ЯEVERSINGLABS

ReversingLabs NVD Analysis 2022: A Call to Action on Software Supply Chain Security

FLAWS IN OPEN SOURCE ARE CONTRIBUTING TO A SHARP RISE IN REPORTS TO THE NATIONAL VULNERABILITY DATABASE IN 2022. BUT EMERGING SOFTWARE SUPPLY CHAIN ATTACKS WARRANT A RE-THINK OF THE NVD—AND YOUR SOFTWARE SECURITY APPROACH—THAT GOES BEYOND COMMON SOFTWARE VULNERABILITIES

Contents

Executive Summary

Vulnerability reports to MITRE Corp's Vulnerabilities and Exposures (CVE) list, part of NIST's National Vulnerability Database (NVD), are accelerating. New vulnerabilities in the first half of 2022 outstripped the same period in 2021. At the current rate, more than 24,000 vulnerabilities will be added to the NVD this year—breaking last year's record of slightly more than 20,000.

Meanwhile, analysis of the data conducted for ReversingLabs by Lemos Associates suggests that the jump in vulnerabilities in recent years is likely to continue, as more private and public sector organizations take part in the CVE program as CVE Numbering Authorities (CNAs) and as open source- and third-party code attract the interest of both security researchers and malicious actors.

Seen in the context of rising tide of software supply chain attacks, the growth in reports to the NVD suggest that the focus of malicious actors is shifting. And yet, the NVD is still dominated by flaws in a handful of legacy platforms by firms including MIcrosoft, Red Hat, Google, Apple and Oracle.

This rise in software supply chain attacks is a call to action for NIST. NVD is a critical resource for both software development and security organizations. To remain relevant, however, the scope of NVD needs to expand to capture the full breadth of vulnerable platforms and applications, as well as the diversity of security exposures (the "E" in CVE)—including malware injections, software tampering and secrets exposure, which threaten supply chain integrity.

Such a shift would empower security and software development teams responsible for software security to likewise expand the scope of their security approach to address software supply chain exposures that are currently being overlooked.

SECTION 1

Vulnerabilities Surge, Leaving Companies Scrambling

ReversingLabs' analysis of vulnerability reports to MITRE Corp's CVE list shows a continuing trend of rapid growth that began five years ago, in 2017, when the number of vulnerabilities assigned an identifier in NIST's Common Vulnerabilities and Exposures (CVE) database more than doubled from the previous year.

That jump coincided with an increase in the number of CVE Numbering Authorities (CNAs), after MITRE began a program of delegating the assignment of CVEs to other prominent technology and security firms, as well as industry groups. In each successive year, vulnerability reports have ratcheted higher, adding an average of 1,400 reports each year over the past four years.

So far, in 2022, the disclosure of vulnerabilities jumped even higher. In fact, if the number of reports in the second half of 2022 keeps pace with the first half of the year, software companies and maintainers will likely have to contend with almost 24,500 reports in 2022—a 22% increase over the previous year.

That steady growth poses a challenge for security and development teams alike.

"Over 24,000 is a daunting number—it's tough to make headway against that sort of number," says Chris Romeo, co-founder and chief security officer for Security Journey, a provider of application security training. Fortunately, all of those vulnerabilities won't be remotely exploitable.

> How many of those issues are buried deep within a library, and the code is never accessible when included in a modern application?

Chris Romeo
Co-founder and chief security officer of Security Journey

Still, the sheer volume of vulnerabilities means that companies that aim to patch every vulnerability are going to struggle to keep ahead of the tide of new vulnerabilities. Not only do organizations have to determine which of the tens of thousands of vulnerabilities need to be patched, but they also need to address any backlog of vulnerabilities from previous years. In many ways, trying to keep up with the growing number of vulnerabilities can be an exercise in futility.

VULNERABILITY TRENDS: MORE SOFTWARE, MORE HOLES

What is driving the surge in vulnerability reports? The answer to that question is complicated, but it is safe to say that the growing number of vulnerabilities in the NVD does not represent a relative decline in the security of software. Rather, it speaks more to the scope and management of the CVE list and NVD.

Over the last two decades, vulnerabilities reached a number of plateaus. Before 2005, the number of vulnerabilities assigned a CVE identifier never passed 2,500, and for more than a decade after that, disclosed issues fluctuated between 4,000 and 8,000 reports. Rather than a measure of the true number of vulnerabilities discovered by researchers and used by attackers, the number of CVE identifiers was often limited by the capabilities of the MITRE CVE team.

In 2016 and 2017, MITRE began inviting more organizations to report vulnerabilities, designating them as CVE Number Authorities (CNAs). Thereafter, the number of vulnerabilities surged—doubling in 2017, and surpassing the previous years every subsequent year. Again, this surge wasn't representative of a marked decline in software quality, but instead loosening access to the system by which CVEs were assigned.

