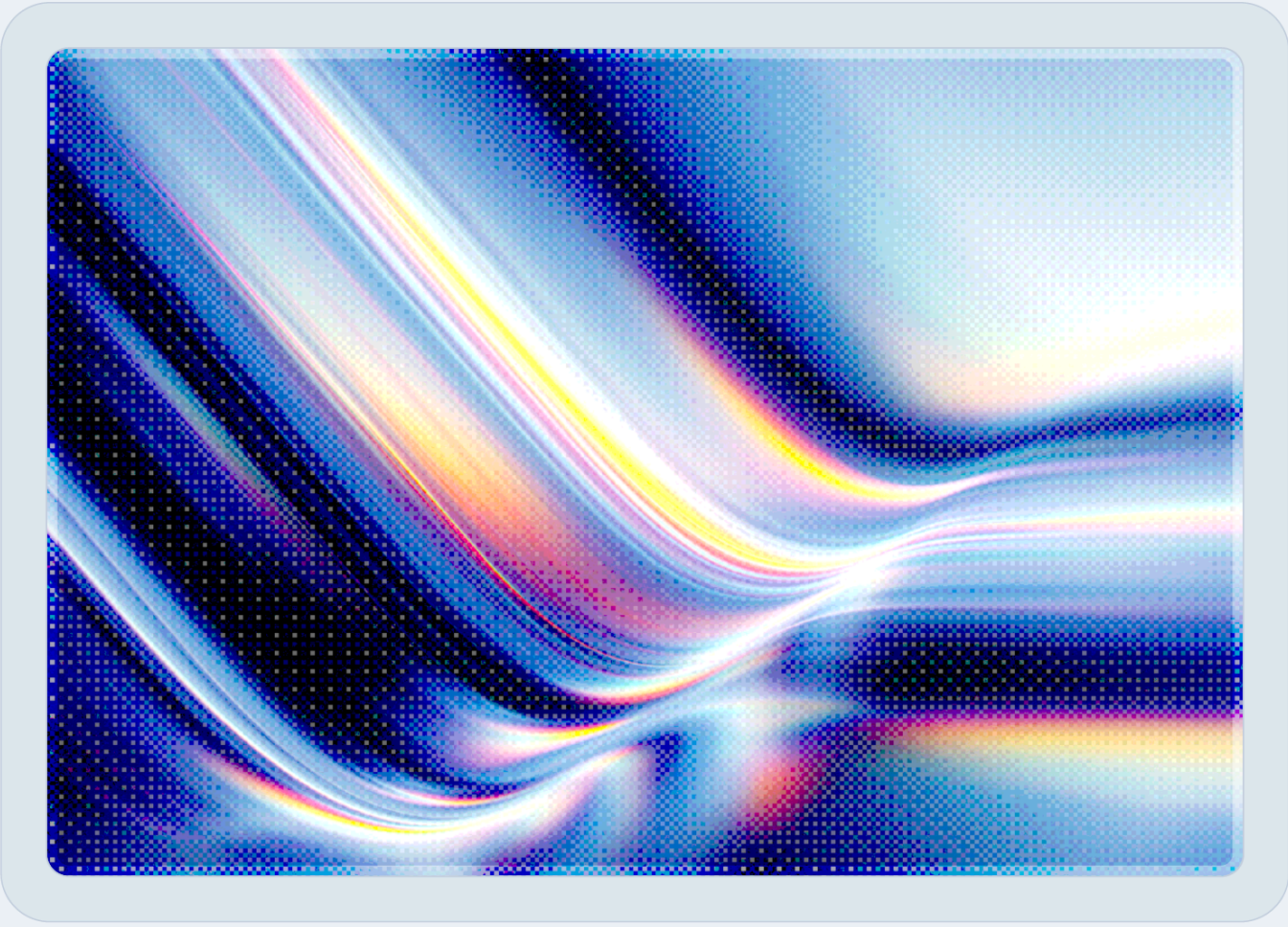
CSAI Foundation | Cloud Security Alliance

The Disclosure Bottleneck: When AI Discovery Outpaces Patch Consumption

Systemic Risk in the Gap Between Machine-Speed Findings and Human-Speed Remediation

2026-11-06

AI-assisted Rapid Research



CSAI cloud security alliance®

© 2026 Cloud Security Alliance. Some rights reserved.

You may download, store, display, view, print, redistribute, and link to this document in its original, unmodified form, provided that attribution to the Cloud Security Alliance is maintained and all trademark and copyright notices remain intact.

This document may not be modified or altered. You may quote portions of the document as permitted by the Fair Use provisions of the United States Copyright Act, provided that attribution is given to the Cloud Security Alliance.

This document may be shared on professional and social media platforms in its original form with attribution.

This document was generated with AI assistance and has not undergone official CSA review and approval processes.

Table of Contents

Executive Summary 5

Introduction and Background 6

- How Coordinated Disclosure Was Designed to Work
- The 2026 Inflection
- Scope and Approach

The Anatomy of the Bottleneck 8

- Discovery Is No Longer the Rate-Limiting Step
- Validation and Triage: The Signal-to-Noise Problem
- Fix Development: The Maintainer Ceiling
- Publication and Enrichment: The Metadata Gap
- Downstream Consumption: The Long Tail

Why the Bottleneck Is a Systemic Risk 13

- Backlog as Latent Exposure
- Collapsing Exploitation Timelines
- Correlated Failure and Monoculture
- Concentration Risk in the New Coordination Layer
- Regulatory Clocks That Assume a Fix Exists
- Summary of Systemic Risk Mechanisms

Measuring Patch Consumption Capacity 18

- From Patch Velocity to Absorption Capacity
- Indicators Worth Tracking
- The Role of Compensating Controls

Responses Across the Ecosystem 19

- What AI Developers Running Discovery Programs Can Do
- What Maintainers and Open-Source Foundations Can Do
- What Enterprises Can Do
- What Policymakers Can Do

CSA Resource Alignment 21

Conclusions and Recommendations 22

- Conclusions
- Recommendations

© 2026 Cloud Security Alliance. AI-assisted Rapid Research.

References 25

Executive Summary

For most of the history of software security, the scarce resource was discovery. Finding a serious, exploitable flaw in mature code required skilled researchers, specialized tooling, and time, and the disclosure system that grew up around that scarcity was built on an implicit assumption that vulnerabilities arrive at a pace humans can absorb. That assumption no longer holds. Over the course of 2026, frontier AI models moved vulnerability discovery from a scarce, labor-intensive activity to a high-volume, automated one, and the bottleneck in software security moved downstream. Anthropic, whose Project Glasswing deployed its Claude Mythos models against critical software beginning in April 2026, stated the shift directly in its May update: progress on software security, it wrote, is no longer limited by how quickly vulnerabilities can be found but by how quickly they can be verified, disclosed, and patched [2].

