

Verification Debt

The change without a signatory — who verifies, accepts and deploys the fixes that machines now help find

*AI can find flaws and help repair and verify them. The open question is how ownership, assurance, approval and deployment affect the time until systems are **protected**.*

REPORT • V1.0 • EVIDENCE CUTOFF 2026-10-23 • PUBLISHED 2026-10-25

CONTENTS

Verification Debt

Lock	Interpretation Lock	3
Ch 00	The finding is cheap	5
Ch 01	The Discovery Ledger	7
Ch 02	Where the queue forms	10
Ch 03	The net balance	12
Ch 04	The verifier paradox	15
Ch 05	The ceiling	16
Ch 06	The mandate	18
Horizon	The world a model learns in	21
Register	Sources	22
Register	Glossary	25

How to read the figures. Each figure carries its own legend. ■ Gold marks AI-assisted work and released fixes; ■ red marks limits, exposure or risk events. Colour never encodes a measured quantity of debt.

How to read the numbers. Figures from AI labs and security vendors are their own reports and are marked as such in the text; counts of findings, attempts and patches are not counts of compromises. Evidence labels in notes: [M] maintainer record · [V] vendor or interested party · [G] government or standards body · [R] independent research · [P] press or secondary · NR not resolved.

HOW TO READ THIS ISSUE

Interpretation Lock

These definitions and refutation criteria were revised after the second adversarial review and are fixed in this edition for any later update. They are interpretive rules, not a preregistered test of the cases below.

Hypothesis. Agentic tools have made writing code, finding vulnerabilities and, increasingly, drafting fixes much cheaper. The rest of the path from a confirmed flaw to a system that is actually safer has not become cheaper at the same rate: settling who owns the flaw, producing evidence a responsible party will accept, obtaining the authority to release, and getting the change onto the machines that run the code. We call the gap that accumulates *verification debt*. We expect it to be uneven: owners who hold their whole pipeline absorb AI-scale findings; fragmented ecosystems, unsupported software and certified systems accumulate them. Public evidence does not yet show which downstream stage dominates; this issue sets out how to find out.

Four boundaries apply throughout. Verification is itself partly automatable. The net security effect of agent-written code is treated as unresolved, not negative. Safety rules are read as constraints on acceptance and release, not on development automation as such. And accountability is not the same as effort: an organisation can remain accountable while the cost of exercising that accountability falls.

Classification before outcomes. Each finding is classified together with its maintained release and deployment cohort before its outcome is examined. An integrated owner controls release and deployment for that cohort; control of an update channel alone does not establish control over independently administered systems. Any other case counts as fragmented.

What would refute it.

1. **Repair remains dominant.** In matched fragmented cohorts, constructing a correct repair takes most of the critical-path time and person-hours, measured separately from waiting, and pre-specified equal-cost help with repair shortens time-to-protection more than equal-cost help downstream.
2. **No fragmentation penalty.** With the equivalence margin, follow-up window, cohort membership and protection threshold declared in advance, fragmented cohorts fall within the margin of integrated ones on time-to-protection and cost. Failing to find a difference does not count as equivalence.
3. **Downstream assurance becomes cheap.** In the same comparison, downstream human effort and time fall at least as much, proportionally, as discovery and repair costs, under unchanged assurance objectives and a declared regression tolerance.
4. **Known backlog grows while risk falls.** On a stable or reweighted estate, severity-adjusted exposure falls over a fixed window even as open findings rise — which would show the backlog to be a poor proxy for risk.

Evidence cutoff: 23 October 2026, 23:59 GST. Later events appear only in a postscript, and only if they change a conclusion.

Conflict rule. This issue was prepared with Claude, an Anthropic model, and Anthropic's Project Glasswing is a central case. Figures from Anthropic, OpenAI, Google, Microsoft and security vendors are treated as self-reports and paired with independent or contrary evidence; a vendor conceding a bottleneck is still a vendor report. Vendors are named alike in favourable and unfavourable cases. A second model acted as opponent through three rounds of review. The issue was prepared independently and without external funding.

Out of scope. Startups and hobby projects; machine consciousness; world models for robot training (see the horizon box).

CHAPTER 00

The finding is cheap

In October 2026 two scenes showed how much work can follow a finding — fixing it, shipping the fix and answering for it — even when the finding itself came quickly.

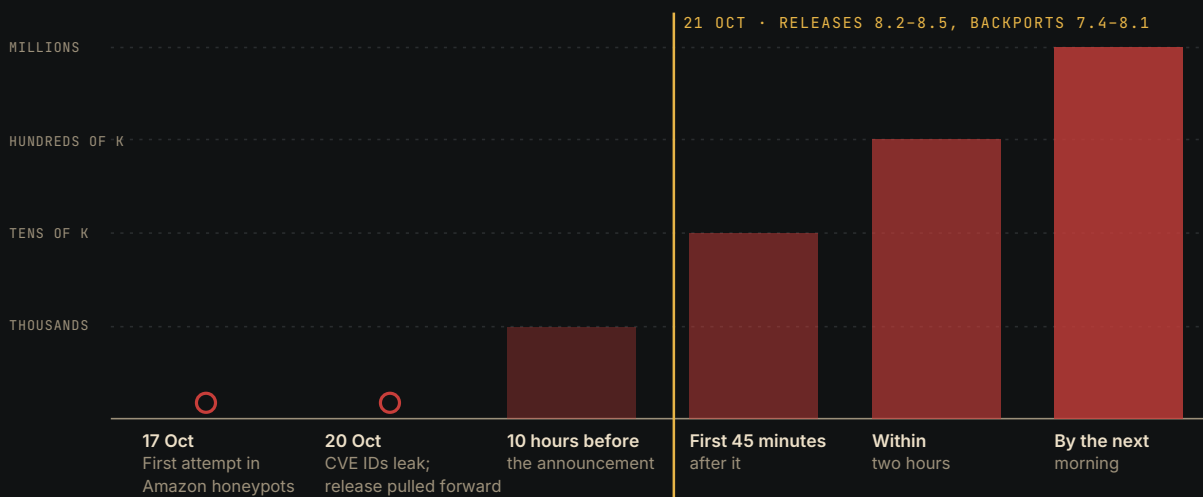
Scene one: PHP. On 21 October the PHP project shipped emergency releases for two critical flaws in the language core, CVE-2026-109618 and CVE-2026-109619, both rated 9.2 under CVSS v4.0. An unauthenticated request could run code on the server before any application logic executed. The bugs had sat in PHP since version 5.4. The **PHP Foundation's account** is unusually candid. A core developer it contracted found the first signs of the out-of-memory flaw, CVE-2026-109618, in 2024, without a concrete attack path. Calif, auditing PHP inside OpenAI's Patch the Planet programme with GPT-5.6 Sol, and OpenAI reported its root cause on 11 August 2026; Wiz, with its own tooling, reported further attack vectors on 24 August. The security team weighed fixes that each broke compatibility, created new denial-of-service risk or required changes across third-party extensions, and the Foundation attributes the delay to "a structural failure, caused by both the security team's decision process and its lack of resources" — a small group of core developers with deep engine knowledge. CVE identifiers **leaked** on 20 October; that evening Calif mistakenly published a proof-of-concept video, later deleted, and the release was pulled forward a day. A distributor, Chainguard, published fixed images more than six hours before the coordinated release, which the Foundation describes as breaking the embargo. Regular releases covered the supported branches, 8.2 to 8.5; php.net lists project-provided minimal **source backports** for 7.4, 8.0 and 8.1 and directs users of older versions to distributor or LTS support, or to an upgrade.