Vulnerability reports to NVD from 2002 to 2022

Source: CVE data from the National Vulnerability Database as of July 10, 2022

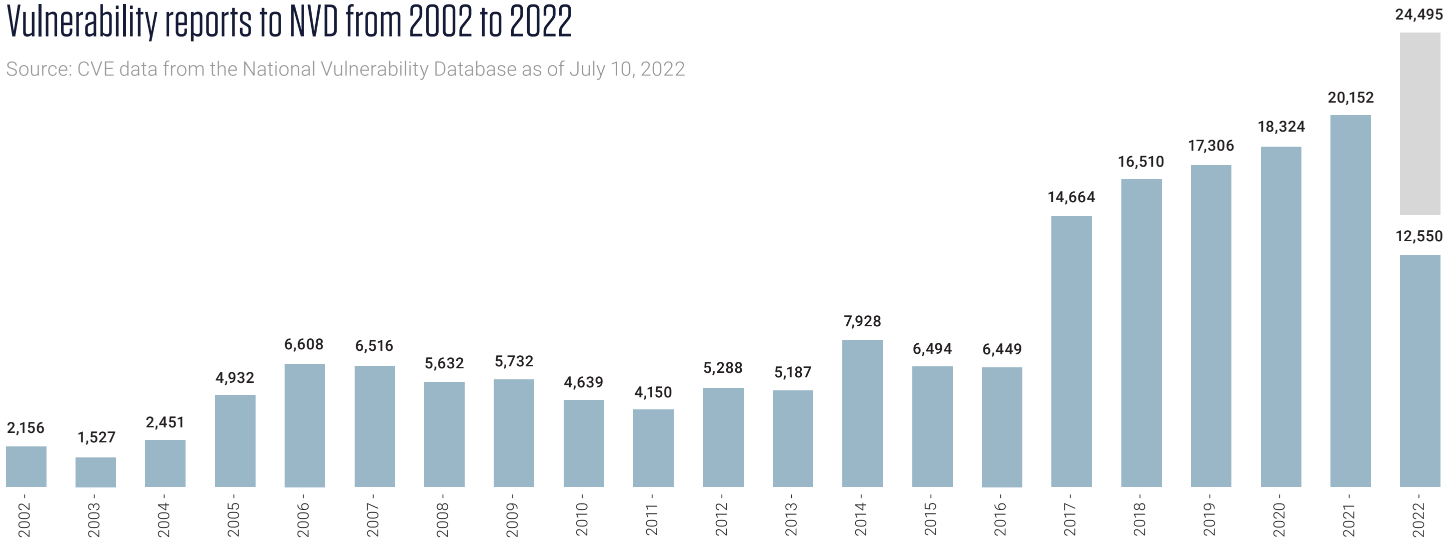


Figure 1. Following the expansion in CVE Numbering Authorities (CNAs) in 2017, vulnerability disclosures took off. By the first half of 2022 vulnerability reports were at 12,550. If the current reporting trend continues, the number of disclosures will have jumped by 22% over the previous year, to 24,495.

The expansion of CNAs continues in 2022, with more coming CNAs representing a broader swath of the software development and security research community.

Monthly CVE volume compared to CNA count

Source: CVE data from the National Vulnerability Database as of July 08, 2022. CNA data collected from Internet Archive versions of cve.mitre.org/cve/cna.html and cve.org

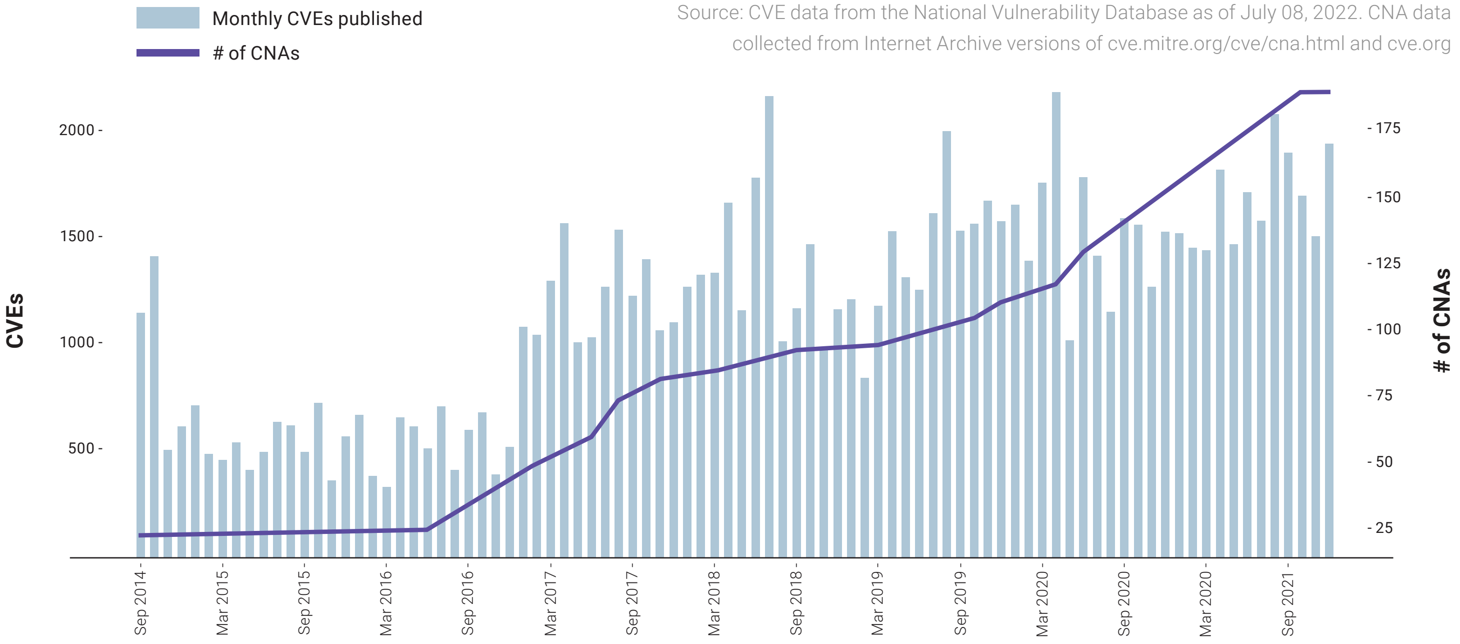


Figure 2. The number of monthly CVEs has grown with the number of CVE Number Authorities (CNAs). There are now 240 participating organizations.

Our analysis suggests that the lion's share of the growth in disclosed vulnerabilities is a function of better reporting, as MITRE distributed the workload for vetting vulnerabilities. As Figure 2 shows, the number of monthly CVEs has grown with the number of CNAs, of which there are now 240 participating organizations.

The number of reported vulnerabilities is directly related to the number of products covered, as shown in Figure 3. In other words, the jump in the number of vulnerabilities witnessed over the last five years did not happen in a vacuum, but arrived as more software companies began working with MITRE, followed by a commensurate increase in the number of products covered.