The numbers published since then illustrate the scale of the imbalance. By early October 2026, Anthropic reported that Glasswing partners had found at least 129,000 verified vulnerabilities between April and July, and that more than 33,000 findings across those partner reports and its own scanning were rated critical or high; it characterized even these figures as a likely undercount [4]. Its open-source disclosure dashboard, updated October 2, recorded 6,157 vulnerabilities disclosed to maintainers across 591 projects, of which 516 were known to be patched upstream [3]. An independent review by VulnCheck argued that the published ledgers do not fully reconcile and that, as of September, fewer than one in ten findings in Anthropic's published ledger had reached a maintainer at all [6]. Whatever the precise figures, the available accounts point in the same direction: by Anthropic's own October dashboard, fewer than one in ten disclosed open-source vulnerabilities was known to be patched, which suggests that discovery is outrunning remediation by a wide margin.

This paper argues that the gap is a systemic risk rather than a collection of individual organizational problems. The backlog of known-but-unpatched vulnerabilities is not inert. It sits in disclosure queues, embargoed clearinghouses, and maintainer inboxes, and because increasingly capable models, including widely available and open-weight models, are being pointed at the same widely used code, the probability that an adversary independently rediscovers a queued flaw plausibly rises the longer it remains unfixed. Meanwhile, the institutions that sit between a fix and a deployed system are under simultaneous strain. The Linux kernel's maintainers have publicly described their security channel as unmanageable [8], and other maintainers have stopped accepting submissions entirely [10]; bug bounty programs have retrenched, with Google pausing product vulnerability submissions to its open-source reward program [9]; and the U.S. National Vulnerability Database ended universal enrichment of CVE records in April 2026 [14]. Each stage of the pipeline has its own human-rate constraint, and the constraints compound.

Regulatory regimes are tightening at the same moment. CISA's Binding Operational Directive 26-04 now requires federal agencies to remediate the highest-risk vulnerabilities within three days [16], and the EU Cyber Resilience Act's 24-hour reporting obligations for actively exploited vulnerabilities took effect on September 11, 2026 [17][29]. These clocks presume that a patch exists and that the information needed to act on it is available. In the disclosure bottleneck, neither presumption is safe.

The paper's central recommendations follow from this analysis. Enterprises should measure and manage their *patch consumption capacity*—the rate at which they can absorb, test, and deploy fixes—as a first-class resilience metric, and should invest in architectural controls that reduce exposure independent of any single patch arriving in time. Maintainers and the foundations that support them need funded triage capacity and shared intake infrastructure such as the Linux Foundation's Akrites initiative [11]. AI developers that run discovery programs should pace disclosure to downstream absorption capacity, ship candidate patches with findings, and publish reconcilable data. Policymakers should treat disclosure and remediation capacity as critical infrastructure, fund it accordingly, and design reporting regimes that account for the period in which a vulnerability is known but not yet fixable.

Introduction and Background

How Coordinated Disclosure Was Designed to Work

Coordinated vulnerability disclosure (CVD) is the set of norms and institutions that govern what happens between the moment someone finds a flaw and the moment users are protected from it. In its conventional form, a researcher privately reports a vulnerability to the responsible vendor or maintainer, the two parties agree on an embargo period during which a fix is developed, and the vulnerability is published—typically with a CVE identifier, an advisory, and a severity score—when the fix ships. Downstream, vulnerability databases enrich the record with structured metadata, distributions and vendors rebuild and repackage affected components, and enterprises consume the resulting updates through their patch management programs. The 90-day deadline popularized by large vendor research teams such as Google Project Zero [30] became a rough industry default, not because ninety days was optimal for every flaw but, in effect, because it balanced maintainer workload against the risk of rediscovery at historical discovery rates.

Every part of this system was calibrated, explicitly or not, to a world in which serious vulnerabilities in mature software were relatively rare and expensive to find. Most maintainer security teams, where they existed at all, were sized to handle a handful of credible reports per month. Bug bounty programs priced rewards on the assumption that a valid critical finding represented significant researcher effort. Public enrichment infrastructure such as the NVD was resourced for far lower annual CVE volumes than it now receives, and its

growth assumed a gradual rather than a step-change increase [15]. Enterprise patch cycles—monthly maintenance windows, quarterly upgrade cycles for operational technology, annual refreshes for embedded systems—assumed that the stream of critical fixes would be steady and roughly predictable.

The pressure on this system did not begin in 2026. CVE volumes had grown steadily for years, and NIST attributed its April 2026 decision to change NVD operations in part to a 263 percent increase in CVE submissions between 2020 and 2025 [15]. Open-source maintainer burnout was a recognized problem well before AI-assisted research became common. What changed in 2026 was not the direction of the trend but its slope, and with it the location of the binding constraint.

The 2026 Inflection

On April 7, 2026, Anthropic announced Project Glasswing, a restricted-access program that gave a group of critical-infrastructure and technology partners access to its Claude Mythos Preview model for defensive vulnerability research, alongside up to $100 million in usage credits and donations to open-source security organizations [1]. The initial results, published on May 22, described a qualitatively different discovery rate. Partners had found more than ten thousand high- or critical-severity vulnerabilities; in Anthropic's own scanning of open-source projects, the model identified an estimated 6,202 high- or critical-severity issues, and of the 1,752 findings that had been independently assessed at that point, 90.6 percent were confirmed as valid [2]. Anthropic had disclosed roughly 530 high- or critical-severity bugs to maintainers, 75 had been patched, and a further 827 confirmed vulnerabilities were waiting to be disclosed [2]. Its public disclosure dashboard, which counts findings of all severities, recorded 97 patched of 1,596 disclosed across 281 projects in its May 22 snapshot [31]. The two sets of figures differ because they measure different populations, not because either is in error.

The program expanded through the summer, and on October 6 Anthropic folded Glasswing into a broader Cyber Verification Program with tiered access for defensive, red-team, and safety-critical users [4]. Its accompanying data was striking. Based on reports from 33 partners, Glasswing participants had uncovered at least 129,000 verified vulnerabilities between April and July, with more than 33,000 findings rated critical or high across those partner reports and Anthropic's own scanning, and Anthropic stated that it expected the true figure to be at least five times higher [4]. It also noted that fewer than half of partners had reported patch numbers, in part because fixes were still in progress [4].

Glasswing is the most visible example, but it is not the only source of machine-speed discovery. Other frontier developers, security vendors, and independent researchers are applying increasingly capable models to the same problem, and the pressure on maintainers predates Glasswing's launch. In January 2026, the curl project ended its paid bug bounty program, citing low-quality AI-assisted submissions [24]. By late October, an OpenAI executive was publicly warning that AI-generated vulnerability reports had grown heavy enough that some maintainers had stopped accepting submissions entirely [10]. The inflection, in other words, is an ecosystem-wide phenomenon in which Glasswing serves as the best-documented case study.

