VulNow Q1 2026:
58 PreCVE Detections Confirmed

Q1 2026 marks VulNow's first full quarter of production PreCVE intelligence. Across 28 open source packages with a combined 13.2 billion monthly downloads, VulNow confirmed 58 detections before public CVE disclosure, with lead times ranging from 14 minutes to 154 days. Average lead time: 6.6 days.

Executive Summary

⚠ Most security programs are blind until a CVE exists. VulNow eliminates that blind spot.

In Q1 2026, VulNow identified **58 verified vulnerabilities** before public CVE disclosure, across a subset of open source packages representing **13.2 billion monthly downloads**. Every detection was later confirmed by a published CVE.

These detections were not isolated to niche components. They occurred in widely deployed libraries including axios, react, django, authlib, undici, lodash, and cryptography, which serve as foundational dependencies across modern software systems.

58

PreCVE Detections

Verified vulnerabilities identified before public CVE disclosure

154d

Max Lead Time

154 days 15 hours before disclosure — maximum lead time in Q1

6.6d

Avg Lead Time

Average days of advance warning across all 58 detections

28

Packages Covered

Open source packages monitored across npm and PyPI ecosystems

Each finding represents a confirmed vulnerability identified before public disclosure. This quarter represents VulNow's first full period of production PreCVE intelligence. The dataset reflects only a portion of the packages currently under analysis as ingestion and validation pipelines scale. Even within this limited scope, the results demonstrate consistent early visibility into real-world risk across widely deployed software dependencies.

✓ This report provides the first production evidence of PreCVE detection at scale.

The Core Finding

> Vulnerabilities in widely used open source packages are often discoverable before any CVE is published. The fix exists in the code. The advisory does not.

Security teams scanning CVE feeds have no signal to act on during this window. Attackers, however, do not rely on CVE publication to identify exploitable conditions.

VulNow operates in that gap.

In Q1 2026, VulNow confirmed 58 PreCVE detections across 28 packages. These include critical layers of the software ecosystem:

Frontend Frameworks

react and react-router

Backend Platforms

django, flask, and aiohttp

Node.js Infrastructure

axios, undici, minimatch, and picomatch

Security-Sensitive Libraries

authlib, cryptography, PyJWT, and pyasn1

Each detection was identified prior to public disclosure and subsequently validated by a corresponding CVE. This represents a shift from reactive vulnerability management to predictive vulnerability intelligence.

Key Outcomes

58 PreCVE Detections

Across 28 packages spanning npm and PyPI ecosystems

Lead Time Range

Maximum lead time: 154 days 15 hours before public disclosure. Average lead time: 6.6 days

52% Arrived 24+ Hours Early

More than half of detections arrived more than 24 hours before public CVE disclosure

High Severity Concentration

Strong alignment to real-world impact with high severity concentration across detections

Full Ecosystem Coverage

Spanning frontend, Python, Node.js infrastructure, and authentication ecosystems

Severity Alignment

Strong alignment between PreCVE severity and confirmed CVE severity.

⚠ More than half of all detections occurred before most organizations would typically begin triage following a new CVE announcement.

The Exploitation Race

After CVE Publication, Defenders Are Already Behind

When a CVE is published, two clocks start simultaneously. They run at different speeds.

Palo Alto Networks' Cortex Xpanse team monitored scanning activity across 50 million IP addresses during Q1 2021. Attackers begin scanning for vulnerable systems within 15 minutes of a CVE announcement. For the Microsoft Exchange Server vulnerabilities disclosed in March 2021, scanning began within 5 minutes. Threat actors inventory exposed assets on cycles of once per hour or less, and accelerate that cadence immediately when a new CVE drops.

The defender clock runs slower by orders of magnitude.

Organizations take an average of 12 hours just to identify which of their own systems are exposed after a new vulnerability is announced (Palo Alto, 2021 Cortex Xpanse Attack Surface Threat Report). The average time to patch a critical vulnerability is 16 days (Ponemon Institute). That is the gap where most breaches occur: between CVE publication and patch deployment. Attackers spend that window scanning, targeting, and exploiting. Defenders spend it triaging, prioritizing, and scheduling.

15m

Attacker Scan Start

Attackers begin scanning within 15 minutes of CVE announcement

12h

Defender Identification

Average time for organizations to identify exposed systems

16d

Avg Patch Time

Average time to patch a critical vulnerability (Ponemon Institute)

5d

Mean Exploitation

Mean time to exploitation in 2023, down from 63 days in 2018–2019 (Mandiant)

The exploitation timeline has compressed sharply. Mandiant's analysis of 138 vulnerabilities tracked as exploited in the wild during 2023 found a mean time to exploitation of 5 days, down from 63 days in 2018 and 2019 (Google Cloud Threat Intelligence, 2024). Among n-day vulnerabilities in that dataset, 29% were exploited within one week of disclosure and 56% within one month. Rapid7's 2026 Global Threat Landscape Report found the median time from CVE publication to CISA Known Exploited Vulnerabilities catalog inclusion dropped to 5 days in 2025, down from 8.5 days the year prior. Exploited high and critical severity vulnerabilities increased 105% year over year.

> ⊗ A 16-day average patching window against a 5-day average exploitation window is not a security posture. It is an assumption that attackers will select other targets during the interim.

