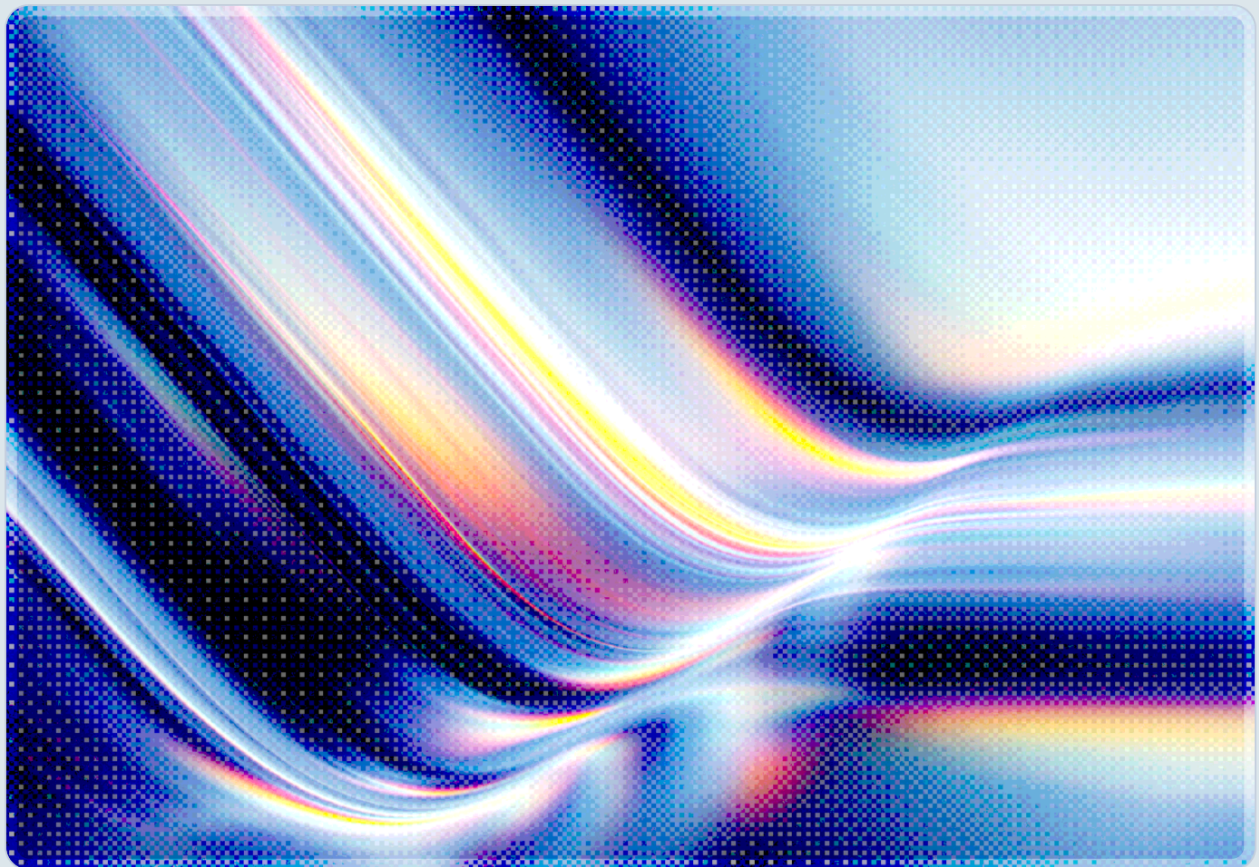


Vulnerability Disclosure Under Strain: Google OSS VRP Pause

AI-Generated Report Floods and the Economics of Open-Source
Bug Bounties

2026-10-09

 AI-assisted Rapid Research



CSAI

cloud
CSA 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.

Key Takeaways

Google paused new product vulnerability submissions to its Open Source Software Vulnerability Rewards Program (OSS VRP) effective October 1, 2026, citing "a significant rise in automated submissions, the vast majority of which are not valid" [1][2]. The pause follows curl's decision to end its bug bounty in January 2026 [4]. Intel also moved to a disclosure program without bounties in September 2026, though it has not stated a reason [8]. The Google and curl cases suggest that programs which pay for reports are encountering a cost problem: AI tooling appears to have lowered the price of producing a plausible submission, while the price of verifying one has not fallen.

The strain does not come from a single source. The evidence points to at least two distinct phenomena that are easy to conflate: unverified, hallucinated reports that consume triage time without containing a real flaw, and accurate but high-volume or duplicated reports from capable AI scanners that exceed maintainer capacity to review and fix [6][7]. The two call for different responses, and policies designed only for the first can penalize the second.