Scope and Approach

This paper focuses on the period between discovery and deployment—what we term the *disclosure bottleneck*—and on the systemic risks that arise when discovery capacity exceeds the capacity of every downstream stage to absorb it. It draws on public data from AI developers, vulnerability intelligence firms, open-source foundations, government agencies, and prior CSA research. Where published figures conflict or have been disputed, the paper notes the disagreement rather than resolving it, and it distinguishes reported data from the inferences drawn from that data. Readers should note that several of the most important data sources in this area are self-reported by the organizations running discovery programs, and that independent verification remains limited.

The Anatomy of the Bottleneck

Discovery Is No Longer the Rate-Limiting Step

To understand why the disclosure bottleneck is systemic, it helps to model vulnerability handling as a pipeline in which each stage has its own throughput. A finding must be generated, validated as a true positive, triaged for severity, reported to the right party, reproduced and understood by a maintainer, fixed with a patch that does not break dependents, released with an advisory and identifiers, enriched with metadata that downstream tools can consume, packaged by distributions and vendors, and finally deployed by the organizations that run the affected software. A pipeline's throughput is set by its slowest stage. For decades, discovery limited the *volume* of serious flaws entering the pipeline, even though deployment, as the Log4Shell data discussed below shows, has long governed how *long* each fixed flaw remains exploitable in practice. What has changed is that discovery no longer limits volume, so every downstream stage now faces inflow it was not sized for.

AI has collapsed the cost of that first stage without proportionately changing any of the others. Software Improvement Group, summarizing Anthropic's testing, reports individual vulnerabilities being found for under $50 in compute [5], and the volume figures cited above show what that cost reduction means at scale. Every subsequent stage still depends largely on human judgment: a maintainer must decide whether a reported memory-safety issue is reachable, design a fix that respects the project's compatibility commitments, and coordinate a release. Anthropic itself has described independent human triage as the rate-limiting step in its disclosure process [3], and its May update reported that several maintainers described themselves as "severely capacity constrained," with some asking Anthropic to slow its disclosure rate because they needed more time to design patches [2].

The table below summarizes the principal stages of the pipeline, the constraint that governs each, and representative 2026 evidence of strain.

Pipeline Stage	Primary Rate Limiter	Representative 2026 Evidence
Discovery	Compute and model access	Glasswing partners reported at least 129,000 verified vulnerabilities, April–July 2026 [4]
Validation and triage	Skilled human reviewers	Anthropic describes independent human triage as its rate-limiting step [3]; VulnCheck estimates fewer than 10% of findings had reached maintainers by September [6]
Intake and coordination	Maintainer and vendor security team capacity	Linux kernel security list described as "almost entirely unmanageable" [8]; Google OSS VRP paused product vulnerability intake on October 1 [9]
Fix development	Maintainer engineering time and domain knowledge	516 of 6,157 disclosed open-source vulnerabilities known to be patched as of October 2 [3]
Publication and enrichment	Identifier assignment and database analysts	NVD moved to prioritized enrichment on April 15, 2026, and placed pre-March backlog in "Not Scheduled" status [14]
Downstream packaging	Distribution and vendor release cycles	Anthropic notes that an upstream patch does not guarantee wide installation [3]
Enterprise deployment	Change management, testing, and asset visibility	Roughly 13% of 2025 Log4j downloads still contained Log4Shell, nearly four years after the fix [19]

Two observations follow from this structure. First, accelerating raw discovery volume further yields diminishing security returns once discovery exceeds downstream throughput; beyond that point, additional findings mainly lengthen the queue unless they also improve prioritization or reduce downstream work. Second, the queue does not form at a single point. It forms at multiple stages simultaneously, because each stage is staffed for historical volumes and because inflow arrives through many parallel channels—independent researchers, multiple discovery programs, and thousands of upstream projects—rather than through a single serial pipeline.

Validation and Triage: The Signal-to-Noise Problem

The first downstream stage to fail visibly was intake. Throughout 2025 and early 2026, maintainers complained primarily about low-quality AI-generated reports—plausible-sounding submissions that described vulnerabilities that did not exist or could not be reproduced. That problem has not disappeared. Google cited "a significant rise in automated submissions, the vast majority of which are not valid" when it paused product vulnerability submissions to its Open Source Software Vulnerability Reward Program on October 1, 2026 [9]. CSA's October analysis of vulnerability intake documented a sequence of similar retrenchments, including the end of curl's paid bounty in January and a pause in Internet Bug Bounty submissions in April [24].

What has changed over 2026 is that the noise problem now coexists with a signal problem. In late October 2026, OpenAI's cyber executive in residence said that AI-generated vulnerability reports had grown heavy enough that some maintainers had stopped accepting submissions, while acknowledging that in most cases the reports likely describe "real bugs of some degree" and questioning whether those bugs are worth maintainers' time [10]. Maintainers, in other words, now face a rising volume of findings that are real but of uncertain importance, layered on top of the reports that are simply wrong. The two problems interact in a damaging way. When a project closes or restricts its intake channel to stem the flood of low-quality reports, it also reduces its capacity to receive valid ones. When it keeps the channel open, every valid report competes for attention with many invalid ones. In both cases, the effective time from discovery to a maintainer's informed decision increases.

Duplication compounds the problem. Because many researchers now use similar models against the same high-value codebases, the same underlying flaw is often reported many times by different parties. In announcing the Linux 7.1-rc4 release on May 17, 2026, Linus Torvalds described the kernel's private security list as "almost entirely unmanageable" and cited "enormous duplication due to different people finding the same things with the same tools" [8]. Because submitters cannot see one another's private reports, the overlap is invisible to them and is absorbed entirely by the maintainers. Torvalds also argued that bugs found by widely available AI tools are "pretty much by definition not secret" [8]—an observation with significant implications for the embargo model discussed later in this paper.

Severity assessment introduces a further distortion. VulnCheck's September analysis of Anthropic's published disclosure ledger found that Claude rated 91.5 percent of findings as critical or high severity, while maintainers rated only 51.3 percent that way [6]. The ledger does not expose the scoring details behind either rating, so the cause of the gap cannot be determined from public data, and it may reflect differences in context about reachability and deployment rather than model error. Regardless of cause, a systematic gap of that size matters operationally. If discovery tools inflate severity relative to maintainer judgment, downstream prioritization systems that consume those ratings will misallocate scarce remediation capacity.

