VULNOW Q2 2026 VULNERABILITY DISCLOSURE REPORT

The Disclosure Gap,

Measured

In 2026 the systems that tell defenders what is vulnerable stopped keeping pace with the systems that produce vulnerabilities. VulNow monitors nearly 20,000 open source repositories and raises more than 100 new findings a day. This report examines the 320 or more of those findings that have since been confirmed by a public security advisory, what the disclosure record around them shows, and why the gap they sit in is widening.

> *Two of the three data sources the vulnerability management industry is built on have publicly stated that they cannot process the current volume. The third, the maintainers themselves, never agreed to be a data source in the first place.*

About VulNow

VulNow B.V. is a predictive software supply chain risk intelligence platform based in Amsterdam. VulNow identifies undisclosed vulnerabilities, silent patches, and Dark Matter Vulnerabilities in open source code before public advisory publication, and delivers that intelligence to enterprise defenders and to security vendors who embed it in their own products.

The confirmed PreCVE dataset described in this report is public, anonymous, and read only at https://precve.vulnow.com, with an OSV format API for programmatic access.

For more information, contact info@vul.now, or Cassie Crossley, CEO / Co-Founder, or Valerio Mulas, CTO / Co-Founder, directly.

Contents

Summary

A PreCVE is a vulnerability that VulNow identified in open source code before any public security advisory for it existed.

VulNow monitors nearly 20,000 open source repositories and currently raises more than 100 new findings per day. Around 8,500 findings are open at the time of writing. Those are initial findings at varying stages of triage. They span security fixes and ordinary bug fixes, they are not a verified count of vulnerabilities, and we do not present them as one.

This report concerns the subset that has since been confirmed by a published security advisory, whether a GitHub Security Advisory, a CVE record, or both. That group currently numbers more than 320. It is the subject here for a simple reason: the outcome is a matter of public record, the full set is published at https://precve.vulnow.com, and anyone can check our work against it.

320+	~8,500	~20,000	100+
Findings confirmed by a published advisory	Open findings, security and non-security	Repositories monitored	New findings raised per day

Three findings stand out.

- Roughly **36 percent** of confirmed findings resolved to a GitHub Security Advisory that carries **no CVE record at all**. For those vulnerabilities, a CVE-driven scanner has nothing to match today, and may never have anything.
- The rate at which findings are raised now materially exceeds the rate at which the public advisory system absorbs them. That gap is the subject of section four.
- Coverage in this dataset spans 97 distinct packages across npm, PyPI, Packagist, Maven, Go modules, and container images. Vulnerabilities that reach production through a dependency graph do not respect ecosystem boundaries, and neither does the disclosure backlog.

> *The question this report tries to answer is not whether vulnerabilities are being found. They are, at record volume. It is whether defenders are being told.*

What We Measured

PreCVE detections

VulNow monitors open source repositories for security-relevant changes that have shipped without a corresponding advisory. Maintainers fix security bugs continuously. They do not always file a CVE, and coordinated disclosure processes routinely run on timelines measured in weeks or months. In the interval between the fix landing and the advisory appearing, the vulnerability is fully described in public source code and entirely absent from every vulnerability database a security team consumes.

These are what VulNow calls **Dark Matter Vulnerabilities™**: real, present, and invisible to instruments calibrated to look for advisories.

Confirmation, and a note on GHSA

Throughout this report, a detection is described as confirmed when a public security advisory for the same vulnerability later appeared. In practice that means one of two identifiers, and often both.

- A CVE record, the identifier most vulnerability management programs are built around, assigned by a CVE Numbering Authority and published through the CVE Program.
- A GitHub Security Advisory, or GHSA. This is the advisory format used by the GitHub Advisory Database, which is the data source behind Dependabot and behind a large share of the software composition analysis tooling in the market. A GHSA can exist with a CVE attached, or entirely on its own.

The distinction matters more than it used to, and section four explains why. Where this report says confirmed, it means confirmed by a public security disclosure, whether that disclosure carries a CVE identifier, a GHSA identifier, or both.

Lead time

Lead time is the interval between VulNow identifying the vulnerability and the first public advisory for it appearing. During that window there is no record for a scanner to match against, so no amount of polling frequency helps: the information a conventional tool needs does not yet exist anywhere.