What the window looked like. Amazon's honeypots recorded a PHP2Shell exploitation attempt on **17 October**, four days before disclosure; by the day of release several actors were probing. Cloudflare's chief executive reported thousands of attempts across its network in the ten hours before the announcement, tens of thousands in the first 45 minutes after it, hundreds of thousands within two hours, and **millions by the next morning** — attempts, not compromises, and a vendor's own count. Amazon's conclusion was blunt: AI-assisted analysis now turns a patch or an advisory into a working exploit within hours, so patch timelines for exposed systems should be measured in hours too.

— FIGURE 01 • PHP2SHELL: ATTEMPTS AROUND DISCLOSURE

Reported attempts rose from thousands to millions around disclosure

PHP2Shell, October 2026 • exploitation attempts, not compromises • Cloudflare and Amazon, their own counts



BAR HEIGHT = ORDER OF MAGNITUDE AS REPORTED, NOT EXACT COUNTS • COLUMNS ARE NOT TO TIME SCALE

Scene two: a bank. A week earlier, Bank of America's chief executive said the bank had brought in **1,000 contractors** in the second quarter to patch flaws exposed by frontier models, on top of 3,000 staff already in that function. The quarter's bill for remediation and separate anti-money-laundering work ran to a few hundred million dollars. The bank did not say how that work divided between diagnosis, repair, testing, supplier coordination and approval. Another bank shows the other side: at Itaú, a case study hosted by the supplier, Cognition, reports that about **70% of static-analysis findings** are now resolved automatically, with engineers reviewing each patch — a vendor account without a controlled comparison.

The question. Neither scene is about whether AI can find vulnerabilities; that is settled. Both are about what happens next — who verifies the finding, who writes and checks the fix, who deploys it, and who is entitled to sign the change. This issue follows that chain through enterprise and state software, from open-source maintainers to banks to nuclear licensing, and asks three questions:

1. Where does remediation now stall — and for whom?
2. Do coding agents reduce the net vulnerability load, or add to it, including through the agentic pipeline itself?
3. Where do the rules of critical infrastructure constrain autonomy, and what would change that?

CHAPTER 01

The Discovery Ledger

The leading AI labs now run vulnerability-finding programmes; the useful comparison is not between models but between stages — candidate, confirmed, reported, patched, deployed — and the public record thins out sharply after discovery.

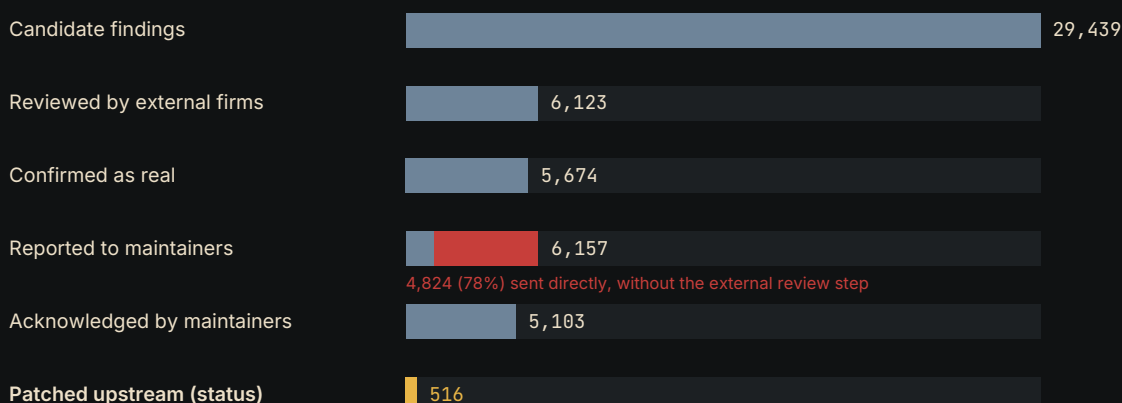
The one public set of stage counts. Only one programme publishes counts for several stages, and even it stops before deployment. Anthropic's **disclosure dashboard**, last updated 2 October, counts 29,439 candidate findings from its models in open-source code. External security firms reviewed 6,123 and confirmed 5,674 — a 92.7% confirmation rate in a reviewed sample, not a measure across all candidates. In total 6,157 findings went to maintainers across 591 projects, and 516 findings were marked “patched upstream” — a status the dashboard's overview describes as a released fix and its glossary as a landed patch. It counts findings, not distinct releases, and neither description establishes installation. The dashboard names independent human triage as the rate-limiting step. These are self-reported figures from an interested party.

One detail changes the reading. Of the 6,157 reports, 4,824 — about 78% — went to maintainers directly, without the external review step, and may include false positives. Verification did not disappear; the first check moved to whoever received the report. The dashboard does not measure that review effort, nor whether the maintainers are paid or volunteers.

— FIGURE 02 · ANTHROPIC'S PUBLISHED DISCLOSURE COUNTS

Anthropic's published counts: 6,157 reports, 516 marked patched upstream

Anthropic models, open-source code, Nov 2025 to 2 Oct 2026 · self-reported · stages are asynchronous counts



92.7% = confirmed share of the sample external firms reviewed. "Patched upstream" counts findings, not releases or installs.

SOURCE: ANTHROPIC COORDINATED VULNERABILITY DISCLOSURE DASHBOARD, SNAPSHOT OF 2 OCT 2026

The stages are separate counts, not one cohort, so no conversion rate follows from them. Of the 5,674 confirmed findings, 1,333 had been reported through the reviewed route by 2 October; of all 6,157 reports, by either route, 516 findings were marked patched upstream. The shape is still informative: discovery is not where the record thins.

Anthropic's own reading, published on 8 October, is that Glasswing partners found many vulnerabilities but the programme has **not yet achieved a sufficient reduction in cyber risk**: months often passed between a flaw being found and fixed, and in operational technology a fix may have to wait until it can be applied safely to running machinery. That is still a vendor's statement, whichever way it cuts.

The other programmes publish points, not funnels.