Fix Development: The Maintainer Ceiling

Even when a valid report reaches a maintainer promptly, writing the fix is the step least amenable to acceleration. A correct patch requires understanding not only the vulnerable code path but the project's API commitments, the downstream consumers that depend on specific behavior, and the regression risk of the change. For widely used libraries, a poorly designed fix can break thousands of dependents, which is why experienced maintainers are cautious about rushing security releases. Anthropic reported in May that a high- or critical-severity bug found by Mythos Preview took on average about two weeks to patch [2], a short interval for an individual issue. The problem is not the latency of any single fix but the aggregate: a maintainer team that can produce a handful of careful security fixes per month cannot keep pace with an inflow measured in dozens or hundreds.

The published Glasswing figures show this ceiling in practice. The share of disclosed open-source vulnerabilities known to be patched rose from 97 of 1,596 (about 6 percent) to 516 of 6,157 (about 8 percent) between Anthropic's May and October dashboard snapshots, even as the absolute number of disclosures roughly quadrupled [3][31]. Because the ratio is cumulative and disclosures accelerated sharply, it understates the patch rate for older disclosures, and the two-week average Anthropic reported likely reflects only the subset of bugs that had been patched. Even allowing for these effects, absolute patch counts grew far more slowly than disclosures. Anthropic notes that its patch count reflects only fixes it knows about and that some fixes ship without advisories, so the true rate may be somewhat higher [3]. VulnCheck's independent analysis, using the published ledger rather than the dashboard summary, counted 202 fixed findings across 113 projects as of early September and argued that the dashboard, ledger, and identifier lists did not reconcile with one another [6]. Bruce Schneier, commenting on the May update, called the low patch counts "weird" and pointed to the lack of released detail as a problem for independent assessment [7].

These disagreements about precise figures are themselves informative. They indicate that the organizations generating findings at machine speed do not yet have reliable, auditable accounting of where those findings sit in the remediation pipeline. If the discovering party cannot reconcile its own ledger, downstream consumers—enterprises, regulators, and other researchers—cannot reliably determine which vulnerabilities in their dependencies are known, which are fixed, and which remain in a queue.

Publication and Enrichment: The Metadata Gap

A fix that ships without usable metadata is only partially consumable. Enterprise vulnerability management depends on structured data—affected product and version ranges, severity scores, weakness classifications—to match vulnerabilities to assets and prioritize work. For two decades, the NVD has been the default source of that enrichment for much of the industry.

On April 15, 2026, NIST changed NVD operations in response to record CVE growth. All submitted CVEs continue to be added to the database, but enrichment is now prioritized for vulnerabilities in CISA's Known Exploited Vulnerabilities catalog, software used by the federal government, and software designated as critical under Executive Order 14028; backlogged CVEs published before March 1, 2026, were moved to a "Not Scheduled" category [14]. NIST also stated that modified CVEs would be reanalyzed only when changes materially affect enrichment data [14]. CSA's analyses of the change noted that organizations relying on NVD-provided CVSS scores or product mappings would face growing blind spots unless they integrated alternative data sources [21][22].

The timing matters. The NVD change and the surge in AI-discovered vulnerabilities arrived within weeks of each other, which means that the volume of vulnerabilities needing enrichment increased just as the default public enrichment service narrowed its scope. The predictable result is a widening population of vulnerabilities that are published, and in some cases patched, but that do not surface in the tools enterprises use to decide what to patch. A fix that an enterprise does not know applies to its environment is, for practical purposes, a fix that has not been consumed.

Downstream Consumption: The Long Tail

The final stages of the pipeline have historically produced the longest delays for individual vulnerabilities, and they are where AI-accelerated discovery is least likely to help in the near term. An upstream fix in an open-source library must be picked up by the projects that depend on it, repackaged by Linux distributions and commercial vendors, incorporated into appliance firmware and container images, and then deployed by organizations that may be several steps removed from the original project and may not know they depend on it. Anthropic's dashboard explicitly cautions that "patched" means maintainers have released a fix and does not guarantee that the fix has been widely installed [3].

Historical evidence suggests how long that tail can be. Sonatype reported in December 2025 that roughly 13 percent of the nearly 300 million downloads of Log4j from Maven Central during 2025—about 40 million downloads—still contained the Log4Shell vulnerability, almost four years after fixed versions became available [19]. The same analysis found that around 95 percent of vulnerable components downloaded had a safer version available at the time [19]. Log4Shell was one of the most heavily publicized vulnerabilities in history. The flaws now emerging from AI discovery programs are, for the most part, far less visible, and there are vastly more of them, which suggests their deployment tail is likely to be at least as long.

The enterprise side of the pipeline is also constrained by asset visibility, testing capacity, change-management processes, and operational risk tolerance. Critical infrastructure operators frequently maintain quarterly or annual update cycles for operational technology, and embedded and medical devices may not be patchable in the field at all. CSA's May 2026 analysis of the enterprise remediation model concluded that

downstream finding volume would rise across all enterprise environments regardless of whether those organizations use AI security tools themselves [22]. In the terms of this paper, every organization's patch consumption capacity is now being tested by inflow generated elsewhere.

Why the Bottleneck Is a Systemic Risk

Backlog as Latent Exposure

The conventional view of an unfixed, privately disclosed vulnerability is that it poses limited risk because only the reporter and the maintainer know about it. That view rested on two assumptions: that discovery is expensive enough that independent rediscovery within the embargo window is unlikely, and that the embargo window is short enough to bound exposure. AI-accelerated discovery weakens both.

When the same capable models are applied to the same widely deployed code, the probability of independent rediscovery is no longer a function of how hard a flaw is to find in absolute terms but of how many parties are searching with comparable tools. Torvalds's observation that AI-found bugs are effectively not secret [8] and the duplicate-report patterns he described [8] are direct evidence of that convergence among defenders. Public evidence on adversary use of frontier models for vulnerability discovery remains limited, but it would be imprudent to assume that adversaries with access to comparable capability—whether through open-weight models, stolen access, or their own development—are not converging on the same findings. Access restrictions on the most capable discovery models, such as those applied to Mythos Preview through Glasswing and the tiered Cyber Verification Program [1][4], may slow this convergence, but they do not eliminate it: the duplication Torvalds describes involves widely available tools, and restrictions on frontier capability have historically eroded over time. VulnCheck made a related point in its September analysis, warning that delays in moving findings to maintainers increase the risk of duplicate discovery [6].

Under these conditions, the backlog of known-but-unpatched vulnerabilities should be understood as latent exposure whose risk increases with time. Every finding in a disclosure queue, a clearinghouse embargo, or an unanswered maintainer inbox is a flaw that at least one party has demonstrated can be found with existing tools, many of which are or will soon be available to others. The longer the queue, the larger the inventory of flaws that adversaries may independently find before defenders can act. This is the core mechanism by which a capacity problem becomes a security problem.

