REVERSINGLABS

4th Annual

Software Supply Chain Security Report 2026

How Sophisticated Malware, AI, and Broken Trust are Reshaping Software Security

Table of Contents

A Message from Our CEO

Software Supply Chain Trust Is Broken. Here's What Comes Next.

Mario Vuksan
CEO AND CO-FOUNDER,
REVERSINGLABS

Software supply chains have quietly become one of the most consequential — and contested — domains in cybersecurity. As the findings in our Fourth Annual Software Supply Chain Security Report make clear: 2025 marked a turning point, as software supply chains became a primary attack surface, deliberately exploited by both cybercriminals and state-sponsored actors to achieve scale, persistence, and impact in targeted organizations from crypto start-ups to critical infrastructure operators.

Our research reveals a troubling acceleration of software supply chain risks. In the past year alone, ReversingLabs observed a 73% increase in malicious open-source packages, with nearly 90% of detections concentrated in npm. Even more concerning: attackers compromised widely used packages and trusted maintainers, transforming routine dependency updates into mass malware distribution events, including Shai-hulud, an npm worm that was the first registry-native malware ever observed.

These incidents underscore a hard truth: that the long assumed trust in software ecosystems is now being actively weaponized and exploited by both cybercriminal and state affiliated threat actors. Our 2026 report shows that adversaries are moving beyond simpler supply chain attacks like typosquatting. We documented growing abuse of OSS repository features and CI/CD workflows, including dependency confusion and the manipulation of GitHub Actions. These techniques allow attackers to hide in plain sight, blending seamlessly into legitimate development activity. Meanwhile, sustained campaigns against cryptocurrency and AI development pipelines revealed how scale and automation, coupled with weak security controls, can amplify downstream software supply chain risks across industries.

Not all the signals are negative, however. Our data suggests that investments in stronger controls, such as mandatory multi-factor authentication and trusted publishing, are beginning to reduce malware prevalence on platforms like Python Package Index (PyPI). But those gains are uneven and often offset by attackers shifting to less mature ecosystems or exploiting blind spots in commercial, closed-source software that remains largely unexamined.

The conclusion of this year's report is unavoidable: commercial and open source software supply chains are now an active, adversarial environment. Defending them requires a fundamental shift in mindset away from implicit trust and toward continuous validation. Organizations must adopt reproducible builds and embrace supply chain transparency and verified trust chains. And both software producers and end user organizations need to obtain the ability to inspect open-source and proprietary software for hidden threats.

This report is intended as both a warning and a guide. The risks are real and growing, but so is our collective ability to address them. With the right visibility, discipline, and commitment, software can once again be something we trust — not by default, but by design.

Sincerely,
Mario Vuksan
CEO and Co-Founder, ReversingLabs

Foreword

Software supply chains are no longer a niche playground for the bad guys. In 2025, they became a primary attack surface for cybercriminals and state-sponsored attackers. This report examines how attackers exploited trust, scale, and automation across open-source and commercial software and emerging AI ecosystems — and what that means for defenders in 2026.

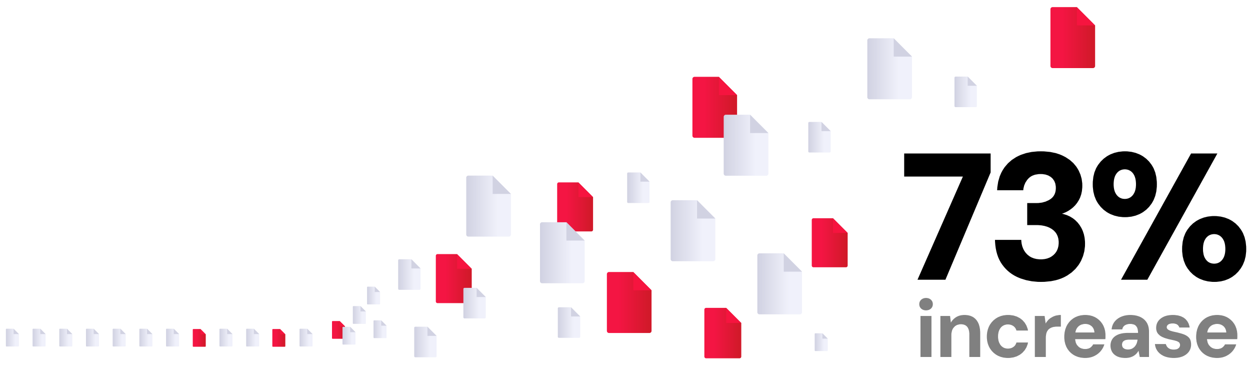
Report Highlights

In 2025, ReversingLabs (RL) observed widening cracks across global software supply chains. Attackers increasingly targeted trusted infrastructure — not just software flaws — to compromise organizations at scale.

Key findings in this year's report include:

- **A 73% increase in malicious open-source packages,** primarily on npm, where more than 10,000 malicious packages were detected - more than double the number in 2024, accounting for nearly 90% of all open-source malware.
- **Compromises of widely used packages and elite maintainers,** turning routine open-source dependency updates into mass malware delivery events.
- **Growing abuse of repository and CI/CD features,** including dependency confusion, typosquatting, and GitHub Actions.
- **Sustained attacks on cryptocurrency and AI development pipelines,** where scale and weak controls amplify risk.

Together, these trends mark a strategic shift: Software supply chains are now actively contested environments that demand precautions such as continuous monitoring and validation, reproducible builds, and verified trust chains that rely on both human and automated (AI) contributors. Read on to learn about what's coming in the arena of software supply chain security in 2026.



Discussion

Wins (and Big Losses): The Year in Open-Source Security

In our fourth annual Software Supply Chain Security Report, RL reveals a mixed picture of the security of open-source software.

Here's a look at the major trends revealed by our analysis of RL Spectra Core's[1] data:

Open-Source Malware Explodes

Open-source malware detections jumped by 73% in 2025 compared with 2024.

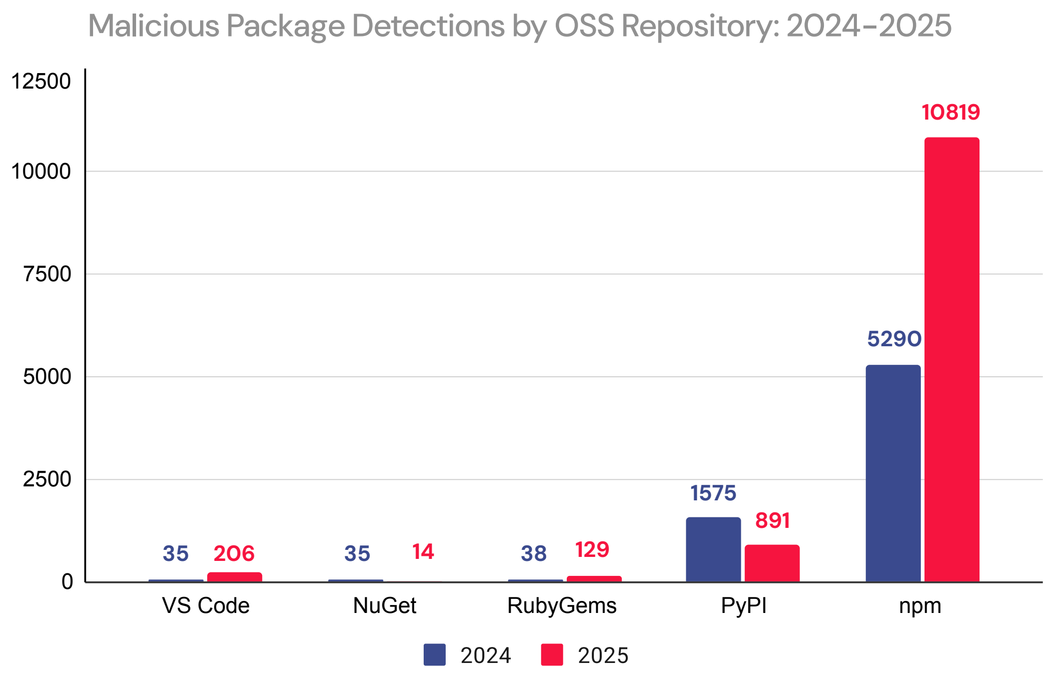


Figure 1: Malicious packages by package manager: 2024-2025.
(Source: RL)

That increase was fueled by attacks on npm, where malware detections more than doubled, from 5,290 in 2024 to 10,819 in 2025.

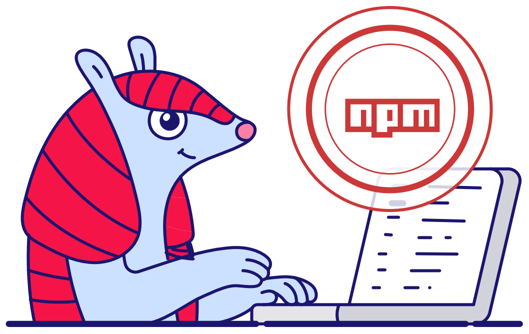
In 2025, secrets exposure increased across four major open-source package managers that RL monitors: npm, PyPI, NuGet, and RubyGems. Exposed developer secrets rose 11% across major repositories. PyPI and npm accounted for about 95% of detected leaks.

Malicious campaigns targeting smaller code repositories such as VS Code demonstrate how attackers continue to evolve their techniques, said Petar Kirhmajer, a threat researcher at RL who discovered a campaign involving 19 Visual Studio Code (VS Code) extensions with malware hidden inside their dependency folders.[2] "They're shifting deeper into the supply chain and abusing trusted components in order to stay hidden," Kirhmajer said.

[1] https://www.reversinglabs.com/products/spectra-core
[2] https://www.reversinglabs.com/blog/malicious-vs-code-fake-image

An OSS Hacker's Playground

The npm repository stands out in 2025 for harboring large quantities of open-source threats and attacks. Detections of npm malware more than doubled to 10,819 in 2025, accounting for close to 90% of all the open-source malware that RL detected in 2025, compared to about 75% during 2024.



