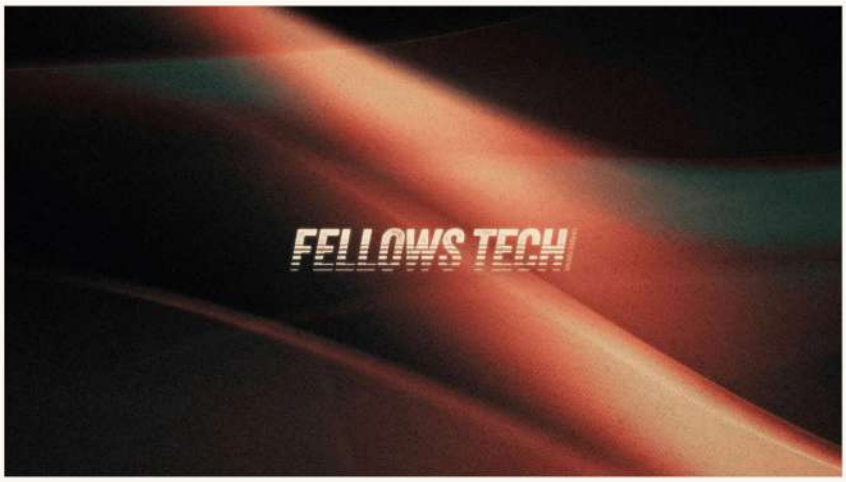
HUSTLE · TECH

The AI Sandbox You Trust Might Be a Year Behind on Security

BY CONTENT CURATOR · JULY 2, 2026

Fact-checked



0:00 6 MIN 1x

Every AI agent that writes and runs its own code rests on a quiet assumption: that whatever box holds that code will actually hold. Companies building agentic products, or just letting models run code on their machines, now carry that assumption as a line item with real liability attached. Researchers at fellows.tech and OrbitaLab tested five of the most widely deployed sandbox products. They found that buyers are mostly asking the wrong questions.

Here is what anyone signing a contract should sit with. How strong is the isolation? That gets the marketing attention, but it barely moves your real exposure. What actually bites you is operational: how fast do fixes reach you, and do a vendor's defaults quietly hand attackers more room than you realized?

The patch gap is a vendor-management problem

Lag was the finding with the most money riding on it. Underlying engines tend to patch fast, often shipping a fix on the same day a flaw gets disclosed. Trouble is, products built on top of those engines pin themselves to one version, and pinned versions rot.

One product had frozen its engine for over 471 days. That window left two

MOST POPULAR

1 TECH
Digital Trust in the Age of Smart Technology
By Content Curator

2 TECH
How Much Companies Spend on AI Tokens in 2026
By Content Curator

3 TECH
iGaming Compliance Systems: The Soft2Bet Data-Driven Approach
By Content Curator

4 TECH
What Is AI Psychosis and Why CEOs Get It
By Benjamin Lewis

serious fixes unapplied, one of them a high-severity escape-class flaw. Another sat 399 days back with an 8.7-severity hole live in its shipping config. So customers ran known-vulnerable software for more than a year while believing someone had their back, all because nobody bumped a version number.

Now the part that should rattle procurement. Whichever product had the toughest isolation underneath turned out worst at actually delivering patches. The flimsiest one did best, purely because it rode an auto-updating runtime it never had to think about. Good plumbing told you almost nothing about whether you were safe in practice. What told you was whether the vendor stayed disciplined, week after dull week. You will not find that on a datasheet, and hardly anyone thinks to ask.

"Zero known vulnerabilities" is a trap

One product runs on an engine with zero published security flaws. It also has no public testing program and no outside audit. The researchers won't treat that clean sheet as a point in its favor, and I think they're right to hold the line. A blank record can mean the code is solid. It can also mean nobody competent has gone looking yet, which is a very different thing to be buying. No history, nothing tested, nobody from outside checking the locks: the real risk just sits there, unknown.

For a business, "unknown" and "low" are not the same word wearing different hats. A vendor waving a spotless record with no audit trail behind it is handing you reassurance and calling it evidence. So flip the question. Don't ask whether they've shipped vulnerabilities. Ask how they hunt for them. Someone who has found ugly problems and fixed them usually beats someone who has simply never searched.