CASE	WHO, WITH WHAT	WHAT WAS FOUND	WHAT HAPPENED NEXT
PHP2Shell	Calif and OpenAI (Patch the Planet, GPT-5.6 Sol); Wiz independently	Two RCE bugs in the PHP core, present since 5.4; first signs of the out-of-memory flaw in 2024	Root cause of that flaw reported 11 August; releases 21 October for 8.2–8.5, source backports for 7.4–8.1
PwnEscape	pwn.ai harness on GPT-5.6 Sol	Guest-to-host escape in Linux networking, present since 2.6.27	Primitive found on day three in August; mainline merge 14 October
FreeBSD CVE-2026-4747	Credited to Nicholas Carlini using Claude	Remote code execution via RPCSEC_GSS packet validation	Advisory and corrections for all supported branches on 26 March
wolfSSL CVE-2026-5194	Anthropic model, triaged by its security team	Certificate-signature forgery, critical	Sent and acknowledged 26 March (Anthropic record); fix merged 6 April; release 8 April
Firefox CVE-2026-8390	OpenAI Preparedness and Bill Demirkapi (credited for this CVE only)	WebAssembly use-after-free, high	Fixed in Firefox 150.0.3 on 12 May
Chrome full chain	XBOW: four researchers with GPT-5.6 Sol and its publicly available predecessors	Three chained bugs, Chrome on Windows	Built 2–7 August after earlier work, about \$40,000 of compute; Google fixed all three; disclosed 13 October
Chrome 149–150	Google: fuzzing, outside researchers and an increasingly AI-assisted pipeline	1,072 security bugs fixed in two milestones	LLMs draft candidate fixes for most; triage to release in one to two days
CodeMender, 2025	Google DeepMind agent; human researchers review	Security fixes for outside open-source projects	72 security fixes upstreamed in about six months
curl, Mythos scan	Anthropic's Mythos	Five reported as confirmed	One low-severity CVE, three false positives, one ordinary bug
curl, AISLE audit	AISLE, working with the maintainers	At least 35 published curl vulnerabilities over roughly the past year	AISLE staff authored dozens of commits; maintainer: "nothing but positive experiences"
AIxCC final, 2025	Seven DARPA finalist systems	54 of 63 injected bugs found (86%)	43 patched (68%); separately, 18 real bugs found and 11 patches provided

Fast fixes happen: wolfSSL released a fix thirteen calendar days after the report — not thirteen days of measured effort — FreeBSD shipped corrections across every supported branch on the day of its advisory, Mozilla fixed its flaw in a point release, and Google reports one to two days from triage to release in Chrome, with automated reproduction, routing and routing to the right owner. They share one feature — a maintainer organisation with security staff and a release train of its own. curl shows the same pattern from the volunteer side: where a capable partner does the triage and writes the commits, a small project can absorb dozens of AI-found vulnerabilities.

— FIGURE 03 · SELECTED REPORT- OR DISCOVERY-TO-FIX MILESTONES

Selected report- or discovery-to-fix milestones

Calendar days between the labelled events, logarithmic scale · endpoints differ between cases



MAINTAINER ADVISORIES, PHP FOUNDATION TIMELINE, ANTHROPIC CVD RECORD, PRESS · CALENDAR TIME, NOT EFFORT

Why the window matters. The same capability serves attackers. On 21 October CrowdStrike reported **four previously unreported actors** who, between May and August 2026, used Claude Code in differently configured intrusion workflows; one compromised at least 15 healthcare organisations in Argentina. In one case the operator had to intervene repeatedly to overcome the model's safety guardrails; CrowdStrike does not describe a bypass method common to all four. It judges, with moderate confidence, that agentic AI is turning baseline offensive operations into a commodity. Every week a finding waits in the queue is a week someone else may find it too.

CHAPTER 02

Where the queue forms

After discovery a finding passes through five more stages before a system is safer; models now do real work in repair and evidence, and integrated owners have begun to automate triage and routing, while for the stages that need authority or access to running machines the sources show less change.

— FIGURE 04 · SIX STAGES FROM A VULNERABILITY SIGNAL TO DEPLOYED PROTECTION

Six stages from a vulnerability signal to deployed protection

A measurement framework · illustrative cases from this issue, not measurements of one pipeline



NOT A RANKING OF STAGES: NO SOURCE MEASURES TIME OR EFFORT STAGE BY STAGE

The figures in the boxes are illustrations from this issue's cases, not measurements of one pipeline. No public source yet reports queue times stage by stage; that gap is itself one of the issue's findings (chapter 6).

Agents now write fixes — and create review work

The best public record of AI-drafted security patches comes from Patch the Planet, where Trail of Bits engineers direct agents and maintainers decide. Of 240 public pull requests, 54 were still open on 14 September; of the 186 decided, maintainers **merged 126 (67.7%)**, and in 91 of those accepted the proposed security fix without a security-relevant revision. Four were explicitly classified as rejected on technical grounds. A later review of more than 130 merged pull requests found **21 regressions** — 16 functional, one performance, four in build or test — none of which broke a security protection. An agent helped find them: 21 of its 58 candidates held up. In a separate comparison, the revised regression hunt produced 25% fewer false-positive findings than a baseline Codex review (95% interval 14–38%), while leaving some known regressions unconfirmed. Trail of Bits is an OpenAI programme partner; all of this is its own account.

The evidence on quality is contested, and both sides have interests. A 1Password study of 6,080 model-generated patches — repeated attempts at six deliberately complex vulnerabilities — found **26% clean fixes**, 20.1% that fixed the flaw but changed behaviour, and 53.9% that left it open or added a new one. Trail of Bits countered that the headline mixed runs told to apply the wrong fix and runs barred from testing; in the remaining trials, 86% of patches blocked the supplied exploit — which is not the same as a complete repair. It also offered a human baseline from its consulting

records: developers' first fixes failed to resolve the reported issue in 283 of 2,265 cases, or 12.5% — different tasks under different conditions, so not a like-for-like comparison. The partner-reported field results and 1Password's benchmark use different tasks too, so their percentages cannot be set against each other. In one freenginx case, an agent and the human maintainer independently introduced the same crash. For new CodeQL alerts in pull requests during the May–July 2024 public beta of its Autofix tool, GitHub reported a median of **28 minutes** to commit a fix, against 1.5 hours by hand — a vendor's non-randomised comparison that ends at the commit, not at deployment.

The defensible reading is narrow. Producing a plausible repair does not complete remediation. Models can also reduce the work of validation, backporting and review, while ownership, authorisation and access to deployed systems remain organisational responsibilities. Public evidence does not yet establish which of these constraints dominates across ecosystems. Integrated owners show what full control buys: Google's Chrome pipeline reproduces, rates, routes, patches and validates findings, and shipped 1,072 security fixes across two releases.

Where the sources show less change

Ownership. In volunteer projects the first question — whose problem is this, and is it real? — lands on a few people. Of Anthropic's 6,157 reports, 78% went to maintainers without external review; curl closed its security intake for five weeks; Linus Torvalds called the kernel's security list close to **unmanageable**. By spring the reports were **good rather than junk**, which raised the load: a valid report cannot be dismissed. Where an owner automates routing, as Chrome does, or a partner absorbs triage, as AISLE did for curl, this stage moves.