Organizations that run disclosure programs, and those that depend on open-source software, should treat intake quality as a control with its own design requirements. Over the near term, the practical measures are to require reproducible evidence, assign human accountability for AI-assisted submissions, and decide explicitly whether financial rewards are the right incentive for the report categories a program receives. Over the longer term, the industry lacks a shared standard for how automated agents act as disclosing parties, a gap CSA has previously identified [9].

Background

Google's OSS VRP launched in August 2022 and pays rewards for security flaws that affect the open-source software supply chain, including projects such as Golang, Angular, Bazel, Protocol Buffers, and Fuchsia [1]. According to BleepingComputer's reporting, reward amounts have ranged from \$100 to \$31,337 [1]. The same report states that Google has awarded more than \$81.6 million across its vulnerability reward programs since 2010 and paid \$17.1 million to about 700 researchers in 2025 [1]. These figures indicate that the pause affects one component of a large and long-established bounty ecosystem, not the ecosystem as a whole.

The announcement states that Google is temporarily no longer accepting product vulnerabilities submitted to the OSS VRP, and it commits to providing an update in the first quarter of 2027 [1][2]. Several details bound the scope of the change. Supply chain vulnerability reports remain in scope, reports submitted before October 1 are reported to continue to be handled, and researchers are pointed toward alternatives such as the Cloud VRP and the Patch Rewards Program [1][2]. Google's stated reason is the volume of automated submissions, most of which it describes as invalid [1][2][3]. Reporting on the announcement indicates it was posted by the @googlevrp account on X and that Google's OSS VRP rules page carries the notice [2]. We could not retrieve Google's original notice at the time of writing; the Google language quoted here is drawn from BleepingComputer [1] and Help Net Security [2], which reproduce it.

The Google decision is the largest by vendor scale of the three cases reviewed here. The curl project announced that its HackerOne-hosted bounty, which had run since 2019, would end on January 31, 2026, with security reports moving to GitHub from February 1 [4]. Daniel Stenberg, curl's lead maintainer, stated that the main goal was "to remove the incentive for people to submit crap and non-well researched reports to us" [4]. In the first 21 days of 2026 the project received 20 submissions, seven of them within a single 16-hour window [4][5]. HackerOne responded that curl's circumstances are not representative of typical enterprise programs, pointing to managed triage for its enterprise clients [4].

Intel's change, reported in September 2026, replaced a bounty program that had paid between \$500 and \$100,000 with a new Intigriti-hosted program described as "a responsible disclosure program without bounties" [8]. Intel has not, per the reporting we reviewed, stated a reason, so any link to AI-generated submissions is an inference by commentators and not an established fact [8]. We include it as a comparable data point on bounty withdrawal while flagging that its causes are unconfirmed.

Security Analysis

Why the economics have shifted

Bounty programs rely on an implicit balance: the cost of finding and writing up a vulnerability is high enough to deter casual submissions, and the reward compensates researchers whose reports are accurate. Large language models change the first half of that balance. A report can now be produced in minutes (CSA's observation, not a measured figure) that has the format, vocabulary, and apparent rigor of a legitimate finding. Coverage of the Google pause describes the problem as speculative, duplicated, or hallucinated findings [3]. By way of illustration, such a report might assert an exploit path that does not exist or a vulnerable function that is not reachable; this example is ours and is not drawn from the cited coverage. Verifying such a claim requires a maintainer or security engineer to read the code and

attempt to reproduce the behavior, which suggests that the cost of rejecting a bad report may approach the cost of accepting a good one, although none of the programs discussed has published triage-cost data.

The curl experience illustrates the asymmetry. A Bugcrowd opinion piece reports that curl's program paid \$86,000 across 78 confirmed vulnerabilities over six years, and attributes to Stenberg the observation that, among AI-only reports he monitored, not a single one discovered a genuine vulnerability [5]. Both the payout figures and that observation are an advocate's characterization and not audited statistics, but they are consistent with Stenberg's public statements that the program's incentive structure was drawing low-effort reports [4]. The same piece also notes that a researcher using AI as a research assistant, with human validation, produced roughly 50 real bugs in 2025 [5]. On this single contrast, the distinguishing factor appears to be the presence or absence of human verification and not whether AI was used, though one case is thin grounds for a general conclusion.