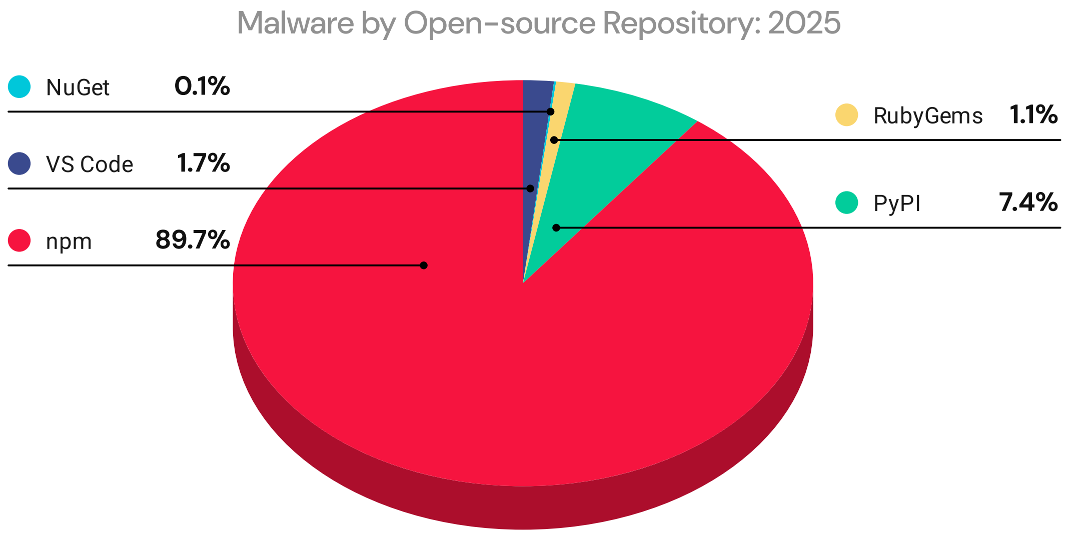


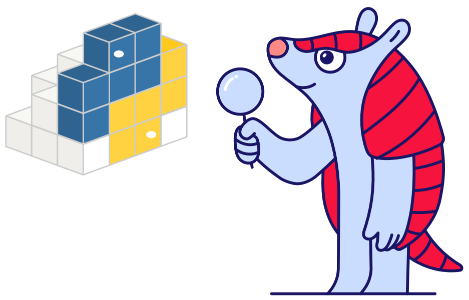
Figure 2: Share of 2025 open-source malware detections by package manager. (Source: RL)

The popularity of the JavaScript language among developers — and the sheer size of the npm repository — explain some of that, but scale isn't the whole story. The slower adoption of security controls at npm, coupled with increased security on other leading open-source package managers, made it a playground for malicious actors in 2025.

For proof of that, look no further than the Shai-hulud worm, which showed how registry-native malware can weaponize stolen credentials to spread at scale. Shai-hulud compromised about 1,000 npm packages in two campaigns, exposing an estimated 25,000 GitHub repositories to the injection of malicious code.[3]

[3] https://www.wiz.io/blog/shai-hulud-2-0-ongoing-supply-chain-attack

Python Malware Drops by Nearly Half



Fortunately, not all trends were negative. Instances of malware detected on PyPI in 2025 were cut nearly in half, from 1,575 detected in 2024 to just 891 in 2025 — a drop of 43%.

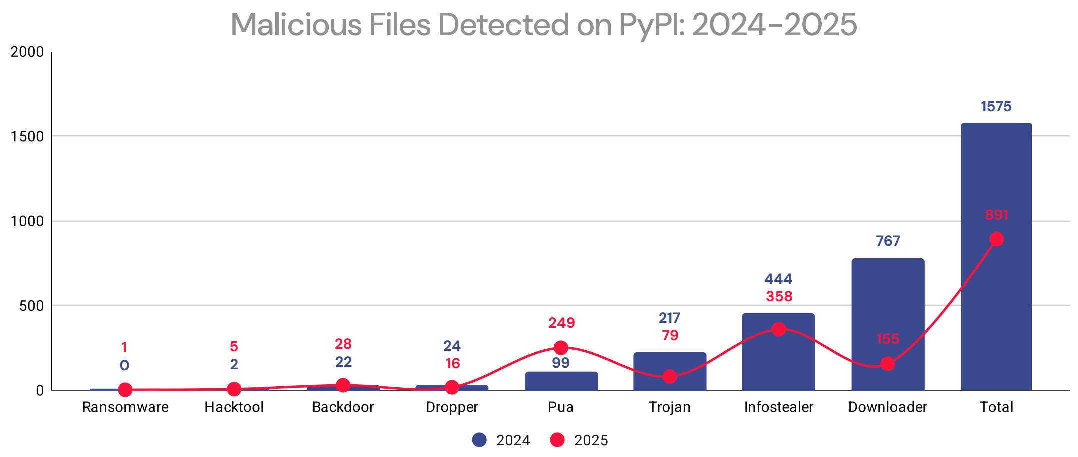


Figure 3: Malicious Python packages by type: 2024-2025. (Source: RL)

An even bigger percentage drop was observed on the NuGet repository, where just 14 malicious packages were detected in 2025, compared to 35 in 2024 — a 60% decline.

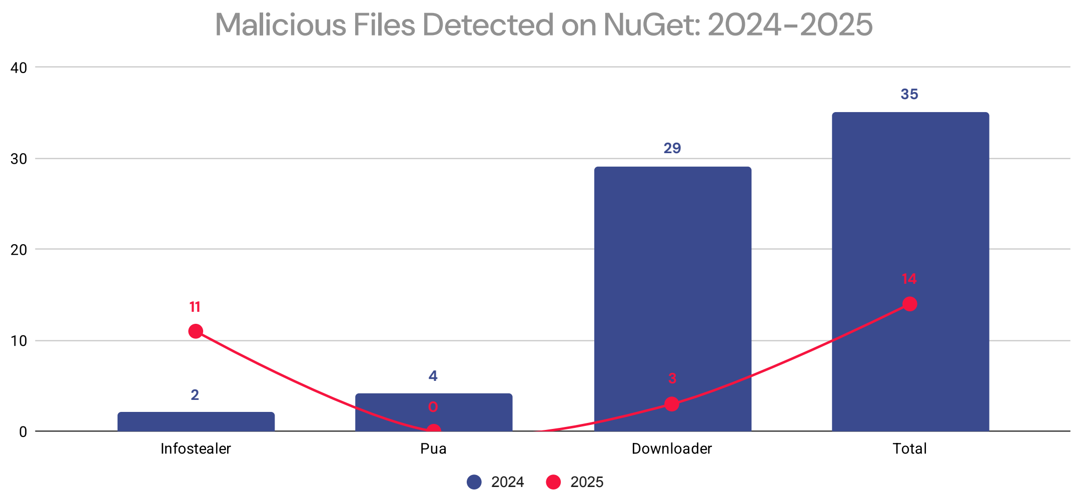


Figure 4: Malicious NuGet packages by type: 2024-2025. (Source: RL)

The reasons for those decreases are open to interpretation. But investments in security features such as mandatory two-factor authentication (2FA) and trusted publishing have increased the impediments to open-source attacks by making it harder to compromise maintainer accounts, steal development secrets through exposed code, and introduce malicious packages. More broadly, security barriers on certain package managers may be pushing malicious actors and campaigns to platforms that have fewer security controls in place.

Legacy Code: The Elephant in the SOC

Over the past year, state-aligned threat actors stormed through legacy software and network infrastructure. Among the attackers were China-backed hacking crews known by their "typhoon" monikers — Volt Typhoon and Salt Typhoon — along with advanced persistent threat (APT) groups tied to Russia, Iran, and North Korea. Their campaigns made it clear that legacy software and network infrastructure remain among the most reliable footholds for sophisticated attackers.

In 2025, Volt Typhoon and Salt Typhoon were repeatedly observed targeting critical infrastructure in the United States and other countries. The goal was not short-term disruption, but long-term, persistent access, with attackers positioning themselves to gather intelligence or sabotage operations during a future crisis.

One notable incident involved the Littleton Electric Light & Water Departments in Massachusetts, with attackers exploiting an unpatched firewall vulnerability. That single weakness allowed the attackers to steal sensitive operational-technology (OT) documentation, including procedures and spatial layouts tied directly to grid operations.[4]

Campaigns against U.S. and Canadian telecommunications providers followed a similar playbook. For example, attackers exploited a critical flaw in Cisco's IOS XE software to implant malicious code directly onto network devices. By configuring GRE tunnels on compromised routers, the group was able to quietly exfiltrate data while blending in with legitimate network traffic. These were not zero-day exploits, but well-documented weaknesses in widely deployed commercial software powering networking equipment and edge devices.

This makes tactical sense. Edge infrastructure is exposed by design and often difficult to monitor. But the larger issue runs deeper: Modern enterprises depend heavily on closed-source, third-party commercial software that is assumed to be trustworthy but is rarely assessed for security risks. That assumption has become a huge security liability.

Joint advisories from the National Security Agency, the Cybersecurity and Infrastructure Security Agency (CISA), and the FBI document how state-sponsored actors consistently exploit known vulnerabilities in device firmware, pivot through trusted network connections, and maintain persistence by modifying routers and edge devices.

Zero-day exploits may grab headlines, but in practice, attackers overwhelmingly succeed by abusing legacy code and long-ago patched flaws that linger, unpatched in the firmware running deployed IT assets.[5]

According to Cisco Talos,[6] for example, Salt Typhoon leveraged a seven-year-old vulnerability, CVE-2018-0171, alongside stolen credentials and living-off-the-land techniques to maintain access on victim networks. It remained undetected in at least one case for more than three years.

Similarly, the China-linked Brickstorm group compromised F5 Networks and stole source code from the BIG-IP product development environment, including information about undisclosed vulnerabilities. F5 later confirmed that proprietary source code and engineering documentation had been taken.[7]

The implications of such breaches extend beyond a single vendor. Following the F5 hack, CISA urged government agencies to inventory their BIG-IP devices, harden public-facing systems, remove end-of-support products, and immediately apply patches.

Yet such recommendations highlight a fundamental weakness: Most security operations centers are reactive. SOC tooling and workflows are focused on phishing, endpoint malware, and exploit attempts against public-facing assets. Vendor-supplied software binaries — especially signed updates — are trusted by default, even though incidents such as SolarWinds and the 3CX Desktop App compromise demonstrate the potential price to be paid for that blind trust.