Release and deployment. A fix protects no one until it runs. PHP2Shell's regular releases covered PHP 8.2 to 8.5; by **Hadrian's** estimate from W3Techs data — a vendor figure and an upper bound — close to half of PHP sites run older versions that the project no longer updates. For 7.4 to 8.1 the project offered source backports and left packaging to distributors and LTS vendors; users of earlier versions depend on distributor or LTS support, or an upgrade. Advice on stopgaps diverged: one vendor recommended disabling a PHP setting as a partial measure, while **Datadog**, having reproduced the exploit, said only patching protects. Where one party runs the service, deployment can be confirmed: GitLab's **notice of 21 October** for a CVSS 10 flaw says GitLab.com was already running the patched version — confirmation for one managed service, not a fleet-wide time to protection. In US healthcare, ARPA-H has put the average time to patch at **491 days**, as reported by Infosecurity Magazine; the measurement's definition is not public in our sources. In operational technology, Anthropic notes, a fix may wait until it can be applied safely to a running plant. Chrome shows the integrated alternative for its own users: from version 150 on macOS it restarts itself to apply a pending update when no windows are open.

Vendor code. Bank of America added people for remediation of internal and supplier code. The public account does not separate diagnosis, repair, testing, supplier coordination and approval, so it does not show which of them was binding.

Producing a fix does not end the problem. Which of the remaining stages binds depends on who holds them — and none of the sources we reviewed publishes the numbers that would settle it.

CHAPTER 03

The net balance

Whether agent-written code reduces or adds to the vulnerability load has not been shown either way at population level; meanwhile the toolchain built around coding agents has become a target in its own right — breached mostly through old failures, with the first signs of agent-specific ones.

Code written by agents

The evidence comes mostly from vendors who sell the cure, and should be read that way; together it documents risk and benefit without settling the net effect. **Veracode** found that about 44% of code-generation tasks in its 2026 tests introduced a known vulnerability, and that the average share of secure decisions — 56% — had barely moved from its previous report. Microsoft's study of tens of thousands of its own engineers found that adopters of command-line agents merged about **24% more pull requests** — company research, on a measure the authors say is not value, and which leaves quality unexamined. **Faros**'s 2025 telemetry from more than 10,000 developers found review time up 91% and pull requests 154% larger in high-adoption teams, and no significant correlation between adoption and company-level gains; most use was autocomplete. The lever exists. In **Arnica**'s benchmark, security rules for agents cut insecure decisions from 58% to 32% on open-source applications held out from rule tuning, and turned the net effect of a session from more vulnerabilities introduced than removed (−6.9) to more removed (+5.6); across all applications, breaking changes per session roughly doubled, from 2.6 to 5.3 — a vendor benchmark graded largely by models. An independent preprint by **Shafique, Miller and Heymann** rebuilt ten Linux utilities with a supervised agent workflow and fuzz-tested them against the human versions: the agent versions were typically as reliable or more so, with fewer memory errors but more hangs. The relevant test is not whether risk reaches zero, but whether an assisted workflow lowers risk and total effort against a stated alternative.

We found no independent, outcome-measured study of agentic development in a regulated industry. That absence is itself a finding: the sectors where errors cost most are the ones where the evidence is thinnest.

The pipeline as a target

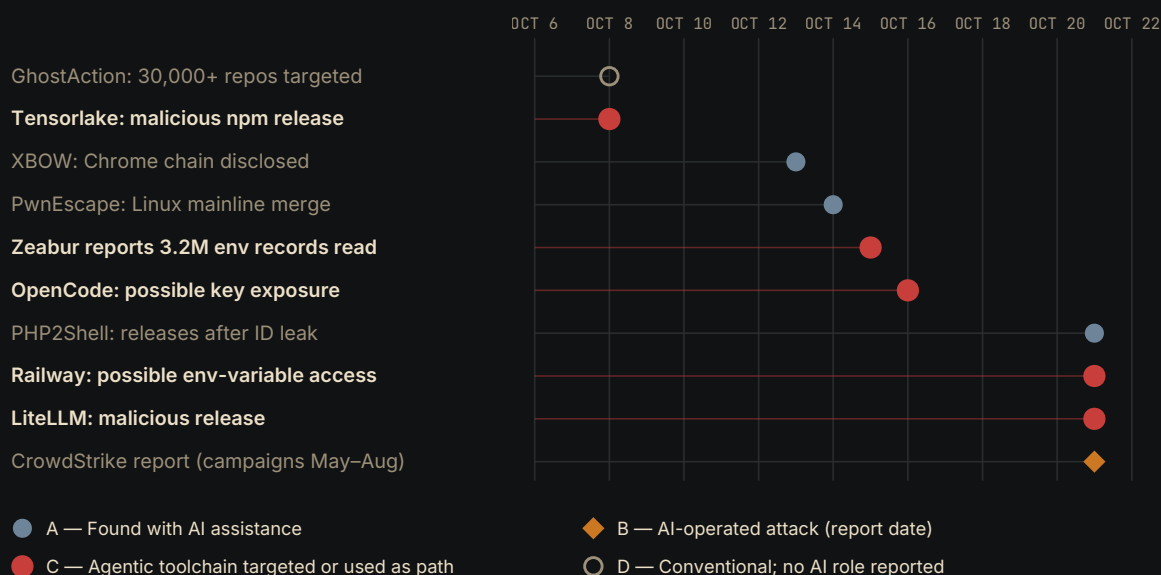
October 2026 was described on social media as a coordinated wave of agent-driven attacks. The record shows incidents that differ on four separate dimensions — how a flaw was found, how the attacker worked, what was targeted, and what technical mechanism was used — and one incident can score on several. Running agents is now itself an opera-

tional risk: in **Harness's** July survey of 700 practitioners at large firms that already run agents, 87% reported an agent-related security event in the past year — a vendor survey of a selected population, about agents as systems rather than about agent-written code.

— FIGURE 05 • SELECTED OCTOBER DISCLOSURES, WITH OVERLAPPING LABELS

Selected October disclosures: AI involvement, targets and attack paths

10 events cited in chapter 3 • incident, disclosure or release dates • one primary label each; dimensions overlap



PRIMARY LABEL FOR THIS ISSUE; INCIDENTS CAN CARRY SEVERAL • DISCLOSURE OR PATCH DATES • CHAPTER 3 SOURCES

Theft of paid-model credentials is not new: Sysdig documented “**LLMjacking**” in May 2024. What October adds is the agentic development toolchain as the target or the path. Agent workflows concentrate keys to paid AI services, cloud accounts and code hosts in a few places — deployment platforms, gateways, agent consoles — and a stolen AI key is spendable within minutes.

- **Tensorlake.** On 7–8 October malicious commits were pushed to the project’s main branch under a maintainer’s name, and the release workflow published them to npm with a **valid provenance attestation** — which records where a package was built, not whether it is safe. StepSecurity reports that the payload steals credentials, including AI-tool configurations, and writes agent and editor settings files into every repository it can reach, so that it runs again when someone opens the project in Claude Code or VS Code.
- **Zeabur**, a deployment platform, **reports** that an attacker read 3,218,965 environment-variable records from its platform database on 15 October, in its second security incident in under two months; it plans to shut down on 31 December. Its report names long-lived admin credentials stored in plaintext and usable from the internet; how the attacker obtained them is not confirmed. The **August incident** began with a known flaw (CVE-2026-7644) in the agent-tool (MCP) functionality of a NextChat service a customer had deployed.
- **OpenCode** said a legacy system behind its hosted console may have exposed the AI-provider keys of **1.4% of console users**; a full technical root cause had not been published in the sources we checked.

