aikido

Whitepaper Report

Autonomous vs. Manual Pentesting

This report benchmarks Aikido Attack (autonomous AI pentesting) against external manual pentests across four web applications. Across the cases, AI tests ran substantially faster and surfaced more deep application-logic vulnerabilities (e.g., IDORs and auth bypasses), while human testers contributed most on configuration hardening and compliance-oriented findings, but missed critical vulnerabilities under time constraints.

The results highlight an emerging access asymmetry: source-code context is costly for humans to fully exploit, but immediately increases AI effectiveness- pointing to a shift from Greybox constraints toward more Whitebox-level testing by default.

The Gist

We ran a head-to-head comparison between **Aikido Attack (Autonomous Pentests)** and external **Traditional Human Pentest (Manual)** on four different web applications.

The Verdict: The AI solution was drastically faster and found deeper logic flaws– like IDORs– due to source code access. The human testers focused heavily on **compliance and configuration standards,** but missed several critical exploits identified by the AI due to time constraints and lack of code visibility.

The Setup

Real-world conditions were prioritized over a scientific control group to reflect how these tools are actually deployed:

- **Aikido AI (Whitebox):** Autonomous, but with full access to the source code.
- **Aikido AI (Blackbox):** Autonomous, but **without** access to the source code.
- **Human Testers (Greybox):** Authenticated user access, but no source code visibility (standard for external engagements due to logistics/NDAs).

Key Concept: Access Asymmetry

In today's pentesting landscape, **Greybox** testing is the norm because it offers the best **compromise between coverage and cost.** While giving an AI tool access to code is instant, the effort for human testers to understand and review a full codebase makes Whitebox prohibitively expensive in most manual engagements.

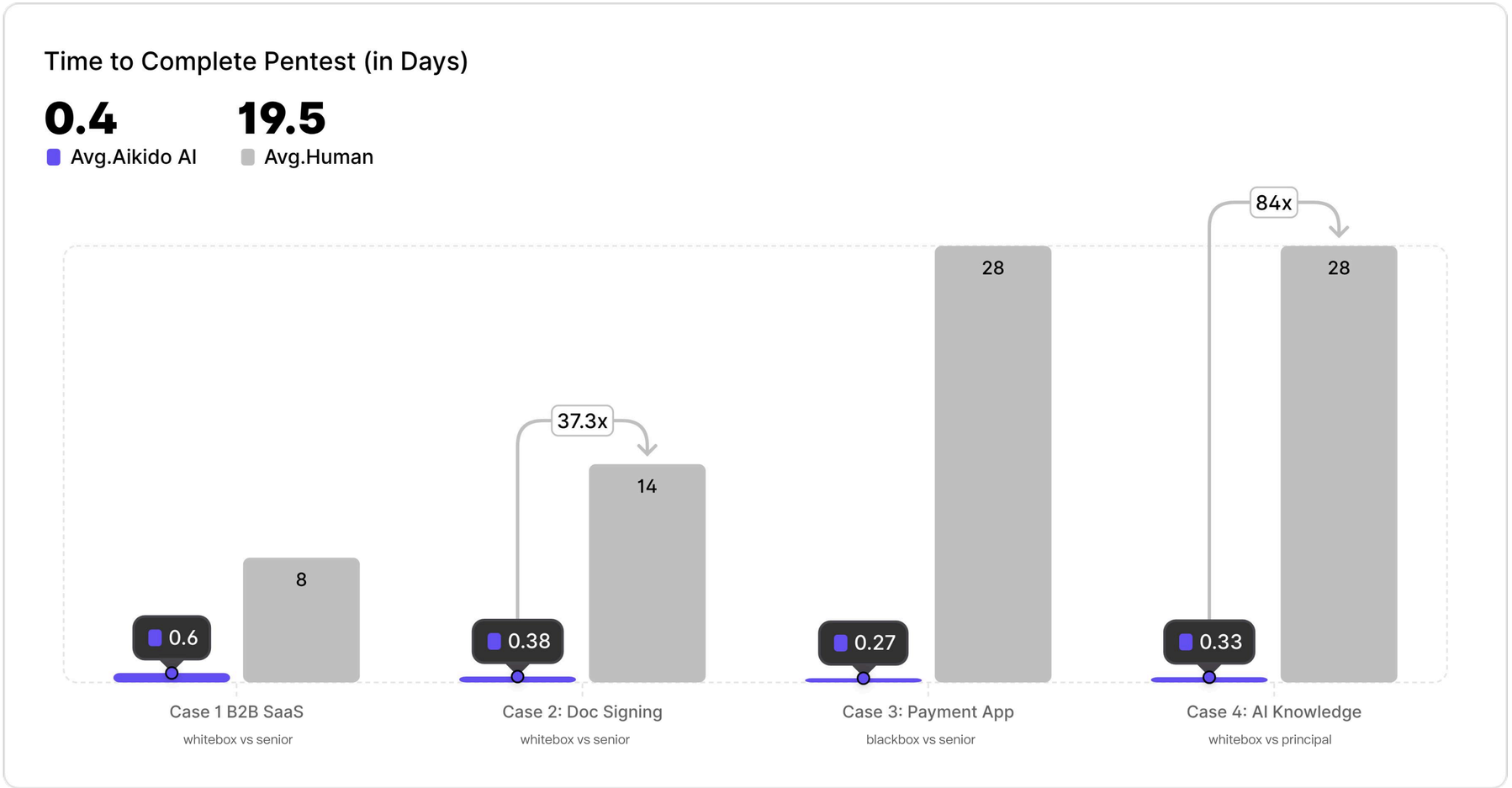
This "asymmetry" allows AI to operate **"Whitebox"** testing while humans are often constrained to "Greybox." With autonomous AI pentesting, this constraint largely disappears; added context increases time and cost for humans**,** but typically improves **AI's effectiveness. AI scales with the richness of the context it ingests,** the most valuable context being the source code itself.

