

THE CVE TRAP

The industry has become very good at finding vulnerabilities. That is not the same as being in control of what your software does.

praedictio.ai • October 2026

Vulnerability management is now the lens through which software risk is seen. Scanners, scores, severity ratings, remediation SLAs, board-level metrics: all of it rests on the same unit of account, the published flaw. The work is necessary. The trouble is what happens when a useful instrument becomes the only instrument, and the map quietly replaces the territory.

The queue has no end

CVE submissions rose 263 percent between 2020 and 2025, and 2026 is steeper still. NIST reported first-quarter submissions nearly a third above the year before. By the end of June the gap was almost 50 percent, and by 30 September the year stood at 72,812 published CVEs, about 267 a day and 106 percent ahead of the same point in 2025. It passed the whole of 2025, a record 48,185, in mid-August. September alone brought 14,943, more than three times the 4,321 of September 2025.

72,812 CVEs published in 2026 to 30 September	+106% year to date against the same point in 2025	0.24% of first-half 2026 CVEs in CISA's exploited catalog by July
--	--	--

NIST enriched close to 42,000 records in 2025, 45 percent more than in any prior year, and still could not keep up. The National Vulnerability Database has carried a backlog since early 2024. A Commerce Department inspector general audit reported in May 2026 put it at more than 27,000 unprocessed records by the end of 2025, up from about 13,000 in June 2024. In April 2026 NIST moved everything published before March that year into a category called Not Scheduled: real vulnerabilities, formally acknowledged, with no date for analysis.

Some of the 2026 surge reflects who is publishing rather than what is breaking, since the five busiest assigners supplied nearly half of September's records. That is a fair caution about the count. It does not change the problem, which is that the queue grows faster than anyone can work it.

That is worth sitting with. The reference system the whole industry prioritizes against has publicly conceded it cannot process its own inbox.

Most of the queue never mattered

The Cyentia Institute examined 237,687 published CVEs against 8.6 million observations of real exploitation. Six percent had ever been seen exploited in the wild.

The 2026 cohort points the same way. Of the 35,364 CVEs published in the first half of the year, only 85 had reached CISA's Known Exploited Vulnerabilities catalog by July, a floor that rises as the cohort ages. CISA added 245 entries in 2025 and about 180 by mid-August 2026, while VulnCheck counted 883 newly exploited CVEs in 2025. Confirmation of exploitation lags exploitation itself.

Read that alongside the backlog and an uncomfortable picture forms. Enormous effort goes into enumerating, scoring and triaging a population of findings, roughly nineteen in twenty of which will never be used against anyone. Meanwhile the handful that matter are often exploited before the queue reaches them. Mandiant put the average time-to-exploit for vulnerabilities disclosed in 2024 at minus one day, meaning attacks on average preceded the patch.

You cannot fix this with better prioritization. Prioritization is already the response to it, and the gap kept widening.

What a vulnerability actually is

Here is the part the vocabulary hides. A vulnerability is not the harm. It is an unintended capability: a way to make software do something it was never built to do. The damage is done by the doing, not by the flaw's existence in a database.

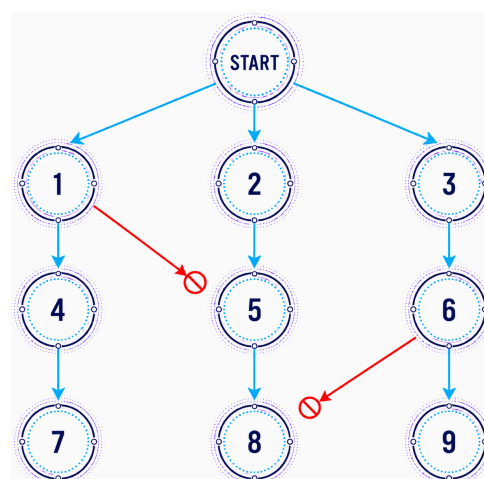
Log4Shell was not harmful because a logging library parsed a string. It was harmful because parsing that string let a remote party execute code. The CVE names the door. The loss comes from what walked through it.

This distinction sounds academic until you notice what follows from it. Every vulnerability, known or unknown, has to express itself as behavior your software would not otherwise have performed: a process it would not spawn, a socket it would not open, a file it would not read. Flaws are infinite and unnamed. Their expressions are neither.

The inversion

Model what the software is supposed to do, then allow only that.

An application built from a known set of components, exercised by a test suite you already run, has a definable envelope of behavior. Define it, enforce it at runtime, and an exploit is out of bounds whether or not anyone has published its CVE, scored it, or enriched the record. The question stops



The set you can define. Paths observed during testing are permitted. A path that was never observed does not run, whether or not anyone has named the flaw that produced it.

being “do we know about this flaw” and becomes “is this action one our software is supposed to perform”.

That question is answerable. The other one never will be.

THE HONEST LIMIT

None of this retires vulnerability management. You still patch, because an unpatched flaw remains a liability, in contract terms and in regulation. What changes is the load it carries. Patching stops being the control that decides whether you are compromised, and becomes hygiene, done on a sane timetable rather than a panicked one.

What it means in practice

Program Behavior Intelligence™ (PBI™) works from the finite side. **PBI Explainer™** builds a deterministic model of how an application behaves, from the tests you already have. **PBI Enforcement™** applies that model at runtime, so behavior outside it does not execute, whatever produced it.

A model built from a 2022 release of Apache ActiveMQ prevented a vulnerability disclosed in 2023, and later a separate vulnerability nobody discovered until 2026, with no signature, rule or update in between. The model existed before the threat did, which is the only sense in which a defense can be said to be ahead.

The problem was never the vulnerability. It was what the vulnerability was allowed to do.

UNDERSTAND EXACTLY WHAT YOUR
SOFTWARE DOES. ENFORCE THAT AT
RUNTIME.

SEE IT ON YOUR SOFTWARE

Sources. NIST, *NIST Updates NVD Operations to Address Record CVE Growth*, April 2026. Cyentia Institute, *A Visual Exploration of Exploitation in the Wild*, July 2024. Mandiant *time-to-exploit analysis*, 2024. J. Gamblin, *2025 CVE Data Review*, January 2026; *CVE Mid-Year 2026 Check-In*, July 2026; *monthlyCVEStats*, September 2026 release (data through 30 September 2026, rejected CVEs excluded). U.S. Department of Commerce Office of Inspector General audit of the NVD, as reported by CyberScoop and Ciphers Security, May 2026. Cyble analysis of the CISA Known Exploited Vulnerabilities catalog, 2025. Axis Intelligence, *CISA KEV Statistics 2026 (additions to 11 August 2026)*. Zafran, *The 2025 Spike in Vulnerabilities Isn't the Full Story (VulnCheck KEV data)*. Counts vary slightly between trackers because of how rejected records are treated.