Two different problems

The Linux kernel provides a contrasting case that complicates a simple "AI slop" narrative. LWN reported that March 2026 saw 171 CVEs assigned for the kernel, compared with 191 in February and 64 in January, and that a security team member noted the team had needed to bring on more maintainers to handle the increase in useful reports [6]. LWN also quoted Anthropic's Nicholas Carlini describing a backlog of more than 500 potentially exploitable kernel crashes awaiting human validation, since model output still contains false positives [6]. In this case the stress comes largely from valid findings arriving faster than a volunteer-staffed process can absorb them.

A secondary account, based on reporting by The Register, attributes to Linus Torvalds a characterization of the kernel security mailing list as close to unmanageable as multiple researchers running similar automated scanners submitted duplicate reports of the same issues, often without patches [7]. The response described in that account was procedural: updated kernel documentation that tightens the definition of a security bug and asks reporters to supply a patch with their disclosure [7]. We have not independently confirmed the exact report volumes or the wording of the documentation change, and readers should consult primary kernel documentation before relying on the specifics. The pattern is nonetheless instructive, because duplication and absent patches are failures of coordination and not of accuracy.

Failure mode	Typical characteristics	Where observed	Primary cost to defenders
Unverified or hallucinated reports	Speculative or hallucinated claims, no reproduction	curl, Google OSS VRP [3][4][5]	Triage time spent disproving claims
Duplicate valid findings	Same issue found by multiple scanners, often without a fix	Linux kernel security list (secondary reporting) [7]	Deduplication and coordination effort
High-volume valid findings	Real vulnerabilities exceeding review and patch capacity	Linux kernel CVE volume [6]	Validation backlog, maintainer burnout
Reward-driven submission	Reports optimized for payout, not for correctness (maintainer's characterization)	curl, per Stenberg's stated rationale [4]	Incentive distortion

Implications for open-source supply chains

These events matter beyond the programs involved because open-source maintainers are often volunteers or small teams, and a disclosure channel that is saturated degrades for everyone, including the researchers who submit accurate findings. If maintainers respond by closing channels, raising evidence requirements, or removing rewards, genuine reporters face higher friction. The risk is that legitimate reports are delayed or discouraged, so that the net effect of an AI-generated flood is a reduction in the rate at which real vulnerabilities are surfaced and fixed. This is an inference from the observed responses and not a measured outcome.

There is also a defensive-use dimension. The Malwarebytes coverage suggests a possible mitigation in which AI is used to validate submissions first, forwarding only credible reports to engineering teams [3]. This is a proposal and not an announced Google measure. It carries its own risk, since an automated filter that rejects reports could silently discard valid findings, and CSA has separately warned about intake processes that create "silent bounties" where findings are consumed without accountable human review [9]. Any automated triage should therefore be paired with sampling and audit of rejected reports.

Finally, the pause arrives alongside a broader capacity mismatch that CSA has analyzed in its work on patch velocity and disclosure. That work argues that the pressure comes from discovery outpacing the ability to triage and remediate, and that the crisis is one of coordination more than uniform exposure [9][10][11]. The Google pause can be read as one symptom of that mismatch, in which the intake end of the pipeline is saturated before the remediation end is reached.

Recommendations

Immediate Actions

Organizations that operate vulnerability disclosure or bounty programs should review their intake data now, measuring the share of reports that are reproducible, duplicated, or invalid, and the analyst time each category consumes. Without these figures, decisions about whether to keep, restrict, or end rewards rest on anecdote. Programs should also require a minimal proof of impact, such as a working reproduction or a failing test, before a report enters the human queue, and should state plainly in their policy how AI-assisted submissions are treated.

Security teams that consume open-source software should recognize that the Google pause does not remove the underlying vulnerabilities in the affected projects. They should confirm which upstream projects they depend on are covered by affected programs, identify alternative channels for reporting issues they discover, and avoid the assumption that reduced bounty activity indicates reduced risk. Teams that submit reports themselves should apply human verification to AI-assisted findings before submission and, where possible, include a proposed patch, since both practices lower the burden on maintainers [5][7].

Short-Term Mitigations

Program owners should consider tiered intake in which reporters with a track record of validated findings receive a faster path, while new or unverified reporters submit through a lightweight pre-screening step. Platform features that score reporter reputation can support this, though they should be evaluated for fairness to new researchers. Maintainers facing duplicate floods can publish known-issue lists and remediation status so that scanner operators can check before submitting, which addresses the duplication pattern described in secondary reporting on the kernel case [7].

Where programs retain rewards, they should examine whether payout structure encourages volume over accuracy. Options that appear in the public reporting include removing monetary rewards, as curl and Intel did [4][8], or conditioning rewards on a confirmed, patched outcome. Each choice has trade-offs:

removing rewards may reduce noise but may also reduce engagement from skilled researchers, and the available evidence does not yet show how that balance resolves. Organizations should plan to monitor the effect on the quality and quantity of reports after any change.

