← BACK TO UPDATES [01] / 2026.05.25 / 8 MIN READ

VIBE CODING SECURITY WEEKLY — MAY 19 - MAY 25, 2026

> **Download:** vibe-coding-security-weekly-may-25-2026.pdf — printable, site-styled.

The week the platforms stopped equivocating. Apple removed the “Anything” vibe-coding app from the App Store under Guideline 2.5.2; the team put it on Google Play in roughly thirty seconds. Cursor shipped its own coding model, Composer 2, deciding it no longer wants to rent the model layer from Anthropic or OpenAI. OpenAI is reportedly negotiating a ~$3B deal for Windsurf, Amazon is drafting a Cursor competitor, and the Veracode “45% of AI-generated code contains OWASP Top-10 flaws” number quietly became the canonical stat that every founder, lawyer, and dealmaker is now repeating. Underneath all of it, the Orchids/BBC zero-click incident finally got a clean breakdown: a journalist’s laptop taken over

via a vibe-coding platform vulnerability, before the platforms even agree on whether AI-generated code is its own security category.

TL;DR — The week in one paragraph

- **Apple pulls "Anything", May 20–21.** Apple removed the "Anything" vibe-coding app under Guideline 2.5.2 (apps that "download, install, or execute code"). Co-founder Dhruv Amin's team put it on Google Play in about 30 seconds. One day earlier at Google I/O, Google's pitch was: "now anyone can be a builder." The platform split that the Apple WWDC framework was drafted to prevent is now visible in production: iOS bans generate-and-execute; Google ships it as a launch keynote.
- **Cursor Composer 2, May 22.** Cursor introduced Composer 2, its own coding-focused model. Cursor reports it beats Claude Opus 4.6 on coding evals and trails GPT-5.4. The strategic move matters more than the benchmark: Cursor is no longer just packaging third-party models in a developer-focused interface — it is buying optionality on the model layer the same week OpenAI is reportedly trying to buy Windsurf for around $3B.
- **OpenAI ~$3B Windsurf deal, May 22.** OpenAI is reportedly exploring a multibillion-dollar deal with Windsurf at roughly $3 billion. Same week: Amazon is reportedly developing a Cursor/Windsurf competitor tied to AWS. The IDE-as-distribution-channel land grab is on.
- **Veracode 45% becomes canon, May 20–21.** Multiple outlets — Master of Code, FrontierNews, and downstream syndications — settled on "45% of AI-generated code contains OWASP Top-10 vulnerabilities" as the headline number. Source: Veracode's 2025 GenAI Code Security Report. The number is now showing up in legal compliance pieces and M&A coverage. **Bain & Company's 2026 M&A report**: one in five strategic dealmakers walked away from a transaction this year because of AI risk on the target's codebase.

- **Orchids/BBC zero-click incident detailed, May 18.** The xhack.net breakdown of the Orchids incident first reported by the BBC: cybersecurity researcher Etizaz Mohsin discovered a vulnerability in Orchids (an AI vibe-coding platform), then demonstrated zero-click takeover of journalist Joe Tidy's laptop *by modifying code in another person's vibe-coding project*. Cloud-side code storage is the attack surface.
- **The framing fight, May 20.** Generative Labs published a clean read on the disagreement that has been growing since April: Vercel and Replit don't agree on what the problem is. Replit's positioning is explicit — AI-generated code is a category and needs its own security model. Vercel's Deepsec is framed as "the first tool that's surfaced the kind of issues we'd actually want a security engineer to flag" — a posture about review quality, not category. Same architecture (deterministic scan → agent investigation → severity rating → human review), opposing market frames.

What did Apple actually do, and why did the app land on Google Play in 30 seconds?

On **May 20, 2026**, developers tied to the "Anything" app — a vibe-coding tool that promises to turn a natural-language prompt into a native iPhone app — said on X that they had put the app on Google Play in roughly **30 seconds** after Apple's App Store removal. The Scout summary, citing TechCrunch reporting, names the rule Apple invoked: Guideline 2.5.2, which prohibits apps that "download, install, or execute code." Per TechCrunch's earlier coverage, Apple had removed Anything twice, briefly restored it once, and then pulled it again after exchanges with the company.