Addressing this blind spot requires a shift in mindset and tooling. Both software producers and the organizations that deploy their products need to take off their blindfold and gain visibility into the integrity of commercial binaries before publishing and deployment. Technologies such as complex binary analysis make it possible to assess closed-source software without access to proprietary code, thereby exposing hidden risks, tampering, and malicious behavior that traditional security tools miss.

That way, if legacy code is still the elephant in the SOC, at least it's not invisible.

[4] https://dragos.brightspotcdn.com/48/3b/5ea916b3459c8b59203b35d1ccb7/littleton-electric-water-dragos-platform-ot-watch-case-study-03-25.pdf
[5] https://media.defense.gov/2025/Aug/22/2003786665/-1/-1/0/CSA_COUNTERING_CHINA_STATE_ACTORS_COMPROMISE_OF_NETWORKS.PDF
[6] https://blog.talosintelligence.com/salt-typhoon-analysis/
[7] https://my.f5.com/manage/s/article/K000154696

NVD Stalls, and the World Moves On

We noted in last year's report that the decades-old Common Vulnerabilities and Exposures (CVE) system and the federally funded National Vulnerability Database (NVD) were faltering: hobbled by a combination of constrained public funding, a lack of automation, and a growing pipeline of software flaws (including AI-powered vulnerability discovery).

The NVD continued to lose relevance in 2025, RL's data shows. NVD-assigned severity scores for open-source CVEs dropped by 70%, while alternative CVE numbering authorities (CNAs) increased their output by close to 30%.

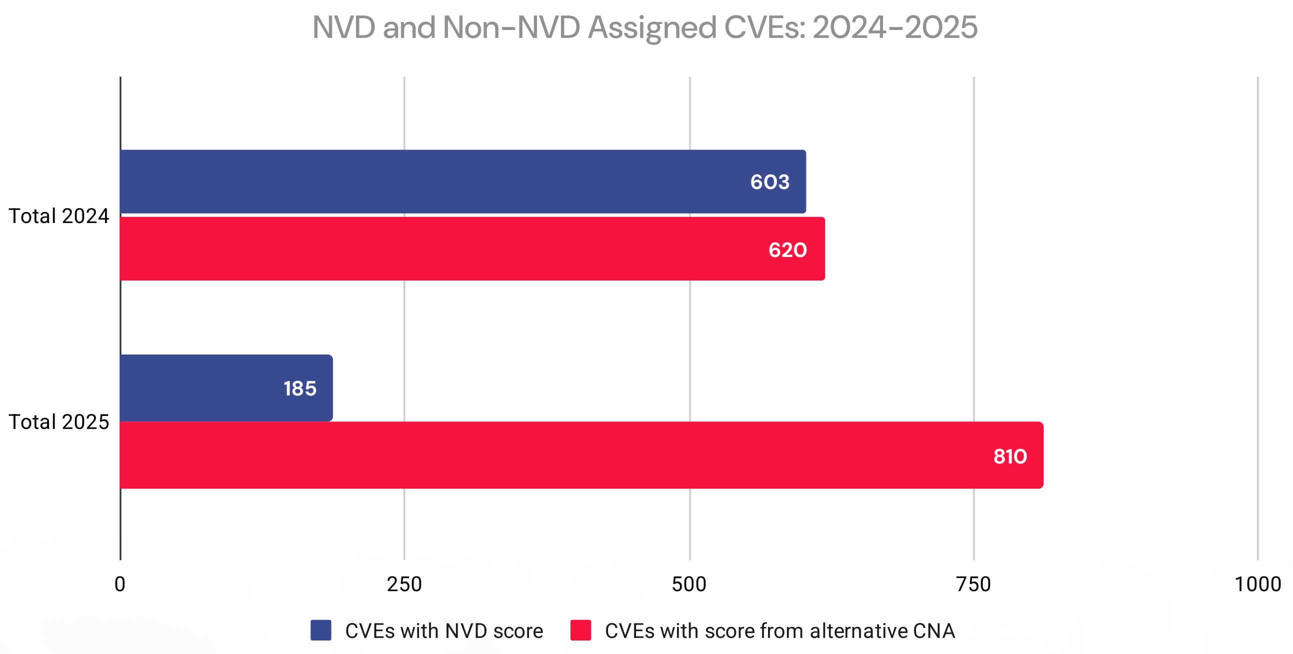


Figure 5: CVE threat scoring NVD vs. alternative CNAs 2024-2025. (Source: RL)

The bottom line: Vulnerability intelligence is decentralizing — and defenders must adapt to that change: aligning their threat intelligence and monitoring operations to leverage information from a growing list of sources. While the NVD is still a source of vulnerability data, it is no longer the gold standard for defenders.

Development Secrets Kept Spilling in 2025

2025 also saw an increase in exposed secrets across four major open-source package managers that RL monitors: npm, PyPI, NuGet, and RubyGems. RL data shows that incidents of exposed developer secrets rose 11% across major repositories. PyPI and npm accounted for roughly 95% of detected leaks.

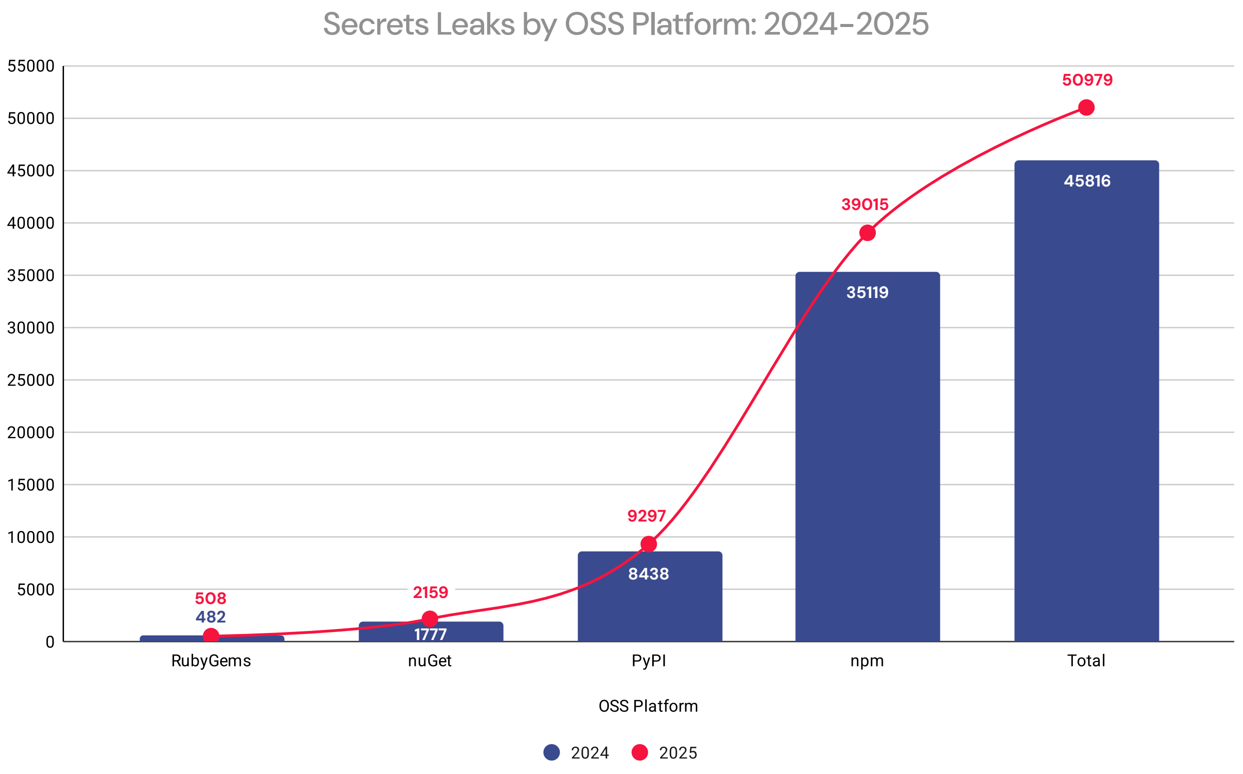


Figure 6: Exposed secrets by OSS package manager: 2024-2025.
(Source: RL)

On npm, the largest open-source repository, RL detected close to 4,000 more secrets in 2025 compared with 2024. A similar trend was observed on PyPI, with incidents of leaked secrets increasing by 10%, though that was slightly smaller than the 12% jump in PyPI secrets leaks seen last year.

Incidents of leaked secrets on the NuGet open-source package manager increased steadily, rising a bit more than 20% in 2025, while exposed secrets on the RubyGems platform increased by 5% this year, from 482 in 2024 to 508 detected secrets in 2025.

The Top Secrets Spillers

As for the source of secrets leaks, our data points the finger at popular cloud-based platforms and applications, including Google, Amazon Web Services (AWS), Slack, and Telegram. Google's cloud was by far the largest single source of leaked developer secrets, accounting for 23% of the more than 39,000 secrets detected on npm and 14% of the nearly 9,300 secrets detected on PyPI. AWS, Amazon's cloud platform, saw a roughly 14% increase in exposed secrets.

Still, the majority of exposed developer secrets are traceable to a long tail of smaller, less well-known applications that together accounted for around two-thirds of leaked secrets detected on both npm and PyPI.