For organizations whose staff or vendors use AI tools to find vulnerabilities, internal policy should assign a named human as accountable for each submitted report. This mirrors the recommendation in CSA's earlier work on autonomous bug hunting that human operators be designated as accountable for AI-generated reports [9].

Strategic Considerations

The longer-term question is whether coordinated vulnerability disclosure norms, which were designed around individual researchers working at human speed, can accommodate automated agents as disclosing parties. CSA's prior analysis calls for updated standards that address autonomous agents explicitly, define operator accountability, and set expectations for machine-volume batches [9]. Standards bodies, platform operators, and large vendors are best placed to coordinate here, and the Google decision to revise its program before reopening it, with an update promised for the first quarter of 2027, offers an opportunity to observe one major operator's approach [1][2].

Funding and capacity matter as much as process. If AI-assisted discovery produces valid findings at higher rates, as the kernel experience suggests [6], the constraint shifts to the humans who validate and patch. In CSA's view, organizations that benefit from open-source software have an interest in funding triage and maintenance capacity, including through shared clearinghouses or programs that provide maintainers with access to the same AI tools used to find flaws. The evidence reviewed here does not establish which model works best, and we recommend that organizations evaluate options against their own dependency risk.

CSA Resource Alignment

CSA's research note *Reforming Coordinated Vulnerability Disclosure for the Autonomous Bug Hunter Era* [9] is the closest prior CSA work on this subject. It argues that CVD norms built for human-speed individual researchers do not address accountability when an AI agent submits findings, how machine-volume batches should be handled, or what happens to reports that never reach public disclosure. Its immediate recommendation that human operators be accountable for AI-generated reports, and its warning about "silent bounties," correspond to the intake-quality and verification measures recommended above. The OSS VRP pause and the curl and Intel changes are consistent with the concerns that note raised.

Project Glasswing and the AI Vulnerability Disclosure Velocity Crisis [10] examines the volume of AI-discovered vulnerabilities in systemically important open-source software and the disclosure infrastructure's limited capacity to absorb it. It complements this note by describing the valid-finding side of the problem, which parallels the kernel experience. CSA's whitepaper *The Bugpocalypse Threshold* [11] addresses the broader mismatch between accelerating discovery and roughly constant patch capacity, and provides context for why intake saturation may affect remediation timelines downstream.

For control mapping, the AI Controls Matrix (AICM) v1.1 [12] provides the relevant Threat and Vulnerability Management (TVM) and Application and Interface Security (AIS) domains. Organizations that run or rely on disclosure programs can use the TVM controls to document how external vulnerability reports are received, triaged, and remediated, and can align their intake policy for AI-assisted submissions with those controls.

References

- [1] BleepingComputer. "[Google halts open-source bug bounty program amid AI spam surge.](#)" BleepingComputer, October 5, 2026.
- [2] Help Net Security. "[AI slop submissions force Google to freeze its open-source bug bounty.](#)" Help Net Security, October 5, 2026.
- [3] Malwarebytes. "[Google pauses open source bug bounty program after rise in AI submissions.](#)" Malwarebytes Labs, October 5, 2026.
- [4] BleepingComputer. "[curl ending bug bounty program after flood of AI slop reports.](#)" BleepingComputer, January 22, 2026.
- [5] Bugcrowd. "[Hacker opinion piece: How lazy hacking killed curl's bug bounty.](#)" Bugcrowd Blog, 2026 (exact date not shown on the page).
- [6] LWN.net. "[A flood of useful security reports \(LWN Weekly Edition, April 16, 2026\).](#)" LWN.net, April 2026.
- [7] AI Weekly. "[Linus Torvalds: AI bug tools broke Linux security list.](#)" AI Weekly, May 18, 2026 (summarizing reporting by The Register).
- [8] Phoronix. "[Intel Appears To End Its Bug Bounty Program.](#)" Phoronix, September 2026.
- [9] Cloud Security Alliance AI Safety Initiative. "[Reforming Coordinated Vulnerability Disclosure for the Autonomous Bug Hunter Era.](#)" CSA Labs, June 7, 2026.
- [10] Cloud Security Alliance AI Safety Initiative. "[Project Glasswing and the AI Vulnerability Disclosure Velocity Crisis.](#)" CSA Labs, May 24, 2026.
- [11] Cloud Security Alliance AI Safety Initiative. "[The Bugpocalypse Threshold.](#)" CSA Labs, May 2026.
- [12] Cloud Security Alliance. "[AI Controls Matrix v1.1.](#)" Cloud Security Alliance, 2026.