Total vulnerabilities as a function of number of products

Source: CVE data from the National Vulnerability Database as of July 10, 2022

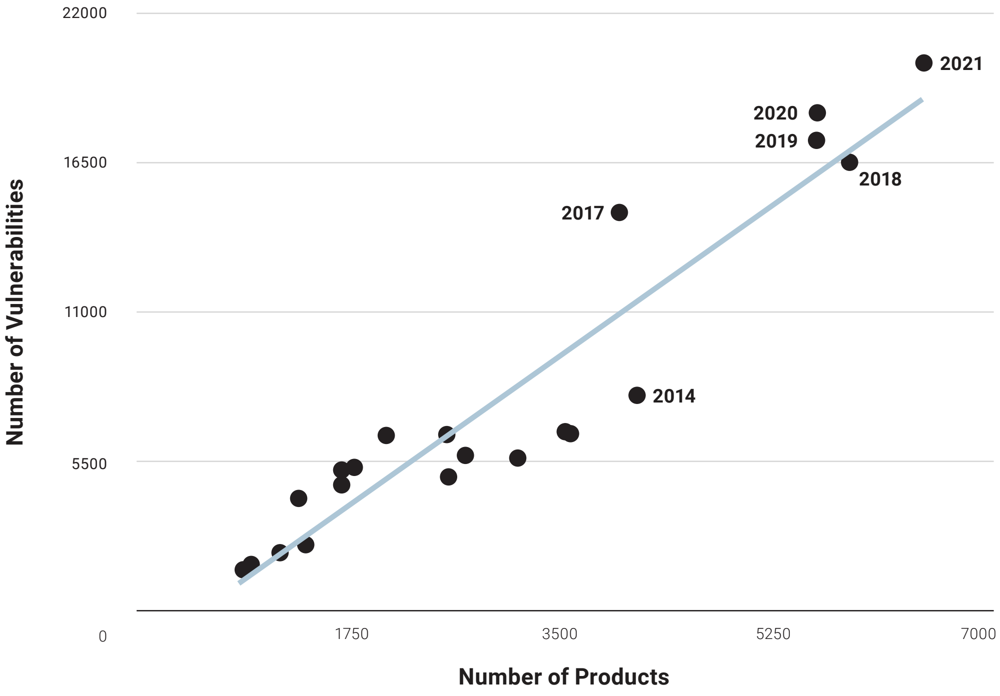


Figure 3. Over the past two years, as the number of covered products increased, so did the number of reported vulnerabilities.

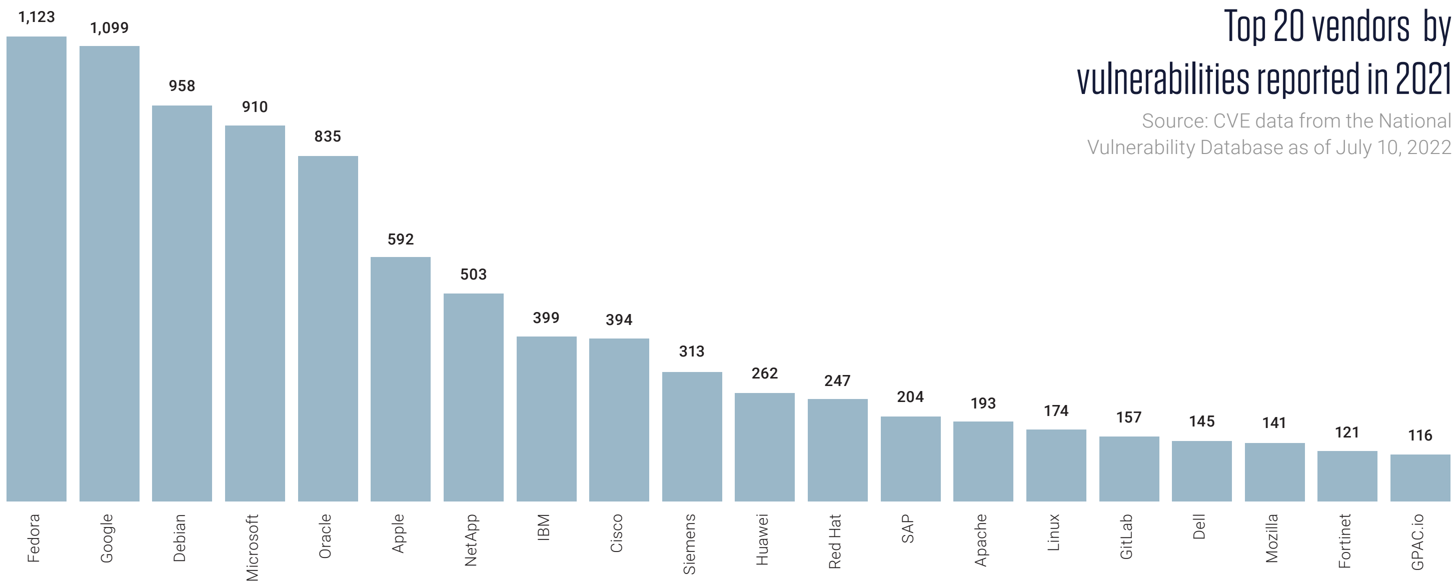
WHY VULNERABILITY COUNTS ARE NOT THE WHOLE STORY

The nearly four-fold growth in reports to the NVD since 2016 doesn’t mean that malicious cyber actors have four times the number of targets in 2022 as they did six years ago. That’s because many of the CVEs assigned under the new system correspond to minor flaws and application bugs that have “no practical security impact,” says ReversingLabs co-founder and Chief Software Architect Tomislav Peričin.

By opening the doors to other organizations to assign CVEs, MITRE enabled the NVD to scale, but that has also meant the organization is more accepting of requests to get a CVE assigned than was the case when all CVE requests were routed through MITRE itself, Peričin said.

ENTERPRISE SOFTWARE MAKERS, LINUX DISTRIBUTIONS TOP VENDOR LIST

As for the vulnerabilities reported in the first six months of 2022? Most accrue to a short list of major software vendors. Linux distributions Fedora and Debian account for 1,123 and 958 vulnerabilities, respectively, and rank first and third on the list of software firms affected by reported issues. Google, Microsoft, Oracle and Apple accounted for more than 500 vulnerabilities each.



Top 20 vendors by vulnerabilities reported in 2021

Source: CVE data from the National Vulnerability Database as of July 10, 2022

Figure 4. Large software companies and popular Linux distributions continue to be targets.

That’s in line with patterns of vulnerability reporting in prior years. With more than 3,500 vendors reporting vulnerabilities, most software development teams only deal with a few to tens of vulnerabilities. The top-20 vendors account for about 8,000 vulnerabilities, just one third of the total in 2021, underscoring the long-tailed distribution of CVEs (and by extension, affected companies, organizations, and projects) that make up the current software ecosystem.

As for attacks: fewer than 800 vulnerabilities are currently being exploited by attackers across all disclosed CVEs, according to the U.S. Department of Homeland Security’s Cybersecurity and Infrastructure Security Agency (CISA). By far, Microsoft products are the most targeted software (see Figure 4), with 234 vulnerabilities exploited by attackers, while Adobe's applications come in a distant second place with fewer than 60 issues targeted by attackers.

