institute of Standards and Technology (NIST) and the Cybersecurity and Infrastructure Security Agency (CISA) release Defending Against Software Supply Chain Attacks, acknowledging that software compromised in supply chain attacks could have widespread consequences. It outlines safeguards to secure the software supply chain, including:

- Developing a written program to address software supply chain risk
- Inventorying organizational reliance on external software and code
- Assessing risk from vendors and adopting contractual and other safeguards
- Monitoring threats and vulnerabilities to the software supply chain

4 **May 2021**

Biden signs a second EO for software supply chains as part of a critical look at the nation's cybersecurity posture. It requires NIST to create guidelines to secure the software supply chain, including the need for all critical software to include a software bill of materials (SBOM).

5 **June 2021**

NIST defines critical software at having at least one of these attributes:

- Is designed to run with elevated privilege or manage privileges
- Has direct or privileged access to networking or computing resources
- Is designed to control access to data or operational technology
- Performs a function critical to trust
- Operates outside of normal trust boundaries with privileged access

6 **July 2021**

NIST publishes security measures for critical software use and minimum standards for vendors' testing of their software source code.

7 **July 2021**

The National Telecommunications and Information Administration (NTIA) releases the minimum elements needed for an SBOM to improve transparency in software supply chains for both technology vendors and government customers.

VULNERABILITY REMEDIATION

In order to mitigate open source risks, it's essential to remediate open source vulnerabilities as soon as they are discovered. However, in most cases it's impractical to fix all vulnerabilities, and some require major development work. You must prioritize vulnerabilities, understand which ones represent a real risk, and provide development and IT teams with the information that they need in order to quickly fix the most critical vulnerabilities first. You must also help teams shift left and discover vulnerabilities earlier, making remediation easier.

The Challenge

Fast response is critical
When a new vulnerability is discovered, you must fix it fast before hackers exploit it. However, many remediations require major development work.

Prioritization
In order to make remediation efficient, you must be able to quickly prioritize vulnerabilities that matter.

Open source component selection
The best remediation is avoiding insecure components when they are initially selected for a project. This requires integrating security at early stages of development.

Locating vulnerabilities in the code
Once a vulnerability is discovered, it's difficult to pinpoint where it is referenced in the code

Mend Offering - Vulnerability Remediation: Essential Features

Prioritization	Analyzes how affected libraries are actually used in your code. Eliminates 70-80% of vulnerability alerts which have no impact on the current project.
Pinpointing Vul-nerabilities in Code	Provides a complete trace analysis for each vulnerability. Shows which part of your code touches the vulnerable functionality, down to the line number.
Shift Left	Scans popular repos, including GitHub repos using its GitHub integration Browser extension gives developers information about components as they browse the web
Shift Right	Tracks the "Bill of Materials" of the latest build of every version you deploy. Alerts teams in real-time if new vulnerabilities are reported, with remediation guidance.
Vulnerability Remediation	Sources patches and suggested fixes from hundreds of trusted open source communities. Offers developers multiple remediation options, with scoring.
Automated Policy Enforcement	Enables granular policies per the project, product, app type, or organization. Allows you to set policies for each stage of the SDLC: planning, development, deployment, production. Policies can be conditional on security parameters such as severity or a specific CVE.

INVENTORY MANAGEMENT

It's crucial to identify which open source components are in use in your software. To fully manage open source security, licensing, and compliance issues, you must gain visibility into open source code usage across all stages of the SDLC—from the selection stage to development, deployment and production stages.

The Challenge

Manual inventory management is extremely time consuming
Developers spend precious time checking licenses and security and requesting approval. These processes can be automated.

Support for multiple languages and frameworks
Automated tools must support all of the languages and frameworks used in your organization to accurately track open source components.

It's difficult to gather information about open source components
Information about open source components, versions and licenses is spread across hundreds of sources. Tracking it manually is extremely complex.

Detecting transitive dependencies
Most open source components are dependant on other open source components, creating complex dependency trees.

Mend Offering - Inventory Management: Essential Features

Languages and frameworks coverage	Supports over 200 languages, frameworks and development environments
Integrations with DevOps tools	Integrates with IDEs, package managers, repos, build tools, CI servers and AST tools. Alerts about problems and guides teams as they select and download components.
Open Source Components Database	Doesn't rely on proprietary data—aggregates information about open source components from hundreds of sources, multiple times a day.
Dependency Detection	Detects open source components based on SHA file signatures. Fully resolves dependency tree and manifest files, including undeclared dependencies.
Automated Policy Enforcemen	Automates selection, approval, and tracking of open source components. Policies defined per SDLC stage, product, and organization.
Reporting	Provides built-in reports at the project, product or organization level. Open source Bill of Materials, due diligence report, risk and attribution reports.