That distinction matters more than it first appears, and section nine returns to it. A vulnerability management program has two separate latencies to manage. One is how long it takes for an advisory to exist. The other is how long it takes the program to notice once it does. Lead time addresses the first, which is the one no amount of tuning on the customer side can shorten.

Figure 1. Conceptual timeline of the disclosure gap between detection and public record.

2026: The Year Disclosure Infrastructure Hit Its Ceiling

The context for this quarter's data is that the public vulnerability disclosure system is now operating beyond its designed capacity, and the organizations running it have said so directly.

The GitHub Advisory Database

In May 2026, the GitHub Advisory Database published 1,560 reviewed advisories. That is more than five times its typical monthly output and the highest total in its history. Two years earlier the same database was publishing roughly 270 advisories per month.

It was still not enough to keep up. Across every input channel, volume rose at once. Private vulnerability reports went from roughly 550 per week in January to more than 3,000 per week for most of May. Repository advisories rose from roughly 650 per week to more than 5,000. CVE requests to GitHub as a CNA reached almost 4,000 in May alone, close to ten times the prior year.

GitHub has been candid about the consequences. Since mid-April, publication times extended first to about a week, and then to multiple weeks for a meaningful share of advisories. In GitHub's own words, longer publication times can increase exposure windows.

Source: GitHub, "Inside the Advisory Database and what happens when vulnerability volume breaks records," 29 June 2026.

The National Vulnerability Database

On 15 April 2026, NIST changed how the NVD operates. Enrichment is no longer universal. The NVD now prioritizes three categories: CVEs in CISA's Known Exploited Vulnerabilities catalog, CVEs affecting software used by the United States federal government, and CVEs affecting critical software as defined under Executive Order 14028. Everything else is labelled lowest priority and not scheduled for immediate enrichment.

At the same time, the entire backlog of unenriched CVEs published before 1 March 2026, roughly 29,000 records, was moved into the not scheduled category. NIST enriched close to 42,000 CVEs in 2025, 45 percent more than any previous year, and still could not clear the queue. Submissions in the first quarter of 2026 ran nearly a third higher than the same period the year before.

The practical effect is that a large and growing share of CVE records now arrive without a CVSS score, without weakness classification, and without the affected product mappings that automated tooling relies on to match a vulnerability to an installed component.

> *Two of the three data sources the vulnerability management industry is built on have publicly stated that they cannot process the current volume. The third, the maintainers themselves, never agreed to be a data source in the first place.*

The Gap, Measured in Our Data

Industry commentary about disclosure backlogs is easy to produce and hard to verify. What follows is the same phenomenon expressed as counts, drawn from the confirmed findings published at precve.vulnow.com.

> **A note on these figures**
>
> The confirmed set is not static. It grows as maintainers and CVE Numbering Authorities publish advisories matching findings already in our queue, and it shifts as we refine the matching and classification logic behind the platform, which means individual entries are occasionally added or withdrawn. The analysis in this section was run against a snapshot of 328 confirmed findings taken in late July 2026. Counts and percentages should be read as accurate to that snapshot rather than as fixed values. The live set is always available at precve.vulnow.com.

How the confirmed findings resolved

Advisory outcome	Findings	Share
Resolved to both a GHSA and a CVE record	196	59.8%
Resolved to a GHSA with no CVE record	117	35.6%
Resolved to a CVE record with no GHSA	15	4.6%

Table 1. Advisory outcome for confirmed findings in the analysed snapshot.

More than a third of these vulnerabilities exist in the GitHub Advisory Database and nowhere in the CVE system. They are real, they are published, they affect widely deployed packages, and a vulnerability management program keyed to CVE identifiers has no record of them.

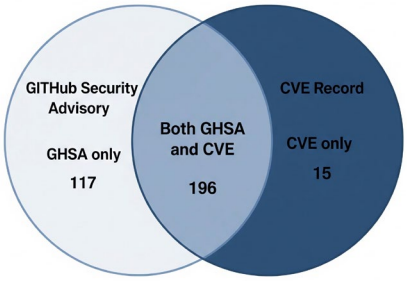

Figure 2. Venn diagram of GHSA and CVE advisories.