However, the deeper problem begins earlier.

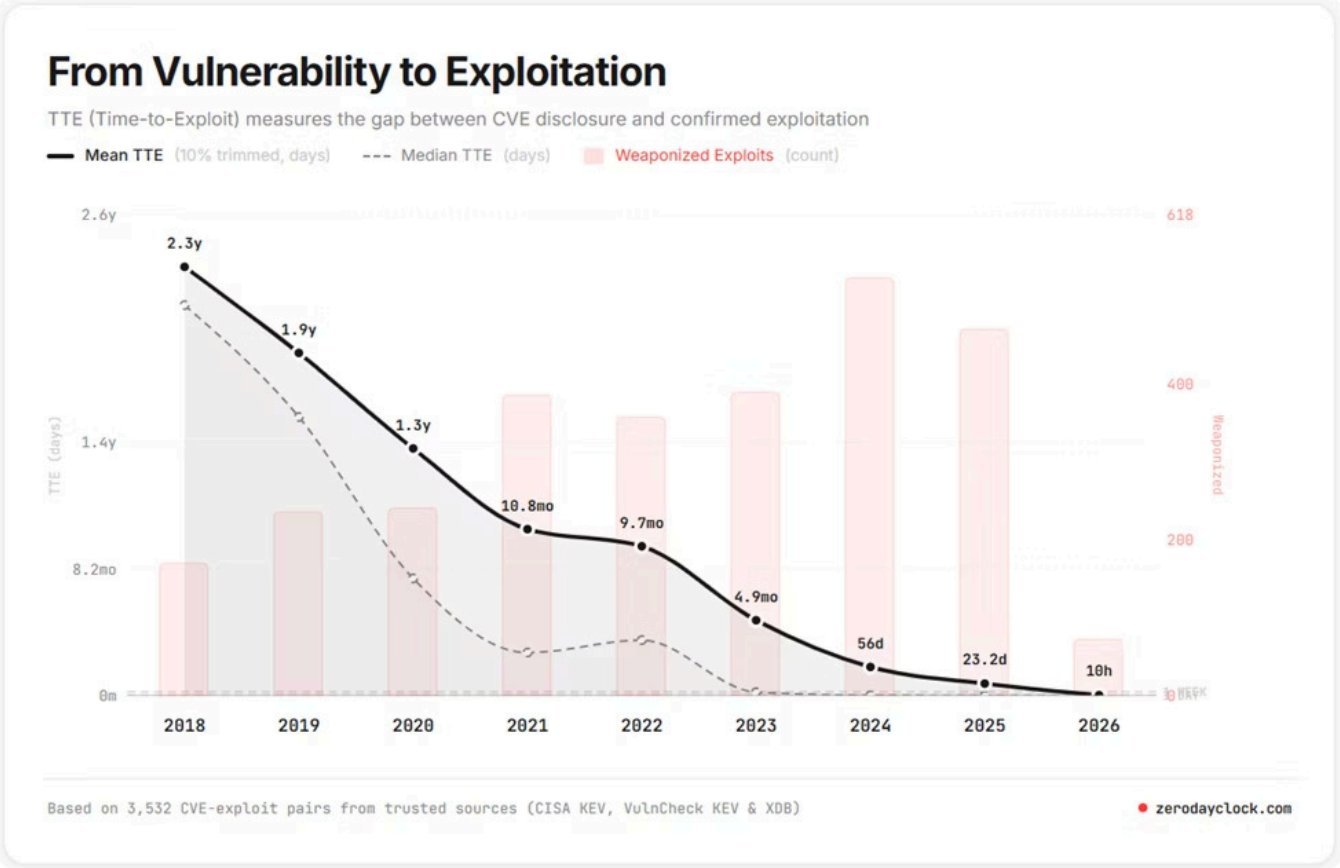


Image: Zero Day Clock. (n.d.). Retrieved April 24, 2026, from https://zerodayclock.com

Attackers Are Not Waiting for the CVE

The post-CVE race is the known problem. The PreCVE window is where the structural disadvantage runs deeper.

Unit 42's analysis of 45,450 public exploits found that 80% of exploits are published before the corresponding CVE is officially released, with an average lead of 23 days (Unit 42, State of Exploit Development, 2024). The lag is partly structural: CVE IDs are assigned but the official CVE record takes an average of 40 days to be published after assignment. The exploit ecosystem operates ahead of the CVE notification system by design.

VulnCheck's 2024 analysis of known exploited vulnerabilities confirms the downstream consequence:

- 23.6% of CVEs ultimately added to CISA's Known Exploited Vulnerabilities catalog were already being weaponized on or before the day the CVE was publicly disclosed.
- In H1 2024, 70% of exploited vulnerabilities had a publicly available proof-of-concept prior to their exploitation being disclosed publicly.

The commercial infrastructure around this dynamic is well-documented. Kaspersky's review of 547 exploit listings across dark web forums and shadow channels from January 2023 through September 2024 found that 51% targeted zero-day or one-day vulnerabilities. Remote code execution exploits averaged approximately $100,000 per listing. The market exists because PreCVE access is the most valuable window: defenders have no signal, no patch, and no advisory to act on.

> ⓘ This is the window VulNow is designed to operate in.

Every one of the 58 confirmed detections in Q1 2026 was identified before CVE publication, including vulnerabilities in widely deployed components such as axios, react-router, django, authlib, and undici.

