REVERSINGLABS

Flying Blind: Software Firms Struggle To Detect Supply Chain Hacks

A SURVEY FINDS CONCERN ABOUT SOFTWARE TAMPERING AND SUPPLY CHAIN ATTACKS, BUT FEW EFFORTS TO DEFEND AGAINST THEM

Contents

Executive Summary

It is accepted wisdom today that almost every company of any size is a software company. Software effectively drives all manner of businesses. It is also the animating force behind a growing population of physical objects - the Internet of Things - ranging from cars and industrial equipment to televisions, home appliances, medical devices and more.

Despite its expanded reach and diversifying use cases, software today is not measurably more secure. Secure software development practices and processes are well understood, but haven't yet penetrated many software development organizations. At the same time, the embrace of open source software and the advent of agile development methodologies have introduced new risks in the form of vulnerable or compromised third party components.

Attacks like the one on the software firm SolarWinds in 2020 raised the profile of what had been considered an obscure threat: attacks on the software supply chains that produce and distribute trusted applications and services used by companies and individuals around the globe. But SolarWinds was not the first supply chain compromise, and judging by the results of a survey of software publishers, it won't be the last.

A survey commissioned by ReversingLabs and conducted by Dimensional Research of 307 employees at firms involved in software publishing shows that software security is too often a back burner issue, even as concerns about software supply chain attacks and the risks that accompany greater reliance on open source and third party code grows.

Surveyed firms admit they regularly release software to customers that is subsequently found to contain security flaws. At the same time, thorough audits of software during and after development are a rarity, with fewer than 4 in 10 companies capable of detecting tampering with developed code. Less than 10% of companies are reviewing software at each stage of the production lifecycle for evidence of tampering or compromises. That doesn't bode well for organizations and industries worried about the prospect of further "SolarWinds" style compromises and attacks.

> The Solar Winds hack shows the effectiveness of an attack targeting software supply chains and tampering with the developed code to achieve attacker's objectives. Unfortunately, this survey suggests companies are struggling to address that risk.

Mario Vuksan
CEO of ReversingLabs

Survey Methodology & Participant Demographics

Dimensional Research conducted the survey on behalf of ReversingLabs. The goal of the survey was to understand current software security risk release metrics and the prevention of software tampering across the build and release processes. The use and related risk of 3rd party software, including open-source repositories, as well as the use of software bills of materials (SBOMs) to reduce software security risk, were all areas of interest.

Participants

The survey was administered to executives, technology, and security professionals at software enterprises representing all seniority levels. The questions asked concerned software security considerations, build and release processes, and tools used. In all, 307 qualified participants completed the survey. All participants had digital product or leadership responsibilities. Participants represented numerous countries from 5 different continents.

Methodology

The survey was administered electronically, and participants were offered a token compensation for their participation.

Definition of Software Tampering

For the purposes of this report, we are defining the term "software tampering" as referring to any change with malicious intent that ends up as a component of a released software package or a container that customers or users deploy.

Background: Software Supply Chain Risk Is Growing

Recent attacks on organizations like SolarWinds, CodeCov, and others signal a fundamental shift in the threat landscape. Sophisticated threat actors, long content to target and compromise the accounts of privileged employees, or vulnerable, public-facing applications, are "shifting left" (to borrow a phrase). Malicious actors have found that they can leverage vulnerable development pipelines to further sophisticated attacks: tampering with developed code to introduce malicious backdoors or other features.

How attuned are software publishers to these risks? And how able are they to detect software tampering and supply chain compromises? To help answer these questions and understand the dynamics shaping modern software development organizations, ReversingLabs commissioned a survey of more than 300 IT professionals working for organizations that develop and publish software.

Supply Chain Attacks: Not New

The notion of manipulating the systems used to develop, test and distribute software to gain privileged access is not a new one. In fact, compromises of software providers date back more than half a century to the depths of the Cold War between Russia and the United States and its allies.

But the practice of supply chain attacks has taken on a new life in the past decade, as sophisticated nation-state actors and even cyber criminal groups have turned from overt attacks on network perimeters and endpoints to stealthy compromises of open source and commercial applications and service providers. In the process, software supply chain compromises have moved from the realm of “rare and extraordinary” to “commonplace,” even as larger trends like the embrace of cloud computing and agile DevOps development strategies have made software supply chains longer and more brittle.

SolarWinds: A Cyber Risk Wake-Up Call

The 2020 attack on SolarWinds, a well known supplier of IT management software, epitomizes the growing risks of supply chain attacks not only to software publishers, but to their customers. As ReversingLabs wrote in its analysis of that attack, attackers believed to be affiliated with a Russian state-sponsored hacking group compromised the build and code signing infrastructure for SolarWinds Orion software and modified the Orion source code to include a malicious backdoor. Ultimately, around 18,000 Orion customers are believed to have received a malicious update from SolarWinds as part of the attack with 100 ultimately attacked. That included U.S. government agencies and Fortune 500 companies in technology, finance and other industries.

Supply Chain Hacks Go Mainstream

The SolarWinds attack stands out for its sophistication, for its high profile victims and the long tenure that attackers enjoyed within the SolarWinds environment. But attacks on software supply chains are becoming commonplace, and not all of them are advanced, nation-state backed operations. Attackers are expanding their target list and lowering the bar for compromises by focusing on development infrastructure used for the creation of proprietary and open source software and services.

For example, ReversingLabs discovered and documented multiple campaigns to place malicious components like back doors inside Node.js packages managed using NPM. Among them, a threat labeled Win32.Infostealer.Heuristics that was found in several versions of the *nodejs_net_server* package using static analysis. Metadata collected from these packages shows that the application was really a ChromePass utility, a tool which can be used to recover passwords stored inside of a Chrome web browser and that was used to steal and exfiltrate credentials from hosts that installed applications using the nodejs_net_server package.

That was similar to another NPM compromise, in 2019, in which ReversingLabs researchers discovered a password recovery tool WebBrowswerPassView in an NPM package called bb-builder. Finding it hidden in bb-builder, for no apparent reason, was evidence of an attempt to leverage the software supply chain to facilitate credential theft from downstream consumers of the bb-builder package.