This is the disclosure backlog made concrete. It is not a prediction about what might happen if the CVE system stays overloaded. It is a count of what already did.

A smaller but equally instructive group runs the other way. Fifteen detections resolved to a CVE record with no GitHub Security Advisory, meaning tooling built on the GitHub Advisory Database alone would miss them. There is no single feed in this dataset that would have caught everything.

Lead time distribution

Advance warning	Findings	Share
More than 30 days	16	5%
7 to 30 days	135	41%
1 to 7 days	127	39%
1 hour to 24 hours	31	9%
Under 1 hour	19	6%

Table 2. Interval between VulNow detection and first public advisory, analysed snapshot.

Roughly **85 percent of confirmed findings** arrived more than a **full day before any public advisory**, and 46 percent arrived more than a week ahead. The mean lead time across the set is approximately 10 days. The longest single window remains axios at 154 days and 15 hours.

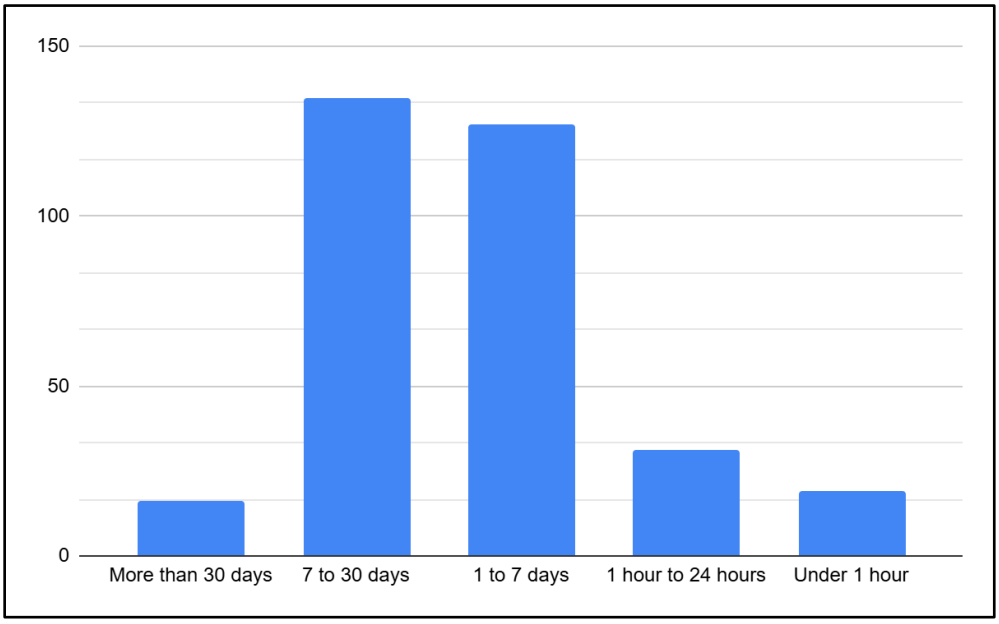

Figure 3. Chart showing the Advance Warning distribution.

Severity

Severity at detection	Findings	Share
Critical	16	5%
High	204	62%
Medium	85	26%
Low	23	7%

Table 3. Severity distribution across confirmed findings, analysed snapshot.

Two thirds of the set is rated High or Critical. This matters for a specific reason. When advisory publication is delayed and enrichment is triaged, the material that gets deprioritized is not reliably the low severity material. Severity is one of the attributes that enrichment is supposed to establish, so it is not available to prioritize with until after the delay it would have helped you manage.