> The asymmetry is direct. An attacker with PreCVE knowledge operates in a window with no defender awareness. An organization with VulNow intelligence has that same window available for remediation instead.

The Silent Fix Dynamic

The 154-day axios detection illustrates the underlying mechanism that makes PreCVE detection possible.

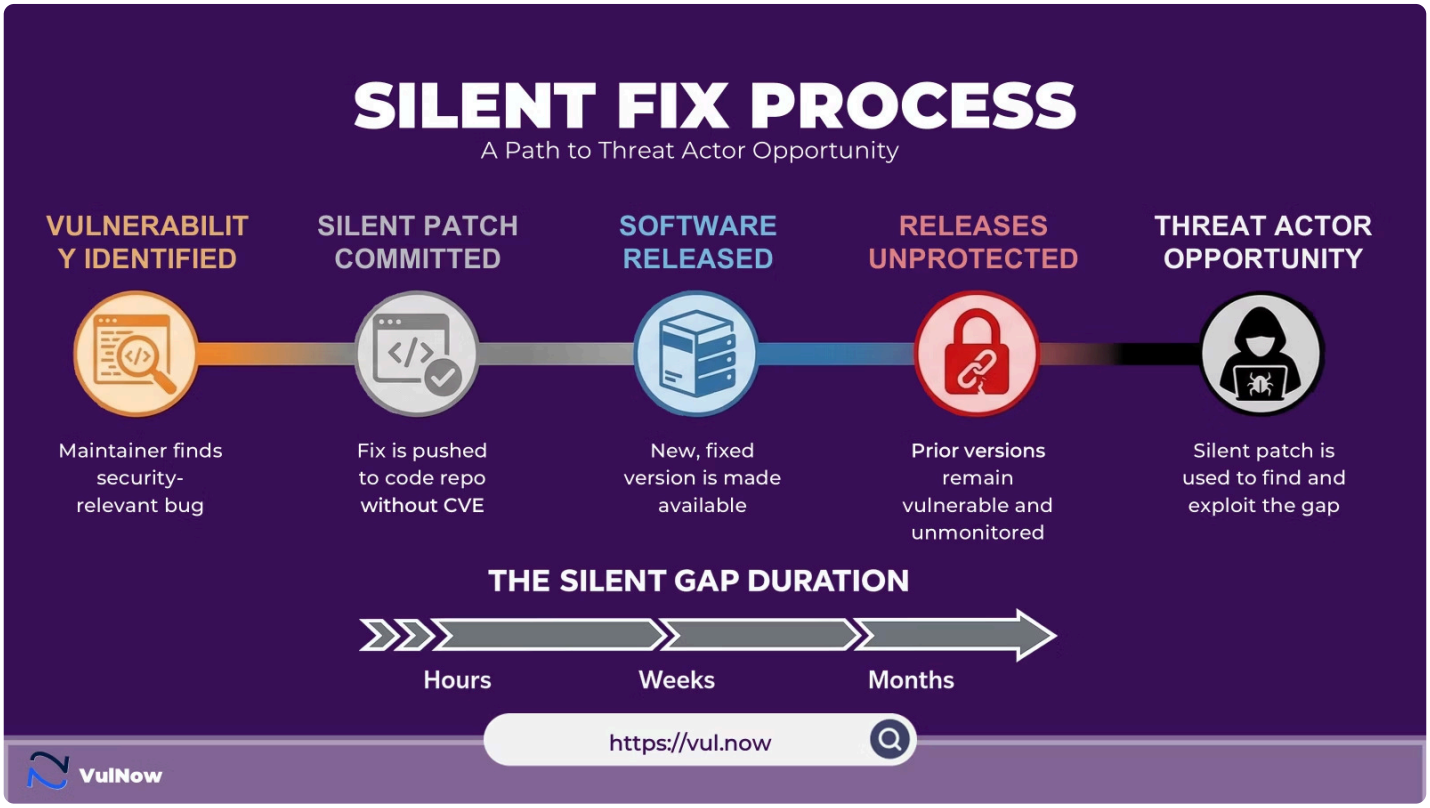
How It Works

Maintainers fix security bugs. They do not always file CVEs immediately, and coordinated disclosure processes frequently run on timelines measured in months.

The Consequence

During that window, some users update organically and receive the fix without ever knowing a vulnerability existed. Most users do not. CVE-based security tools have no signal to surface.

⚠ This is not an edge case. It is a structural feature of how open source security maintenance works in practice.



Why This Matters Now

The industry is consolidating around a narrative that AI will find and fix vulnerabilities automatically. In practice, the opposite pattern is emerging.

Vulnerabilities are found but not always disclosed

Fixes are made quietly, without coordinated advisories.

Security teams remain blind until CVEs appear

The CVE system, which was designed to provide timely notification, increasingly lags behind both the fix timeline and the exploit timeline.

The gap is structural, not temporary

The gap between when a vulnerability is patched and when it is publicly named is not a temporary friction in the disclosure process. It is a structural feature of how open source security maintenance operates, and it is the window where the most asymmetric risk accumulates.

> VulNow operates in that gap. The Q1 data is the first production evidence that operating there at scale is achievable.