Unlike SolarWinds, however, the NPM compromises do not demand access to sensitive development environments nor extensive actions to cover up the activities of attackers. Instead, they function more as “watering holes”: waiting passively for victims to download and implement compromised packages before springing to life.

Sophisticated or not, supply chain compromises are more frequent now than ever before. As our Partial History of Supply Chain Attacks illustrates, there have been more attacks targeting software supply chains in the last two years than the 40 years before that. They include attacks on AsusTek[1], a software vendor used by managed service providers, and CodeCov, a maker of software development tools.

As organizations embrace software as a service (SaaS), cloud and DevOps methodologies, reliance on extended software supply chains - and the risks that come with them - is bound to increase in the years to come.

Uncle Sam Is Watching

Amid growing threats, the U.S. Government is showing increasing interest in the issue of software supply chain security. NIST, the National Institute for Standards and Technology, published Version 1.1, of SP 800-218, the Secure Software Development Framework (SSDF) in February. In May, NIST published updated guidance, SP 800-161r1, on Cybersecurity Supply Chain Risk Management.

The documents were published in response to a May 2021 Cyber Executive Order (EO 14028)[2] which cited a "pressing need to implement more rigorous and predictable mechanisms for ensuring that products function securely, and as intended."

SP 800-218 includes a number of requirements focused on spotting threats in the software development process. Among other things, it asks makers of commercial off-the-shelf (COTS) and government off-the-shelf (GOTS) software to:

- Collect, maintain, and share provenance data for all components and other dependencies of each software release (e.g., in a Software Bill of Materials [SBOM]).
- Obtain provenance information (e.g., SBOM, source composition analysis, binary software composition analysis) for each software component, and analyze that information to better assess the risk that the component may introduce.
- Check code for backdoors and other malicious content

Other requirements articulated in the Executive Order and incorporated into the document require software producers to attest their conformity with secure software development practices and also to attest to the security of any open source software they use.

SP 800-161r1 provides guidance to organizations that wish to identify, assess, and mitigate cybersecurity risks in the supply chain by integrating cybersecurity supply chain risk management (C-SCRM) with existing risk management activities.

Key Findings

Software publishers find themselves buffeted by these changing winds. On the one side there are mounting attacks on software supply chains. On the other: heightened concerns (and demands) by customers - the federal government chief among them. At the same time, software publishers must contend with all the old challenges: attracting and retaining top development talent, meeting customers' demands, and getting products and features to market in a timely manner.

[1] https://www.csoonline.com/article/3384259/asus-users-fall-victim-to-supply-chain-attack-through-backdoored-update.html

[2] https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

Our survey of executives, technology, and security professionals at software publishers sought to expose the state of play in the industry by asking forthright questions about the role of software security and security reviews in the software development process and measuring publishers' awareness and concern about issues like software supply chain attacks.

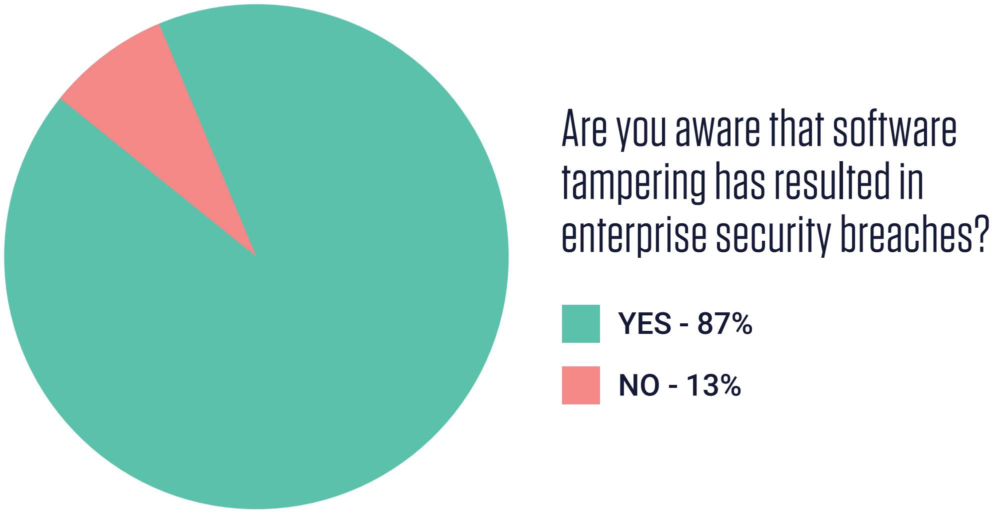
The results paint an intriguing picture and reveal some of the challenges that lay ahead as organizations across the economy look to insulate themselves and their customers from software supply chain risks and attacks.

Here are the top takeaways from our survey.

Software Supply Chain Fears Are Growing

Ten years ago, supply chain attacks may have been considered cyber security arcana, but no longer. The individuals responding to our survey made clear that software supply chain concerns are on the radar and considered a real cyber risk.

For example, asked whether they were aware that software tampering has resulted in enterprise security breaches the vast majority of those we surveyed (87%) answered in the affirmative.



And, when we asked our respondents which software security issues posed a risk to their organization, 63% said that threats and malware lurking in open source repositories or like those seen in the attacks on SolarWinds and CodeCov were a concern. That's just behind the 66% who said "exploitable software vulnerabilities" posed a risk. Other supply-chain related issues also ranked highly. More than half (51%) said that the inability to detect software tampering posed a security risk. 40% said that vulnerabilities in CI/CD toolchains were a concern.