- ▶ **Railway** disclosed on 21 October that an intruder may have obtained **decrypted environment variables** for some users. Its belief that this was part of a series of attacks aimed at AI inference providers is the company's own assessment.
- ▶ **LiteLLM**, a widely used gateway for model APIs, shipped a malicious release on 21 October. Attackers used **stolen GitHub CLI tokens** of three developers: one account changed the release pipeline, a second approved the production release, a third deleted workflow records. Branch protection rejected the first unauthorised change; five seconds later a privileged developer credential overrode it. Seven malicious wheels were live on PyPI from 11:39 to 13:23 UTC; container tags were withdrawn from GitHub's registry by 13:51 and from Docker Hub by 14:20. LiteLLM later said, as reported by **The Hacker News**, that it had observed situations in which Claude Code embeds a user's GitHub CLI token in a response body, which can then appear in gateway logs, and suspects the intruder took tokens from the logs of an internal deployment exposed to the internet; how that deployment was reached is not established.
- ▶ **GhostAction** used about 1,900 hijacked GitHub accounts to **target more than 30,000 repositories** with credential-stealing workflows; GitHub began blocking the workflow on 9 October. This is a conventional CI compromise in which AI keys are one item among many; GitGuardian suspects some access relied on previously stolen, unrotated secrets.

None of these sources ties a flaw to code written by an AI system, and the issue makes no such claim. Two different propositions are worth separating. **Concentration of value** — agentic development gathers keys to paid models, cloud accounts and code hosts, plus the right to execute — is well supported by October. **A new trust-boundary failure** — an attack that depends on an agent-specific feature — has its first plausible examples: Tensorlake's reported use of agent configuration files as a persistence path, Zeabur's August entry through an agent-tool feature, LiteLLM's suspected token leak through an agent's responses, and an issue in the agent-tool protocol's SDKs that **OX Security** disclosed in April and describes as systemic, in packages it says have more than 150 million downloads; Anthropic, by OX's account, treats the behaviour as expected. The null hypothesis — ordinary platform and supply-chain breaches with AI keys as the latest loot — still explains most initial access. Verification debt accrues in the toolchain as well as in the code it produces.

CHAPTER 04

The verifier paradox

Demand for verification is rising while firms that adopt generative AI hire fewer juniors — and today's juniors are, in the ordinary course, the people who would sign changes in ten years. Whether that becomes a shortage of verifiers is a risk, not yet an observation.

The evidence on hiring. The most detailed study so far, by Hosseini and Lichtinger at Harvard, uses résumé and job-posting data on about 65 million workers at more than 280,000 US firms ([version of 6 May 2026](#), p. 1). It flags 10,433 firms as adopters because they advertised roles to integrate generative AI into their workflows (p. 4). Junior employment at those firms fell by about 9% relative to non-adopters six quarters after adoption, while senior employment kept its earlier trend (pp. 5, 33). The decline came mainly through slower hiring rather than more departures; junior separation rates at adopters actually fell (pp. 1, 7). It is concentrated in the occupations most exposed to generative AI (p. 38).

Why it matters for this issue. Verification is learned largely by doing the work being verified. The person who signs a change to a bank's payment system or a plant's control logic usually spent years writing, breaking and fixing such code first. Persistent reductions in junior recruitment could weaken future assurance capacity, if they affect the relevant occupations and are not offset by training, recruitment from elsewhere or productivity gains. In open source, the [Cloud Security Alliance](#) observes that burnout falls on the maintainers who stay responsive — because they receive the most reports.

What the study does not show. Its adoption measure catches firms that hired AI integrators, not firms whose staff simply use the tools; it does not isolate software engineering; and it does not measure the chain from fewer junior hires to fewer qualified signatories. The authors note that the early timing suggests firms cut hiring in anticipation of automation, an adjustment that could prove temporary if expectations were too high. There are counter-examples: IBM says it is [expanding entry-level hiring](#) in 2026 after rewriting junior roles around analysis and oversight of AI.

CHAPTER 05

The ceiling

Safety rules do not cap how much development work agents may do; they govern what can be accepted and released, on what evidence and under whose authority. AI authorship does not by itself waive the applicable assurance objectives; the evidence and work needed to meet them depend on the development method, tool qualification and the approved assurance case.

The rules that already bind. Safety-critical software is approved on evidence. In civil aviation, DO-178C asks that code trace to requirements and back; a tool that generates code earns credit only through qualification under DO-330, and a code generator at the highest level must have **operational requirements** that define how it translates a model into source. A non-deterministic language model does not fit that template easily, and we found none qualified at that level. That does not close the door: the FAA's **AC 20-115D** treats DO-178C as an acceptable means of compliance, not the only one, and recognises supplements for tool qualification, model-based development and formal methods. In the nuclear sector, US guidance has long held, following NASA, that automatically generated safety-critical code must be inspected and tested as **rigorously as hand-written code**. The practical effect today: an agent can draft certified code, and better tools could lower the effort of meeting the objectives, but authorship alone does not lower the objectives.

How regulators have moved in 2026.

BODY	WHAT IT DID	WHAT IT IMPLIES FOR AGENT-WRITTEN CODE
UK NCSC, 18 June	Advisory guidance: calibrate oversight to consequence; for safety-critical and national-infrastructure code, review, understand, check and verify what the AI produces — another AI agent may help with that oversight	AI may assist verification; it does not replace it
CISA, BOD 26-04, 10 June	Federal civilian executive-branch agencies must remediate or mitigate vulnerabilities by risk, with three days for the most urgent class; agencies must apply its timelines from 7 December 2026 ; it cites AI-driven growth in findings	Governs fixing, not authorship
Fed, OCC, FDIC, SR 26-2, 17 April	Revised model-risk guidance; generative and agentic AI are outside its scope, and banks' own practices are to guide controls for tools it does not cover	A gap in specific guidance, not an absence of obligations; the FSB's non-binding sound practices (final, 15 October) partly fill it
EU, Cyber Resilience Act Art. 14, from 11 September	Manufacturers must give an early warning within 24 hours of becoming aware of an actively exploited vulnerability, a notification within 72 hours, and a final report within 14 days after a corrective measure is available	Reporting clocks, not installation deadlines; silent on authorship, but names who must answer, and when
US NRC, RIL 2026-06, September	Tabletop safety evaluation of an AI condition-monitoring application	AI as a component of the plant, not as the author of its code
EASA	AI rulemaking and concept papers centred on machine learning in aircraft systems	Again AI as a component; generative AI not yet covered in final rules

Regulators also face their own capacity limits. According to people familiar with the matter cited by CNBC, the Federal Reserve still lacked access to the model its banks were using to hunt flaws **as of mid-July**, while urging those banks to patch.