What This Means: For Security Teams and Partners

A vulnerability management program built on CVE feeds had no coverage for any of these 58 events until the moment of public disclosure.

Broken down by window:

30+ Days Early

2 vulnerabilities with more than 30 days of undetected exposure

1–30 Days Early

28 vulnerabilities with 1 to 30 days of undetected exposure

1 Hour–24 Hours Early

23 vulnerabilities with 1 hour to 24 hours of undetected exposure

Under 1 Hour Early

5 vulnerabilities with under 1 hour of undetected exposure

During these windows, organizations relying on CVE-based tooling would have reported a clean bill of health, despite the presence of exploitable conditions.

For organizations that experienced an adversary independently identifying and exploiting these vulnerabilities prior to disclosure, there would have been no defensive signal and no indication of exposure.

For organizations operating with VulNow intelligence, the operational difference is concrete:

- Earlier prioritization before vulnerability disclosure noise begins
- Reduced exposure window for exploitable conditions
- More effective allocation of remediation resources
- Focus on validated risk rather than volume of CVE alerts
- A measurable advantage over purely reactive vulnerability management programs

The underlying shift is operational, not theoretical.

> The question shifts from "when did the CVE drop?" to "when did we receive the signal and what was our response time?"

The Full Q1 2026 Confirmation Table

The table below represents the complete set of confirmed PreCVE detections for Q1 2026. Each row corresponds to a detection that was identified by VulNow prior to public CVE disclosure and subsequently validated by a published CVE.

Table 1: Full Q1 2026 PreCVE Confirmation Data — all 58 confirmed detections with package name, VulNow detection ID, CVE ID, lead time, severity, and monthly download figures. This table will be populated with the full dataset as provided in the production report.

Column	Description
Lead Time	Time between VulNow detection and public CVE disclosure
CVE	Corresponding CVE published after VulNow detection
Package	Open source package name
CVE Severity	Final severity rating from published CVE record
PreCVE Severity	Severity rating assigned by VulNow at detection time
Earliest PreCVE	Internal VulNow identifier assigned at time of detection

Q1 2026 Confirmation Table (rows 1–29)

#	Lead Time	CVE	Package	CVE Severity	PreCVE Severity	Earliest PreCVE
1	154d 15h	CVE-2026-39865	axios	Med	Med	VULNOW-2025-00485
2	93d 16h	CVE-2025-61686	react-router	Crit	Med	VULNOW-2025-00050
3	26d	CVE-2026-28802	authlib	Crit	High	VULNOW-2026-01604
4	14d 5h	CVE-2026-28498	authlib	High	High	VULNOW-2026-02264
5	14d 4h	GHSA-7432-952r-cw78	authlib	High	High	VULNOW-2026-02262
6	14d 4h	CVE-2026-27962	authlib	Crit	Crit	VULNOW-2026-02261
7	8d 6h	CVE-2025-61920	authlib	High	Med	VULNOW-2025-00029
8	4d 5h	CVE-2026-33532	yaml	Med	High	VULNOW-2026-02644
9	4d 2h	CVE-2026-34518	aiohttp	Med	Med	VULNOW-2026-02808
10	4d 2h	CVE-2026-34525	aiohttp	Med	High	VULNOW-2026-02806
11	3d 57m	CVE-2026-27606	rollup	Crit	Crit	VULNOW-2026-01711
12	2d 4h	CVE-2025-69229	aiohttp	Med	Med	VULNOW-2026-01243
13	2d 4h	CVE-2025-69226	aiohttp	Med	High	VULNOW-2026-01240
14	2d 4h	CVE-2025-69223	aiohttp	High	High	VULNOW-2026-01241
15	2d 2h	CVE-2026-22030	react-router	Med	High	VULNOW-2026-01265
16	2d 2h	CVE-2026-22029	react-router	High	High	VULNOW-2026-01255
17	2d 2h	CVE-2026-21884	react-router	High	Med	VULNOW-2026-01263
18	2d 1h	CVE-2026-34043	serialize-javascript	Med	High	VULNOW-2026-02727
19	1d 21h	CVE-2026-33672	picomatch	Med	Med	VULNOW-2026-02683
20	1d 20h	CVE-2026-33671	picomatch	High	High	VULNOW-2026-02686
21	1d 18h	CVE-2026-34073	cryptography	Med	High	VULNOW-2026-02735
22	1d 12h	GHSA-9ppj-qmqm-q256	node-tar	High	High	VULNOW-2026-02378
23	1d 10h	CVE-2026-27903	minimatch	High	High	VULNOW-2026-01829
24	1d 10h	CVE-2026-27904	minimatch	High	High	VULNOW-2026-01835
25	1d 4h	CVE-2026-33151	socket.io	High	Med	VULNOW-2026-02550
26	1d 3h	CVE-2026-3304	multer	High	High	VULNOW-2026-02222
27	1d 3h	CVE-2026-2359	multer	High	High	VULNOW-2026-02221
28	1d 3h	CVE-2026-33036	fast-xml-parser	High	Med	VULNOW-2026-02529
29	1d 14m	CVE-2026-1527	undici	Med	High	VULNOW-2026-02472

Q1 2026 Confirmation Table (rows 30–58)