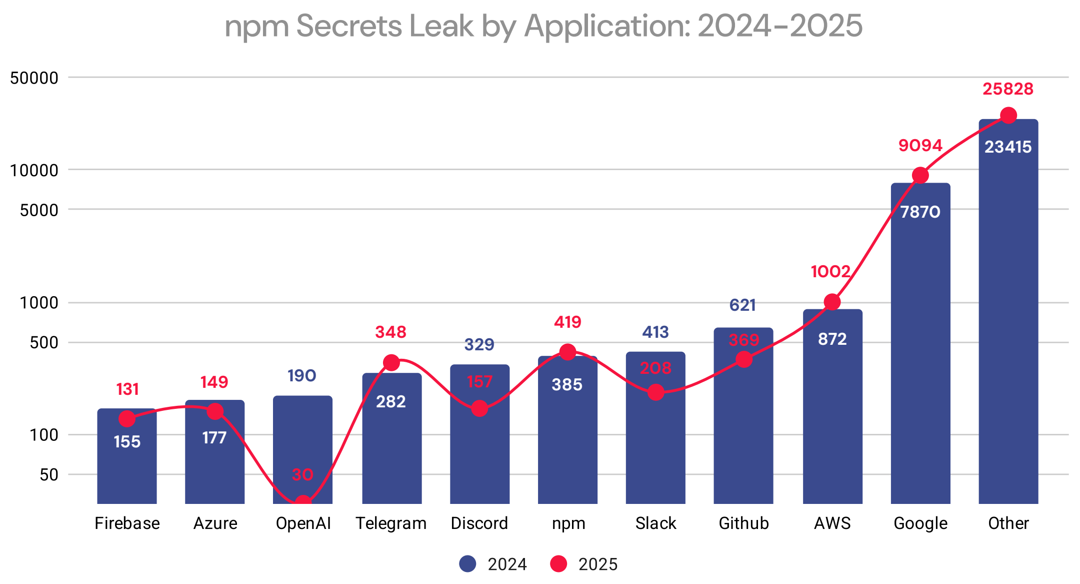


Figure 7: npm exposed secrets by application: 2024-2025.
(Source: RL)

Not that there wasn't good news on secrets. The widely used applications Discord, GitHub, and Slack saw a roughly 50% drop in secrets detected between last year and this year.

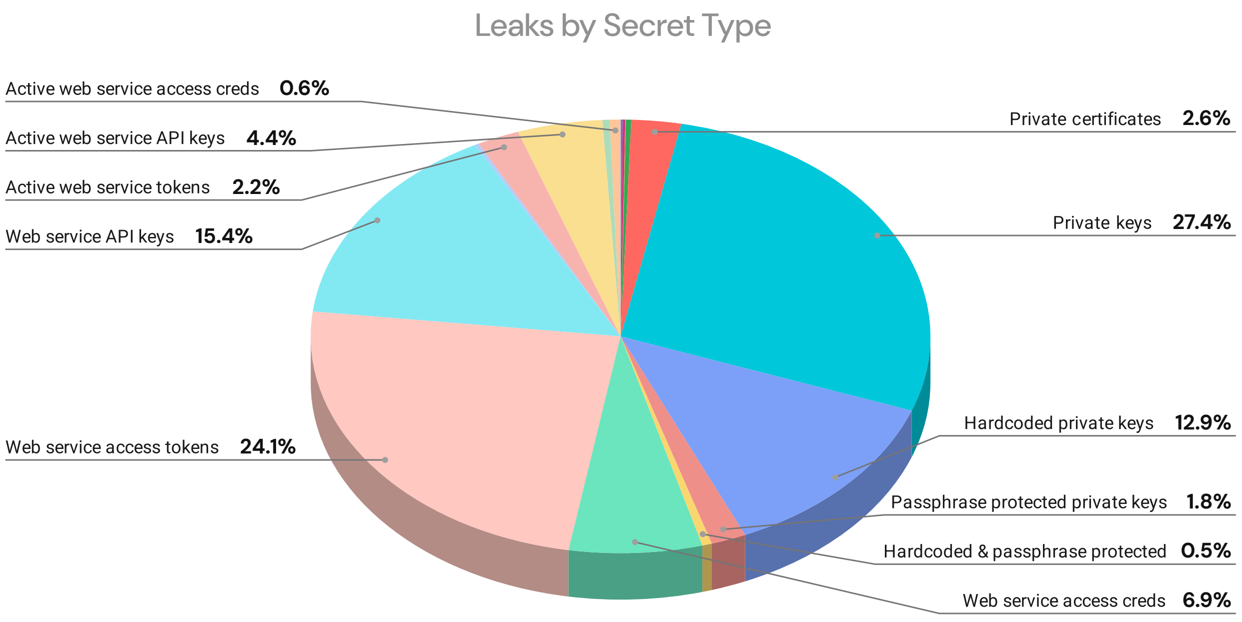


Figure 8: Leaked developer secrets by type: 2025.
(Source: RL)

As for the kinds of secrets most often detected, private keys, web-service API keys, web-service access tokens, and embedded private keys made up more than three quarters of all leaked secrets (79.8%).

The Year in Open-Source Malware

When it comes to the malware lurking on open-source repositories, RL's Spectra Intelligence data shows major shifts in both the supply and makeup of open-source malware in the past year.

Overall, malware detections on open-source repositories rose sharply, up 73%, from 5,290 in 2024 to 10,819 in 2025. But downloader malware, once dominant, fell by more than half (52%) across five monitored repositories: npm, PyPI, RubyGems, NuGet, and the VS Code Marketplace.

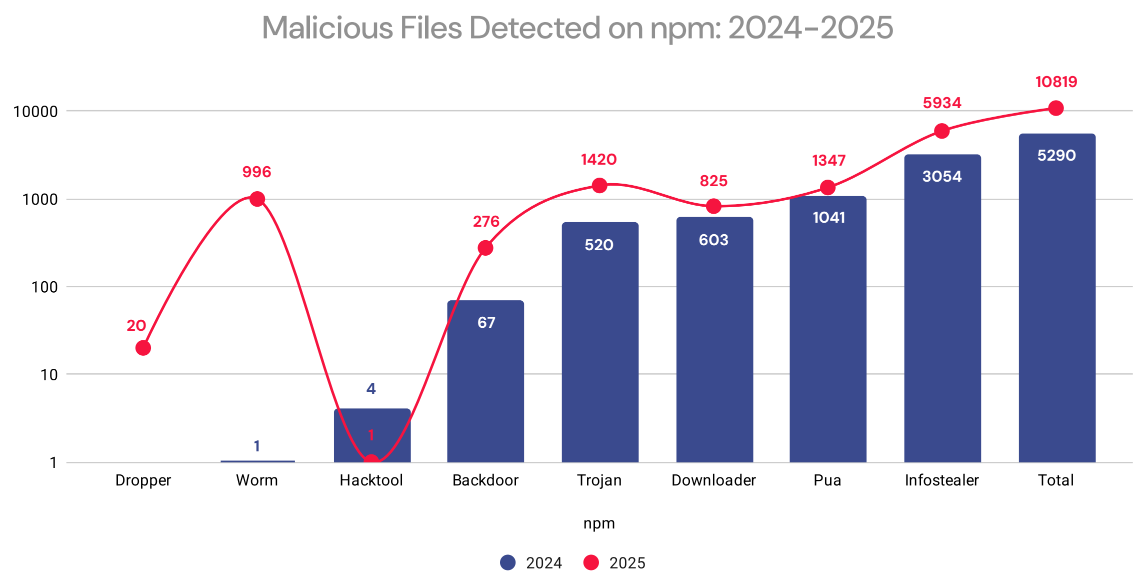


Figure 9: Open-source malware detections by type: 2024-2025.
(Source: RL)

The data signals a pivot away from downloaders toward more sophisticated, data-harvesting malware across open-source ecosystems. Those changes align with shifts in the kinds of open-source supply chain malicious campaigns observed by RL in the last year.

Open-Source Compromises Hit the Big Time

RL's annual reports have shown that open-source packages have been a prime target for malicious actors, including cybercriminal groups and nation-backed hacking crews. Historically, such attacks have focused on small and obscure open-source projects with single maintainers. In 2025, that changed, as the scale and scope of open-source attacks exploded, with a steady stream of successful attacks on popular and widely used open-source packages.

Shai-hulud: A Registry-Native Worm

No incident underscores the radical transformation of the supply chain threat landscape in 2025 better than Shai-hulud, a registry-native, self-replicating worm that first appeared in September using stolen credentials to compromise npm developer accounts and inject malicious code into packages maintained by those compromised accounts.

The first known worm to operate within the open-source supply chain at scale, Shai-hulud combined token-stealing, the exposure of private-code repositories, and automated propagation in two distinct campaigns, first in September and then in November (dubbed "The Second Coming or Shai-hulud 2.0"). Together, the two incursions compromised close to 1,000 npm packages, including widely used modules such as *@ctrl/tinycolor*[8] (~2.2 million weekly downloads) and *@asyncapi/specs*[9] (~1.4 million weekly downloads).

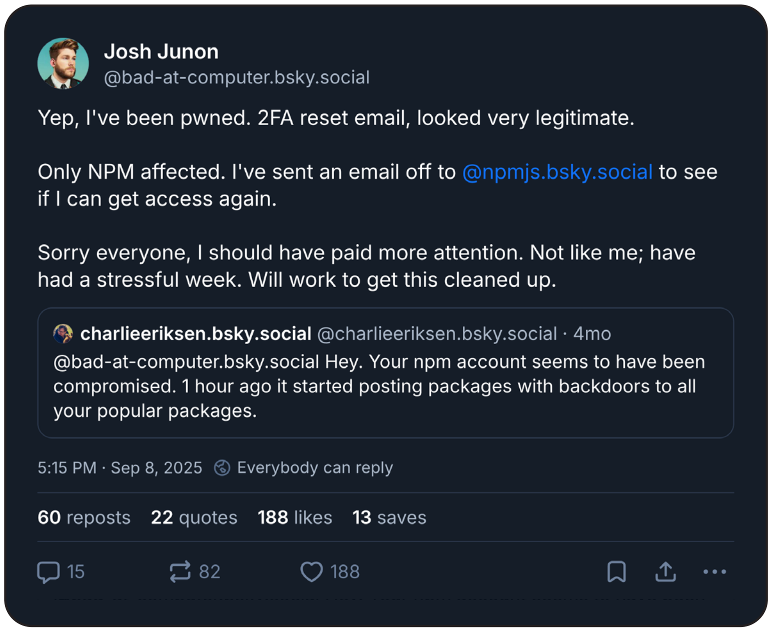
The impact of the Shai-hulud campaigns was felt widely by both open-source and commercial software publishers, with several npm packages from endpoint detection vendor Crowdstrike compromised[10], as well as packages managed by workflow automation vendor Zapier, development tools vendor PostHog, and others.[11]

On December 30, for example, the cryptocurrency wallet vendor Trust Wallet disclosed that a hack of its Google Chrome extension that resulted in the theft of approximately $8.5 million in crypto assets was the result of the Shai-hulud outbreak in late November, according to a company blog post.