Collapsing Exploitation Timelines

The risk of latent exposure is amplified by the speed at which disclosed or rediscovered flaws are now exploited. Mandiant's M-Trends 2026 report, based on its 2025 incident response engagements, estimated the mean time to exploit vulnerabilities at negative seven days—meaning exploitation, on average, began

before a patch was available—and found that exploits were the leading initial infection vector, accounting for 32 percent of intrusions [18]. Other data points to windows measured in hours rather than days; CSA's June 2026 analysis cited figures indicating that 28.3 percent of CVEs were exploited within 24 hours [20], although methodologies and definitions vary considerably and individual figures should be treated as indicative rather than precise.

When exploitation routinely precedes patch availability, the conventional disclosure model's core promise—that users will have a fix in hand when a vulnerability becomes public—is already broken for a meaningful share of vulnerabilities. Adding a large, slowly draining queue of AI-discovered flaws to that environment increases the population of vulnerabilities for which the race between attacker and defender begins with the defender already behind.

Correlated Failure and Monoculture

Systemic risk differs from aggregate risk in that it involves correlation: many parties failing at once for the same reason. The disclosure bottleneck creates correlation in at least three ways.

First, the targets are concentrated. Discovery programs naturally focus on the most widely used and most critical software, because that is where findings have the greatest defensive value. The consequence is that the backlog is disproportionately composed of flaws in components that are present in a very large number of systems. A single unpatched library vulnerability, if exploited, can affect organizations across sectors and geographies simultaneously.

Second, the discovery tools are concentrated. A small number of frontier models are widely regarded as the most capable at vulnerability research, and defenders and, plausibly, attackers draw on the same handful of capabilities. This produces correlated discovery—different parties finding the same flaws at roughly the same time—which reduces the protective value of secrecy and embargo.

Third, the remediation infrastructure is concentrated. A small number of overstretched maintainers are responsible for the security of components on which large portions of the digital economy depend, and a small number of organizations now operate the coordination layers through which pre-disclosure findings flow. That last point warrants separate treatment.

Concentration Risk in the New Coordination Layer

The ecosystem has responded to the bottleneck by building new intermediaries. In June 2026, Chainguard launched Athena, a coalition that pools AI-discovered findings and remediation work from multiple organizations, including financial institutions and infrastructure providers, to produce patches for open-source projects [23]; Chainguard had separately committed engineering resources to open-source remediation earlier that month [13]. IBM and Red Hat announced Project Lightwell, a $5 billion commitment

to build an AI-driven security clearinghouse for open-source software [12], and Red Hat later reported having remediated more than 400 previously unknown open-source vulnerabilities through the project [28]. The Linux Foundation launched Akrites on June 25, 2026, as a shared security incident response team intended to give maintainers a single, trusted place to coordinate, remediate, and disclose, replacing duplicate reports with shared handling and coordinating fixes before public disclosure, with seed funding from the Alpha-Omega project [11].

These initiatives address real problems, and the shared-intake model in particular responds directly to the duplication and triage burden described above. They also introduce new forms of concentration. CSA's July 2026 analysis of the clearinghouse model observed that a pooled repository of embargoed, pre-disclosure vulnerabilities in widely used software is an extraordinarily valuable target, and that dependency on a small number of commercial clearinghouses raises questions about access equity, governance, and the treatment of organizations outside the membership [23]. A clearinghouse that operates on a commercial membership basis may, by design, deliver fixes or advance notice to members before non-members—a reasonable business model that nevertheless creates tiers of exposure across the ecosystem.

The following table compares the principal coordination models now operating, based on public descriptions.

Model	Example	Primary Function	Principal Benefit	Principal Systemic Concern
Discoverer-led disclosure	Anthropic Glasswing / Cyber Verification Program [3] [4]	Discovering party validates and reports directly to maintainers	Direct accountability for each finding	Disclosure pace set by discoverer, not by maintainer absorption capacity; ledger reconciliation disputed [6]
Shared community SIRT	Linux Foundation Akrites [11]	Neutral intake, deduplication, and coordination for open-source maintainers	Reduces duplicate reports and triage load on maintainers	Funding sustainability; operational maturity still emerging
Pooled commercial coalition	Chainguard Athena [23]	Members pool findings and remediation work	Shares fix-development labor across organizations	Concentration of pre-disclosure data; member/non-member asymmetry

Model	Example	Primary Function	Principal Benefit	Principal Systemic Concern
Vendor clearinghouse	IBM/Red Hat Lightwell [12] [28]	Vendor-operated remediation and coordinated handling	Large dedicated engineering capacity	Single-vendor governance; initial restriction to specific sectors [23]

None of these models is inherently inadequate, and a resilient ecosystem will likely require several of them operating together. The systemic concern is that the coordination layer is consolidating rapidly, under commercial and time pressure, without shared standards for security of the pooled data, equitable access to fixes, or transparency about what is held in embargo and for how long.

Regulatory Clocks That Assume a Fix Exists

Regulators have responded to collapsing exploitation timelines by compressing remediation and reporting deadlines. CISA's BOD 26-04, issued on June 10, 2026, replaced calendar-based, severity-anchored federal patching rules with a risk-based model that evaluates whether an asset is publicly exposed, whether a vulnerability is known to be exploited, whether exploitation can be automated, and whether successful exploitation grants full control of the system; the highest-risk combination requires remediation within three days [16][25]. In the European Union, the Cyber Resilience Act's reporting obligations for actively exploited vulnerabilities and severe incidents began applying to manufacturers on September 11, 2026, with an early warning due within 24 hours of awareness [29], submitted through ENISA's Single Reporting Platform [17].

In CSA's view, both measures are well-founded responses to the exploitation data, and BOD 26-04's move toward risk-based prioritization aligns with what CSA has recommended for enterprise programs [22][25], a direction also argued in industry commentary published on the CSA blog [27]. Both, however, operate most effectively when a fix exists and when the information needed to identify affected assets is available. The disclosure bottleneck produces a growing population of vulnerabilities for which neither condition holds: flaws that are known to a discoverer and perhaps to adversaries, but that are not yet patched, not yet published, or published without the enrichment that would let an organization determine its exposure. Compressed deadlines applied to that population risk generating compliance activity without security benefit, or pushing organizations toward emergency changes that introduce operational risk.

The CRA's reporting regime also creates a coordination question that has not been fully resolved. When a manufacturer becomes aware that a vulnerability in its product is being actively exploited, it must report within 24 hours. If the vulnerability originates in an upstream open-source component whose maintainer has not yet produced a fix, the manufacturer's report enters the regulatory system before the upstream fix exists.

How regulatory channels, open-source maintainers, and clearinghouses share information in that window—and how they protect it—will shape whether the new reporting obligations accelerate remediation or merely relocate the bottleneck.