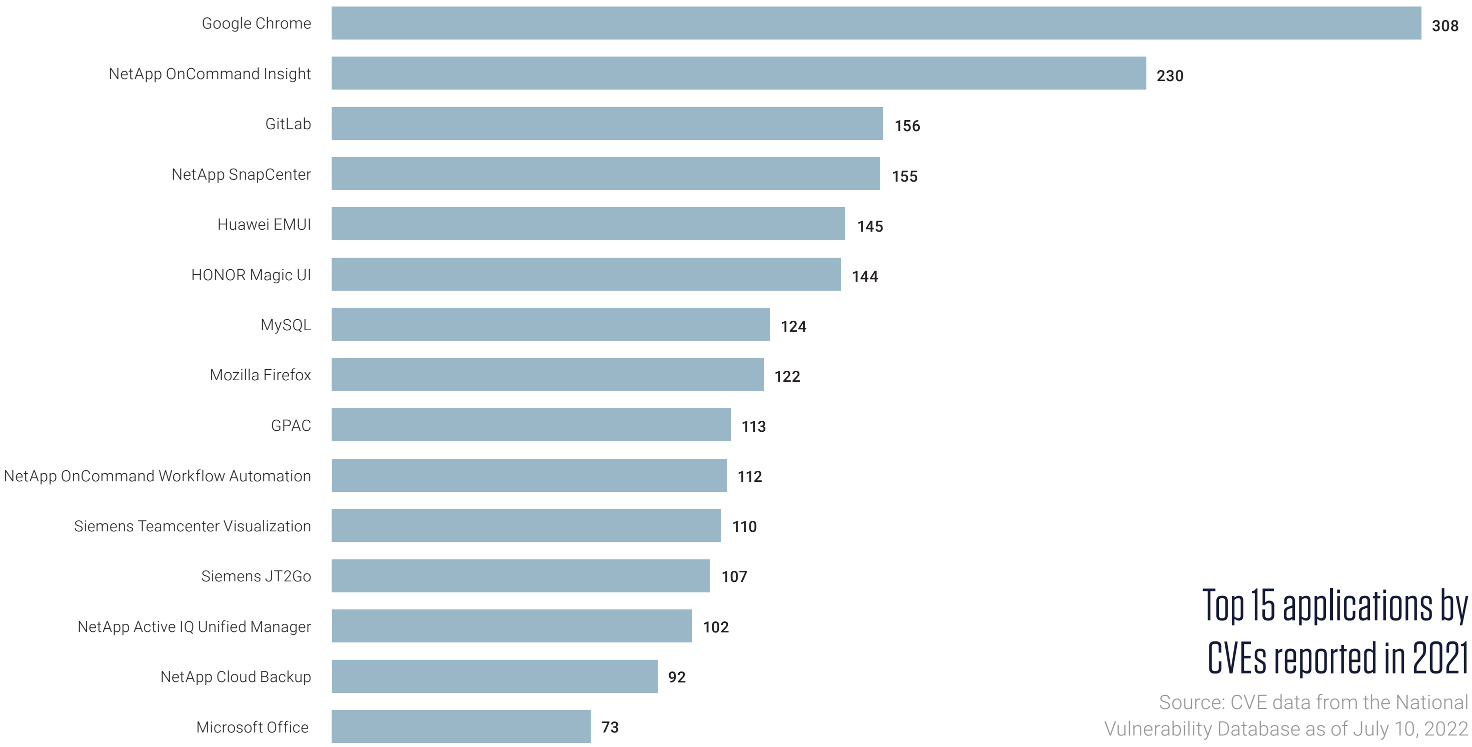
But there is evidence that the ground is shifting. For example, NetApp, an enterprise software vendor, and Siemens, a maker of industrial control software, both ranked among the top 20 vendors based on the number of reported vulnerabilities, as security researchers scrutinized their products.

GITLAB AND GITHUB AND JENKINS…OH MY!

GitLab is another notable addition to the list of vendors with the most vulnerabilities. The popular DevOps platform provider was the (non-OS) application with the third highest number of vulnerabilities reported, behind Google Chrome, but ahead of both Mozilla Firefox and Microsoft Office.

The company is a relative newcomer to the list of vendors with the most reported vulnerabilities. It is also one of the new CVE Numbering Authorities (CNAs). GitLab's presence on the list of top applications underscores the impact of security researchers (and, presumably, attackers) broadening their horizons and focusing on finding ways into the enterprise. Those forays include assessments of not just open source modules, but also the DevOps tools and platforms that are integral to most software development teams these days.

The company sits at that nexus, while the scope of its position as a CNA, which encompasses both vulnerabilities in the GitLab platform and "any project hosted on GitLab.com in a public repository, and any vulnerabilities discovered by GitLab that are not in another CNA's scope," ensures that vulnerabilities in public GitLab hosted projects will receive CVEs and be reported and added to the NVD. GitLab joins several other DevOps platforms and applications as CNAs, including Microsoft-owned GitHub, Jenkins, Node.js and more.



Top 15 applications by CVEs reported in 2021

Source: CVE data from the National Vulnerability Database as of July 10, 2022

Figure 5. Google Chrome tops the list, but the list has shifted to less common software as security researchers shift their focus to enterprise applications.

OPEN SOURCE FLAWS FEED SOFTWARE SUPPLY CHAIN ATTACKS

Indeed, focusing on vulnerabilities in commercial and in-house products excludes a growing source of insecurity: the open-source software libraries and components that form the foundation of, by some estimates, 75% of applications. More than 90% of applications use at least one open-source component.

Despite that, many software developers do little to track the vulnerabilities in the components and libraries on which their applications depend. In the case of Apache Log4J 2, for example, the vulnerabilities discovered in December 2021 will be around for years because developers are slow to update their applications. And Log4J is often not imported directly into a software project but from a component that is five layers deep, an analysis conducted by Google found. The company found some dependency chains for the component descended through nine different code imports before researchers found the library that added Log4J.

For attackers, vulnerabilities in such dependencies give them the ability to exploit a far wider swath of applications than targeting a single codebase. Google originally found nearly 36,000 programs that used Log4J, potentially making those applications vulnerable. With such a force multiplier, common—and often overlooked—software components are becoming a favored target of attackers.

A report by the U.S. Cyber Safety Review Board (CSRB) warned that the threat posed by Log4J will linger for decades, with big implications for development organizations and those who maintain open source projects. CSRB recommended that development organizations ramp up secure software development practices by adhering more closely to standards like ISO 27034:2011160 and NIST's Secure Software Development Framework, while also embracing practices like Software Bills of Materials (SBOMs).