Top OSS Maintainers Hacked

Another clear trend in 2025 was the targeting of specific maintainers with access to widely used open-source packages. That included a late-2025 campaign targeting cryptocurrency credentials and developer secrets.

That incident first came to light after Josh Junon, a highly respected developer who uses the handle ~qix signaled that his npm maintainer account had been hacked via a phishing attack involving a fake 2FA email that "looked shockingly authentic." Shortly thereafter, security researchers identified malicious code updates to packages that the ~qix account helped maintain.


Josh Junon
@bad-at-computer.bsky.social

Yep, I've been pwned. 2FA reset email, looked very legitimate.

Only NPM affected. I've sent an email off to @npmjs.bsky.social to see if I can get access again.

Sorry everyone, I should have paid more attention. Not like me; have had a stressful week. Will work to get this cleaned up.

charlieeriksen.bsky.social @charlieeriksen.bsky.social · 4mo
@bad-at-computer.bsky.social Hey. Your npm account seems to have been compromised. 1 hour ago it started posting packages with backdoors to all your popular packages.

5:15 PM · Sep 8, 2025 Everybody can reply

60 reposts **22** quotes **188** likes **13** saves

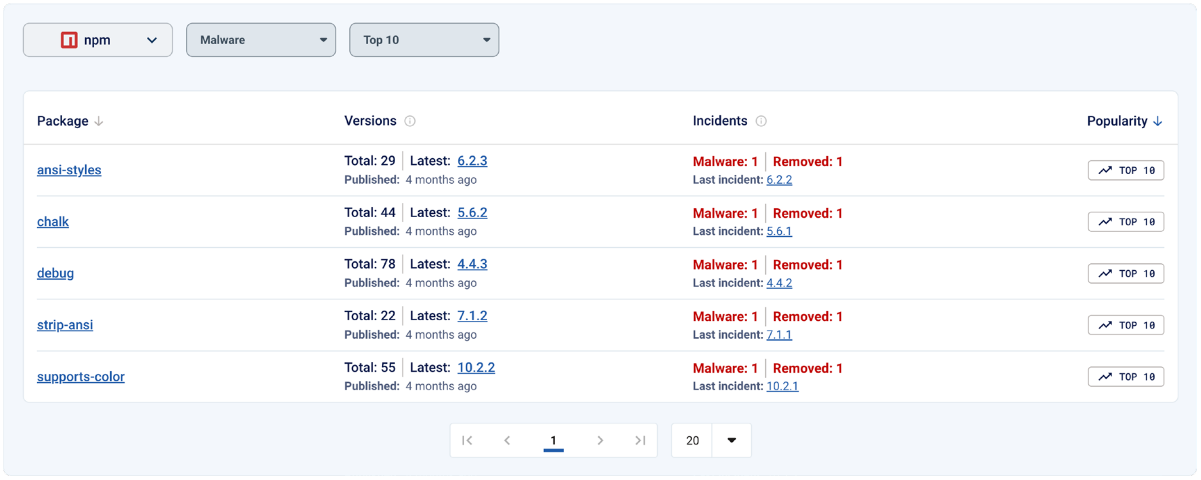
15 82 188

A post by Josh Junon (~qix) to his Bluesky account acknowledging the compromise of his npm maintainer account.

RL documented 18 distinct npm packages compromised following the takeover of the ~qix maintainer accounts. Among those were five of the top 10 npm packages, each with between 1 billion and 1.6 billion monthly downloads: *ansi-styles*, *debug*, *chalk*, *supports-color*, *strip-ansi*.[12]

[8] https://secure.software/npm/packages/@ctrl/tinycolor
[9] https://secure.software/npm/packages/@asyncapi/specs
[10] https://www.reversinglabs.com/blog/shai-hulud-worm-npm
[11] https://www.reversinglabs.com/blog/new-shai-hulud-worm-spreads-what-to-know
[12] https://secure.software

Analysis of the affected packages revealed suspicious behaviors, including the presence of a heavily obfuscated malicious JavaScript designed to steal funds from Bitcoin, Ethereum, Solana, and other cryptocurrency wallets.



Package	Versions	Incidents	Popularity
ansi-styles	Total: 29 | Latest: 6.2.3 Published: 4 months ago	Malware: 1 | Removed: 1 Last incident: 6.2.2	TOP 10
chalk	Total: 44 | Latest: 5.6.2 Published: 4 months ago	Malware: 1 | Removed: 1 Last incident: 5.6.1	TOP 10
debug	Total: 78 | Latest: 4.4.3 Published: 4 months ago	Malware: 1 | Removed: 1 Last incident: 4.4.2	TOP 10
strip-ansi	Total: 22 | Latest: 7.1.2 Published: 4 months ago	Malware: 1 | Removed: 1 Last incident: 7.1.1	TOP 10
supports-color	Total: 55 | Latest: 10.2.2 Published: 4 months ago	Malware: 1 | Removed: 1 Last incident: 10.2.1	TOP 10

Figure 10: Malicious top five npm packages.
(Source: secure.software)

The broad outlines of the attack on Junon and his ~qix maintainer account tracked closely with an earlier compromise of the npm package *eslint-config-prettier*[13], in July.[14] That attack began with a phishing email to maintainer Joun Qin (~jounqin) that spoofed the legitimate support@npmjs.[org] address. The email served up a domain that is a full copy or proxy of npm's actual website and a tokenized URL.[15] Compromised packages were modified to include a postinstall script that dropped a Windows DLL file containing the Scavenger RAT malware.

"These attacks underscore the need for improved security and monitoring on widely used platforms such as npm," said Tomislav Peričin, RL's chief software architect and co-founder, "and the need for more comprehensive software supply chain security controls to identify and stop sophisticated and less noisy campaigns seeking access to trusted development infrastructure."

"If you don't know which packages you're building software with — the content of your build pipeline — then someone else will figure it out and use that knowledge against you."

Tomislav Peričin | Chief Software Architect & Co-founder

Automating Insecurity: Dependabot and Maintainer Hacks

The hacks of influential open-source maintainers such as Junon (~qix) and Qin (~jounqin) also exposed the downside of software supply chain automation, such as dependency updates, which are intended to improve the security of applications by narrowing patch windows.

[13] https://secure.software/npm/packages/eslint-config-prettier/10.1.7
[14] https://www.reversinglabs.com/blog/eslint-hack
[15] https://socket.dev/blog/npm-phishing-email-targets-developers-with-typosquatted-domain

For example, Dependabot, the GitHub security tool designed to monitor and manage a project's open-source dependencies, played a big role in the supply chain compromise of the popular eslint-scope project.[16] That's because Dependabot routinely scans for dependency updates and submits automated PRs to keep projects current. After an attacker gained control of the ~jounqin npm maintainer account and published a malicious package version of *eslint-config-prettier*, Dependabot's automated pull requests unknowingly introduced the tainted package into downstream repositories, expanding the reach of the malicious update and allowing the attacker to access authentication tokens and potentially sensitive developer information.

The incident highlights how trusted automation tools such as Dependabot and Renovate, which are intended to strengthen the integrity of applications with open-source dependencies, can inadvertently amplify the impact of a supply chain attack when adversaries compromise upstream dependencies.

Attackers Exploit OSS Tools and Platforms

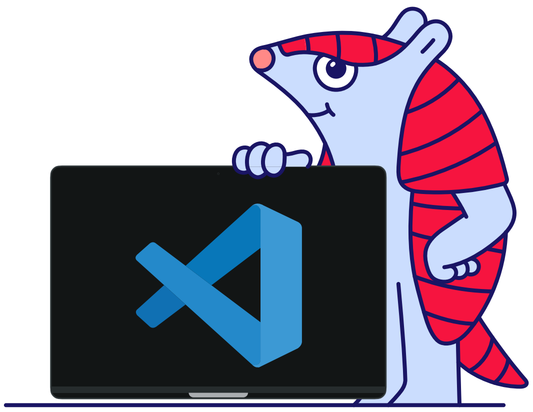
The most common software supply chain attacks of the past few years used social engineering attacks, such as malicious packages designed to be easily mistaken for legitimate open-source modules. Techniques like that, often referred to as "typosquatting," are low-hanging fruit for supply chain hackers.

But the incidents identified by RL this past year show that attackers are also exploiting open-source software platforms' features as well as common tools and automations to cast a wider net on OSS than ever before.

VS Code Threats Multiply

An example of this shift has been the targeting of VS Code Marketplace, a Microsoft platform that offers a source-code editor for all operating systems. According to Stack Overflow's 2025 Developer Survey[17], roughly 76% of developers use VS Code, making its extension ecosystem a high-value target.

As seen in attacks on Python and npm packages, attackers that wish to exploit VS Code are happy to exploit weaknesses in the platform itself to do so.



Marketplace Loopholes Pose Risks

For example, RL software threat researcher Lucija Valentić discovered an exploitable loophole in VS Code Marketplace's naming convention for extensions: an anomaly in which two extensions shared the same name within their unique IDs. That's something that would enable a threat actor to reuse the name of a legitimate but discontinued VS Code extension to upload a malicious one — although, according to VS Code's official documentation[18], that shouldn't be possible.

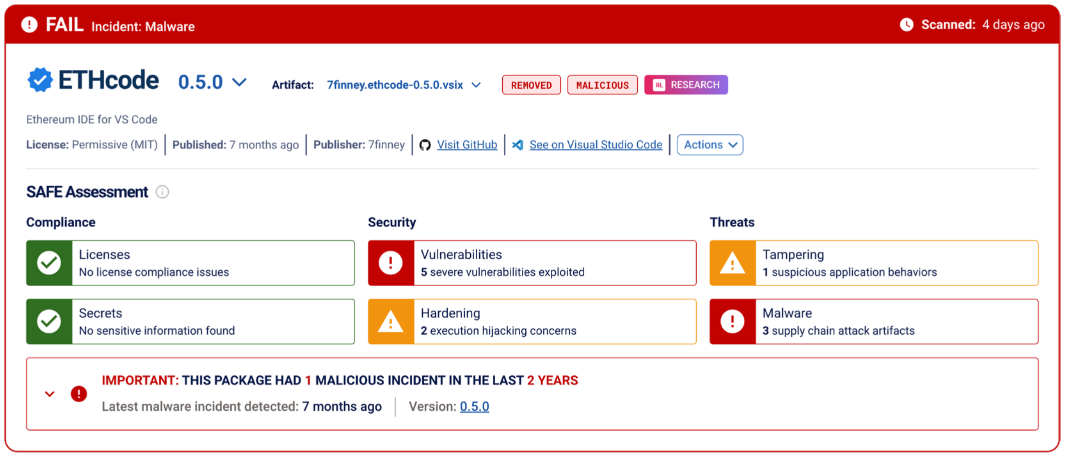
More recently, in December, RL's Kirhmajer uncovered a campaign containing 19 VS Code extensions that hide malware inside their dependency folders. The campaign had been active since February 2025, with malicious files abusing a legitimate npm package to avoid detection and crafting an archive containing malicious binaries that posed as an image with a PNG extension.

