A(I) future of Bug Bounty

BY CHRIS HOLT · APRIL 10, 2026 · LAST UPDATED ON APRIL 22, 2026

What you will learn

- How AI is changing bug bounty
- Where AI helps security teams
- Why human hackers matter
- What the future of bug bounty looks like

AI and all the tools built around related technologies have been working their way into the Bug Bounty community for a little over a year now and by around March 2025 we started seeing notably AI-written reports. It is time to take stock of what impact they have wrought already so we can look to the future and begin to address the reality and some of the fears surrounding this new technology.

This article is part of a series bringing together different perspectives on how AI is reshaping bug bounty and vulnerability disclosure. To reflect the reality of how quickly this space is evolving, we spoke with our colleagues across the company who encounter these shifts daily in their work, from program operations to triage and researcher engagement.

For this first perspective, we turn to Chris Holt, who sits at the intersection of researcher community, program operations, and the real-world mechanics that make responsible disclosure work at scale. For two decades, Chris has worked across software engineering, penetration testing, application security, and finally in bug bounty program management, building and running large enterprise programs.

Today, Chris operates as a strategic partner for customers: less interested in “more reports” as an outcome, and more interested in a healthy pipeline: clear proof, standards, sustainable triage, and incentives that keep skilled researchers engaged and coming back. In practice, that often looks like engagement ideas, experiments, and community-driven levers (events, campaigns, feedback loops) designed to improve signal-to-noise and produce repeatable value for both customers and researchers.

That lens is also why Chris is so interested in the “AI era” question. When the cost of generating findings drops, the bottleneck shifts to validation, prioritization, and trust. The future of bug bounty, if it is done well, won’t be defined by volume. It will be defined by disciplined operations, stronger proof, and platforms and programs that scale without burning out the humans on either side of the marketplace.

Introduction

> **“Bug bounty is not a tool. It is a market mechanism.”**

For the last decade, bug bounty has been one of the most effective “pay for outcomes” models in security. It turns uncertainty into a measurable pipeline.

- Organizations externalize discovery to a diverse pool of motivated experts.

- Researchers apply creativity and persistence that do not fit neatly into traditional security testing.
- Everyone gets a clear unit of value: a validated vulnerability with enough context to fix it.

Now AI is changing the economics of software, the volume of findings, and the expectations of speed. The obvious question that keeps coming up at every conference:

> **"Will AI harm the responsible disclosure economy built around human ingenuity?"**

Perhaps a more useful answer is:

AI will force researchers, VDP, and bug bounty to evolve.

Below is a practical view of how that evolution may happen.

The core value of bug bounty survives

AI will make finding "something" easier. It will not make knowing what matters trivial.

Bug bounty's persistent advantage is that it is anchored to real-world exploitation and real-world risk.

- Truth over theory: Many security tools produce "could be vulnerable" outputs. Bug bounty pays for what is demonstrably vulnerable.
- Incentives create effort: When risk and reward align, people push further than "best effort." That effort is where high-impact bugs come from.
- Edge cases remain human territory: Novel business logic flaws, complex chains, and contextual abuse paths are still hard to automate end-to-end.

In other words, AI vulnerability discovery increases the baseline capability, but it does not remove the need for an outcome-based, adversarial proof model.

We have been here before. AI raises the baseline, just like scanners did, but at a new scale. Vulnerability scanners and modern SAST did not kill pentesting or bug bounty. They raised the floor. The shift was not "automation replaces expertise." The shift was that more people could generate plausible leads, while the work that matters (validation, threat-model fit, impact analysis, and clean reproduction) stayed stubbornly dependent on methodology.

AI follows the same arc, but it does so with far more parallelism. There are now dozens of frontier and commodity models, hundreds of wrappers and agents, and a large population of people running them against the same technologies. The result is predictable: the baseline goes up, but the top-of-funnel widens faster than most programs can absorb, and the duplication surface area explodes.

That is why the early signal looks messy. Many AI-assisted submissions will be "tool output" dressed as a report: partially correct, often missing context, and frequently indistinguishable from a thousand near-identical variants.

Meanwhile, the researchers who actually know what they are doing will use the same tools as pair programming/hacking assistants: they will tune the prompts, verify the behavior, fill the gaps with their own testing, and only submit once the report clears the bar for truth and exploitability.

The historical pattern remains consistent:

1. A new capability becomes widely available.
2. The industry initially panics about signal-to-noise.
3. Processes adapt, and new categories can be created.
4. Expectations change.
5. The work shifts up the stack.

AI accelerates step 1 and dramatically increases step 2.

We are already starting to see this play out. Next, we probably will see AI tools begin effectively to automate the discovery of entire classes of some vulnerabilities on certain technologies. The immediate impact will be a higher volume of low-to-medium severity reports, plus a wave of AI-assisted duplicates. The long-term impact is that programs that adapt will get stronger faster.

The closest shark

The biggest near-term risk is not “AI finds more bugs.” It is operational overload.

AI-assisted submission volume behaves like a runaway inflation spiral: once the marginal cost of producing reports drops, volume can accelerate faster than triage and remediation systems can respond…until capacity and trust become the constraint.

Most programs are not designed for a world where submissions are cheap. Which means the constraint moves to triage capacity, proof requirements, remediation bandwidth, and internal decision-making speed.