→ As a result, AI pentesting will drive a **structural shift** from Greybox towards **Whitebox as the default model.**

Key Takeways

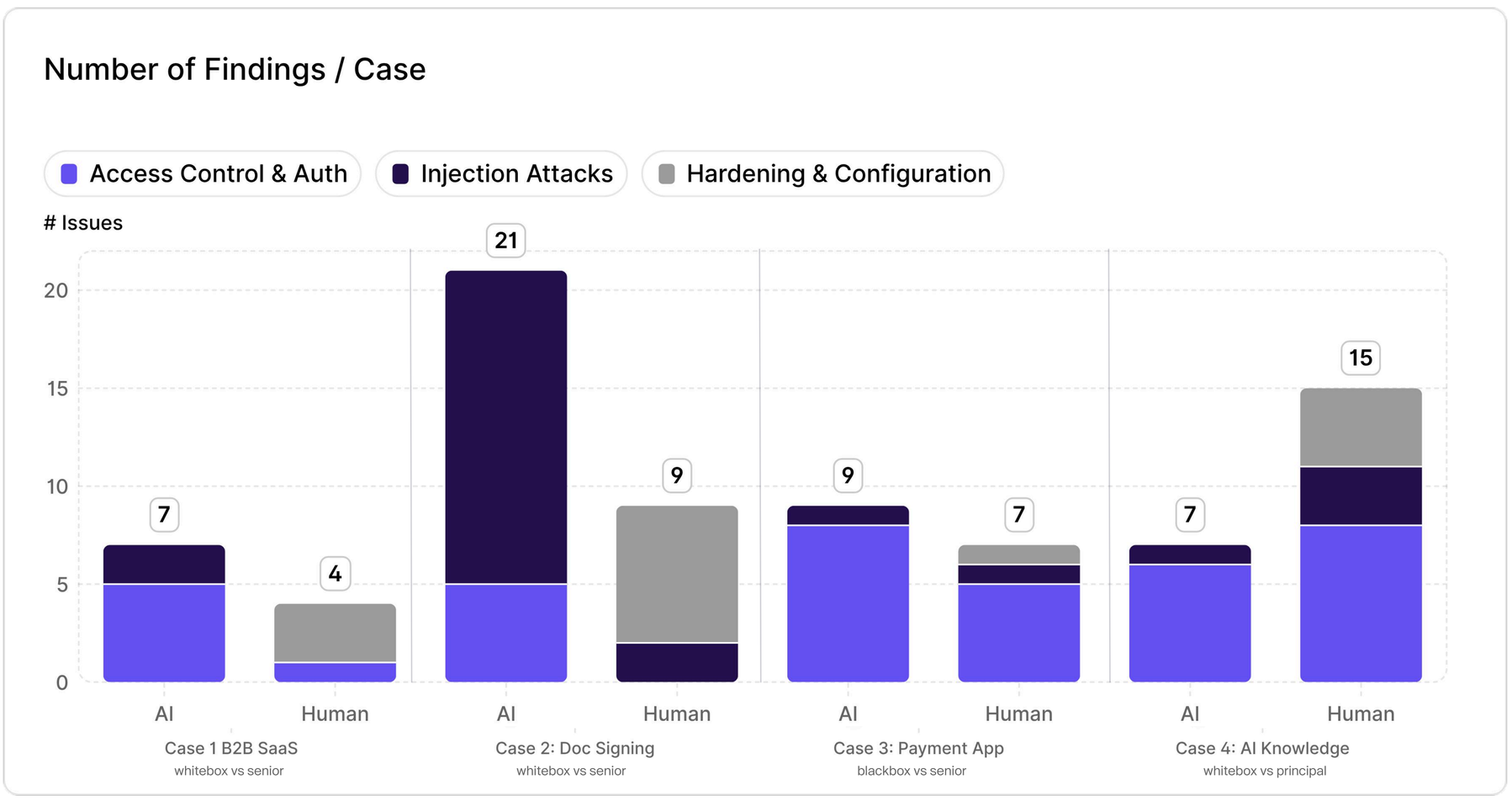
1. Speed Advantage

- **Autonomous**: Completed test in hours (approx. 6.5 to 14.5 hours).
- **Manual**: Took weeks (up to 4 weeks) for testing and reporting, from start to finish.



2. Depth vs. Compliance

- **AI Strengths:** Found deep logic flaws like IDORs, Authentication Bypasses, and E-signature forgery that humans missed.
- **Human Strengths:** Focused heavily on compliance, configuration hardening, and general security hygiene.



3. The “Access” Gap

- **Whitebox vs. Greybox**: AI tools instantly utilized full source code access ("Whitebox") to find hidden bugs . Humans were limited to "Greybox" (no code) access due to logistical hurdles, preventing them from seeing deeper flaws.
- **Blackbox Surprise:** Even without code access, the AI outperformed humans on finding Broken Access Control issues in one case study.

Taken together, these findings illustrate a consistent pattern: **access is the differentiator.** When source code is available, AI can capitalize on it immediately, with fast, scaleable, cost-effective Whitebox testing; when it isn’t, strong results in blackbox conditions suggest the advantage is not purely source code-dependent, but rooted in how effectively AI scales analysis across whatever context it can obtain. In other words, the “access gap” is not just operational- it’s structurally reshaping what “default” pentesting can look like.

Overall Verdict