[16] https://www.reversinglabs.com/blog/eslint-hack
[17] https://survey.stackoverflow.co/2025/technology/#1-dev-id-es
[18] https://code.visualstudio.com/api/references/extension-manifest

Trust No Code

Flaws in code-sharing platforms aside, many malicious campaigns targeting developers using VS Code, npm, PyPI, and other platforms hinge on slipping malicious code past a developer who is anxious to complete a task and therefore is not willing to dig into the code that promises to help them do that. That's a lapse with potentially huge consequences.

For example, in July, Kirhmajer examined attackers' strategy of exploiting known and legitimate VS Code extensions to inject malicious code into legitimate applications. Specifically, he discovered a malicious VS Code extension, *ETHcode*, that was originally intended for Ethereum smart-contract development.


FAIL Incident: Malware   Scanned: 4 days ago

ETHcode 0.5.0   Artifact: 7finney.ethcode-0.5.0.vsix   REMOVED   MALICIOUS   RESEARCH

Ethereum IDE for VS Code

License: Permissive (MIT) | Published: 7 months ago | Publisher: 7finney | Visit GitHub | See on Visual Studio Code | Actions

SAFE Assessment

Compliance	Security	Threats
Licenses No license compliance issues	Vulnerabilities 5 severe vulnerabilities exploited	Tampering 1 suspicious application behaviors
Secrets No sensitive information found	Hardening 2 execution hijacking concerns	Malware 3 supply chain attack artifacts

IMPORTANT: THIS PACKAGE HAD 1 MALICIOUS INCIDENT IN THE LAST 2 YEARS

Latest malware incident detected: 7 months ago | Version: 0.5.0

Figure 11: ETHCode report showing compromise.
(Source: secure.software)

Developed and maintained by 7finney, a small GitHub organization that develops tools related to the Ethereum blockchain platform, *ETHcode* enables Ethereum developers to streamline their smart-contract development by providing a set of dev tools that can be used to test, debug, and deploy smart contracts and decentralized applications across all EVM-based blockchains. The package was first published in 2022 and has seen nearly 6,000 user installs to date, with regular updates through September 2024, when it received its last legitimate update.

In June, Kirhmajer observed threat actors abusing the GitHub pull request feature to turn the longstanding legitimate and benign *ETHcode* extension into a malicious one.[19] This prompted RL threat researchers to warn the VS Code Marketplace user community about the risks in automatically updating extensions when a new version is published.

Hugging Face's Rise Comes with Price

AI and machine-learning platforms have become integral to modern software supply chains — and attractive targets. You see this when comparing supply chain risks on public package repositories to those on platforms like Hugging Face, an AI-focused open-source package manager that has emerged as a new attack source.

[19] https://www.reversinglabs.com/blog/malicious-pull-request-infects-vscode-extension

In February of last year, for example, RL threat researcher Karlo Zanki discovered a novel attack in which attackers abused one kind of AI model file format, known as Pickle (PKL). The technique that Zanki discovered, dubbed "NullifAI," exploits the Pickle format to store malicious code while evading detection by Picklescan, an open-source tool used by Hugging Face to detect suspicious PKL files. The NullifAI attack highlights a broader trend: Attackers are bypassing platform security features, not just software vulnerabilities. Even more telling, attackers are eager to exploit the features and processes behind AI and machine learning-focused platforms such as Hugging Face in order to hurt the software supply chain.

PowerShell Gallery Abuse Targets Dev Processes

PowerShell Gallery, another code repository, is essential to the Windows ecosystem — and an emerging target for supply chain actors. In the last year, RL threat researchers observed so-called command-jacking campaigns, in which PowerShell is used to enable "Clobbering" while autoloading modules in the PowerShell Gallery repository. This allows commands within a session to be overwritten by malicious implementations with the same name.

The process can be abused as an attack chain, and RL threat researcher Vladimir Pezo has detailed how PowerShell Gallery's simple *Install-Module* command makes the process even simpler. Attackers are finding that when developers rely heavily on code hosted on open-source platforms and the processes that let them easily integrate such code into their projects, they are able to work smarter, not harder, to compromise legitimate applications and development pipelines.

OSS Platform Security Investments Pay Dividends

While attacks on OSS platforms continued in 2025, not all repositories received the same amount of attention from threat actors. One possible reason for this is the security investments that OSS platforms have made in recent years.

For example, PyPI announced in 2023 that it would require[20] all project maintainers to enable 2FA in a bid to curb account takeover attacks. The platform also took steps to enable trusted publishing, an OpenID Connect (OIDC)[21] standard that exchanges short-lived identity tokens between a trusted third-party service and PyPI, thus reducing the overhead 2FA brings to publishing, PyPI admin Donald Stufft wrote[22].

Then there is Project Quarantine[23], a new initiative that allows maintainers on the PyPI platform to quarantine attacked projects rather than remove them, which was the only option before.

"The impact of these removals can be disruptive, and removals are pretty much irrevocable," said Mike Fielder[24], a PyPI admin and safety and security engineer.

For its part, npm has been more reactionary. Roughly one week after the initial discovery of Shai-hulud, npm owner GitHub announced its "plan for a more secure npm supply chain,"[25] including mandatory use of 2FA, trusted publishing, and granular tokens.

RL's Zanki said that while granular tokens give more control than legacy tokens, they have a greater number of configuration options, which automatically increases the risk of misconfiguration. What's more, he noted, granular access tokens, with their short lifespan, wouldn't have prevented Shai-hulud from spreading because the worm was publishing new packages immediately upon infection.

[20] https://blog.pypi.org/posts/2023-05-25-securing-pypi-with-2fa/
[21] https://openid.net/connect/
[22] https://blog.pypi.org/posts/2023-05-25-securing-pypi-with-2fa/
[23] https://blog.pypi.org/posts/2024-12-30-quarantine/
[24] https://www.linkedin.com/in/miketheman/
[25] https://github.blog/security/supply-chain-security/our-plan-for-a-more-secure-npm-supply-chain/

How AI Is Transforming Cyber Risks

AI: The New Shadow IT?

CISOs are well acquainted with the threat of shadow IT. Having increased significantly in 2020 with the COVID-19 pandemic and the shift to remote workforces, shadow IT started to recede as IT security leaders pushed to tighten work-from-home policies. Then came the release of OpenAI's ChatGPT in 2023 — a major setback. In short: While rogue Dropbox accounts, unauthorized Slack deployments and management apps are risky, "shadow AI is riskier. Way riskier," the Cloud Security Alliance (CSA) noted.

There are many ways in which unsanctioned AI use within enterprises can cause cybersecurity risks. Stack Overflow's 2025 Developer Survey found that 84% of developers are using or plan to use AI tools. Hugging Face, the open-source platform designed for machine-learning development and collaboration, hosts more than 2 million models — twice the number of models hosted on the platform in September 2024. AI is not just a tool used to power software supply chains. It is deeply intertwined with modern software development — and developers' reliance on AI is only going to increase.

More than ever, enterprise governance and controls are needed to curb shadow IT and prevent new, AI-flavored software supply chain threats from taking shape, including phishing, malware, information leaks, and more.

"While you don't want to stifle … the adoption of AI in companies and organizations, the reality of these models and data that are potentially poisoned is something that just demands that companies be more vigilant about shadow AI."

Doug Levin | Black Duck Software founder and former CEO

AI Supply Chain Threats: What 2026 Has in Store

Death by Pickle
As RL researchers have shown in the last year, the Hugging Face platform has become a critical resource for developers working on AI and ML projects — and a playground for attackers.

In addition to the NullifAI campaign in February, in May RL threat researcher Karlo Zanki identified three malicious PyPI packages that posed as a Python SDK (software development kit) for users of Alibaba AI labs. Once installed on a victim's machine, the malicious packages deliver an infostealer payload hidden inside a PyTorch model, which is basically a zipped Pickle file.

Despite those incidents, and Pickle being a known unsafe data format, developers and others in ML operations (MLOps) continue to use it. At the same time, security tools have been slow to detect malicious behavior in ML files, which have not traditionally been viewed as a medium for distributing executable code.

With more AI supply chain attacks in the works, and Hugging Face a popular platform for launching such attacks, that thinking needs to change. Development and MLOps teams need to vet Pickle files for threats, just as they would other pieces of software.

Model Corruption

With so many of the software supply chain's stakeholders relying on AI models to ensure that processes run smoothly, model manipulation — an attempt to manipulate an AI model to produce inaccurate or misleading results — is a growing concern.

One way attackers attempt to carry out this manipulation is with prompt injections: exploits of flaws in the AI model's underlying logic. In 2025, for example, the GenAI Security Project of the Open Worldwide Application Security Project (OWASP) released its Large Language Model (LLM) Top 10 list — something like OWASP's Top 10 AppSec Risks list.

At No. 1 was prompt injection, which can lead to a model violating guidelines, generating harmful content, enabling unauthorized access, or influencing critical decisions. Multimodal injection is just one scenario in which prompt injection could be carried out by an attacker. OWASP defines this type of incident as occurring when an attacker embeds a malicious prompt within an image that accompanies benign text. When the targeted multimodal AI model processes the image and text concurrently, the hidden prompt alters the model's behavior, potentially leading to unauthorized actions or the disclosure of sensitive information.

Malicious MCP Servers

Model Context Protocol (MCP) servers are booming in popularity, particularly with software developers. MCP is an open-source standard for connecting AI applications to external systems. MCP servers expose specific capabilities to AI applications through standardized protocol interfaces. For example, GitHub MCP servers are used by developers for code management.