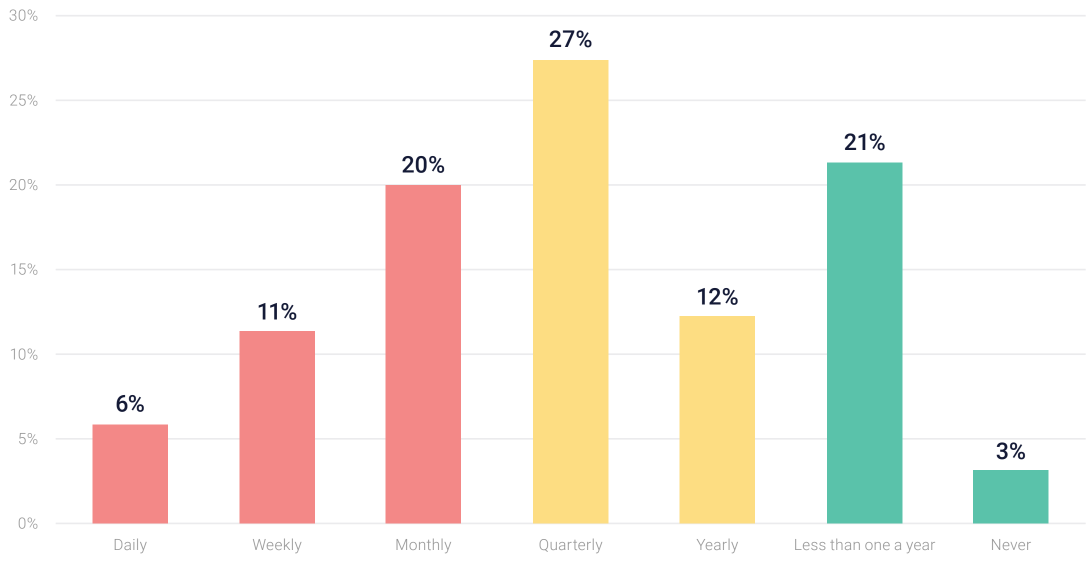
Firms Push Vulnerable Software

Software supply chain risk ultimately traces back to software publishers, themselves. Their security practices - more than anything - determine the security of the code they produce and its susceptibility to attack. And, according to the results of our survey, the security practices of development organizations are a major source of cyber risk for software publishers and downstream users of that developed code.

First, there's the issue of software quality. Companies that license and deploy software implicitly trust its security. The assumption is (understandably) that software producers have assessed their code for flaws and vulnerabilities prior to release - finding and fixing any glaring issues. But responses from the professionals who took our survey suggest that customers' trust is somewhat misplaced.

In fact, 37% of respondents said their company released software monthly that was subsequently found to have a security vulnerability. Widen the aperture to quarterly software releases, and almost two third of respondents (64%) admitted that software released during that period was subsequently found to contain vulnerabilities following either internal or external review.

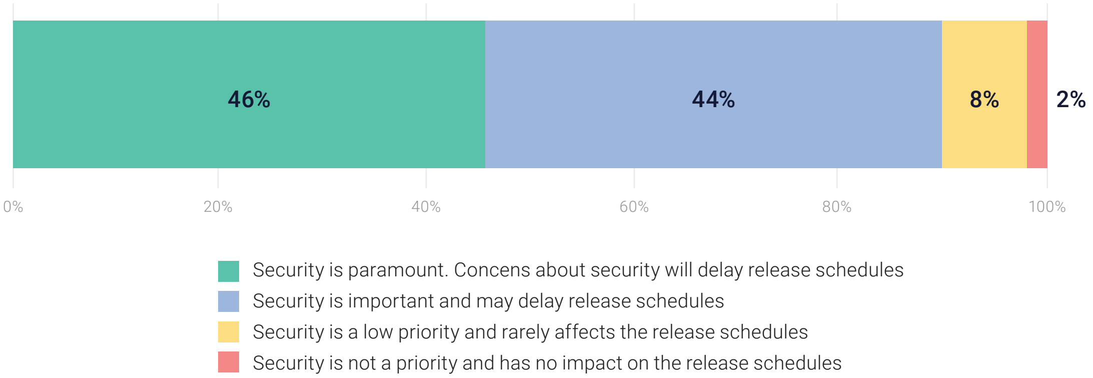
At your company, approximately how frequently is software released that is later discovered (internally or externally) to have a security vulnerability?



Some of those vulnerabilities may simply have been overlooked during quality assurance and security testing prior to final release. But our survey results suggest that a not-trivial percentage of them may not have been overlooked. Specifically: 10% of those surveyed admitted that security issues rarely or never affect the release of software by their organization. That's right: 10% of respondents suggested that it was likely that their employer would ship software even with knowledge of security issues in the code.

Another 44% of those surveyed admitted that security issues were important and could (but may not) delay the release of software. A plurality (46%) said security issues with developed software would definitely delay its release by their employer.

In general, how does software security impact releases at your company?



In other words: a majority of professionals we surveyed (54%) acknowledged that their employer was at least open to the possibility that it would ship software with a known security issue in order to meet a delivery schedule. That's a sobering statistic for downstream organizations that purchase and deploy finished products and services. It also underscores an important, contributing factor to supply chain attacks: a high tolerance for security issues among software publishers.

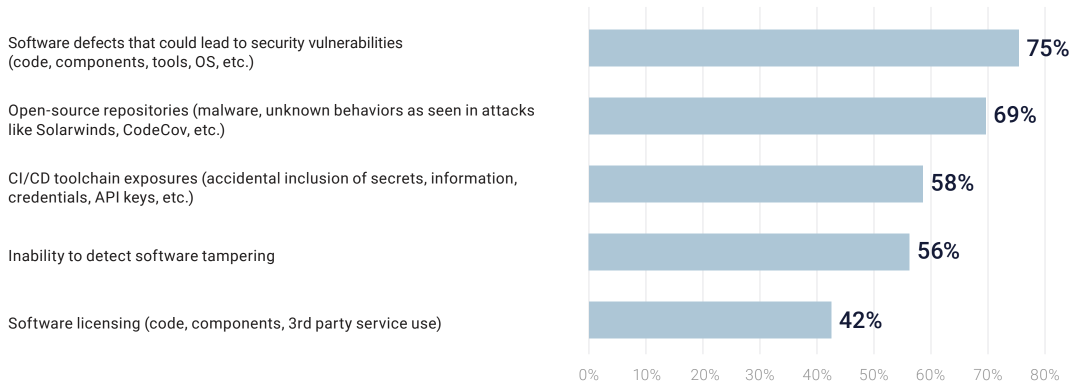
Such attitudes may not be new in software publishing circles. What is new is the willingness of sophisticated cyber criminal and nation-state groups to target vulnerable software supply chains. According to the National Institute for Standards and Technology's (NIST)[3] Framework for Defending Against Software Supply Chain Attacks, for example, advanced persistent threat (APT) actors are "likely to have both the intent and capability to conduct … highly technical and prolonged software supply chain attack campaigns." Lax secure development practices and a willingness to look the other way at vulnerabilities in developed applications and services don't ensure that a given publisher will be the victim of a software supply chain compromise, but they do make it easier for malicious actors to achieve their objectives.