Summary of Systemic Risk Mechanisms

The mechanisms described in this section reinforce one another. The table below summarizes them and indicates where in the pipeline each originates.

Risk Mechanism	Origin in Pipeline	Why It Is Systemic
Latent exposure from backlog	Validation through fix development	Queued flaws are plausibly discoverable by others using comparable tools; risk grows with queue length
Intake collapse	Intake and coordination	Channel closures reduce capacity for valid reports along with invalid ones
Severity distortion	Triage	Inflated ratings misallocate remediation capacity across all consumers of the data
Metadata gap	Publication and enrichment	Fixes exist but cannot be matched to affected assets at scale
Long-tail consumption	Downstream packaging and deployment	Patched upstream does not mean protected in production; tail persists for years
Coordination-layer concentration	Intake through disclosure	Pooled embargoed data creates high-value targets and tiered access to fixes
Regulatory clock mismatch	Disclosure through deployment	Deadlines presume fix availability that the bottleneck does not deliver

Measuring Patch Consumption Capacity

From Patch Velocity to Absorption Capacity

Enterprise vulnerability management has long measured mean time to remediate and the share of vulnerabilities closed within service-level targets. Those metrics remain useful, but they describe outcomes rather than capacity, and they implicitly assume that the inflow of vulnerabilities is roughly stable. Under the disclosure bottleneck, the more important question is how much inflow an organization can absorb before its backlog begins to grow without bound.

We use the term *patch consumption capacity* to describe that rate: the number of fixes an organization can identify as relevant, test, and deploy across its environment in a given period, weighted by the risk they address. Patch consumption capacity depends on asset inventory accuracy, the degree of automation in build and deployment pipelines, the availability of test environments, change-management latency, and the organization's ability to deploy compensating controls when a patch cannot be applied. It is bounded both by internal factors and by the upstream pipeline: an organization cannot consume a fix that has not been released or that it cannot match to its assets.

Framing remediation in these terms has practical consequences. It makes visible that the relevant comparison is between inflow and capacity, not between the organization's current performance and an industry benchmark. An organization that remediates critical vulnerabilities in a median of ten days may still be accumulating backlog if critical inflow has tripled. Conversely, it focuses investment on the factors that actually raise capacity—automation, inventory, architectural simplification—rather than on exhortations to patch faster.

Indicators Worth Tracking

Organizations seeking to manage patch consumption capacity can begin with a small set of indicators that reveal whether the backlog is stable. The most important is the ratio of new applicable vulnerabilities to closed vulnerabilities over a rolling period, segmented by risk tier; a ratio persistently above one indicates that capacity is being exceeded. A second is the age distribution of open vulnerabilities in internet-exposed and high-impact systems, which reveals whether the backlog is being worked from the top or is accumulating at the highest-risk tier. A third is the share of applicable vulnerabilities for which the organization lacks sufficient metadata to determine exposure—an indicator that has become more important since the NVD change.

Two further indicators address dependencies outside the organization's direct control. The first is time-to-consume for upstream fixes: the interval between an upstream open-source release containing a security fix and its deployment in the organization's environment. The second is the inventory of critical dependencies

whose upstream disclosure or remediation channels have changed—for example, projects that have closed bounty programs, restricted intake, or joined a clearinghouse. CSA's October analysis of vulnerability intake recommended exactly this kind of dependency-channel monitoring [24].

The Role of Compensating Controls

Because no realistic increase in patch consumption capacity will fully match machine-speed discovery in the near term, organizations must reduce their dependence on any single patch arriving in time. Anthropic's own guidance to defenders makes this point, recommending controls such as enforced multi-factor authentication and comprehensive logging that improve security independent of specific patches [2]. CSA's April 2026 strategy briefing on the "AI Vulnerability Storm" framed the objective as minimum viable resilience, measured by the cost of exploitation, early detection of compromise, and containment of blast radius, rather than by the completeness of patching [21].

In practice, this means investing in network segmentation that limits which systems are reachable from untrusted networks, least-privilege designs that limit the consequences of a compromised component, immutable and rapidly rebuildable infrastructure that reduces the cost of each deployment, virtual patching and runtime protection for components that cannot be updated promptly, and detection engineering focused on the behaviors that exploitation produces rather than on specific vulnerability signatures. These controls do not eliminate the bottleneck, but they convert some fraction of the latent exposure in the backlog from exploitable to contained.

Responses Across the Ecosystem

What AI Developers Running Discovery Programs Can Do

Organizations that operate large-scale AI discovery programs are in an unusual position: they generate the inflow that strains the pipeline, and they possess the tools most likely to help relieve it. Several practices would reduce the systemic risk their programs create.

The first is to pace disclosure to absorption capacity. Anthropic has acknowledged maintainer requests to slow disclosure [2], and responsible discovery programs should treat each maintainer's demonstrated capacity as a constraint rather than an obstacle, prioritizing within each project the findings with the greatest real-world exploitability. The second is to reduce the work required to act on each finding by delivering validated reproductions and, where feasible, candidate patches alongside reports. A finding accompanied by a tested candidate fix can shift the maintainer's task from design toward review, which is often faster—provided the candidate patch is of high enough quality that reviewing it costs less than writing a fix

independently. The third is calibration: given the gap VulnCheck identified between model and maintainer severity ratings [6], discovery programs should calibrate their severity assessments against maintainer judgments and publish the methodology behind them.

The fourth is transparency and reconcilability. Public disclosure ledgers are a welcome development, but their value depends on being internally consistent and independently auditable. Discovery programs should publish data in machine-readable form with stable identifiers, clear state definitions, and timestamps that reflect when events actually occurred. Finally, programs should route findings through shared intake infrastructure such as Akrites where it exists [11], rather than adding another independent stream to each maintainer's inbox.

What Maintainers and Open-Source Foundations Can Do

Maintainers cannot absorb machine-speed inflow through effort alone, and the open-source community has begun building the institutions needed to share the load. Shared intake and triage, as Akrites proposes, is among the most promising structural changes because it addresses duplication and validation—the stages where maintainer time is most often wasted—before reports reach the project [11]; its effectiveness will depend on sustained funding and operational maturity that are still being established. Projects can complement this by publishing clear security policies that specify required evidence, such as a working reproduction or a candidate patch, before a report enters human review, a practice CSA's intake analysis also recommended [24].

Funding remains the underlying constraint. The commitments made in 2026 by AI developers, clearinghouse operators, and foundations are significant, but they are concentrated in a relatively small set of critical projects and in time-limited programs. Durable improvement will require sustained funding for security staff, triage support, and release engineering in the projects on which critical infrastructure depends. Foundations are well positioned to identify those projects, broker funding, and set expectations for how discovery programs interact with maintainers.