*An attribution report tells you which components require you to publish attributions, crediting the project creators

OPEN SOURCE LICENSE COMPLIANCE

You must establish an open source usage policy, and block inappropriate or overly restrictive open source licenses. You must also ensure that you are meeting license requirements for all of the open source components used in your software. Management, security, legal teams, and third parties such as investors conducting due diligence need complete visibility over open source licensing.

The Challenge

Manual license management is extremely time consuming
Applications use thousands of open source components. Managing their licenses manually creates overhead and slows down development.

Expert review is required in some cases
Even when using automated tools, there must be a way to perform a review for some types of licenses.

Inaccurate detection
Less advanced automated SCA tools do not always accurately detect licenses. Associating libraries and licenses is not trivial. Missing problematic licenses can incur major costs later on.

False positives
Some SCA tools raise false alarms, placing a burden on teams, or miss problematic licenses leading to liability.

Mend Offering - License Compliance: Essential Features

Languages and frameworks coverage	Supports over 200 languages, frameworks and development environments
Accuracy	100% license detection accuracy unique ability to associate folders and libraries. Shows licenses for the entire dependency tree. Supports multiple licenses for one component.
Real-Time Alerts	Alerts which license types are in your software, with a full risk analysis. Offers suggestions for resolution based on organizational policies.
Shift Left	Provides a native Github integration—scans GitHub repos and discovers license information. Provides a browser plugin that helps avoid problematic licenses at the selection stage. Integrates with DevOps tools at all stages of the SDLC and alerts about license issues.
Automated Policy Enforcemen	Automates the license approval process. Extends license tracking and approval to the full dependency tree of each package. Can automatically approve, reject, or initiate a manual workflow, depending on license type.
Reporting and Auditing	Offers a wide range of reports built for all relevant organizational roles. Provides visibility for internal teams—R&D or IT management, security, legal, management. Provides visibility for compliance auditors and due diligence investigators.

EMPOWERING OPEN SOURCE AUTOMATION WITH EFFECTIVE USAGE ANALYSIS

Automation of open source vulnerability management and license compliance is critical to solving the challenges we outlined above. However, automated tools have their limitations.

Previous generations of SCA tools were able to detect open source components and tell organizations they have vulnerabilities but were not able to identify the potential impact of those vulnerabilities. This required security and development teams to invest many resources in investigation
and response to a large number of vulnerability reports.

EFFECTIVE
Vs.
INEFFECTIVE

Research has shown that of the vulnerabilities discovered, 70-85% were not really critical, because
the vulnerable open source code was not actually accessed or used by the organization's proprietary code.

Mend pioneered the next generation of SCA with Effective Usage Analysis. This technology doesn't just tell you what open source components you have, but also how you use them. It lets you hone in on the 15-30% of open source vulnerabilities that are actually in active use within your product.

It also shows exactly where a vulnerable functionality is referenced within the code,
making it much easier for developers to fix vulnerabilities.

Effective Usage Analysis provides automatic prioritization that can reduce remediation efforts,
and help teams fix important vulnerabilities much faster.

THE MEND EDGE

We have covered numerous considerations for selecting an SCA tool for your organization. The bottom line is that you should select an SCA tool that enables you to minimize risk and reduces the effort for all of your teams—from management, legal and security through ops, developers and QA engineers.

Mend is the leading SCA platform. It helps you minimize risk with the lowest possible effort while fostering a DevSecOps mindset and cooperation across your organization:

Completeness

A one-stop-shop for all of your open source usage regardless of your languages or environments, including containers and serverless. We also support all groups in your organization: Security, Developers, DevOps.

Prioritization

Focus on what really matters. We help you to prioritize the security vulnerabilities that actually impact your products, and ensure there are no false positives.

Remediation

Alerts are great, and we also provide the fix. Automatically generate fix pull requests, get direct links to suggested fixes, optimize remediation with full trace analysis and initiate automated workflows including issue tracker integration.

Simplifying the World of Open Source Usage

We believe the only way to use open source without compromising on security or quality and without slowing down your developers — is to make this complex process of risk mitigation as simple as possible.

Using open source code should be easy. That's why we created Mend. Our technology takes care of the heavy lifting that comes with open source usage. We alert you only when something demands your attention, providing you with all the information that you need to make the right choices.

How We Bring Order to Chaos

The Mend platform continuously detects all of the open source components used in your products. It then compares these components against Mend's database. This database is built by collecting up-to-date information about open source components from numerous sources, including various vulnerability feeds and hundreds of community and developer resources. The Mend platform is designed for security and compliance professionals, to give managers everything that they need in order to control and manage the open source usage within their organization. It allows them to enforce their policies automatically throughout the SDLC, get real-time alerts on critical issues, and generate up-to-date reports.

It includes a set of tools that fit into the developers' ecosystem, empowering them to make educated choices, speed up integration, and quickly find and remediate problematic open source components. Because knowledgeable developers make better software.