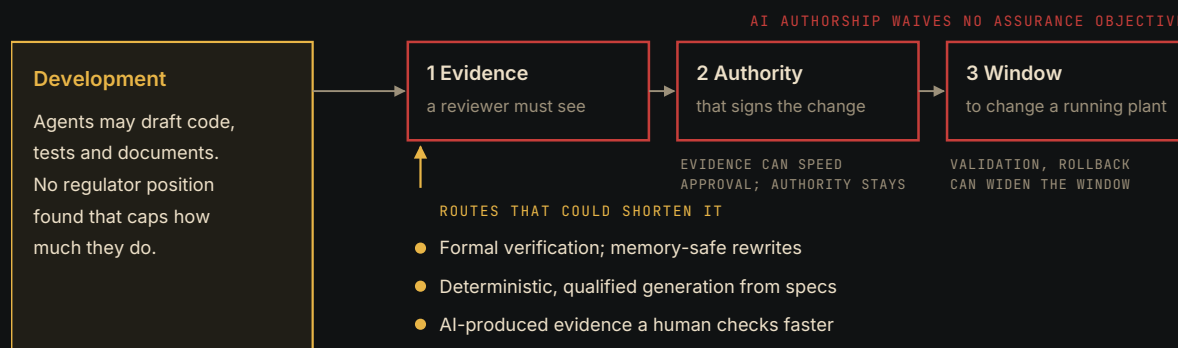
Where AI is entering the certified loop. Not through code, but through documents and checks. In a US Department of Energy demonstration with Idaho and Argonne national laboratories, Microsoft and Everstar, an AI system **converted an existing preliminary safety analysis** for a generic high-temperature gas reactor into 208 pages of sections equivalent to a licence application in one day — work the programme says takes a team four to six weeks; an expert then evaluated it for accuracy. A reviewing agent that checks such documents against NRC guidance is planned, not demonstrated, and no licence was approved. In the UAE, the Cyber Security Council and Open Innovation AI **announced on 17 September** the launch of a specialised model intended for vulnerability analysis, secure code analysis and remediation; on 2 October, **WAM reported** that Microsoft, the Council and Core42 had announced plans to deploy AI-assisted vulnerability management across government entities.

Where the constraint actually binds. In three places: the evidence a reviewer must see, the authority that signs, and — in operational technology — the window in which a running plant can be changed. Several routes could shorten the first: formal verification and memory-safe rewrites of critical components, both named by OpenAI and Anthropic as directions; deterministic, qualified generation from specifications that a model helps write; and AI-produced evidence — tests, proofs, regression hunts — that a human can check faster than the code itself. Better evidence does not transfer authority, but it can shorten the time an authorised person needs to approve; better validation and rollback can widen what fits into an operating window.

— FIGURE 06 • WHERE SAFETY RULES BIND

Rules govern acceptance, not drafting; the assurance objectives remain

Safety-critical software, 2026 • as read from DO-178C/DO-330, NRC guidance and 2026 regulator statements



OUR READING OF THE RULES CITED IN CHAPTER 5; NOT LEGAL ADVICE

We found no regulator position that sets a limit on how much of safety-critical development an agent may perform. Standards bodies have begun to draw the line this chapter draws: a draft amendment to the European railway software standard, **EN 50716:2023/prA1:2026**, proposes clarifying how its AI and machine-learning provisions apply to support tools as distinct from the software architecture of safety functions. It is a draft, not a permission.

CHAPTER 06

The mandate

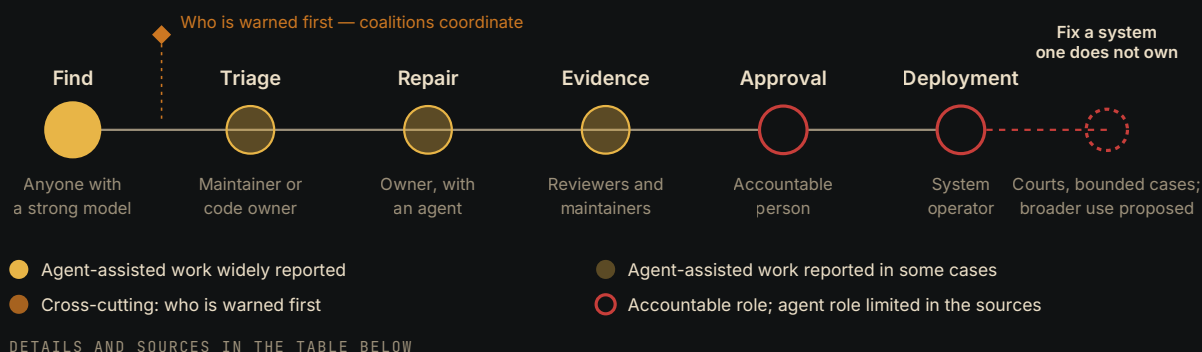
The issue's central question is not who finds a flaw but who is entitled, and obliged, to change the system it lives in. Agents now assist discovery, repair and verification; this chapter maps the responsibilities that connect that work to deployed protection. It does not establish which stages have become cheaper fastest.

Shadow 05 described the decision without a decider — public choices made by systems no official owns. Its counterpart in software is the change without a signatory: a fix that exists, may even be written by a model, but that no one with authority has verified, accepted and put into service.

— FIGURE 07 · A RESPONSIBILITY MAP

A responsibility map: who holds each stage, October 2026

The six stages of figure 4 and two questions that cut across them · not a measure of cost or effort



Who holds each stage. The six stages of chapter 2, plus two questions that cut across them.

STAGE	WHO HOLDS IT TODAY	WHAT AGENTS CHANGED
Find	Anyone with a strong model; the strongest through access programmes such as Glasswing and Daybreak	Discovery became cheap and plentiful
Triage and ownership	The maintainer or code owner	Integrated owners automate routing; elsewhere the first check falls on whoever receives the report
Repair	The maintainer or code owner, increasingly with an agent	Drafts are cheap; deep expertise can still be short, as PHP showed
Evidence and review	Reviewers and maintainers	Agents produce tests and regression hunts; humans still decide what counts
Approval and release	An accountable person and their supervisor	SR 26-2 leaves agentic AI outside its scope; banks' risk-management and governance practices should guide controls for it
Deployment	The system operator	Remediation deadlines for US civilian agencies (BOD 26-04); reporting clocks, not install deadlines, for EU manufacturers

STAGE	WHO HOLDS IT TODAY	WHAT AGENTS CHANGED
<i>Across the stages: who is warned first</i>	Under OpenAI's proposed framework, the maintainer — with a company and its coalition coordinating	Private coalitions now allocate the defender's window
<i>Across the stages: fixing a system one does not own</i>	Courts and governments in bounded cases	Broader, continuing remediation is now being proposed

The proposal on the table. Bounded remediation of other people's systems has precedent: in a court-authorised operation in January 2024, announced on 15 February, the US Justice Department **deleted malicious files from compromised routers** and reversibly changed their firewall settings. In its PHP2Shell post, OpenAI asked the security community when, if ever, governments or operators should use a vulnerability to patch systems they own or are authorised to manage after losing administrative access — and sketched an end state of continuous internet-wide scanning, with owners **notified or systems patched**. It is a question, not a policy. The newer proposal concerns broader, continuing remediation, and it puts the mandate problem in plain view.