However, our respondents made it clear that they see a link, and that poor security practices by publishers can set the stage for such an attack, and make it more likely to succeed.

For example, when we asked them to name the contributors to software supply chain risk, more than two thirds of respondents (69%) identified threats and malware lurking in open source repositories or like those seen in the attacks on SolarWinds and CodeCov, just behind software defects that could lead to security vulnerabilities. More than half (58%) named CI/CD tooling flaws as a risk.

[3] https://www.cisa.gov/sites/default/files/publications/defending_against_software_supply_chain_attacks_508_1.pdf

In your experience, what are the contributors to software supply chain risk?



Of course, an attack on an exploitable overflow or injection vulnerability in a developed application is different from (and more common than) the kind of stealthy code manipulation that turned SolarWinds' Orion application into a malicious back door on the networks of more than 100 of the company's customers. But both the flaws and the malicious backdoor are likely to be discovered in a software development organization that employs frequent static and dynamic analysis of both source code and finished binaries. Organizations like the 10% of those who admitted in our survey that they would knowingly ship vulnerable code might be presumed to be less capable of- or interested in catching such security lapses.

Understandably, the intention for enterprises is to reach their business goals and to constantly improve their operations. Other priorities, such as security, often move to the back burner. However, the growing drumbeat of software supply chain attacks - SolarWinds, CodeCov, AsusTeK - show how even successful and wealthy firms can suffer if supply chain is not placed at the forefront of executives' priorities.

This tension between agility and security will only grow as DevOps methodologies continue to spread across industries. According to a Forrester study[4], the COVID-19 Pandemic has accelerated the focus on digital offerings, pushing software enterprises to speed up their release schedules in an effort to stay competitive in the industry[5]. This continued cadence has and will give APT actors substantial opportunity to carry out software supply chain attacks.

Publishers Increasingly Concerned About Software Risks

For software publishers, commonly-used components, such as open source repositories and third-party software pose serious risks for enterprises. The software industry is now realizing that these vital components, if not protected properly, could lead to costly and disruptive software supply chain attacks.

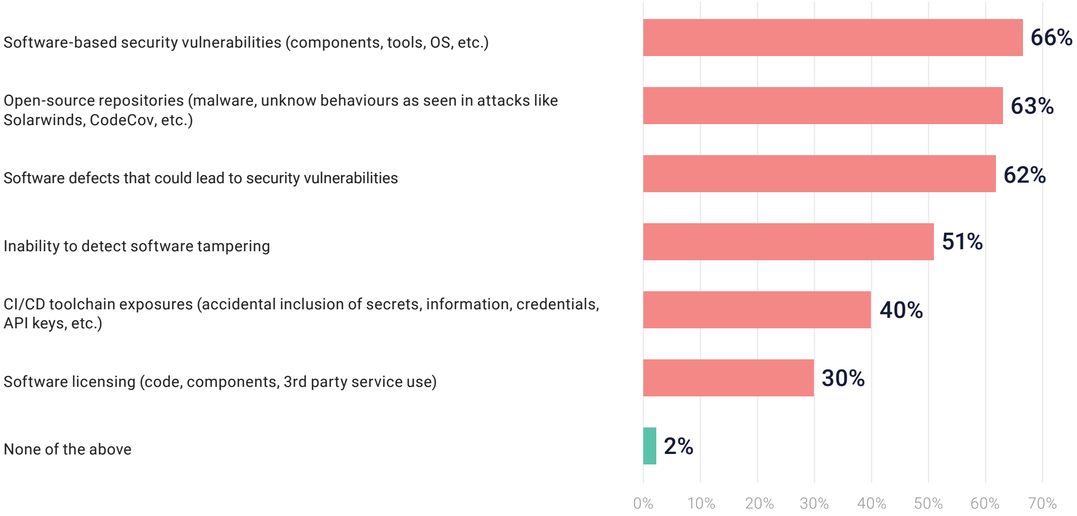
[4] https://www.forrester.com/blogs/covid-19-accelerates-digital-business-in-2021/

[5] Cigniti.com, "Agile Devops Continuous Testing," https://cigniti.com/blog/agile-devops-continuous-testing/

The heavy reliance on open source libraries and components is a case in point. Anywhere from 85-97% of enterprise codebases come from open-source repositories[6]. And open source flaws figured prominently among a list of the most exploited software holes in 2021, including Log4j and a vulnerability in Sudo. It's also clear that companies are not fully aware of the third-party software services they rely on: in a Ponemon study, 54% of responding organizations said they lack a comprehensive inventory of the third parties with access to their network[7].

Our survey suggests that professionals working within software publishing organizations are aware of the risks posed by open source and third party components.

Which of the following software security issues pose a risk for your company today?



Asked what software issues pose the biggest risk to their firms, two thirds (66%), said software-based security vulnerabilities in components and development tools and operating systems. Flaws in open source repositories were a close 2nd place, with 63% of respondents citing open sources as posing a software security risk.

Similarly, when we asked respondents what exactly is increasing software security risk, nearly all of them (98%) indicated that third party software, open source software, and software tampering are contributors.