So the question is not “How do we stop AI?”

The question is:

> **“How do we keep the pipeline healthy when the top-of-funnel has explosive growth in its width?”**

The winning response I give: intelligent friction and stronger proof-of-concept standards.

If programs do nothing, AI will flood them. The fix is not a “deny list vulnerability classes.” That approach worked poorly even in the scanner era, and it will fail harder with AI-assisted reporting.

Instead, programs will add intelligent friction:

- Higher-quality PoC requirements for certain categories.
- Clear reproduction standards that force the report to be actionable immediately.
- Better evidence expectations like short videos, accurate impact narratives, and environment-specific steps.

The goal is not to make reporting painful. The goal is to make low-effort, low-signal submissions uneconomic. This is how you preserve researcher time, customer time, and program credibility.

You get an AI, and you get an AI

If researchers can use AI to generate reports, platforms and programs must use AI to manage the workflow too, especially in triage and PSIRT-style workflows. Not to replace humans, but to keep humans focused on judgment and decision-making; the areas where human synthesis vastly outperforms algorithms.

Key areas where AI should be expected to land:

- Deduplication and clustering: Group near-identical issues across assets and report variants.
- Automated context enrichment: Pull logs, code references, endpoint ownership, and change history into the triage view.
- Risk scoring that adapts: Not static severity labels, but exploitability and business impact signals.
- Smart routing: Send the right class of issue to the right reviewer.
- Pre-fix guidance: Provide remediation suggestions and validation steps to speed time-to-fix.

The competitive edge will be less about who can accept the most reports and more about who can turn valid reports into fixes in a timely fashion without burning out everyone involved.

How does bug bounty evolve?

Segmentation

AI will widen the gap between program models. Some organizations will optimize for broad coverage and accept higher volume, supporting more good-faith reporting. Others will choose a more selective model with dedicated communities specializing in certain technology classes or proprietary customer tech.

The “one size fits all” approach becomes less viable than it already is. Expect more segmentation to appear, such as:

- Volume reporting programs: lower payout, higher volume, strong automation, stricter PoC
- Expert programs: higher payout, lower volume, more collaboration

Red Teaming & Penetration Testing

Red teaming and bug bounty will continue to converge, but not merge.

As AI drives down the cost of finding broad classes of vulnerabilities, security leaders will do what they always do when a capability becomes cheaper and more commoditized: they will re-audit the spend. Some of that budget pressure will land on traditional penetration testing, pushing it further toward compliance-shaped engagements and away from open-ended assurance.

But the underlying need does not go away. Organizations still need evidence grounded in real attacker behavior that products and services hold up under modern threat models. That is where red team exercises and bug bounty continue to earn their place: both are designed to simulate adversaries, surface practical risk, and produce outcomes that map to how systems are actually broken. There is a real

possibility that some organizations will choose either red teaming or bug bounty for certain threat coverage, especially as budgets compress.

However, the three activities still produce different value:

- Red teaming is objective-driven, time-bound, limited by knowledge and capability.
- Penetration testing is (already too often conflated with red teaming), constrained, scenario-driven, compliance-oriented, and time-bound.
- Bug bounty is open-ended, collaborative, persistent, and incentive-driven.

AI will compress some of the differences, but it will not erase them. What will change is the expectation that bug bounty can, and will begin to, produce more continuous, campaign-like outcomes, especially when paired with strong program design and tooling.

The next era

Researcher skill does not become irrelevant. It becomes more leveraged. The “AI replaces researchers” narrative misses how real work happens.

Even when AI is involved:

- Tools still need tailoring and tuning.
- Quality still depends heavily on what the operator knows.
- The best results come from people who can:
  - define the right questions,
  - recognize nonsense,
  - and chain findings into real impact.

AI makes great researchers faster. It also makes mediocre reporting cheaper. Program design must reward the first outcome and discourage the second. Then a modern bug bounty program will look less like an inbox and more like a pipeline.

“Good” will mean:

- Clear requirements for impact and reproduction.
- Fast, consistent triage decisions.
- Automation that reduces busywork.
- Remediation that is measured and actively managed.
- Researcher experience that stays fair, predictable, and respectful.

And the most mature programs will treat AI as normal; not as a threat to block, but as an amplifier to harness.

Conclusion

Bug bounty does not disappear; it becomes more disciplined. AI will not destroy bug bounty. It will expose which programs were relying on goodwill and manual heroics.

The programs that survive and thrive will raise proof standards, add intelligent friction, use AI for workflow scaling, and keep the incentive model aligned with real outcomes.

Bug bounty remains one of the few security mechanisms that reliably converts adversarial effort into measurable improvement.

That is not going away.

If you disagree with anything I've said here, good. That tells me that you are paying attention to what is happening in the world and in our community and are passionate about what you see. So let's debate it. These are conversations we need to have if we want to stay ahead of where this is going.

AUTHOR

Chris Holt

Chris Holt is a seasoned bug bounty program leader with over 15 years of experience in application and product security. Certified by GAIC, NTISSI, Guinness, PADI, and the USSF, he has spent the last 8 years building and scaling some of the industry's most respected bug bounty programs. His expertise spans the full spectrum of vulnerability management—from offensive security and researcher engagement to program operations and strategic growth.