Open source maintainers should also step up their games: making greater use of source code scanning tools and formalizing communications with researchers and the broader community around security issues, CSRB said in its report (PDF).

SOFTWARE SUPPLY CHAIN WOES WILL DRIVE VULNERABILITY COUNTS HIGHER

Already, efforts are afoot to stem the flood of vulnerabilities and other security issues in open source code. The Linux Foundation and its Open Software Security Foundation (OpenSSF) along with the U.S. government, and private companies, such as Google and Microsoft, are focusing on providing resources to at least 1,000 critical open-source projects.

An immediate side effect of such efforts will be to better identify and document vulnerabilities in these projects, which will expand the number of products covered by CVE programs and increase reported vulnerabilities. In fact, the greatest room for expanding the CVE program lies in the coverage of the critical projects that make up the software supply chain—and that expansion will fuel a rise in yearly vulnerabilities for the near future.

That’s especially true as more popular and widely-used CI/CD platforms like CodeCov, CircleCI and Bamboo join the likes of GitLab, GitHub and Jenkins by becoming CNAs. As incidents like the supply chain attack on CodeCov illustrate, vulnerabilities and exposures in these critical platforms can have wide-reaching implications for the broader open source and developer community. In that attack, an error in CodeCov’s Docker image creation process allowed a malicious actor to extract the credential required to modify the company’s Bash Uploader script. The compromise of CodeCov resulted in a massive supply chain attack against CodeCov customers, including the theft of developer credentials, and so on.

OpenSSF is encouraging the use of features such as two-factor authentication to protect maintainer accounts from hijacking attacks. At the same time, projects like sigstore.dev, a collaboration between OpenSSF, Cisco, Google, Red Hat, HP and other firms, is promoting technology to streamline digital code signing to verify the authenticity of software supply chains that use open source components.

SECTION 2

Attackers Shift Their Focus to the Software Supply Chain

In the past, exploits tended to focus on standalone applications and operating systems running on desktops, laptops, and servers. Since 2017, the focus of malicious threat actors has shifted from attacking vulnerabilities in desktop, mobile or web applications to the software components used to develop all of those programs.

In March 2020, for example, researchers discovered that attackers infiltrated the development environment of network management firm SolarWinds, placing a backdoor in the company's software that infected more than 18,000 companies as the company regularly updated its systems.

Over the past few years, attackers have also used a variant of typosquatting (registering common misspellings of a target organization's domain) to create malicious packages that have a name similar to popular libraries—a technique known as dependency confusion. In 2017, for example, the Python Package Index (PyPI) discovered 10 different malicious programs posing as common software dependencies, after being warned by the Slovakian Computer Security Incident Response Team (CSIRT).

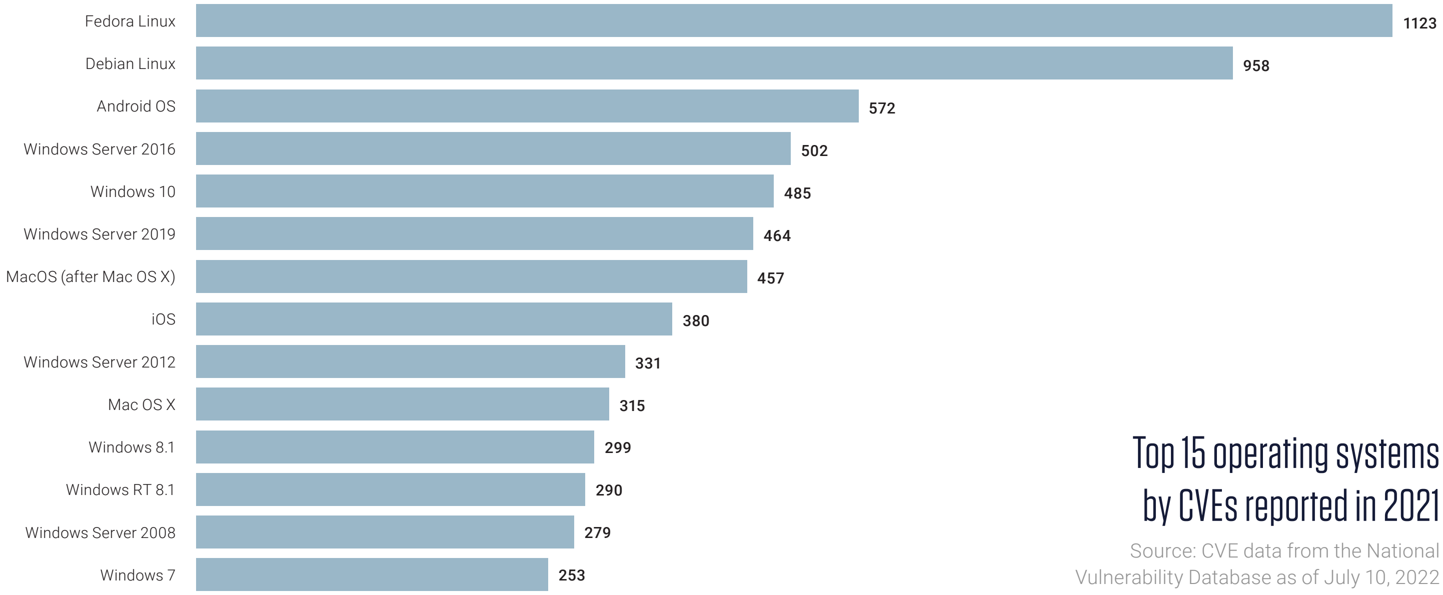
While this shift adds complexity to the work of the vulnerability hunter, the “upside” is considerable. Instead of finding one-off vulnerabilities in specific products, researchers or malicious actors who can find and exploit flaws in software components and infrastructure find they can undermine the security of many software programs and services that rely on them.

> As an attacker, I'm going to go after enterprises to see if they have exposed instances of their development pipeline. We need to get the right tools and find the problems in our own supply chains. We, as an industry, are not using the tools consistently and correctly.

Chris Romeo
Co-founder and chief security officer of Security Journey

MIND THE (LONG) TAIL OF VULNERABILITIES

Looked at from the perspective of assigned CVEs and vulnerability reports, the shift under way in the source of CVE reports to NVD may be hard to appreciate. A glance at the software flaws documented in the NVD for 2022 looks like more of the same. A hefty portion of vulnerabilities continue to be found in the most popular operating systems, such as the major Linux distributions, Android and iOS, various flavors of Windows, and Apple's MacOS.

