EXECUTIVE SUMMARY

Same Code, Same Access, 3x the Findings

A whitebox benchmark of AI security tooling on AI-generated applications

zauth research | July 2026 | zauth.inc

The assumption we tested

A hundred thousand applications ship every day on AI development platforms like Lovable, Replit, and Bolt, built by founders, designers, and students who are not security engineers. They handle real users, real data, and real money, and almost none of them have ever been security tested.

Whitebox testing is sold as the fix. In a whitebox assessment the scanner is handed the application's entire source code, every file, every database rule, every server function, and asked to find what is exploitable. It is the most generous starting position possible, the opposite of how a real attacker works from the outside, and it is what security vendors charge the most to provide. The premise is simple: if a tool can read all of the code, it should find all of the holes.

We tested whether that premise holds. We took ten production applications built on Lovable and scanned each one with Vector, zauth's agentic security scanner, in whitebox mode with full source access. Then we scanned the same ten with Aikido, with the same full access to the same code.

Aikido is not a fringe tool. It is a security company backed at a billion-dollar valuation, and it is the pentesting engine Lovable chose to build directly into its own platform, the default its developers are handed when they want their app checked. If any whitebox scanner should catch what these applications are shipping, it is the one the platform itself picked.

Same applications. Same source. Same access. The results were not close.

The results

	Total findings	Critical	High	Medium	Low
Vector	227	18	45	147	17
Aikido	71	4	13	28	26

Vector found 227 vulnerabilities to Aikido's 71, more than three times as many, on identical code with identical access. Counting only findings that represent real application risk and setting aside low-severity configuration hygiene, Vector found 210 to Aikido's 45, a 4.7x gap.

What 227 looks like

These are not linter warnings. Across ten apps, Vector surfaced exposed credentials, disabled access controls on live databases, unauthenticated admin panels, and broken authorization across nearly every table and endpoint. One crypto betting platform alone carried 11 critical findings, including hardcoded admin credentials, encrypted Solana private keys exposed to anyone, and an unauthenticated path straight to draining the treasury.

Why the gap

This is not Vector burying Aikido in low-severity noise. Vector found more of what matters most. On the same code, it surfaced more than four times the critical vulnerabilities and more than three times the high-severity ones. Below that, it mapped the full population of authorization gaps, missing access controls, and unvalidated inputs across every table, endpoint, and function, where Aikido reported a fraction. On the invoicing app alone, Vector found eleven distinct authorization and validation flaws to Aikido's two.

Both tools read the same code. Source access put every line in front of each of them. Finding what is exploitable in it is a different problem, and Vector is far better at it.

What we found across the ten

The pattern repeated across the set. On nine of the ten applications, Vector found critical or high severity flaws that Aikido did not report, even though both tools held the same source. These were not obscure edge cases. They included a login system where a user's public wallet address doubled as their password, medical records any doctor could quietly reassign to another patient, invoicing tenants that turned out to share a single admin boundary, and admin panels mounted with no authentication at all.

In several cases Aikido was reading the exact file that carried the flaw and stopped one step short, reporting a smaller issue in the same function while missing the one that mattered. The consistency is the point. This was not one unlucky application or one lucky find. Given identical access to ten codebases, one scanner surfaced what was exploitable and the other did not.

Methodology

Every finding was normalized to root cause and scored under CVSS v3.1 with strict banding, applied identically to both tools. Aikido's reports contain no CVSS vectors, so we assigned one to each of their findings from their own published evidence, using stricter and more fully documented scoring than our prior benchmark. Every assigned vector is published in the full study for independent verification.

The takeaway

Whitebox testing works, and full source access is exactly the advantage it is sold as. But an advantage is only worth what the tool can do with it. Given the same code and the same access as a billion-dollar competitor, Vector found more than three times the vulnerabilities, and nearly five times as many that actually matter. The access was identical. The scanner was not.

The full benchmark, with every finding on every application, comes next.

team@zauth.inc