- **AI testing** was drastically faster and better at pinpointing critical code-level exploits
- **Manual** testing excelled at broad compliance checks but missed catastrophic backdoors.

Conclusion

Times are changing, and AI is moving fast.

We are already seeing automation surpass human efforts in speed and deep logic detection. While this benchmark highlighted a distinction where manual efforts excelled at configuration, we have in the meantime closed the 'completeness gap' on hardening and compliance checks. Our goal is clear: to relentlessly improve until the automated engine outperforms traditional methods on all facets of security testing.

***Note**: Since December 15, 2025, Aikido autonomous pentests also include automated hardening and configuration checks, closing the gap observed in earlier benchmark results.*

Case Study 1: B2B SaaS Platform

A ClimateTech management platform for large enterprises.

Performance Data

Metric	Aikido (Autonomous)	Manual (Human)
Time	14,5 Hours	8 Days (Test + Validation)
Total Findings	7	4
Critical/High	3	1

Seniority: Senior Tester

Detailed Vulnerability Breakdown

Vulnerability Type	AI (Whitebox)	Human (Greybox)
IDOR	1	-
Broken Access Controls	2	1
Sensitive Info Disclosure	1	-
Missing Webhook Verification	1	-
XSS	2	-
Hardening Checks	-	3

Analysis:

- **AI:** Found **IDORs** and **XSS** by analyzing the code logic.
- **Human:** Missed the IDORs but found 3 specific **Hardening/Configuration** issues. The manual assessment focused on security best practices **but missed critical vulnerabilities.** The greybox setup of the test effectively prevented them from finding these deeper logic flaws.