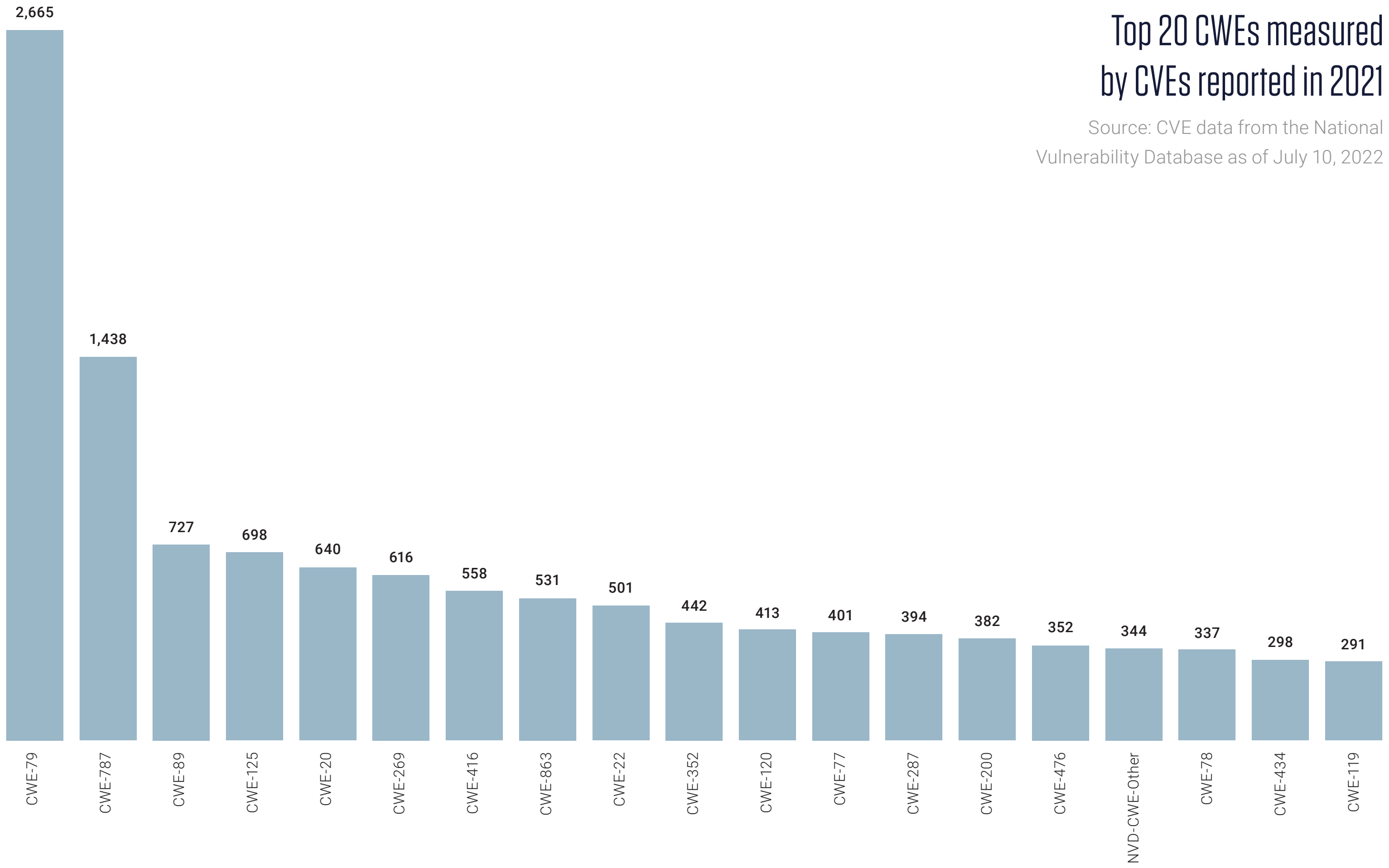


Top 15 operating systems by CVEs reported in 2021

Source: CVE data from the National Vulnerability Database as of July 10, 2022

Figure 6. Unsurprisingly, the most common operating systems—Linux, Android, Windows, MacOS and iOS—attract the most research and disclosures.

Ubiquitous flaws like cross-site scripting (XSS) flaws (CWE-79); SQL injection weaknesses (CWE-787); path traversal (CWE-22) cross-site request forgery (CWE-352); and buffer overflows (CWE-119) all continue to hover at the top of the list of identified vulnerabilities.



Top 20 CWEs measured by CVEs reported in 2021

Source: CVE data from the National Vulnerability Database as of July 10, 2022

Figure 7. While most vulnerabilities disclosed each year are easily fixed, a growing number of less common issues, such as improper privilege management, are increasing. [See the full description for each CWE in the index at the end of this article.]

Dig a bit deeper, however, and new trends can be spotted in the long tail of reported vulnerabilities. These include classes of vulnerabilities that were less common in past years such as issues with authentication and authorization. Improper privilege management (CWE-269) accounted for 616 software flaws in 2021, while 531 vulnerabilities had their foundation in incorrect authorization (CWE-863).

In addition, Web application programing interfaces (APIs) have become an increasing vector of attack, leading the Open Web Application Security Project (OWASP) to launch an API Security Top-10 list to enumerate the most common security weaknesses discovered in application architectures that use APIs. Authorization and authentication security issues affect APIs as well, with three of the top five weaknesses encompassing those problem areas.

On the issue of vulnerabilities, open-source organizations such as the Linux Foundation and the Open Source Security Foundation (OpenSSF), along with companies that rely on open source, including Google, have banded together to identify and help secure common and critical open-source components.

Under its Alpha Omega project, the OpenSSF aims to identify the most critical projects on which software and internet infrastructure rely, and the organization's Scorecard and Best Practices Badge aim to set security standards for projects, and give them the ability to communicate their adherence to those standards to developers.

However, only a combination of the community improving overall software security, as well as vendors giving companies the tools they need to evaluate the threats posed by software vulnerabilities, will improve software security more broadly.

SOFTWARE SUPPLY CHAIN SECURITY GETS MESSY

But it is also worth noting what can't be found in the NVD data. For example, as scrutiny of both open source and common development tools and platforms grows, the security picture for development organizations and their customers is becoming divorced from issues around specific vulnerabilities (like Log4J), and also increasingly messy.