#	Lead Time	CVE	Package	CVE Severity	PreCVE Severity	Earliest PreCVE
30	1d 14m	CVE-2026-2229	undici	High	Med	VULNOW-2026-02474
31	23h 45m	CVE-2026-1525	undici	Med	High	VULNOW-2026-02473
32	23h 45m	CVE-2026-1528	undici	High	Med	VULNOW-2026-02475
33	23h 31m	CVE-2026-32597	pyjwt	High	Med	VULNOW-2026-02476
34	12h 14m	CVE-2026-30922	pyasn1	High	High	VULNOW-2026-02546
35	11h 16m	CVE-2026-4923	path-to-regexp	Med	High	VULNOW-2026-02752
36	11h 16m	CVE-2026-4926	path-to-regexp	High	Med	VULNOW-2026-02753
37	11h 2m	CVE-2026-27901	svelte	Med	High	VULNOW-2026-01871
38	11h 2m	CVE-2026-27902	svelte	Med	High	VULNOW-2026-01870
39	10h 4m	CVE-2026-27205	flask	Med	Med	VULNOW-2026-01688
40	9h 50m	CVE-2025-68480	marshmallow	Med	Med	VULNOW-2025-01034
41	8h 23m	CVE-2026-32274	black	High	Med	VULNOW-2026-02463
42	5h 45m	CVE-2026-29074	svgo	High	High	VULNOW-2026-02326
43	5h 45m	CVE-2025-13465	lodash	Med	High	VULNOW-2026-01444
44	5h 31m	CVE-2026-3520	multer	High	High	VULNOW-2026-02330
45	5h 16m	CVE-2020-7660	serialize-javascript	High	High	VULNOW-2026-02224
46	4h 48m	CVE-2026-27119	svelte	Med	High	VULNOW-2026-01687
47	4h 48m	CVE-2026-27121	svelte	Med	High	VULNOW-2026-01686
48	4h 48m	CVE-2026-27122	svelte	Med	High	VULNOW-2026-01685
49	3h 36m	CVE-2026-27199	werkzeug	Med	Med	VULNOW-2026-01700
50	1h 40m	CVE-2026-33349	fast-xml-parser	Med	High	VULNOW-2026-02616
51	1h 40m	CVE-2026-4800	lodash	High	High	VULNOW-2026-02905
52	57m	CVE-2026-1526	undici	High	High	VULNOW-2026-02471
53	43m	CVE-2026-27942	fast-xml-parser	High	Med	VULNOW-2026-02114
54	28m	CVE-2026-23864	react	High	High	VULNOW-2026-01489
55	28m	CVE-2026-25645	requests	Med	Med	VULNOW-2026-02728
56	14m	CVE-2026-1285	django	High	Med	VULNOW-2026-01584
57	14m	CVE-2025-14550	django	High	Med	VULNOW-2026-01585
58	14m	CVE-2025-13473	django	Med	Low	VULNOW-2026-01586

Lead Time Analysis

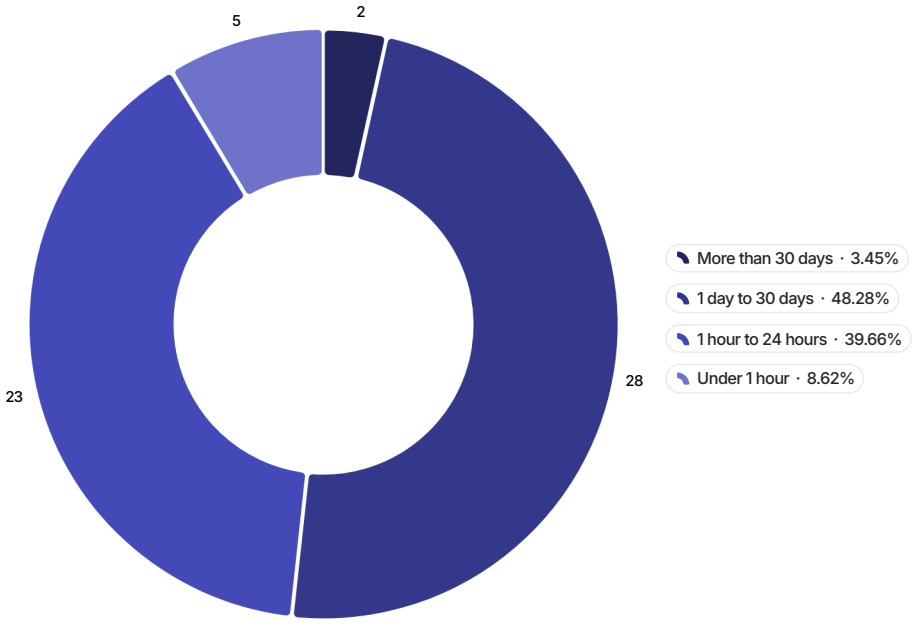
Lead time is the primary measure of value. A detection that arrives a day before public disclosure gives a security team one day to act. One that arrives 154 days before gives a security team a full quarter and then some.

Distribution Across Q1

Table 2: Lead Time Distribution — breakdown of all 58 detections by lead time bucket, showing a percentage of total detections in each window category.

Window	Count	Share
More than 30 days	2	3%
1 day to 30 days	28	48%
1 hour to 24 hours	23	40%
Under 1 hour	5	9%

More than half of all detections arrived more than a full day before public CVE disclosure. Two arrived over a month early. And VulNow has many, many more PreCVEs that have not been disclosed in a CVE yet.