Lack Of Tampering Detection A Major Concern

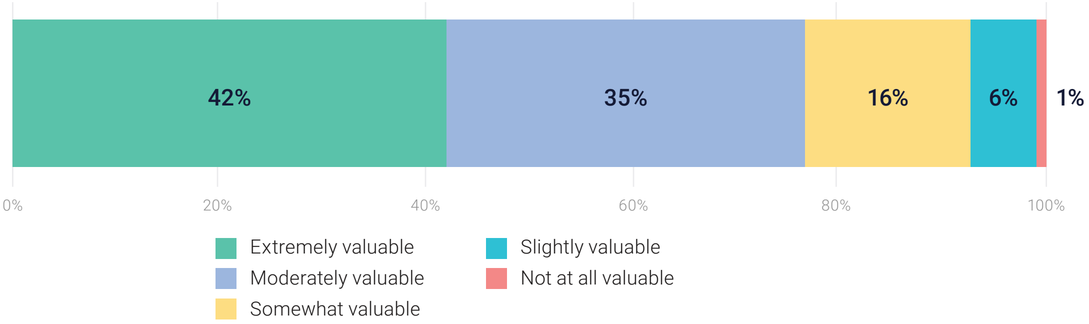
While compromises of internally developed, third party and open source components was a major concern for the individuals who took our survey, few said they had an easy way to detect and block such attacks - a major gap in current protections.

For example, 51% agreed that the inability to detect software tampering poses a risk for their companies while more than three quarters of respondents (77%) indicated that they would welcome a tool to detect software tampering.

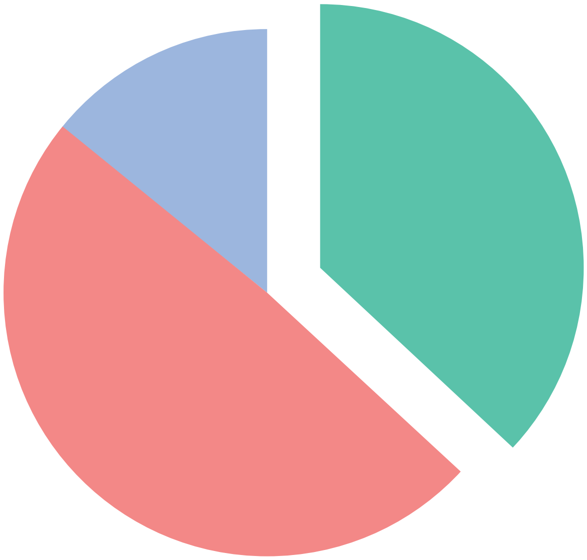
[6] https://github.blog/2020-09-02-secure-your-software-supply-chain-and-protect-against-supply-chain-threats-github-blog/

[7] https://www.forbes.com/sites/forbestechcouncil/2021/11/03/why-managing-third-party-access-requires-a-better-approach/

In your experience, how much value would your company attach to a solution that could detect software tampering?



Still, that desire for a way to detect software tampering is - for most firms - unrequited. Despite the majority of software companies understanding the risks posed by software supply chain attacks, these same companies are falling short in being able to protect themselves:



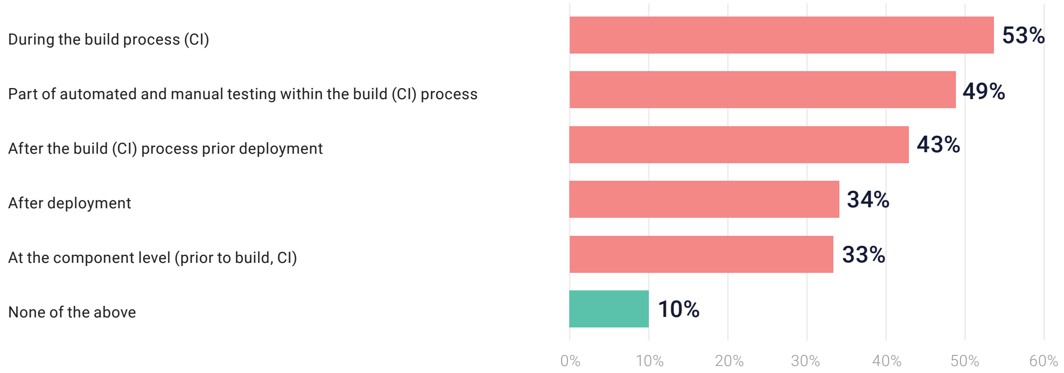
Does your company have a way to detect software tampering across your company's software supply chain?

For example: just 37% of software companies indicate they have a way to detect tampering across their supply chain[8]. Almost half (49%) admitted they did not have a way to detect software tampering across their software supply chain, or that they "didn't know" (14%).

Organizations engaged in software development need to be able to detect tampering at any and all stages of development, including post-build and post-deployment. Our survey suggests that most software publishers are not checking for tampering at each stage of the development lifecycle.

[8] Dimensional Research, "COMPANIES FAIL TO LOOK FOR SOFTWARE TAMPERING: A Global Survey of Security, Technology Professionals, and Executives" (February, 2022)

During which of the following software development lifecycle phases does your company check for software tampering?



Note: Only 7% of participants selected all 5 phases options

In fact, of those professionals surveyed about checks for software tampering, a shockingly small share (7%) 'checked every box,' indicating that they look for evidence of tampering at each phase of the software development lifecycle[9].

While scans for tampering were fairly common during the build process (53%) and after build but prior to deployment (43%), much lower percentages of survey respondents said they scanned code post deployment (34%) or that they scanned individual components prior to build (33%). As recent supply chain compromises like SolarWinds, and CodeCov indicate, such spotty checks leave a great deal of room for threat actors to operate[10], exploiting publishers' privileged access to customer environments to push malicious executables or exfiltrate sensitive data. Research done by ReversingLabs, for example, showed how the SolarWinds attackers tampered with the software to include a malicious backdoor in an upcoming SolarWinds Orion software update, which specifically targeted the build/pre-distribution stage[11].

Improvement in this area also means detecting software tampering not just on internally developed code, but for the various components used to make a company's software: internal development, open source software, and third-party software. This is a hefty task that prioritizes security in the development process, which will in turn shore up the defensive measures that organizations can take against threat actors looking to pull off a software supply chain attack.

Software companies know that they can do better, and they want to. But they lack the bandwidth and guidance to take the next step in improving defense measures across the board.