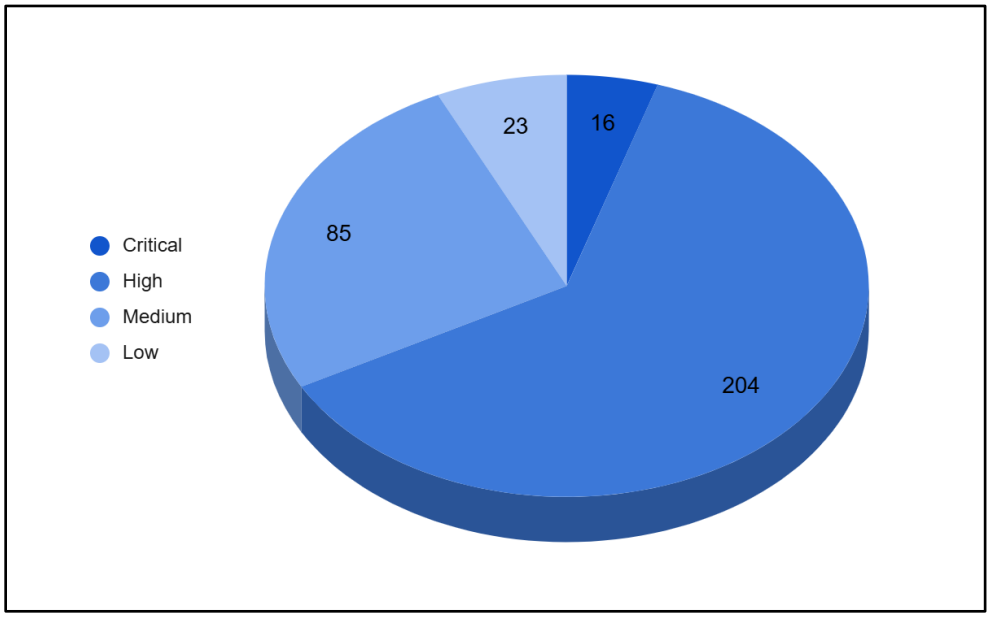

Figure 4. Chart showing the Severity distribution of disclosures.

What has not surfaced yet

Everything above describes the confirmed portion. Behind it sits a larger body of open findings, around 8,500 at the time of writing, at varying stages of triage.

We are deliberately careful about what that number means. Those findings are **candidate signals**, not verified vulnerabilities, and the set is not exclusively security material. It includes ordinary bug fixes alongside security fixes, because at the point of detection the two are not always distinguishable. Some findings will be confirmed as vulnerabilities. Some will resolve to issues that were never security relevant. Some are already inside a coordinated disclosure process and will surface on their own schedule. Some describe real problems that **will never receive an advisory** at all, because nobody files one.

What the figure does support is a directional claim, and it is the one that matters here. More than 100 new findings are raised per day across roughly 20,000 repositories, and those 20,000 are a fraction of the repositories in active use. Whatever proportion of that flow eventually proves out, it is arriving faster than the public advisory system is currently absorbing it. The confirmed findings in this report are the part of the gap that has closed, not a measure of its width.

Why the Quality Problem Compounds the Volume Problem

Throughput is only half of what has changed. The other half is that the average advisory now takes longer to process than it used to, because the average advisory arrives in worse condition.

GitHub's curation team has described the work in detail. A well formed advisory names the affected package and its ecosystem, documents the version range, and identifies the fix. A curator validates and publishes that in a few minutes. A growing share of incoming advisories require considerably more: working out whether a package named in an advisory refers to the npm, PyPI, or Maven artifact of the same name; reconstructing affected version ranges from commits, changelogs, and tags because the submitted range is missing or does not match release history; verifying vulnerabilities that span multiple registries independently; and adjudicating cases where the CVE record, the maintainer's advisory, and the commit history disagree about what is actually affected.

Historically the straightforward advisories dominated and the difficult ones could be absorbed. At current volume the queue fills with both, and the difficult ones take disproportionately longer. The effect compounds.

The reporting side has changed too

The upstream pressure on maintainers is now well documented. In January 2026 the curl project ended its bug bounty program, with its maintainer citing the unsustainable cost of triaging AI generated submissions. In March 2026 HackerOne halted new submissions to the Internet Bug Bounty, citing the mismatch between AI accelerated vulnerability discovery and the capacity of open source maintainers to validate and remediate what is reported. Linus Torvalds has described the Linux kernel security mailing list as almost entirely unmanageable, with enormous duplication from different people finding the same issues with the same tools.

This is not an argument that AI assisted vulnerability research is bad. Verified, reproducible findings produced with AI assistance are landing in real projects and are genuinely valuable. It is an argument that the cost of separating those from the noise has been pushed onto volunteers, and that the disclosure pipeline is where that cost shows up as delay.