What Enterprises Can Do

Enterprises are the final consumers of the pipeline and bear many of the consequences of its failures. Their priorities follow from the analysis above: measure patch consumption capacity and treat a persistently growing backlog as a material risk to be reported to leadership; diversify vulnerability intelligence sources beyond the NVD, incorporating vendor advisories, the GitHub Advisory Database, KEV, exploit-prediction data, and commercial feeds; adopt risk-based prioritization along the lines of BOD 26-04 even where it is not mandatory [25]; and invest in the compensating controls that reduce dependence on any single patch.

Enterprises that depend heavily on open-source components should also take a more active role upstream. That includes maintaining an accurate software bill of materials so that upstream fixes can be matched to deployed assets quickly, identifying the critical dependencies whose maintainers are most capacity-constrained, and considering direct support for those maintainers or participation in shared remediation efforts. When evaluating clearinghouse memberships, enterprises should assess not only the speed of access to fixes but the security controls protecting pooled pre-disclosure data and the governance arrangements that determine who receives what information and when [23].

What Policymakers Can Do

Public policy has so far focused on the ends of the pipeline: encouraging or funding discovery, and compressing reporting and remediation deadlines. The analysis in this paper suggests that the middle of the pipeline—validation, coordination, fix development, and enrichment—deserves comparable attention, because it is where the binding constraint now sits.

Several measures follow. Public vulnerability infrastructure, including the NVD and the CVE program, should be resourced to match current volumes, or its narrowed scope should be complemented by publicly funded alternatives so that enrichment does not become available only to those who can pay for commercial feeds. Governments that depend on open-source software for critical services should treat maintainer security capacity for that software as critical infrastructure and fund it accordingly. Reporting regimes such as the CRA should clarify how information about actively exploited vulnerabilities in upstream components flows to maintainers and how it is protected in the period before a fix exists. Finally, governments that authorize or partner with large-scale AI discovery programs should consider requiring disclosure pacing, data-security standards for pooled pre-disclosure findings, and published, auditable statistics as conditions of that partnership.

CSA Resource Alignment

CSA has published a substantial body of work on AI-accelerated vulnerability discovery during 2026, and this paper builds directly on it. The most specific antecedent is CSA's June 2026 analysis, *Project Glasswing: AI Discovery Outpaces Open Source Patching Capacity* [20], which documented the early Glasswing disclosure figures and first characterized the gap between machine-speed discovery and open-source remediation as a structural deficit. This whitepaper extends that analysis with the later October data, independent critiques of the disclosure ledgers, and a pipeline model that locates the deficit at each stage rather than at the maintainer alone. Readers looking for a practical response to the immediate Glasswing findings, including dependency audits against affected projects, should consult that note alongside this paper. Two further CSA works address the same subject from complementary angles. *Project Glasswing and the AI Vulnerability*

Disclosure Velocity Crisis [32] examined the strain that Glasswing's disclosure pace placed on coordinated disclosure in May 2026, and *AI-Accelerated Vulnerability Discovery and the Patch Debt Crisis* [33] treated the accumulating backlog as systemic patch debt, the framing this paper develops through its pipeline model.

The systemic-risk mechanisms described in this paper draw heavily on two more recent CSA analyses. *The Clearinghouse Rush: Concentration Risk in AI Patching* [23] examined the emergence of Athena and Lightwell and the concentration risk created by pooling pre-disclosure data; the coordination-model comparison in this paper situates those clearinghouses alongside discoverer-led disclosure and the community-run Akrites model. *Vulnerability Intake Collapse: Bounty Shutdowns and Maintainer Capacity* [24] documented the sequence of bounty pauses through October 2026 and argued that fewer reports do not mean fewer vulnerabilities—a conclusion this paper adopts in its treatment of intake collapse as a systemic, rather than a local, failure.

For enterprise program design, CSA's *The "AI Vulnerability Storm": Building a "Mythos-ready" Security Program* [21] provides the operational framework that this paper's recommendations on compensating controls and patch consumption capacity are intended to complement. Its emphasis on minimum viable resilience—raising the cost of exploitation, detecting compromise early, and containing blast radius—is the appropriate response to a backlog that will not drain in the near term. CSA's *The AI-Driven Patch Wave* [22] and its analysis of BOD 26-04's risk-based patching model [25] provide further detail on adapting enterprise remediation models and risk-based prioritization.

Finally, the controls recommended here map to the AI Controls Matrix (AICM) v1.1 [26]. The Threat and Vulnerability Management domain addresses vulnerability identification, prioritization, and remediation, including the risk-based approaches discussed above; the Application and Interface Security domain addresses secure development practices relevant to maintainers and to organizations building software on open-source components; and the Supply Chain Management, Transparency, and Accountability domain addresses the dependency visibility and third-party governance that clearinghouse participation and upstream engagement require. Organizations using AICM to structure their AI and cloud security programs can use the patch consumption indicators in this paper as evidence for the operating effectiveness of those controls.

Conclusions and Recommendations

Conclusions

The central finding of this paper is that the binding constraint in software security has moved. Discovery, long the scarce resource, is now abundant; the scarce resources are validation, coordination, fix development, enrichment, and deployment—every stage that still depends on human judgment. The evidence for this shift is

consistent across sources that otherwise disagree on precise figures: discovery programs report findings in the tens or hundreds of thousands, known patch rates for disclosed open-source vulnerabilities remained in the single-digit percentages as of October 2026, maintainers describe their intake channels as unmanageable, and public enrichment infrastructure has narrowed its scope.

This gap is a systemic risk because the backlog it creates is not inert. Many queued vulnerabilities are discoverable with tools that are or will soon be available to adversaries. Exploitation timelines have collapsed: Mandiant's 2025 incident-response data put the mean time to exploit at negative seven days, meaning that among the vulnerabilities it observed being exploited, exploitation frequently began before a patch was available [18]. The backlog is concentrated in the most widely used software, the discovery capability is concentrated in a small number of models, and the new coordination layer is concentrating pre-disclosure data in a small number of institutions. Regulatory regimes are tightening deadlines in ways that presume a fix exists. Each of these factors would be manageable alone; together they describe a system whose failure modes are correlated.

None of this argues against AI-assisted vulnerability discovery. The vulnerabilities being found are real, many have existed undetected for years, and defenders are better off knowing about them. The argument is that discovery capacity has been scaled without commensurate investment in the rest of the pipeline, and that the gap must now be closed from the downstream side.

Recommendations

The recommendations below are organized by the party best positioned to act. Each is intended to increase pipeline throughput at a constrained stage or to reduce the exposure created by the backlog while it persists.