[9] Dimensional Survey

[10] Dimensional Survey

[11] https://blog.reversinglabs.com/blog/sunburst-the-next-level-of-stealth

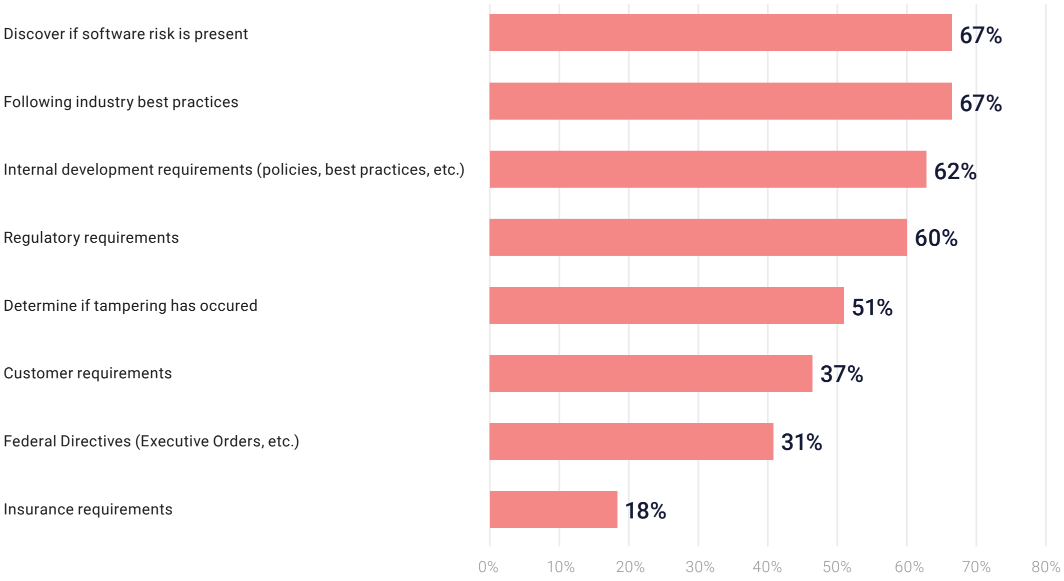
Fears About Complexity Keep SBOM Adoption Low

The use of Software Bill of Materials (SBOM) is a critical step to improving the integrity of software supply chains. SBOMs, for example, allow software publishers to provide a standardized and consumable list of ‘ingredients' for published software binaries. They allow publishers and downstream consumers of software to identify the impact of- and respond to security issues, such as the appearance of remotely exploitable vulnerabilities like Log4Shell.

SBOMs can also be used to help detect software tampering. For example, software publishers can sign SBOM files cryptographically. Downstream organizations can then check the hash values of the signed SBOM files at run time or as part of security assessments to make sure that the software delivered to them has not been modified on its journey from the publisher to the consumer.

With the backing of the Executive Branch and federal regulators, SBOMs are beginning to be looked at seriously by firms that develop applications and services. Asked about reasons to generate and review SBOMs, more than two thirds of the professionals we surveyed cited the need to identify software risk and follow industry best practices, for example.

What are your company's drivers to generate and review a SBOM?

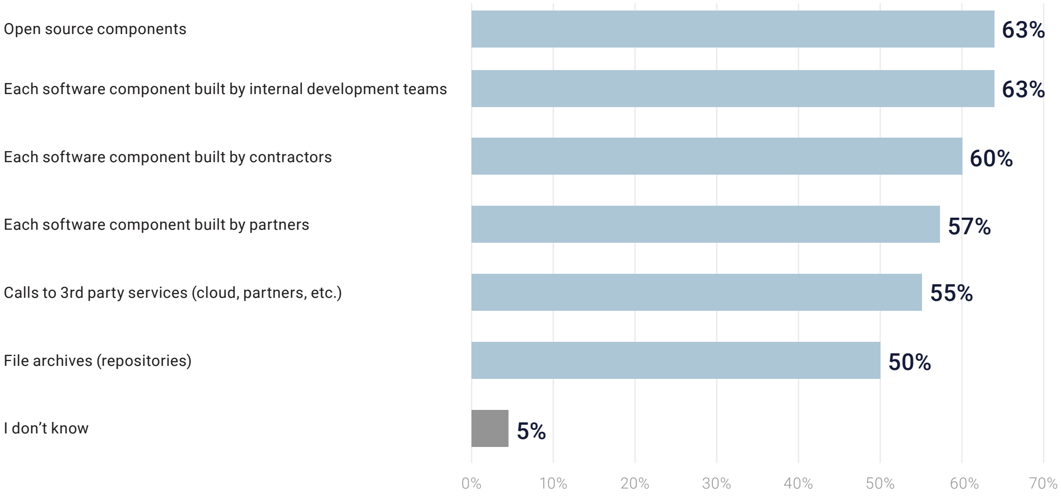


However, our survey also underscored the reality: adoption of SBOMs by software publishers is still meager and efforts to increase use of SBOMs may be hampered by concerns about management and complexity.

Focus On Internal, Open Source Software

For example, only 27% of the IT professionals we surveyed said their employer generates and reviews SBOMs prior to releasing software. Of those, almost two thirds (63%) said that open source components and internally developed components were reviewed. Slightly smaller percentages cited software components built by contractors (60%) and partners (57%) as the focus of their review. A little more than half (55%) said third party services (55%) and file archives (50%) were part of their SBOM review.