> *The number of vulnerabilities being found has gone up. The number being clearly described, correctly scoped, and promptly published has not kept pace. The distance between those two numbers is the exposure window.*

What Verification Actually Costs

A platform that identifies more than 100 vulnerabilities a day has to answer an obvious question: how many of them are real, and how precisely are they scoped? Volume without verification is exactly the problem described in the previous section, not a solution to it.

Every VulNow detection that reaches the confirmed set has been through a documented analysis process. The standard we hold to is the following.

- Affected version ranges are bounded by testing rather than by assumption. We establish both ends of the range empirically against real published artifacts, and we test beyond the apparent boundary to confirm the range does not in fact extend further. Releases between verified boundaries are treated as affected. A single finding can span dozens of releases across many minor branches.
- The introduction point is treated as its own question. Where a vulnerability was introduced is a separate question from where it was fixed, and it is the one that determines whether a given deployment is exposed. An advisory that reports only the fixed version leaves every earlier release ambiguous, which is precisely the ambiguity that downstream curators and security teams then have to resolve for themselves.
- Exploitability is demonstrated, not asserted. Findings are reproduced end to end against the real artifact, and the patched version is tested against the identical input to confirm the fix actually closes the path.
- Weakness classification is validated rather than inherited. Where an upstream advisory assigns a CWE that does not match the actual exploit path, we correct it and say why.
- Severity is derived from the path we reproduced. Where our assessment differs from a published score, the difference is documented with the vector and the reasoning, not simply asserted.

This is deliberately the opposite of the pattern straining the disclosure system. GitHub's curators spend a significant share of their time correcting incomplete or inaccurate upstream data. Intelligence that arrives already carrying verified version ranges, a validated weakness classification, and reproduction evidence reduces that work rather than adding to it.

Notable Findings

The full set of confirmed findings is published and searchable at https://precve.vulnow.com. The selection below is illustrative rather than a ranking, chosen to show the range of lead times, ecosystems, and advisory outcomes present in the data.

Package	Ecosystem	Severity	Lead time	Advisory outcome
axios	npm	Medium	154d 15h	GHSA and CVE
react-router	npm	Medium	93d 17h	GHSA and CVE
authlib	PyPI	Medium	90d 9h	GHSA and CVE
sqlparse	PyPI	High	73d 4h	GHSA only, no CVE
authlib	PyPI	Medium	55d 21h	GHSA only, no CVE
python-dotenv	PyPI	Medium	50d 21h	GHSA and CVE
undici	npm	High	48d 16h	GHSA and CVE
v8	Source repo	High	39d 6h	CVE only, no GHSA
pillow	PyPI	High	37d 16h	GHSA and CVE
laravel/framework	Packagist	High	28d 2h	GHSA only, no CVE
ws	npm	High	24d	GHSA only, no CVE
markdown-it	npm	High	22d 11h	GHSA only, no CVE
cassandra-all	Maven	Medium	20d 18h	GHSA and CVE
dompurify	npm	Critical	20d 7h	GHSA only, no CVE
symfony/symfony	Packagist	High	20d 7h	GHSA only, no CVE
shell-quote	npm	Critical	17d 23h	GHSA and CVE
activemq-core	Maven	Critical	17d 15h	GHSA and CVE
twig/twig	Packagist	Critical	16d 8h	GHSA only, no CVE
jdbi3-freemarker	Maven	Critical	7d 15h	GHSA only, no CVE

Table 4. Selected confirmed findings. Full dataset at precve.vulnow.com.

What the long windows have in common

The detections at the top of Table 4 are not obscure components. axios, react-router, authlib, undici, and pillow are foundational dependencies pulled into an enormous share of production applications, usually transitively and often without the developer being aware the dependency exists.

The 154 day axios window remains the clearest single illustration of the mechanism. The fix shipped. Users who happened to upgrade during that period received it and never knew there had been a vulnerability. Users who did not upgrade stayed exposed for five months, with no advisory anywhere to tell them, while the fix commit sat in a public repository for anyone to read.

What the Critical findings without CVEs have in common

Four of the sixteen Critical severity detections in this dataset resolved to a GitHub Security Advisory with no CVE record. They affect a DOM sanitization library, a PHP templating engine, and a Java database utility. All three are the kind of component that sits directly on an untrusted input boundary.

For an organization running CVE-keyed tooling, those vulnerabilities do not exist.