Case Study 2: Document Signing App

A workflow-heavy application involving e-signatures.

Performance Data

Metric	Aikido (Autonomous)	Manual (Human)
Time	~9 Hours	~2 Weeks (Test + Reporting)
Total Findings	21	9
Key Win	Detected E-signature Forgery	Detailed Config Audit

Seniority: Senior Tester

Detailed Vulnerability Breakdown

Vulnerability Type	Aikido (Whitebox)	Human (Greybox)
Forging of e-signatures	1	-
XSS	12	1
SSRF	3	1
Open Redirect	1	-
Improper Access Control	3	-
Exposure of Information	1	-
Hardening Checks	-	7

Analysis

- **AI:** Detected a critical **Workflow Integrity** flaw (allowing forged signatures) and a high volume of XSS (12 instances).
- **Human:** Found 1 XSS and 1 SSRF, but focused heavily on **Hardening Checks** (7 out of 9 findings). This highlights that the human testers prioritized **compliance and configuration hygiene** over deep vulnerability detection, missing several critical issues found by the AI.

Case Study 3: Agentic Payment App

An application involving AI agents to manage payments.

Performance Data

Metric	Aikido (Autonomous)	Manual (Human)
Time	~9 Hours	~2 Weeks (Test + Reporting)
Total Findings	21	7
Key Win	Detected E-signature Forgery	Detailed Config Audit

Detailed Vulnerability Breakdown

Seniority: Senior Tester

Vulnerability Type	AI (Blackbox)	Human (Greybox)
Broken Access Control	8	4
XSS	1	-
CSRF	3	-
Information Disclosure	-	1
HTML Injection	-	1
General Hardening	-	1

Analysis:

- **AI (The "Blackbox" Surprise):** Even without source code access, the AI proved it can outperform human "Greybox" testing on deep logic flaws. It discovered **8 Broken Access Control** vulnerabilities (double the human findings) along with **CSRF** and **XSS** issues that were completely missed during the manual test.
- **Human:** The manual testers continued the trend of excelling at specific compliance and configuration checks, finding unique issues regarding **Information Disclosure**, **HTML Injection**, and **General Hardening**.

Case Study 4: AI Knowledge Platform

A platform to gather information and visualize it with AI.

Performance Data

Metric	Aikido (Autonomous)	Manual (Human)
Time	~8 Hours	~4 Weeks (Test + Reporting)
Total Findings	7	15
Key Win	Critical Code Flaws (Auth Bypass)	Breadth of Logic & Hardening

Seniority: Principal Tester

Detailed Vulnerability Breakdown

Vulnerability Type	Aikido (Whitebox)	Human (Greybox)
Improper Access Controls	3	7
XSS	1	2
Missing State Parameter (OAuth)	1	-
Hardcoded Auth Bypass	1	-
Sensitive Info Disclosure	1	1
GraphQL Hardening	-	3
Open Redirect	-	1
General Hardening	-	1

Analysis:

- **AI:** Leveraging Whitebox access, the system detected specific implementation vulnerabilities often missed by external assessments, including a **Hardcoded Authentication Bypass** and a **Missing State Parameter** in the OAuth flow, both severe vulnerabilities that compromise the application's core security.
- **Human:** The senior human team identified a wider range of business logic and configuration issues, uncovering an **Open Redirect**, specific **GraphQL hardening** gaps, and more **Improper Access Control** findings (7 vs. 3) compared to the AI.
- **The Trade-off:** This benchmark demonstrates that while Principal Testers can compete with automation when given extensive time and budget, the speed difference is stark. Automation offers a strategic advantage by delivering critical security insights in **~8 hours**, eliminating the **4-week lead time** required for human experts to achieve similar depth.

Update on Hardening Coverage:

Since these assessments were conducted, Aikido has closed the previously observed gap in configuration and hardening checks. As of **December 15, 2025,** autonomous pentests include systematic validation of common hardening and security hygiene requirements alongside exploit-driven testing. This ensures continued coverage of configuration-related findings without changing the focus on high-impact vulnerabilities.