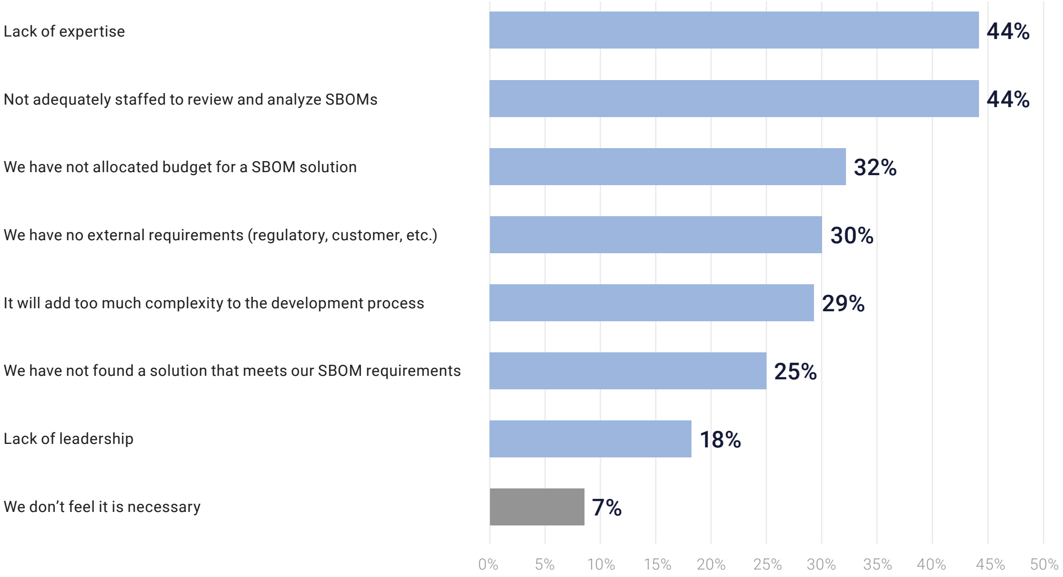
At your company, what elements are specifically reviewed in a SBOM?



Lack of Expertise, Staff, Budget Cited At Firms That Don't Generate SBOMs

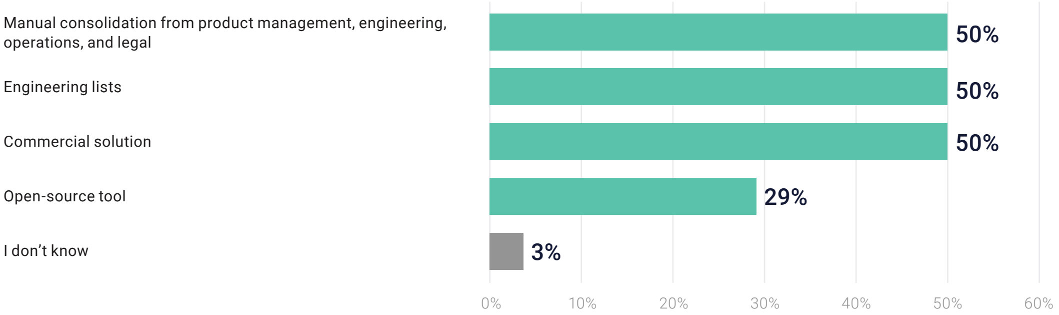
Half of the professionals we surveyed admitted their company does not generate an SBOM. Of those, large percentages (44%) cited a lack of expertise and staffing needed to do so. Lack of budget for implementing SBOM was also cited as a contributing factor by almost a third of respondents (32%). Just 7% of respondents at companies that don't produce SBOMs said the reason was that an SBOM wasn't needed.

Why does your company not generate and review an SBOM?



Feelings about the impediments to implementing SBOMs may reflect the persistence of outdated and manual processes for integrating SBOM data. For example, when we asked our respondents about their process for generating an SBOM, half responded that they were generated manually from information provided by product engineering, operations and legal departments. Another 50% said they used a commercial solution for generating their SBOM.

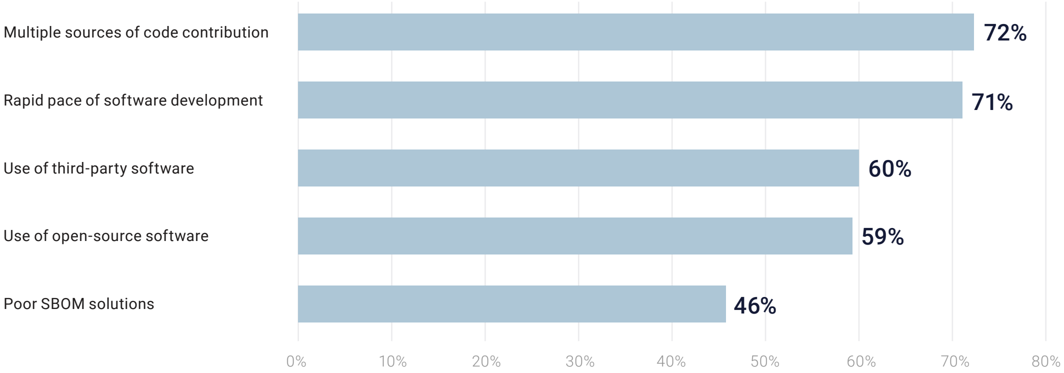
What is your process or technology for generating an SBOM?



And respondents were overwhelmingly of the opinion that the work of creating SBOMs was getting harder not easier. When asked, fully 90% of respondents to our survey said that creating and reviewing SBOMs is growing more complex, not less.

The increasing speed and complexity of software development processes is the biggest driver of SBOM complexity, according to our respondents. Large majorities cited the "rapid pace of software development" (71%) and the existence of "multiple sources of code contribution" (72%) as making SBOM creation and review more complex. The use of third party (60%) and open source (59%) components were also commonly cited as contributing to SBOM complexity.

Why is creating and reviewing a SBOM becoming more complex?



Whatever the benefits of SBOMs, the industry's shared judgment that they are time consuming may explain why adoption of SBOMs remains low. There is clearly a need for a shared understanding of SBOM best practices, processes and tool use if the industry wants to better protect itself from software supply chain attacks.

Conclusion

The results of the survey gave us reason for optimism - but also concern. On the one hand, software publishers are aware of the risks posed by software supply chain attacks, given the growing reliance on open source and third party software. Almost everyone we surveyed said third party and open source code contributed to their cyber risk, while around two thirds of the professionals surveyed said these elements posed the biggest risk to their firms.

But Dimensional Research's survey of IT professionals working within organizations engaged in development organizations show that gaps exist in the ability of publishers to detect and respond to supply chain attacks and software tampering. Fewer than 10% of respondents who took our survey said they had an easy way to detect and block attacks at each stage of the development and release process - a major gap in current protections. More than half of those we surveyed admitted that the inability to detect software tampering poses a risk for their companies.