Coverage Across the Dependency Graph

The Q1 2026 report covered 28 packages across two ecosystems, npm and PyPI. The matched set now spans 97 distinct packages across six.

Ecosystem	Findings	Share	Representative packages
npm	157	48%	axios, undici, minimatch, ws, dompurify
PyPI	84	26%	authlib, aiohttp, django, pillow, cryptography
Packagist	57	17%	symfony, laravel, twig, guzzle
Maven	21	6%	activemq, log4j, cassandra, jackson
Go and source repos	7	2%	prometheus, golang.org/x/net
Container images	2	1%	alpine and wolfi base packages

Table 5. Ecosystem distribution across confirmed findings, analysed snapshot.

The expansion beyond the JavaScript and Python ecosystems matters for a practical reason. Most organizations do not run one ecosystem. A typical production estate combines a JavaScript front end, a Python or PHP application tier, Java infrastructure services, and container base images, and the dependency graph crosses all of them. Vulnerability intelligence scoped to a single registry leaves the rest of that estate uncovered.

The concentration visible in the table is also worth noting. A small number of packages account for a large share of detections: symfony, undici, axios, django, aiohttp, authlib, multer, and minimatch together account for well over a third of the matched set. These are not the packages teams choose. They are the packages that arrive underneath the packages teams choose.

What This Means for a Security Program

Most vulnerability management programs share an unstated assumption: that the absence of a finding means the absence of a vulnerability. That assumption was always imperfect. **In 2026 it is measurably wrong**.

Three consequences follow from the data in this report.

A clean scan is a weaker signal than it was

For 36 percent of the vulnerabilities in this dataset, a **CVE-keyed scanner would have reported nothing**, because there was nothing to report. Not a delay in matching, an absence of the underlying record. Any program whose coverage claim rests on CVE feeds alone should understand what fraction of published advisories that actually represents.

Enrichment gaps break prioritization, not just detection

With the NVD now enriching only a prioritized subset, a growing number of CVE records carry no CVSS score and no affected product mapping. Tooling that matches on those fields silently under-reports, and severity thresholds calibrated against NVD scores apply to a shrinking share of the catalog. This is a configuration question worth asking of any vendor: which fields does the matching logic depend on, and what happens when those fields are empty?

Lead time is a controllable variable

The interval between a fix landing and an advisory appearing is not something a security team can influence. Whether they know about it during that interval is. In this dataset, **46 percent of detections offered more than a week of advance notice**, and **5 percent** offered more than a **month**. That is time available for inventory queries, patch scheduling, and compensating controls, applied before the disclosure noise starts rather than during it.

Polling cadence is the second constraint, and it binds too

There are **two separate delays** between a vulnerability existing and a security team seeing it, and they have different remedies.

The first is the one this report has been describing. No advisory exists, so there is nothing for any tool to match against. For 36 percent of the findings in this dataset that delay is still running, because no CVE record has ever been created. No amount of polling frequency reaches data that has not been published.

The second delay applies **after publication, and it is architectural**. Most vulnerability tooling is built on a **polling model**. **Advisory feeds** are pulled on a schedule. **Workloads** are assessed on a schedule. **Agentless** cloud security platforms that copy customer workloads and analyse them out of band **run on a scan cycle**, and where that cycle is measured in hours, the cycle length sets the floor on detection time no matter how current the underlying feed is. The advisory can be perfectly fresh and still sit unread until the next pass.

The arithmetic does not favour the defender. **Scanning** for newly disclosed **vulnerabilities** has been observed beginning **within minutes of publication**, and mean time to exploitation has **compressed from years to days**. Against that, a scan cycle measured in hours does not lose narrowly. Halving the cycle does not change the conclusion either, because the real comparison is not between one cycle length and a shorter one. It is between running on a cycle at all and an adversary who is not on one.

This is why **VulNow delivers intelligence as an event rather than as a feed to be polled**. A notification that fires when the signal appears, rather than when the next cycle comes round, is the only structure that puts a defender on the same clock as the person they are defending against. Combined with a lead time measured in days rather than minutes, it changes what is achievable: not a faster response after disclosure, but a completed one before it.