While MCP servers offer innovation and convenience, the technology is subject to compromise. In September 2025, for example, researchers discovered the first-ever instance of a malicious MCP server that was distributed via npm.

While MCP server threats began making news in late 2025, there is no doubt that attackers will continue to exploit the MCP ecosystem in the months ahead as an easy means of attacking software supply chains in 2026.

"The rapid pace of innovation surrounding MCP leaves little space for security considerations. We've already seen some non-traditional malware attack vectors targeting MCP users. As MCP servers become widely deployed the number of creative malware attacks will only increase."

Tomislav Peričin | Chief Software Architect at ReversingLabs

Common Open-Source TTPs Observed in 2025

In 2025, threat actors combined increasingly sophisticated techniques with longstanding methods such as social engineering. Together, these approaches defined the most common tactics, techniques, and procedures (TTPs) observed across OSS ecosystems.

Developer Secrets in the Crosshairs

After malicious OSS packages, secrets exposure remained one of the most damaging risks to software supply chains in 2025.

Open-source platforms saw high levels of secrets exposures and thefts driven by malware. The malicious *solana-token* Python package that Zanki identified in May targeted Solana blockchain developers with the goal of stealing application code and gathering sensitive information or developer secrets in the code.[26]

Then, in September, the Shai-hulud worm campaign on npm deployed TruffleHog, an open-source password-sniffing tool, onto victims' machines. To facilitate secrets theft, Shai-hulud "migrated" private-code repositories belonging to compromised GitHub accounts to publicly accessible repositories named after the worm, giving the public access to secrets hardcoded within the exposed code.

In the second wave of Shai-hulud, in November, Wiz Security confirmed, roughly 400,000 raw secrets were exposed after attackers published the stolen data in 30,000 GitHub repositories.

Local 'Patching' That's Anything But

This past year, RL researchers discovered a first-of-its-kind attack on npm. Malicious, locally installed packages were designed to infect other legitimate packages residing on a victim's machine. The two malicious packages identified, *ethers-provider2* and *ethers-providerz*, appeared as simple downloaders.

The malware's second stage did something researchers hadn't seen before: It "patched" the legitimate npm package *ethers* when installed locally. Once patched by the malicious packages, *ethers* included a malicious payload that served up a reverse shell that gave attackers persistent access to the victim's system. Soon after, RL researchers discovered another malicious campaign using the same technique targeting the Atomic Wallet and Exodus crypto wallet software.

The repeated success of these campaigns suggests local patching is likely to reappear as an open-source attack vector in 2026.

The Phishing Is Always Good

Social engineering and phishing attacks on open-source developers were a cornerstone of OSS attacks again in 2025. In addition to the phishing attacks that compromised the accounts of the high-profile open-source maintainers ~qix and ~jounqin, which fueled some of the largest open-source security compromises on record, there were more subtle social-engineering campaigns designed to gain access to legitimate code repositories.

[26] https://www.reversinglabs.com/blog/same-name-different-hack-pypi-package-targets-solana-developers

In April, for example, RL researchers detected two Python libraries that posed as fixes for a popular cryptocurrency library, *bitcoinlib*. The package names referenced an issue that was raised related to error messages being generated by bitcoinlib during bitcoin transfers. Developers who rely on the library specifically asked maintainers to address the issue, which attackers saw as the perfect social-engineering opportunity.

Another malicious campaign discovered in June was carried out by a threat actor known as Banana Squad, which posted more than 60 GitHub repositories posing as legitimate security and penetration testing tools but contained trojanized look-alikes of other identically named repositories.

Then there was the compromise of the *ETHcode* extension, in which the attacker using the handle Airez299 opened up a GitHub pull request for *ETHcode* that, at face value, looked like a legitimate code update. However, upon further investigation, RL researchers discovered two new lines of code amid a series of legitimate code updates. When used together, those lines could compromise the entire project and the corresponding VS Code extension.

These incidents reinforce that "layer 8" threats remain a serious concern, with even highly experienced OSS maintainers falling victim to sophisticated phishing and social engineering attacks.

Cryptoattacks Rage On in 2025

Several of the incidents RL researchers identified in 2025 targeted cryptocurrency wallets and applications, as well as blockchain infrastructure. This makes sense, given a global cryptocurrency market capitalization of more than $3 trillion as of December 2025.[27]

They include the two Python libraries that posed as fixes for the open-source *bitcoinlib* library and were designed to exfiltrate sensitive database files from victims once downloaded, as well as the malicious npm package *pdf-to-office* and the malicious Python campaign involving the package *solana-token*, which pretended to be a utility for developers working on applications that leverage the Solana blockchain.

A hunger for exploiting crypto also drove malicious campaigns on NuGet, the OSS repository for Microsoft's .NET. That includes the discovery of 14 malicious NuGet packages, each with a malicious payload that either stole crypto-wallet credentials, crypto funds, or OAuth client IDs and client secrets.

Then there were the two npm packages — *colortoolsv2* and *mimelib2* — that abused Ethereum smart contracts to conceal malicious commands that installed downloader malware on compromised systems. Smart contracts, which allow users to interact with the Ethereum blockchain, became a tool for delivering second-stage malware to unsuspecting victims.

[27] https://www.forbes.com/digital-assets/crypto-prices/

What Comes Next

SOCs Shift Left

One unmistakable lesson of the past year is that the scope and cadence of software supply chain compromises are growing — and fast.

Despite that, both software publishers and end-user organizations struggle to acknowledge the downstream impact of supply chain compromises, take the appropriate steps to identify malicious software extensions, and mitigate other supply chain threats in their development pipelines or IT environments.

With growing consequences stemming from such incidents, 2026 is likely to see this magical thinking about supply chain integrity fade as organizations take steps to better protect their IT infrastructure from the fallout of supply chain hacks.

One consequence may be that security operations centers (SOCs) will evolve from reactive threat response hubs focused on monitoring the security of deployed technology to continuous assurance programs that span development, build, and deployment pipelines.

We may see traditional SOC boundaries blur as organizations promote closer collaboration between security and development teams to uphold DevSecOps processes: ingesting telemetry from source-code repositories, build systems, package registries, and developer-focused resources such as Spectra Assure Community. This "shift left" of SOC visibility will see alerts arise not just from malicious network traffic, but also from anomalies in developer behavior, build signatures, and dependency graphs.

SOC analysts will need to work closely with development teams, making their understanding of software composition and provenance as deep as their understanding of network and endpoint activity. On the other hand, developers will need to embrace the security mindset of SOC analysts, showing greater awareness of threats such as phishing, code tampering, and malware that may compromise CI/CD pipelines.

The Worm–ification of Supply Chain Threats

When it comes to open-source risks, the trends we observed in 2025 are almost certain to continue in 2026, with malicious actors sharpening their focus on open-source maintainers and widely used open-source projects. The tactics of leveraging social engineering attacks and exploiting security loopholes in widely used open-source package managers and directories are also certain to stretch through 2026 and beyond.

As for the concerted and (occasionally) coordinated efforts to improve security by the private and nonprofit entities managing open-source repositories? These will surely erect obstacles to malicious actors that complicate or even hamper some flavors of supply chain hacks.

However, as the Shai-hulud outbreaks in 2025 suggest, the supply chain threat landscape is starting to look an awful lot like the larger threat landscape, with a steadily growing stream of malicious files, self-propagating malware, automated attacks, live-off-the-land techniques, and so on. So, too, is the reactive, "Whac-a-Mole" nature of supply chain defenses, with new threats and attacks causing new stopgap defenses to pop up — prompting attackers to innovate.

Consider this: The “Second Coming” of Shai-hulud in November arguably caused *greater damage* and disruption than the initial outbreak a couple of months earlier — despite the first round’s clear preview of what an outbreak would look like.

“Shai-hulud proves that increasing security measures in an open-source repository doesn’t by itself solve the problem,” said RL’s Peričin. In the end, the job of preventing compromises falls most heavily on developers and software publishers, not platform owners.

“The fact is that no one else can mitigate or reduce the impact of a software supply chain attack for you.”

Tomislav Peričin | Chief Software Architect at ReversingLabs

Cryptocurrency Attacks Continue

Fewer open-source hacks or not, one trend is almost certain to continue into 2026 and beyond: attacks targeting cryptocurrency developers, applications, and infrastructure.

The reason for that is easy to grasp: Cryptocurrency wallets provide easy, remote access to high-value digital assets that can easily be transferred to accounts controlled by attackers. And crypto wallets are in the crosshairs of both cybercriminal groups and nation-backed hackers such as Lazarus Group (APT 38). Add to that the decentralized and unregulated nature of the cryptocurrency industry, with its heavy reliance on open-source code, and you have a recipe for continued malicious activity targeting cryptocurrency software supply chains.

“Recent crypto attacks should serve as a reminder that malicious actors continue to exploit the weakest link in the supply chain: trust. What appears to be a small and harmless dependency can in reality act as a silent backdoor.”

Petar Kirhmajer | Threat Researcher at ReversingLabs

AI Powers Security Processes — And Disrupts Them

As AI continues to reshape how software is built, attackers will keep refining their ability to corrupt models, tooling, and ecosystems. Fast-evolving threats like these will prove disruptive to software development teams and others, requiring a greater focus on supply chain security measures such as extending trust verification beyond code — into the AI systems that increasingly write it.

That will include more instances of model-embedded malware (MEM). Unlike traditional malware, MEM can be hidden directly inside AI models and activated only under specific conditions — a powerful tool for malicious actors. Because MEM operates below the surface of conventional code, it can evade many existing security controls—allowing malicious functionality to persist undetected inside trusted AI systems.

Look for attackers to continue efforts at model poisoning, particularly for AI systems embedded in the software development lifecycle (SDLC) and the model development lifecycle (MDLC). By corrupting training data, fine-tuning processes, or pre-trained models, adversaries can influence the behavior of AI tools at scale. In development environments, the impact can be severe: poisoned models may generate vulnerable code, introduce backdoors, or embed logic bombs that are difficult to detect once deployed.