Just as worrying, the survey found that software publishers are regularly publishing vulnerable code. Almost two third of respondents to the Dimensional survey admitted that software their firm published on a quarterly basis was subsequently found to contain vulnerabilities following either internal or external code review. The combination of lax secure development practices and an inability to detect tampering and supply chain attacks suggests that malicious actors will continue to find fertile ground attacking or compromising enterprise software applications and services.

Fortunately, efforts to address supply chain risk are afoot. In the U.S., a 2021 Executive Order (#14028) on Improving the Nation's Cybersecurity (PDF) prompted the National Institute for Standards and Technology (NIST) to publish a Secure Software Development Framework for software publishers selling to federal government agencies. The Frameork includes a series of requirements that producers of commercial off-the-shelf and government-off-the-shelf software have to meet if they are licensing their products and services to federal agencies. Among those requirements are that comprehensive and current SBOMs are available for "all classes of software" used by the federal government, including purchased, open source and internally developed software.

NIST also revised their Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations, which provides guidance to federal agencies and contractors for identifying, assessing, and responding to cybersecurity risks throughout the supply chain at all levels of an organization.

However, government issued mandates, guidelines and frameworks are not enough. As the survey indicated, significant impediments exist to implementing secure development and software supply chain directives. For example, IT professionals within development organizations say they appreciate the utility of software bills of materials (SBOMs) in identifying supply chain risk, but lack the budget, staff and expertise to employ them. The result: fewer than a third of respondents (27%) worked for organizations that used SBOMs to manage software supply chain risk.

In short: software development organizations don't just need orders and suggestions to address the next generation of supply chain and software tampering attacks; they need help.

Tools from firms like ReversingLabs can help both software publishers and their customers become attuned to supply chain risks: inspecting the discrete software components that make up modern applications for signs of tampering and malicious components such as backdoors and malware.

ReversingLabs has unmatched expertise in malware analysis and software supply chain risk. We're thinking about the big challenges that lay ahead for software development organizations, and look forward to discussing our viewpoints and helping our customers and the larger IT community reduce organizational software supply chain risks.

Don't hesitate to contact us if you'd like to learn more. You can use the button below to schedule a meeting!

REVERSINGLABS

Additional Resources:

Why Malware Detection Isn't Enough Protection Against Software Supply Chain Attacks

Download Solution Brief

REVERSINGLABS SOLUTION BRIEF

Malware detection isn't enough to stop software supply chain attacks

Software development and supply chains are the new battleground for sophisticated adversaries. Development teams need to up their game on release acceptance testing to prevent compromises.

PAGE 1 / 3

The common wisdom in information security circles is that cyber defenses need to "shift left" - moving from post-deployment "layered protections" into the application development process. Compromises of vendors like SolarWinds and CodeCov make clear that software development organizations and software supply chains are the new battlefield on which sophisticated cyber criminal and nation state groups are operating.

In response, organizations developing software are looking to fortify their CI/CD systems and workflows against supply chain threats and risks. As incidents like the SolarWinds "SunBurst" attack show, these risks, if undetected, can severely impact customers and damage both bottom lines and corporate reputations.

One common approach to supply chain risk is to simply extend endpoint- and perimeter anti-malware scanning tools into the development process. Anti-malware engines that are already scanning email attachments or file downloads can also inspect release packages and container images produced by development systems and workflows. That's a sensible approach - and malware scanning of development binaries can reduce cyber risk. However, the range of attack types and threat models observed in known supply chain attacks make clear that relying on anti-malware alone will not address some of the biggest cyber risks to development organizations, leaving them with little real protection against cyber adversaries.

Here are a few reasons that antivirus isn't enough to secure your development process:

KNOWN MALWARE IS A FRACTION OF SOFTWARE SUPPLY CHAIN RISK

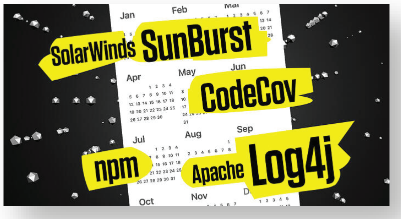
There are millions of known malware families circulating on the Internet. But known malware accounts for a fraction of software supply chain compromises. That means anti-malware engines are likely to miss the vast majority of non-malware attacks. For example, attackers like Cozy Bear, the Russian APT group behind the SolarWinds hack, masked their attack using behavioral changes that mimicked developer's coding styles. Such changes are invisible to malware detection and have proven to be very hard to spot even with code reviews and vulnerability scans. Additionally, nation-sponsored hacking groups have the time and resources to develop and test novel or company-specific attacks to evade detection based on understanding malware families used in past attacks.

YOU WANT TO STOP ATTACKS BEFORE MALWARE IS PLACED IN SOFTWARE, NOT AFTER!

The attack on CodeCov's customers began with developer credentials that were left exposed in a container image (see Table 1). Assessing container images for exposed secrets like credentials or private keys before release provides the opportunity to stop data theft and avoid future CI/CD or build system compromises. But anti-malware tools aren't designed to conduct such assessments or detect exposed secrets, making this important early detection impossible.

A look back at 2021: The year supply chain threats went mainstream

Read Blog



ReversingLabs supports many languages and repository packages to deliver software supply chain protection for CI/CD workflows, containers and release packages.

docker .NET python Package Index Maven JS npm

What You Need to Know: NIST's Secure Software Development Framework

Watch Video

REVERSINGLABS
What You Need to Know:
NIST's Secure Software Development Framework
REVERSINGLABS CHIEF SOFTWARE ARCHITECT TOMISLAV PERICIN DISCUSSES THE RECENT NIST GUIDANCE ON SECURE SOFTWARE DEVELOPMENT AND SOFTWARE BILL OF MATERIALS
Tomislav Pericin
CHIEF SOFTWARE ARCHITECT & CO-FOUNDER AT REVERSINGLABS
Paul Roberts
CYBER CONTENT LEAD AT REVERSINGLABS

Not all SBOMs Are the Same. Choose Wisely!

Read Blog

Contact ReversingLabs to learn more about malware analysis solutions

REQUEST A DEMO

Worldwide Sale:
+1.617.250.7518
sales@reversinglabs.com

REVERSINGLABS