> *The useful question is no longer how quickly the team responds after a CVE is published. It is how much of the relevant risk ever gets a CVE at all, how long the rest waits, and whether anything in the stack is listening when it finally lands.*

A Note on Our CNA Appointment

In July 2026, VulNow was appointed a CVE Numbering Authority (CNA) under the ENISA CVE Root, with a scope covering our own products and third party open source vulnerabilities not already covered by another CNA.

We raise it here for what it addresses rather than for what it proves. The vulnerabilities in this report share a structural problem: they occur in projects whose maintainers are not CNAs themselves, so the path from a quietly shipped fix to a public record runs through the queues described earlier in this report, which are now several weeks deep and lengthening. Somebody has to assign an identifier for that material, and the set of organizations authorized to do so for a package they do not own is small.

That is the gap the appointment speaks to. Over time it allows findings in our queue to become advisories that anyone can act on, including organizations that will never be VulNow customers, and it puts the assignment step inside the same process that already verifies affected versions and reproduces the flaw.

The work is ahead of us rather than behind us. We would rather describe it once there are published records to point at than claim credit for the authority alone.

Method, Limitations, and Sources

How to read these numbers

- The confirmed count is cumulative to date, not a count of findings made during Q2 2026. It includes findings first detected in 2025 whose advisories appeared later, which is why the axios finding from the Q1 2026 report appears again here.
- Findings are counted, not advisories. A single published advisory can cover several distinct vulnerable behaviours that VulNow identified and tracked separately, so the number of distinct GHSA and CVE identifiers in this set is smaller than the number of confirmed findings.
- The Q1 2026 report described its 58 findings as confirmed by a published CVE. This report uses the broader and more accurate criterion described in section two, confirmation by any public security advisory. The two figures are therefore not directly comparable.
- Lead time is measured to the first public advisory of any kind. Where a GHSA preceded a CVE, the GHSA sets the clock.
- Every count and percentage in this report reflects a point-in-time snapshot, for the reasons set out in the note at the start of section four. Figures quoted elsewhere by VulNow may differ, and the live dataset is authoritative.
- The figure of roughly 8,500 open findings and the rate of more than 100 per day are platform totals covering findings at all stages of triage, including changes that will prove to be ordinary bug fixes rather than security fixes. They are not a count of verified vulnerabilities, they are not individually published, and no true positive rate is claimed for them here. Only the confirmed subset is public.

External sources

- GitHub, "Inside the Advisory Database and what happens when vulnerability volume breaks records," 29 June 2026. https://github.blog/security/supply-chain-security/inside-the-advisory-database-and-what-happens-when-vulnerability-volume-breaks-records/
- NIST, "NIST Updates NVD Operations to Address Record CVE Growth," April 2026. https://www.nist.gov/news-events/news/2026/04/nist-updates-nvd-operations-address-record-cve-growth
- Daniel Stenberg, "The end of the curl bug-bounty," 26 January 2026. https://daniel.haxx.se/blog/2026/01/26/the-end-of-the-curl-bug-bounty/
- HackerOne, Internet Bug Bounty program policy, submissions paused 27 March 2026. https://hackerone.com/ibb
- InfoWorld, "Internet Bug Bounty program hits pause on payouts," 3 April 2026. https://www.infoworld.com/article/4154210/internet-bug-bounty-program-hits-pause-on-payouts.html
- Help Net Security, "AI is drowning software maintainers in junk security reports," 18 May 2026. https://www.helpnetsecurity.com/2026/05/18/problems-with-ai-assisted-vulnerability-research/
- Palo Alto Networks Unit 42, Cortex Xpanse Attack Surface Threat Report, on attacker scanning beginning within minutes of CVE publication. https://unit42.paloaltonetworks.com/
- Google Cloud Threat Intelligence and Mandiant, "How Low Can You Go? An Analysis of 2023 Time-to-Exploit Trends," 2024, on the compression of mean time to exploitation. https://cloud.google.com/blog/topics/threat-intelligence/time-to-exploit-trends-2023
- ENISA, coordinated vulnerability disclosure and the EU Cyber Resilience Act, Regulation (EU) 2024/2847. https://www.enisa.europa.eu/topics/vulnerability-disclosure
- VulNow confirmed PreCVE dataset, public beta. https://precve.vulnow.com

Predictive Vulnerability Intelligence