Finally, 2026 will see the further expansion of AI-driven supply chain attacks known as slopsquatting, in which malicious actors take advantage of AI model "hallucinations" such as the naming of package dependencies for packages that do not exist. That enables attackers to register malicious packages using the hallucinated names — turning AI recommendations into a delivery mechanism for compromise.

Of course, generative AI also promises improved security and defense measures. For example, companies including Google, OpenAI, and Meta are using AI to automate the discovery and disclosure of CVEs.

These efforts are bearing fruit in the form of large numbers of newly identified code vulnerabilities, but that comes with consequences for security teams and open-source maintainers. A growing chorus of experts have warned that, thanks to these efforts, developers and maintainers are drowning in a flood of new-code vulnerability reports created by AI-powered vulnerability analysis tools.

This means that defenders and maintainers have to spend time they likely don't have trying to remediate a flood of vulnerabilities that may not even need to be prioritized. The next year will see software publishers, end-user organizations, and the larger security community wrestling with both the benefits and threats that generative AI presents.

"We take security very seriously, but at the same time is it really fair that trillion-dollar corporations run AI to find security issues on people's hobby code [and] then expect volunteers to fix [them?]"

FFmpeg

Calls for Software Transparency, Quality

In the United States, the White House has shifted the government's cybersecurity priorities. Two older presidential Executive Orders that pertain to software supply chain security — the May 2021 EO 14028 and the January 2025 EO 14144 — now show up as 404s on the White House's website.

However, new precedents pertaining to the supply chain are poised to take shape in the United States in 2026. Those include the U.S. Department of Defense (DoD) Software Fast Track (SWFT) initiative and the U. S. Cybersecurity and Infrastructure Security Agency's 2025 Minimum Elements for a Software Bill of Materials (SBOM). These efforts illustrate that software supply chain security is still front and center when it comes to U.S. cybersecurity policy.

Additionally, final rules and enforcement mechanisms of the Cyber Incident Reporting for Critical Infrastructure Act (CIRCIA) will come into force in 2026.[28] Among other things, CIRCIA requires U.S. operators of critical infrastructure to report cyber incidents within 72 hours, and ransomware payments within 24 hours, to CISA.[29]

Even bigger changes are afoot outside the United States. The EU's Cyber Resilience Act (CRA), which went into effect at the end of 2024, is a landmark regulation that mandates and harmonizes security requirements for products with digital elements, including both hardware and software sold in the EU. That law requires manufacturers, importers, and distributors to ensure that digital products are secure by design and by default, to manage vulnerabilities throughout the product lifecycle, and to provide transparency about cybersecurity features, with conformity demonstrated through the CE marking.[30]

Starting in September 2026, manufacturers of products with digital elements must begin reporting actively exploited vulnerabilities to authorities such as a computer security incident response team or the European Union Agency for Cybersecurity (ENISA) within 24 hours of their detection.

The United Kingdom's Department for Science, Innovation and Technology's "Open Source Software Best Practices and Supply Chain Risk Management," released in March 2025, is another important policy document. It maps and evaluates existing best practices for managing and mitigating risks by focusing on the use, production, security, and licensing of OSS, in addition to the management of software supply chain risks in general. The guidance builds on the effort made by the United Kingdom a year earlier to establish a Code of Practice for Software Vendors, which serves to protect U.K. businesses against harm and disruption from software supply chain attacks and poor software resilience.

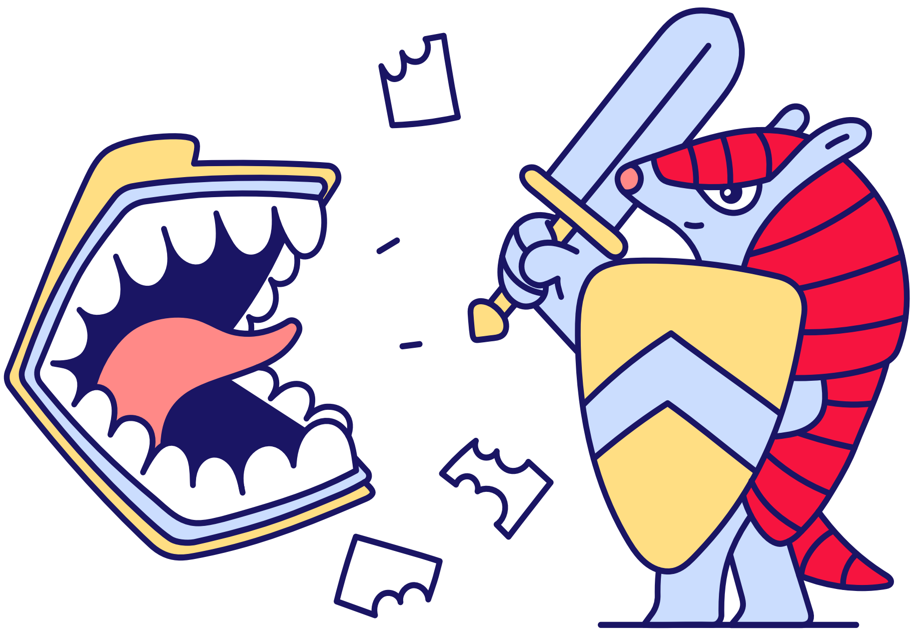
[28] https://www.hklaw.com/en/insights/publications/2025/12/2026-legislative-regulatory-outlook/
[29] https://www.reversinglabs.com/blog/software-supply-chain-security-compliance-get-out-front
[30] https://3375217.fs1.hubspotusercontent-na1.net/hubfs/3375217/Documents/Compliance-Card-CRA.pdf

OSS Security Gets Real

Malicious packages and extensions surface on OSS platforms daily. What RL threat research has shown is that while the majority of these attacks are distributed via npm and PyPI — the two biggest public package managers — they are found on smaller platforms as well. This means that developers and maintainers need to vet packages on all kinds of OSS infrastructure before they are downloaded and incorporated into software products.

Doing that will require greater coordination between individual developers and the larger open-source ecosystem. And RL's Spectra Assure Community is a valuable resource for that. It provides free software supply chain risk assessments for open-source modules on npm, PyPI, RubyGems, NuGet, VS Code and PowerShell, allowing anyone to review over 7 million packages before deployment. Those checks include valid licenses and secrets exposure, security concerns such as vulnerabilities and application hardening, as well as threats such as tampering and malware.

As software supply chain threats expand, security assessments of both open-source and third-party software will need to evolve to manage both the greater scale and complexity of supply chain attacks. That will include attention to risks on a growing list of code-sharing platforms and a widening landscape of open-source risks, from code vulnerabilities to license violations to tampered-with and malicious software.



Moments That Defined Software Security in 2025

There were a plethora of mandates, frameworks, and industry calls that further defined software supply chain security policy this past year. Here are five that you should be aware of.

EU Enacts The Digital Operational Resilience Act (DORA)

DORA is designed to enhance the overall digital operational resilience of the EU financial sector and ensure that these firms, as well as third-party firms supplying them, can withstand, respond to, and recover from cyberattacks or system failures, including software supply chain attacks.

CISA & ENISA Signal Support for Vulnerability Disclosure & Management

After a disruption in how software vulnerabilities are centrally tracked and reported, CISA and ENISA both made commitments in 2025 to vulnerability disclosure and management programs in the form of revitalizing the National Vulnerability Database (NVD) and unveiling of the EU Vulnerability Database (EUVD), respectively.

U.S. EO 14306: Refocusing Federal Cyber Policy

The White House released Executive Order (EO) 14306 that updates two earlier cybersecurity directives, and it refines priorities around defending the nation's digital infrastructure, secure software development, and resilience against malicious cyber campaigns.

CISA Issues 2025 SBOM Minimum Elements

CISA published an updated list of SBOM minimum elements with requests for public comment, and it refreshed earlier baseline guidance to reflect rapidly maturing software bill of materials practices and tooling by both software producers and end user organizations.

GitHub Tackles Its Platforms' Security

Microsoft's GitHub announced several changes to its platforms' security protocols, including an update that enables explicit blocking and SHA pinning support to GitHub Actions policy, as well as changes to npm in response to the registry-native Shai-hulud worm.

For a full list of software supply chain security's defining moments in 2025, head to RL Blog.

CLICK HERE ›

Conclusion

The findings in this year's report make one thing clear: Software supply chains have moved from being an obscure risk to a primary battleground for malicious actors and the organizations they target. In 2025, attackers repeatedly demonstrated their ability to exploit trust, automation, and scale: compromising open-source ecosystems, abusing CI/CD workflows, and targeting high-impact domains such as cryptocurrency and AI. While investments in platform security and maintainer protections have seen progress in some areas, overall supply chain risks continue to grow as adversaries shift laterally to weaker controls, smaller ecosystems, and trusted infrastructure that was never designed to be actively defended.

Looking ahead to 2026, the challenge for defenders is not simply to patch faster, but to rethink how trust is established, verified, and continuously enforced across the software lifecycle. As the events of 2025 make clear, static, perimeter-focused approaches and vulnerability-only intelligence are no longer sufficient in an environment defined by shared dependencies, automated contributors, and decentralized risk signals.

Organizations that succeed in defending against supply chain threats will look beyond threat volumes and headlines, focusing on the threats that are relevant to their specific environment and treating the software supply chain as a living system — one that requires continuous inspection, reproducible builds, and verified trust chains spanning both human and machine inputs.

The software supply chain is now an active, contested domain, and defending it demands a corresponding shift in strategy, tooling, and thinking. Embrace the change!

Methodology

RL's fourth annual Software Supply Chain Security Report brings together the findings of public reports and data with non-public, anonymized data compiled by RL analysts and powered by Spectra Core, the world's fastest and most comprehensive software platform for automated static decomposition and analysis of binary files. You can read more about our report methodology here:
https://www.reversinglabs.com/sscs-report/methodology

About RL

ReversingLabs is the trusted name in file and software security. We provide the modern cybersecurity platform to verify and deliver safe binaries. Trusted by the Fortune 500 and leading cybersecurity vendors, RL Spectra Core powers the software supply chain and file security insights, tracking over 422 billion searchable files daily with the ability to deconstruct full software binaries in seconds to minutes. Only ReversingLabs provides that final exam to determine whether a single file or full software binary presents a risk to your organization and your customers.

Worldwide Sales +1.617.250.7518
sales@reversinglabs.com