Stakeholder	Recommendation	Pipeline Stage Addressed
Enterprises	Measure patch consumption capacity and report sustained backlog growth to leadership as a material risk	Deployment
Enterprises	Diversify vulnerability intelligence beyond the NVD and adopt risk-based prioritization aligned with BOD 26-04	Enrichment, triage
Enterprises	Invest in segmentation, least privilege, rebuildable infrastructure, and behavior-based detection to reduce reliance on any single patch	Deployment (compensating)
Enterprises	Maintain accurate SBOMs and identify capacity-constrained upstream dependencies for monitoring or	Downstream consumption

Stakeholder	Recommendation	Pipeline Stage Addressed
	direct support	
AI discovery programs	Pace disclosure to maintainer absorption capacity and route findings through shared intake where available	Intake, coordination
AI discovery programs	Deliver validated reproductions and candidate patches with findings; calibrate severity against maintainer judgment	Validation, fix development
AI discovery programs	Publish reconcilable, machine-readable disclosure data with clear state definitions	Transparency across stages
Maintainers and foundations	Adopt shared triage and deduplication infrastructure and require reproducible evidence before human review	Intake, validation
Maintainers and foundations	Secure sustained funding for security staff and release engineering in critical projects	Fix development, publication
Clearinghouse operators	Publish security controls for pooled pre-disclosure data and policies governing access for non-members	Coordination
Policymakers	Resource public enrichment infrastructure to current volumes or fund public alternatives	Enrichment
Policymakers	Treat maintainer capacity for critical open-source software as critical infrastructure	Fix development
Policymakers	Clarify information flows under reporting regimes for upstream vulnerabilities without fixes	Disclosure, coordination

The period during which discovery dramatically outpaces remediation is likely to persist for some time, and possibly to widen as model capabilities continue to improve. Organizations that recognize the bottleneck as a defining feature of the current vulnerability landscape—and that invest in absorption capacity and resilience rather than in further acceleration of discovery alone—will be better positioned to withstand it. The broader ecosystem's ability to do the same will depend on whether the institutions that sit between a finding and a deployed fix receive investment proportionate to the load now placed on them.

References

[1] Anthropic. "Project Glasswing." Anthropic, April 2026.

[2] Anthropic. "Project Glasswing: An Initial Update." Anthropic, May 22, 2026.

[3] Anthropic. "Coordinated Vulnerability Disclosure Dashboard (October 2, 2026 Snapshot)." Anthropic Frontier Red Team, October 2, 2026.

[4] Anthropic. "Introducing the Cyber Verification Program." Anthropic, October 6, 2026.

[5] Software Improvement Group. "Mythos and Project Glasswing Explained (October 2026 Update)." Software Improvement Group, October 2026.

[6] Garrity, Patrick. "Anthropic's Project Glasswing Receipts." VulnCheck, September 8, 2026.

[7] Schneier, Bruce. "Anthropic's Project Glasswing Update." Schneier on Security, June 8, 2026.

[8] GamingOnLinux. "Linux Head Says AI Tools Are Great, but They're Making the Security List 'Almost Entirely Unmanageable'." GamingOnLinux, May 18, 2026.

[9] Help Net Security. "AI Slop Submissions Force Google to Freeze Its Open-Source Bug Bounty." Help Net Security, October 5, 2026.

[10] CyberScoop. "Before You File That AI Bug Report, OpenAI Has Some Advice." CyberScoop, October 27, 2026.

[11] Linux Foundation. "Linux Foundation and Industry Leaders Launch Akrites to Defend Critical Open Source Software Against AI-Enabled Cyber Threats." Linux Foundation, June 25, 2026.

[12] Red Hat. "IBM and Red Hat Expand Lightwell with New Offerings to Build the Trust Infrastructure for AI-Era Open Source." Red Hat, July 8, 2026.

[13] Tech Newsday. "Chainguard Joins Push to Protect Open-Source Software from AI-Powered Vulnerability Hunting." Tech Newsday, June 4, 2026.

[14] NIST. "NIST Updates NVD Operations to Address Record CVE Growth." National Institute of Standards and Technology, April 2026.

[15] The Hacker News. "NIST Limits CVE Enrichment After 263% Surge in Vulnerability Submissions." The Hacker News, April 2026.

[16] CISA. "BOD 26-04: Prioritizing Security Updates Based on Risk." Cybersecurity and Infrastructure Security Agency, June 10, 2026.

[17] ENISA. "CRA Single Reporting Platform (SRP)." European Union Agency for Cybersecurity, September 2026.

[18] Google Cloud / Mandiant. "M-Trends 2026: Data, Insights, and Strategies From the Frontlines." Google Cloud Threat Intelligence, April 2026.

[19] Sonatype. "Exploring Unnecessary Risk: The Persistence of Open Source Vulnerabilities." Sonatype, December 10, 2025.

[20] Cloud Security Alliance. "Project Glasswing: AI Discovery Outpaces Open Source Patching Capacity." CSA, June 2026.

[21] Cloud Security Alliance. "The 'AI Vulnerability Storm': Building a 'Mythos-ready' Security Program." CSA, April 2026.

[22] Cloud Security Alliance AI Safety Initiative. "The AI-Driven Patch Wave: How Machine-Speed Vulnerability Discovery Is Breaking the Enterprise Remediation Model." CSA, May 2026.

[23] Cloud Security Alliance. "The Clearinghouse Rush: Concentration Risk in AI Patching." CSA, July 2026.

[24] Cloud Security Alliance AI Safety Initiative. "Vulnerability Intake Collapse: Bounty Shutdowns and Maintainer Capacity." CSA, October 6, 2026.

[25] Cloud Security Alliance. "CISA's Risk-Based Patching Mandate Meets an AI Platform." CSA, July 13, 2026.

[26] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." CSA, June 22, 2026.

[27] Gill, Jennifer (Skyhawk Security). "Patching Faster Is Not the Answer to Mythos; Patching Smarter Is." CSA Blog (guest post), May 14, 2026.

[28] Poireault, Kevin. "Red Hat's Lightwell Project Remediates 400 Open-Source Vulnerabilities." Infosecurity Magazine, October 6, 2026.

[29] European Parliament and Council of the European Union. "Regulation (EU) 2024/2847 (Cyber Resilience Act)." Official Journal of the European Union, November 20, 2024.

[30] Google Project Zero. "Vulnerability Disclosure Policy." Google, accessed November 2026.

[31] Anthropic. "Coordinated Vulnerability Disclosure Dashboard (May 22, 2026 Snapshot)." Anthropic Frontier Red Team, May 22, 2026.

[32] Cloud Security Alliance. "Project Glasswing and the AI Vulnerability Disclosure Velocity Crisis." CSA, May 24, 2026.

[33] Cloud Security Alliance. "AI-Accelerated Vulnerability Discovery and the Patch Debt Crisis." CSA, June 17, 2026.