Vulnerabilities in platforms like GitLab have created openings for impersonation attacks and account takeovers—even ransomware attacks that hold code repositories hostage. Beyond that, account hijacking subsequent to phishing or other attacks on maintainers has stung prominent firms, including GitHub owner Microsoft, and resulted in the theft of proprietary code and sensitive data.

There have also been numerous incidents in which manipulation of open source modules has sown chaos among downstream developers and applications. For example, there are incidents of so-called "protestware," in which maintainers of legitimate applications decide to weaponize their software in service of some larger cause (be it personal or political).

In March, for example, Brandon Nozaki Miller, the developer of node.ipc, pushed an update of his popular open source library that sabotaged computers in Russia and Belarus in retaliation for Russia's invasion of Ukraine (and Belarus's support for that invasion). The new release included an obfuscated function that checked the IP address of developers who used the node.ipc module in their projects. IP addresses that geolocated to either Russia or Belarus saw node.ipc wipe files from their machine and replaced them with a heart emoji, published reports note.

Then in July, the developer Markus Unterwaditzer temporarily deleted code for his popular and widely used atomicwrites Python library from the popular code registry PyPI in protest over mandated two-factor authentication for maintainers of what are deemed "critical" projects—a requirement that is in no small part due to incidents of maintainers' accounts being hijacked and abused. Unterwaditzer said he found the requirement "annoying" and "entitled."

Such incidents make a strong case for the need for increased security and scrutiny of the code hosted on platforms like GitLab, GitHub or npm, that goes beyond research on software vulnerabilities and exposures.

Efforts by OpenSSF and others to promulgate code signing or two-factor authentication are important, but they're not going to solve the problem of trust, says ReversingLabs' Peričin.

> They ignore the most problematic bit of it, which is 'how do you trust code behaviors?' And that's the problem these efforts don't have an answer for: You still basically just have to trust the maintainers.

Tomislav Peričin
Co-founder and Chief Software Architect, ReversingLabs

SECTION 3

Keeping Pace as Software Supply Chain Attacks Mount

Attackers are already taking advantage of weak links in the modern software development pipeline. Attacks on open source repositories, for example, have skyrocketed over the past decade, outpacing vulnerabilities found in those repositories, especially vulnerabilities with a CVE identifier assigned.

SolarWinds, a legitimate software maker, unwittingly allowed a malicious software update to infect its customers. Similar attacks have affected application providers including Kaseya, CodeCov and others. Detecting such changes before they compromise enterprise systems requires a focus on the code and its behavior, rather than whether the code came from a trusted party.

In the past four years, for example, attacks on two popular open-source repositories, npm (for JavaScript) and the Python Package Index (PyPI), both grew dramatically. From 2018 to 2021 attacks on npm and PyPI increased by 271% and 414 %, respectively, or a combined 289%. (see chart in Figure 8, next page).

Growth in Software Supply Chain Vulnerabilities

Source: ReversingLabs compilation of internal CVE/vulnerability data, with data from Snyk's vulnerability database, and the GitHub Advisory Database

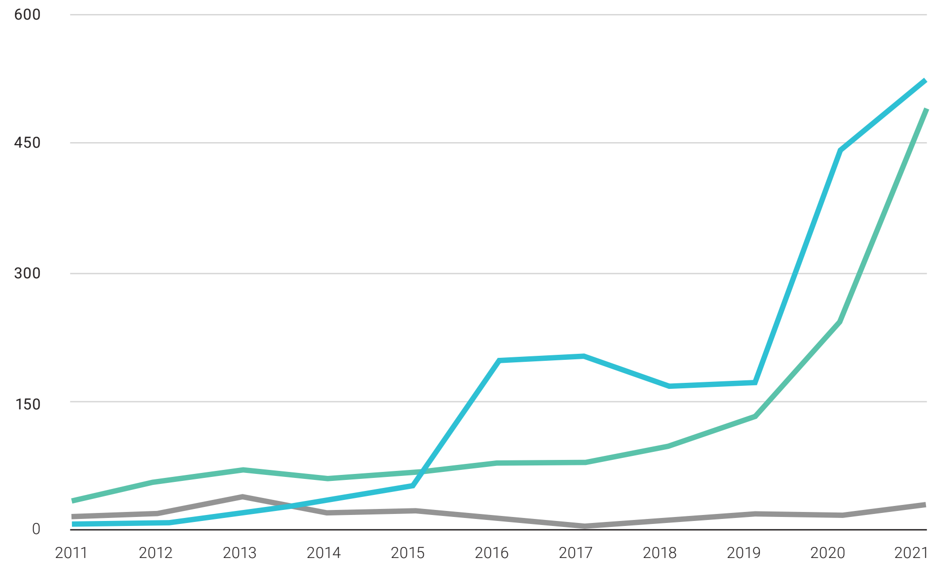


Figure 8. Reported vulnerabilities affecting leading open source repositories.

THE NVD IS OUT OF STEP WITH SOFTWARE SUPPLY CHAIN THREATS

The NVD is expanding as open source increasingly joins the CNA mix. And as noted above, emerging software supply chain threats are an area of concern not reflected in the NVD. However, ReversingLabs' analysis also found the NVD not keeping pace with vulnerabilities from third-party sources. As shown in Figure 9, below, public reports and internal ReversingLabs research shows vulnerabilities outpacing the NVD—and malware/known attacks mounting.

Software Supply Chain Threats

Source: ReversingLabs compilation data from public reports, and internal ReversingLabs research.

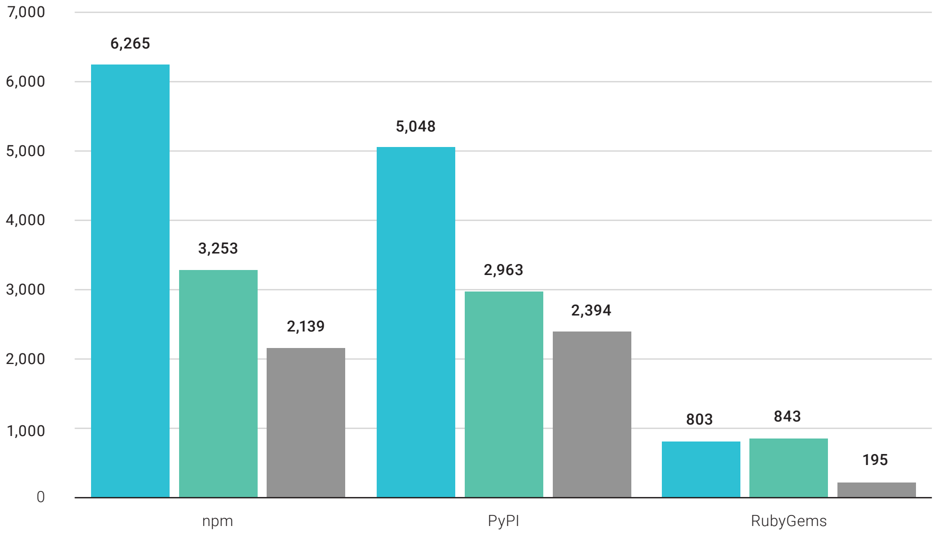


Figure 9. Vulnerabilities listed in the CVE database don't fully reflect software supply chain threats. The blue bar shows malware/known attacks on the supply chain since 2010 for perspective.