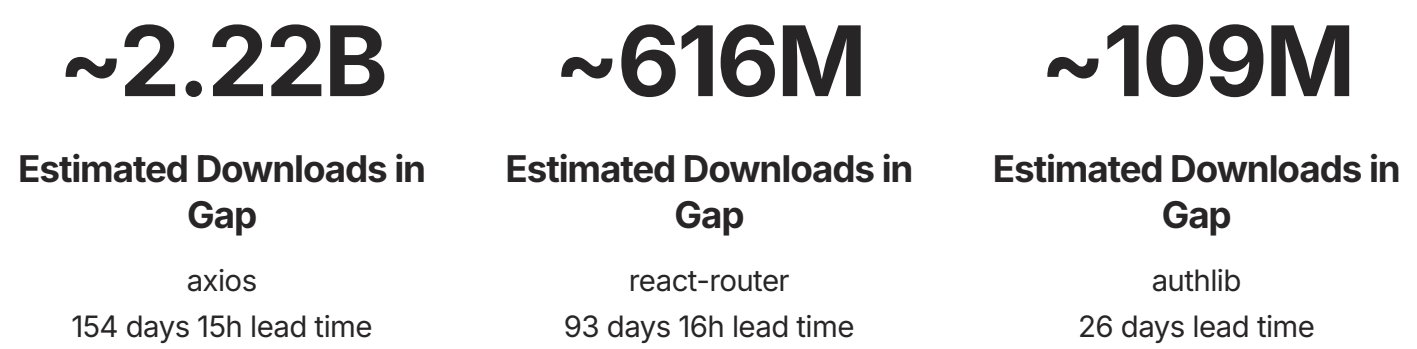


Download Exposure per Detection Window

For the longest lead times, the volume of unprotected downloads during the gap is significant.

Table 3: Download Exposure per Detection Window — for each major lead time bucket, the estimated number of package downloads that occurred while VulNow held an active PreCVE detection and CVE-based scanners had nothing to flag.

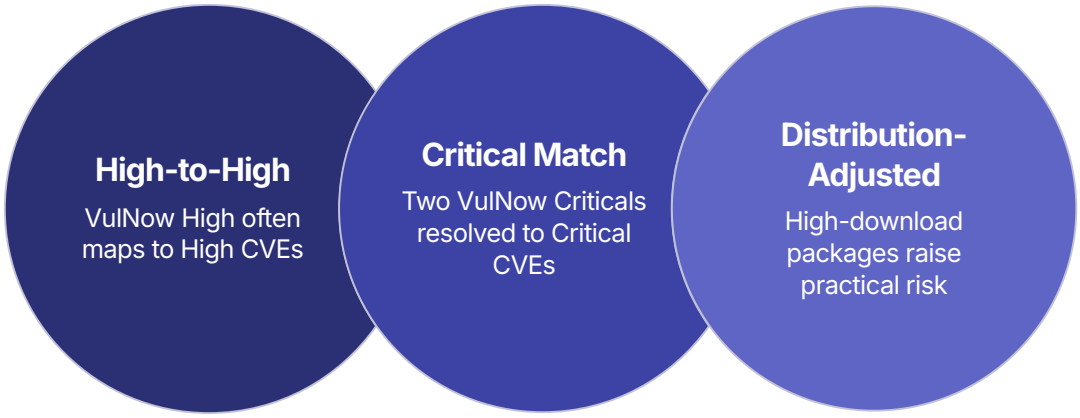
Package	Lead Time	Monthly Downloads	Estimated Downloads in Gap
axios	154 days 15h	429,947,258	~2.22 billion
react-router	93 days 16h	197,209,993	~616 million
authlib	26 days	125,547,108	~109 million



Severity Correlation

PreCVE intelligence is only useful if the severity assessments are reliable. Alert fatigue from low-quality signals erodes trust faster than no signals at all.

The Q1 data shows strong correlation between VulNow PreCVE severity and final CVE severity:



High-to-High Alignment

Where VulNow rated High, the resulting CVE was High in the majority of cases.