Signatures can be forged. Two October incidents show why a mandate has to attach to verified people and evidence, not to a step in a workflow. At LiteLLM, branch protection rejected an unauthorised change and a privileged credential overrode it seconds later; a second stolen account gave the production approval. At Tensorlake, the malicious release carried a valid provenance attestation, because the project's own pipeline had built it. One control was overridden, compromised accounts satisfied another, and build provenance did not establish benign intent or code safety.

Three scenarios to 2028

These are analytical sketches, not forecasts; each is tied to signposts readers can watch.

SCENARIO	HOW IT UNFOLDS	SIGNPOSTS	WHO CARRIES THE COST
Verification as a utility	Labs, platforms and states fund triage, patch-writing and backports as shared infrastructure; maintainers are paid; patch ratios rise	Paid maintainer programmes scale; partnerships like AISLE-curl multiply; disclosure dashboards show patched share climbing; regulators fund capacity, not only deadlines	Labs and large platforms, through subsidy
Debt spiral	Discovery keeps outrunning repair; leaks shorten embargoes; end-of-life software stays exposed; commodity attackers use the same tools	Patched share flat or falling; more leaks like PHP2Shell; mass exploitation of unsupported versions; deadlines met on paper only	Operators of old systems, their users, and volunteers
Mandated patching	States or designated operators gain wider rights to scan and fix systems they do not own, at least in emergencies	Legislation or directives on emergency remediation; internet-wide scanning programmes; disputes over sovereignty and liability	Owners lose control; whoever patches gains responsibility — and liability

The scenarios are not exclusive. If current patterns hold, a mix by sector is plausible: integrated vendors and well-funded institutions move towards a utility model, while the long tail of unsupported and volunteer-maintained software risks the spiral. The third becomes plausible only after a failure large enough to make owners' consent look like a luxury.

What would settle it. We propose a stage-level test. The unit is a unique actionable finding, linked to the affected release and to the deployed estate, and classified before its outcome is known. For each, record the events — confirmation, owner acceptance, repair candidate, evidence complete, approval, release, and a defined protection threshold; the

work and the waiting — human hours, machine cost, queue time, deliberate embargo, operating-window delay, re-work; and the outcome — time and cost to protection, regressions, and exposure over a fixed follow-up window. Mitigation and retirement count as protection; documented risk acceptance is reported separately as administrative closure, because it does not by itself reduce exposure; unresolved cases stay in the data. None of the sources reviewed for this issue reports these linked measurements. Without them, no one can establish where the dominant delays occur, how they vary by ecosystem, or whether unresolved risk is rising.

HORIZON • SHADOW 08 PREVIEW

The world a model learns in

The next verification problem sits one layer up: code that builds the simulated world in which a model learns to act in the physical one.

In 2026, alumni of frontier labs raised some of the largest early rounds on record to build world models — systems that predict how an environment responds to an action, so that a robot can try options before it moves. Emulate, founded by former members of Google DeepMind’s world-model team, was reported in talks for a seed round of **up to \$700 million**; AMI Labs, co-founded by Yann LeCun, raised \$1.03 billion in March. Increasingly, the code that builds these worlds is itself written with agents.

That creates a loop this issue only glimpses: a model writes the world, another model learns in it, and the gap between the simulation and reality decides whether the robot is safe. Verifying code was hard enough. Verifying that a world is true enough to learn from is the subject of the next issue.

REGISTER

Sources

All sources consulted. Vendor and lab figures are marked as self-reported in the text. Web sources were checked between 20 and 22 October 2026.

01	OpenAI — Patch the Planet: global response coordination
02	PHP Foundation — Today's PHP security release and how we're strengthening PHP security, 21 October
03	PHP — news archive 2026: security advisory and source backports
04	Huntress — php2shell rapid response
05	Hadrian — php2shell
06	Datadog Security Labs — PHP2Shell
07	GitLab — Patch release 19.5.1, 21 October
08	Amazon Threat Intelligence — PHP2Shell exploitation before disclosure
09	OfficeChai — Cloudflare CEO on PHP2Shell attempt volumes (reproduces his post)
10	CNBC — Bank of America and Mythos remediation
11	Cognition — Itaú customer case (supplier-hosted)
12	Anthropic — coordinated vulnerability disclosure dashboard
13	Anthropic — Introducing the Anthropic Cyber Mission, 8 October
14	CSO Online — PwnEscape
15	The Register — XBOW Chrome full chain
16	XBOW — Building a Chrome full-chain exploit, 13 October
17	Google — Chrome: stronger with every update, 30 July
18	Google DeepMind — Introducing CodeMender
19	Daniel Stenberg — Mythos finds a curl vulnerability, 11 May
20	Daniel Stenberg — A curl audit by AISLE, 14 October
21	AI Cyber Challenge — final competition results (corrected denominator)
22	DARPA — AIXCC final results, 8 August 2025
23	Infosecurity Magazine — AIXCC and ARPA-H
24	CrowdStrike — agentic AI campaigns, 21 October
25	ThreatCluster — Linux security list
26	The Register — curl, better reports, more work
27	Trail of Bits — 1Password's AI patching benchmark is misleading
28	Trail of Bits — Your AI agent fixed the bug, but what else did it change?
29	1Password Off-by-1 Labs — FLAWED patch study
30	GitHub — Secure code more than three times faster with Copilot Autofix

31	Veracode — 2026 GenAI Code Security Report
32	Murphy-Hill, Butler, Savelieva — CLI coding agents at Microsoft (arXiv 2607.01418)
33	Faros — AI Productivity Paradox Report 2025
34	Arnica — AI coding agents security benchmark
35	Shafique, Miller, Heymann — A Study of the Reliability of Agentic AI-Generated Programs (arXiv 2609.18298)
36	Harness — State of Agent DLC 2026
37	Sysdig — LLMjacking, 6 May 2024
38	StepSecurity — Tensorlake npm compromise
39	Zeabur — October 2026 security incident report, 19 October
40	Zeabur — August 2026 security incident report, 7 September
41	The Hacker News — OpenCode console
42	Railway — Security disclosure, 21 October
43	LiteLLM — incident report, 21 October
44	The Hacker News — LiteLLM suspects GitHub token leak in AI gateway logs
45	The Hacker News — GhostAction
46	GitGuardian — GhostAction returns, updated 13 October
47	OX Security — MCP SDK disclosure, 15 April
48	Hosseini and Lichtinger — Generative AI as Seniority-Biased Technological Change, version of 6 May 2026
49	Cloud Security Alliance — open-source disclosure strain
50	SocialTalent — IBM entry-level hiring
51	TQL-1 qualification of a model-based code generator
52	FAA — AC 20-115D, 21 July 2017
53	US NRC — evaluation of guidance for software tools
54	US NRC — RIL 2026-06
55	UK NCSC — The vibe coding spectrum
56	Inside Government Contracts — CISA BOD 26-04
57	FedRAMP — NTC-0014 on BOD 26-04, 16 June
58	Federal Reserve — SR 26-2 attachment, model risk management guidance, 17 April
59	European Commission — CRA reporting obligations
60	TechBuzz — Fed access to Mythos (citing CNBC)
61	US DOE — AI in reactor licensing, 26 March
62	EVS — EN 50716:2023/prA1:2026 (draft)
63	Open Innovation AI — Cyber Security Council model launch, 17 September
64	Emirates 24/7 (WAM) — UAE Cyber Security Council model
65	