SHIFT YOUR SOFTWARE SECURITY FOCUS TO TRUST

What does this disconnect mean for the future of public resources like the NVD? Change, for one thing. While many tools designed to help secure software-development pipelines focus on rating the projects, programmers, and the open-source components and their maintainers, recent events—such as the hijacking of the popular ua-parser-js project by cryptominer—expose that that even seemingly secure projects can be compromised, or otherwise pose security risks to organizations.

ReversingLabs' Peričin says the lesson from emerging software supply chain threats is that software security teams need to expand their focus beyond vulnerabilities alone—and even source code analysis—to what the actual code is doing, and the runtime behavior of the software's components.

> As long as we keep ignoring the core of the problem—which is how do you trust code—we are not handling software supply chain security

Tomislav Peričin
Co-founder and Chief Software Architect, ReversingLabs

A Call to Action for the NVD—and Your Software Security Team

The rise of software supply chain attacks—brought to the fore by attacks like SolarWinds and followed by attention from the White House Executive Order on SBOMs and MITRE's more recent System of Trust—is a clear call to action.

NIST is already advocating for supply chain security. On June 6, MITRE, which manages the CVE list for NIST, put the issue front and center when it announced its System of Trust, a framework that helps organizations protect against vulnerabilities in the software supply chain by standardizing how supply chain security is assessed.

NIST and the Federal government are also slowly pushing forward to implement the White House's year-old Executive Order on Improving the Nation's Cybersecurity which requires all federal government contractors and software providers to create a software bill of materials (SBOM) that can be reviewed.

To keep pace with these larger changes, the NVD public database also needs to evolve. At the very least, its scope should expand to consistently include software supply chain exposures. Only then will the NVD move closer to representing the full breadth of threats facing modern organizations.

Our analysis of submissions to the NVD so far in 2022 underscores how the growing profile of open source modules and CI/CD platforms are inspiring vulnerability researchers and driving submissions to the National Vulnerability Database ever higher. However, while vulnerabilities like Log4J and attacks like those on CodeCov and other firms make the connection between vulnerabilities and supply chain attacks clear, the current NVD public database is out of step with a fast evolving landscape of vulnerabilities and exploits.

True, the delegation of CVE assignment authority to CNAs has greatly improved the fidelity of the NVD, but our research suggests that there is more to be done to enlist prominent CI/CD platform and application providers as CNAs to better capture supply chain vulnerabilities such as those in open source repositories , development tools, and platforms.

Finally, the fact that vulnerabilities from such important software supply chain players aren't in the NVD today shouldn't stop organizations from expanding their security teams' scope to include software supply chain security. Software security is built on a shared responsibility model that requires application development teams, application security teams—and security operations teams —to work together to deliver and run secure software.

ReversingLabs worked with Lemos Associates LLC on the analysis and visualization of CVE data from the National Vulnerability Database.

INDEX

Figure 7. CWE Definitions.

CWE-79	Improper Neutralization of Input During Web Page Generation ("Cross-site Scripting")
CWE-787	Out-of-bounds Write
CWE-89	Improper Neutralization of Special Elements used in an SQL Command ("SQL Injection")
CWE-125	Out-of-bounds Read
CWE-20	Improper Input Validation
CWE-269	Improper Privilege Management
CWE-416	Use After Free
CWE-863	Incorrect authorization
CWE-22	Improper Limitation of a Pathname to a Restricted Directory ("Path Traversal")
CWE-352	Cross-Site Request Forgery (CSRF)
CWE-120	Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
CWE-77	Improper Neutralization of Special Elements used in a Command ("Command Injection")
CWE-287	Improper Authentication
CWE-200	Exposure of Sensitive Information to an Unauthorized Actor
CWE-476	NULL Pointer Dereference
NVD-CWE-Other	Other
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
CWE-434	Unrestricted Upload of File with Dangerous Type
CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer

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.

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

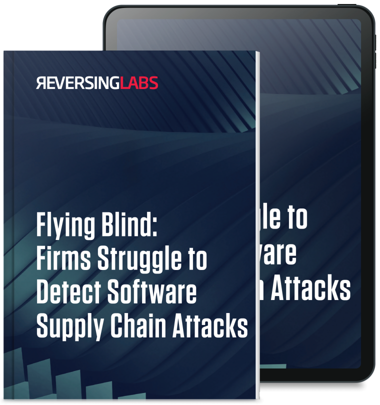
There are millions of known malware families circulating on the Internet. But known malware accounts for a fraction of software supply chain compromises. That means anti-malware engines are likely to miss the vast majority of non-malware attacks. For example, attackers like Cozy Bear, the Russian APT 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!

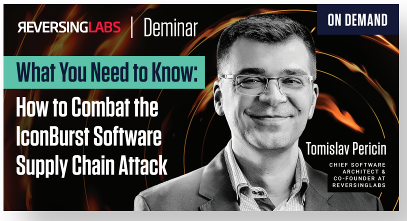
The attack on CodeCov's customers began with developer credentials that were left exposed in a container image (see Table 1). Assessing container images for exposed secrets like credentials or private keys before release provides the opportunity to stop data theft and avoid future CI/CD or build system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

PAGE 1 / 3

Flying Blind: Software Firms Struggle
To Detect Supply Chain Hacks
Download Report



What You Need to Know: How to Combat
the IconBurst Software Supply Chain Attack
Watch Deminar



Not all SBOMs Are the Same.
Choose Wisely!
Read Blog

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

Learn how ReversingLabs secure.software can help you perform critical security checks before you deploy software

LEARN MORE

Worldwide Sale:
+1.617.250.7518
sales@reversinglabs.com

REVERSINGLABS