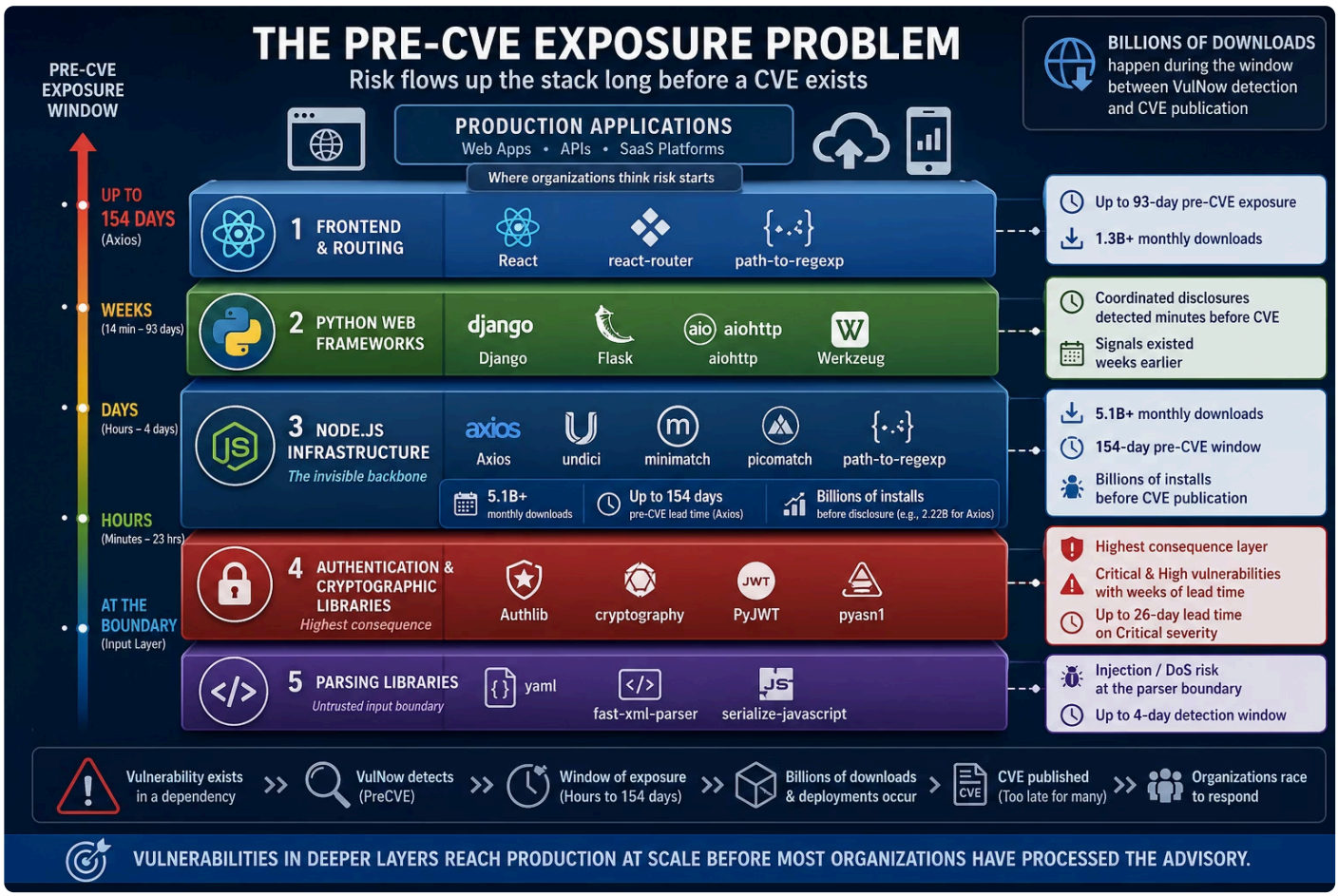
Critical-to-Critical Exact Match

Two detections were rated Critical by VulNow (entries 6 and 11); both resolved to Critical CVEs, an exact match. This reflects precise calibration at the upper end, which is the correct posture for a pre-disclosure signal.

Distribution-Adjusted Severity

A further pattern reinforces the value of PreCVE signals: in some cases, VulNow rated detections higher than the eventual CVE score. This reflects how standard CVE scoring systematically underweights impact for packages with very high distribution. At 500 million or 1 billion monthly downloads, a vulnerability that scores Moderate in isolation carries a materially different risk profile in practice. PreCVE signals calibrated to distribution scale are more operationally accurate than post-disclosure scores alone.

Ecosystem Breakdown



Frontend and Routing

React, react-router, and path-to-regexp together produced 7 detections in Q1. React-router yielded 4 confirmed PreCVE detections with lead times from 2 days to 93 days. Combined monthly downloads across these three packages exceed 1.3 billion. Vulnerabilities in this layer reach production deployments at scale before most organizations have processed the advisory.

Python Web Frameworks

Django, Flask, aiohttp, and Werkzeug appeared repeatedly across the quarter. Django's three CVEs (entries 56, 57, 58) were all detected 14 minutes before publication, indicating a coordinated disclosure that VulNow's signals had pre-dated at the earliest-detection level (VULNOW-2026-01584 through 01586). Aiohttp yielded 5 detections with lead times from 2 days to 4 days, with all five clustered within a coordinated disclosure batch.

Node.js Infrastructure

Axios, undici, minimatch, picomatch, and path-to-regexp are the infrastructure layer of the npm ecosystem. Most developers have no direct relationship with these packages, yet they appear in virtually every Node.js project through HTTP clients, build tooling, test runners, and frameworks. Their combined monthly download count exceeds 5.1 billion. Axios registered the single longest lead time in the Q1 dataset at 154 days 15 hours with 429 million monthly downloads, meaning approximately 2.22 billion downloads of axios occurred in the window between VulNow's detection and CVE publication. Minimatch's two High-severity CVEs (entries 23 and 24) arrived with over 34 hours of pre-detection lead time, representing approximately 81 million daily installs without CVE coverage. Undici alone yielded 5 CVEs in a single coordinated disclosure event.

Authentication and Cryptographic Libraries

Authlib, cryptography, PyJWT, and pyasn1 are the highest-consequence cluster in this dataset. Authlib produced 5 separate detections in Q1, including one exact Critical-to-Critical severity match (entry 6) and a 26-day lead time on CVE-2026-28802, which was ultimately rated Critical. Cryptography, at over 1 billion monthly downloads, produced a High-severity detection. PyJWT's High CVE (entry 33) was detected 23 hours and 31 minutes before public disclosure.