WAM — Microsoft Digital Defence Report and MDASH, 2 October

66 US Department of Justice — court-authorised disruption of a GRU-controlled router botnet, 15 February 2024

67 The Next Web — Emulate

MAINTAINER ADVISORIES

68 FreeBSD — Security Advisory SA-26:08.rpcsec_gss

69 Mozilla — MFSA 2026-45, Firefox 150.0.3

70 Anthropic CVD — ANT-2026-7DG11TPA, wolfSSL

71 wolfSSL — v5.9.1-stable release

REGISTER

Glossary

Evidence labels: [M] maintainer record · [V] vendor or interested party · [G] government or standards body · [R] independent research · [P] press or secondary · NR not resolved.

Verification debt The gap between a confirmed flaw and a system that is demonstrably protected: ownership settled, evidence accepted, release authorised, fix or mitigation in service. This issue's term, used as a hypothesis. Administrative closure — such as documented risk acceptance — is reported separately, because it does not by itself reduce exposure. Public data do not yet show which step dominates.

Integrated owner vs fragmented ecosystem Classified per finding, before outcomes: an integrated owner controls release and deployment for the maintained cohort concerned, as Google does for Chrome's own update channel. Control of an update channel alone does not establish control over independently administered machines. Any other case counts as fragmented.

CVE and CVSS A CVE is the public identifier for a vulnerability. A CVSS base score (0–10; version 4.0 is current) describes intrinsic severity; CVSS 4.0's Threat and Environmental metrics can add exploitation intelligence and deployment context [G]. Severity is not a count of exposed systems.

Coordinated disclosure and embargo The finder and the maintainer coordinate the timing of disclosure and remediation, usually so that a fix is available when details become public — though a fix does not always precede disclosure. For PHP2Shell, CVE identifiers leaked on 20 October and the release moved from 22 to 21 October [M].

Project Glasswing Anthropic's programme giving a limited group of organisations access to a frontier model for defensive security work. Its disclosure dashboard publishes counts for several stages of findings from Anthropic models, but not deployment; "patched upstream" counts findings, which its overview describes as released fixes and its glossary as landed patches. All figures are self-reported [V]; this issue was prepared with an Anthropic model and treats them accordingly.

Patch the Planet OpenAI's programme in which partners — Trail of Bits, Calif and others — direct agents to find flaws and draft fixes, while maintainers decide what is merged. Partner-reported acceptance and regression results [V] and 1Password's separate benchmark [V] use different tasks and conditions; their percentages are not directly comparable.

Separate counts, not a cohort Each stage of the dashboard counts its own items as of the snapshot date. A finding confirmed in September may be reported in October, so dividing one stage by another does not give a conversion rate. The 516 findings marked patched upstream belong to the 6,157 reports, across both routes.

Backport and end of life A backport applies a fix to an older release line. For PHP2Shell the project issued regular releases for 8.2–8.5 and php.net lists project-provided minimal source backports for 7.4, 8.0 and 8.1; users of earlier versions must rely on distributor or LTS support, or upgrade [M].

Provenance attestation A signed record of where and how a package was built — for npm, which repository and which workflow produced it. It attests a build relationship, not benign code: Tensorlake's malicious release carried a valid one, because the project's own pipeline built it.

MCP (agent-tool protocol) The Model Context Protocol, an open standard for connecting AI agents to tools and data. Zeabur's August incident began with a known flaw (CVE-2026-7644) in the MCP functionality of a NextChat service a customer had deployed [M]. OX Security disclosed in April an SDK issue it describes as systemic; Anthropic, by OX's account, considers the behaviour expected [V]. Downloads are not installations.

Environment variables Settings passed to running software, commonly including API keys, database passwords and registry tokens. Deployment platforms store them for their customers, which makes those stores a concentrated target; the Zeabur and Railway disclosures concerned these records.

Branch protection and approval gates Rules in a code host or release pipeline that can block changes to protected branches and require reviews, checks or a second approver before release. They verify that an authorised account acted, not who was behind it. At LiteLLM branch protection rejected an unauthorised change, a privileged credential overrode it five seconds later, and a second stolen account approved the release [M].

DO-178C, DO-330 and AC 20-115D DO-178C is the standard civil aviation authorities use to approve airborne software: requirements, traceability and verification evidence scaled to how dangerous a failure would be. DO-330 covers tool qualification; a code generator at the highest level (TQL-1) needs its own. The FAA's AC 20-115D treats DO-178C as an acceptable, not the only, means of compliance and recognises supplements for tool qualification, model-based development and formal methods [G].

Assurance objectives and the assurance case Objectives are what a safety standard requires to be shown — that requirements are met, that code traces to them, that testing is adequate. The assurance case is the argument and evidence that they are met. AI authorship does not waive the objectives; the evidence and effort needed depend on the method and tools.

CISA BOD 26-04 A binding operational directive of the US Cybersecurity and Infrastructure Security Agency (10 June 2026) for federal civilian executive-branch agencies. It sets risk-based deadlines to remediate or mitigate vulnerabilities — three days for the most urgent class — with agencies applying its timelines from 7 December 2026 [G]. It governs fixing, not who wrote the code.

EU Cyber Resilience Act, Article 14 From 11 September 2026, manufacturers must give an early warning within 24 hours of becoming aware of an actively exploited vulnerability, a notification within 72 hours, and a final report within 14 days after a corrective measure is available [G]. These are reporting clocks, not deadlines for customers to install a patch.

SR 26-2 (US model-risk guidance) Model-risk guidance issued on 17 April 2026 by the Federal Reserve, FDIC and OCC. Its footnote 3 puts generative and agentic AI outside its scope and says banks' own practices should guide governance and controls for tools it does not cover [G]. A gap in specific guidance, not an absence of obligations.

Operational technology Control systems that run physical processes — plants, grids, pipelines. A fix may have to wait for a maintenance window or for equipment replacement, because the process cannot simply be restarted.

AIxCC DARPA's AI Cyber Challenge. In the 2025 final, after the denominator was corrected from 70 to 63, finalist systems found 54 injected bugs (86%) and patched 43 (68%); separately they found 18 real bugs and provided 11 patches [G] — patches provided, not accepted or deployed.

Not an allegation against any named body. Nothing here constitutes legal, investment or security advice. Statements of fact are limited to matters recorded in the public sources cited and are reports of what those sources state. Prepared independently and without external funding by the Foresight Future Institute. AI assistance and adversarial review are disclosed in the Interpretation Lock.