What this means for buyers

The authors decline to crown a winner, and that refusal is the whole point. They split the call into four business-facing questions, and nothing scored well across all of them.

How hard is it for hostile code to break out? While it runs, what can it learn about your environment? Can your team stack its own controls on top? And the one that quietly does the damage: do fixes actually land in your deployment on time?

A product can nail the first handful and whiff on the last, so the security you paid for bleeds away over months unless somebody actually owns version currency as a standing job. The picture also gets worse as agents get sharper. The study points to recent evaluations where offensive AI capability climbs steadily with compute. In one government test, an agent worked through 22 of 32 steps of a simulated attack on a corporate network. An operator running untrusted, maybe hostile, machine-written code on their own systems used to be a thought experiment. It isn't one now.

There's a quieter trap in the defaults too. One product shipped with a setting that handed sandboxed code far more reach than rivals gave it, not from any coding mistake but from how somebody dialed the out-of-the-box experience. Buyers who hear "secure by default" and never check are living with choices nobody showed them.

The practical checklist

This work turns sandbox security from a spec you skim into a pile of tradeoffs you have to manage. For whoever makes the call, a few questions cover most of the danger.

Who owns version currency, and how quickly does your vendor shrink the distance between an upstream fix and your live deployment? A frozen pin is an incident playing out in slow motion, and you won't feel it until it's already cost you. Next, how does the vendor go looking for vulnerabilities, and will they actually walk you through the program? Quiet is not the same as clean. Last, whatever the default configuration grants, have you gone and confirmed it yourself instead of trusting the brochure?

None of this is engineering work. It's governance, the unglamorous kind, and it decides whether the sandbox on your invoice is guarding you this morning or guarding the version of your company that existed last spring. OrbitaLab has published the full methodology and the per-product results.

FEATURED

§ § §

GREY

Content Curator

CONTRIBUTOR

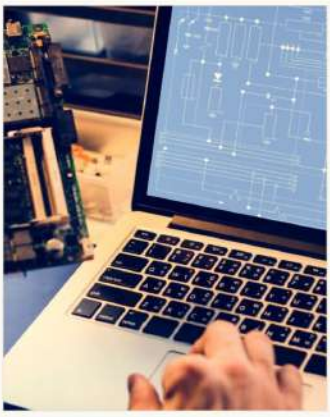
GREY Journal's editorial desk account, used for curated explainers, daily news summaries, and contextualized third-party reporting. Every curated piece is reviewed by a human editor on our newsroom team before publishing and follows our Editorial Policy. For tips or corrections, email care@greyjournal.net.

MORE FROM CONTENT CURATOR →

CORRECTIONS

Spotted something off? See our corrections policy or email care@greyjournal.net.

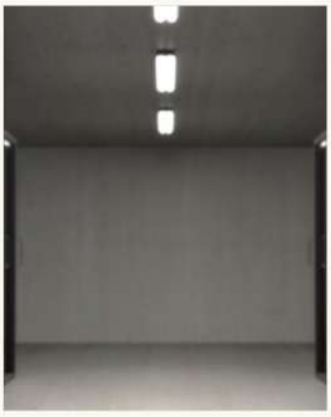
Read More FROM THE TECH DESK



TECH

Why Predictive Maintenance Software Fails Without Data Standardization

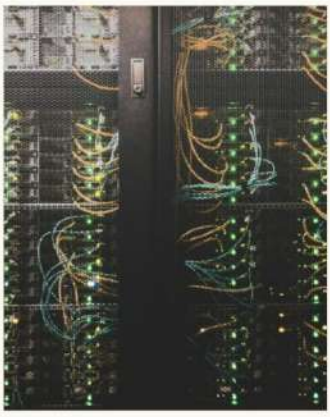
By Content Curator · June 3, 2026



TECH

Meta's Georgia Data Center Water Crisis Explained

By Benjamin Lewis · May 26, 2026



TECH

What Is the NVIDIA Vera CPU and Why Founders Should Care

By Benjamin Lewis · May 25, 2026