Parsing Libraries

Fast-xml-parser, yaml, and serialize-javascript handle untrusted input at system boundaries. Fast-xml-parser yielded 3 detections across the quarter. The yaml package had a 4-day detection window before its CVE. Vulnerabilities in parsing libraries frequently translate to injection conditions or denial-of-service at the parser boundary, making early visibility particularly valuable.

Download Exposure: What Was at Stake

The packages involved are not niche libraries. They are foundational dependencies present in the majority of production software stacks.

PyPI Ecosystem (monthly downloads, April 2026)

Table 4: PyPI Ecosystem Monthly Downloads — package-by-package breakdown of monthly download volumes for all PyPI packages included in Q1 2026 detections, sourced from the PyPI Stats API as of April 2026.

Package	Monthly Downloads	Role
requests	1,400,159,721	The Python HTTP library
cryptography	1,030,366,128	Cryptographic primitives, pervasive transitive dependency
pyjwt	542,090,195	JSON Web Token authentication
aiohttp	472,741,074	Async HTTP client and server for Python
pyasn1	455,611,225	ASN.1 types, dependency chain for SSL/TLS
werkzeug	240,416,305	WSGI utilities, the foundation under Flask
flask	197,839,658	Python web microframework
marshmallow	132,712,602	Object serialization and deserialization
authlib	125,547,108	OAuth 2.0 and OpenID Connect
black	116,195,251	Python code formatter
django	44,340,110	Python's batteries-included web framework

Combined: approximately 13.2 billion downloads per month across 28 packages. Every one of these packages was inside an active VulNow detection window before its CVE existed. For any organization running these dependencies in production, that gap was the exposure window.

npm Ecosystem (monthly downloads, April 2026)

Table 5: npm Ecosystem Monthly Downloads — package-by-package breakdown of monthly download volumes for all npm packages included in Q1 2026 detections, sourced from the npm Registry API as of April 2026.

Package	Monthly Downloads	Role
minimatch	2,301,162,973	Glob matching, pulled by virtually every Node.js build tool
picomatch	1,419,215,687	Glob pattern matching, universal transitive dependency
path-to-regexp	643,720,360	URL routing, powers Express.js and most Node.js frameworks
lodash	579,227,043	JavaScript utility library
yaml	565,801,464	YAML parser used across CI/CD and config tooling
react	505,903,135	The dominant frontend UI framework
axios	429,947,258	HTTP client for JavaScript and Node.js
rollup	409,320,729	JavaScript module bundler
undici	315,680,434	Node.js HTTP client, the engine under fetch
tar (node-tar)	~306,000,000*	Tar archiving, used by npm CLI and package managers
fast-xml-parser	293,586,788	XML and HTML parsing
serialize-javascript	208,228,076	Safe serialization, used in Webpack
react-router	197,209,993	Client-side routing for React
svgo	124,233,818	SVG optimization
multer	57,628,504	Multipart file upload for Express
socket.io	52,409,290	Real-time bidirectional communication
svelte	17,690,731	Compiler-first frontend framework

Weekly downloads confirmed at 70,588,127 via npm Registry API; monthly figure estimated from weekly cadence. The npm package is published as tar. The node-tar package is a deprecated redirect stub with 1,039 weekly downloads.

Summary

Based on a limited subset of open source packages as VulNow launched production in Q1 2026.

Table 6: Q1 2026 Summary Statistics — aggregate summary of all key metrics from the quarter including total detections, packages covered, ecosystems represented, lead time distribution, severity breakdown, and total download exposure during active detection windows.

Metric	Value
PreCVE detections confirmed	58
Average lead time	6.6 days
Packages covered	28
Combined monthly downloads	~13.2 billion
Maximum lead time	154 days 15h (axios, CVE-2026-39865)
Detections with more than 1 day lead time	30 of 58 (52%)
Package with most confirmed CVEs	authlib, aiohttp, svelte, undici (5 each)
Package with longest single lead time	axios (154 days 15h)

This is an early but meaningful signal of what becomes possible when vulnerability discovery shifts from reactive to predictive. We are just beginning to scale.

References

VulNow is an early-warning vulnerability intelligence platform specializing in PreCVE detection across the open source ecosystem. Q1 2026 data reflects confirmed detections from a subset of analyzed packages as ingestion pipelines scale toward full production coverage. Download statistics sourced from the npm Registry API and PyPI Stats API as of April 2026.

External sources cited:

- Palo Alto Networks 2021 Cortex Xpanse Attack Surface Threat Report
- Unit 42 State of Exploit Development (2024)
- Unit 42 Incident Response Reports (2022, 2024, 2026)
- Google Cloud Threat Intelligence / Mandiant "How Low Can You Go? An Analysis of 2023 Time-to-Exploit Trends" (2024)
- Rapid7 2026 Global Threat Landscape Report
- VulnCheck 2024 Exploitation Trends and H1 2024 State of Exploitation
- Kaspersky dark web exploit market analysis (2024)
- Ponemon Institute patch management research
- zerodayclock.com

For more information about VulNow, contact **Cassie Crossley, CEO/Co-Founder** or **Valerio Mulas, CTO/Co-Founder** on LinkedIn.