Co-founder Dhruv Amin's framing: the company had been through "emails, calls, appeals and multiple rewrites" trying to satisfy App Review. Apple's framing: "if the App Store model does not fit a business, there is always the open Internet."

This is the live test case for the WWDC framework we covered last week. The reported direction was: App Intents as the substrate, structured declared capabilities permitted, generate-and-execute on-device blocked. Anything was the precise category the framework is designed to deny. The 30-second jump to Google Play is the receipt that the platform split is now product-defining, not policy-aspirational.

The Google I/O timing is not coincidence. At I/O on **May 19**, Google said "now anyone can be a builder" and rolled out more agentic Gemini and Android creation tools. The implicit pitch to vibe-coding founders: if iOS will not have you, Android will. Sundar Pichai's keynote put the usage numbers at 8.5 million developers actively coding with Gemini.

What this means for product roadmaps in the vibe-coding stack:

- **Anything on a phone is now a two-platform decision, not a build decision.** iOS is closed to the generate-and-execute pattern unless App Intents threads the needle.
- **Web is the default escape hatch.** Apple explicitly points there. The category is reverting to browser-tab distribution for the iOS userbase.
- **The TestFlight / enterprise-cert workaround surfaces will get scrutinized.** Apple's removal of Anything was not a sweep — TechCrunch reported Replit and Vibecode were affected in the same scrutiny wave. Expect more.

What is Composer 2, and why did Cursor decide to build it?

On **May 22, 2026**, Cursor introduced Composer 2, a coding-focused model built specifically for agentic software development inside the Cursor editor. Cursor reports it outperforms Claude Opus 4.6 on relevant coding evaluations while still landing behind GPT-5.4. The benchmark is fine. The structural move is bigger.

Cursor built its growth by being the best wrapper around the best models — first Claude, then a multi-model picker. Composer 2 is the first time Cursor is owning the model layer that powers its assistant. The reason matters: in an editor that depends on tool use, multi-file editing, terminal output, and apply/diff/review interfaces, "the same model with our prompts on top" hits a ceiling. Cursor needs the model to know what Cursor wants.

That move is happening at the same moment OpenAI is reportedly trying to acquire Windsurf for around $3B — which is the model-owner buying the editor — and Amazon is drafting an AWS-native Cursor competitor — which is the cloud-owner building the editor inside its console. The IDE is the distribution channel now. Three companies, three different directions of vertical integration, one shared bet: the next platform fight happens in the editor.

The GitHub-leads-the-enterprise piece names the constraint Cursor still has: in regulated buyers, GitHub already owns identity integration, audit trails, data controls, SSO, SOC 2, and the surrounding workflow. Cursor wins on bottom-up developer enthusiasm and loses the standardization vote unless it can match the governance posture. Composer 2 does not fix that gap; it makes Cursor less dependent on a model vendor that could decide tomorrow to ship their own editor.

For our own posture: the editor war is also a security-context war. The model that knows your repo, your secrets, your CI config, and your `apply` diffs is the model that is best positioned to either prevent or amplify the integration-layer failures we keep documenting. The week Cursor shipped Composer 2 is the week to ask which model touches your codebase and whether your data controls match.

What is the Orchids/BBC incident, and why does it change the threat model?

On **May 18, 2026**, xhack.net published a detailed breakdown of an incident first reported by the BBC: BBC journalist Joe Tidy's laptop was taken over by

cybersecurity researcher Etizaz Mohsin via a vulnerability in **Orchids**, an AI vibe-coding platform. The xhack piece's framing is the part worth holding onto:

01. In December 2025, Mohsin was experimenting with vibe-coding systems and spotted a vulnerability in Orchids.
02. He performed a zero-click demo targeting an Orchids project that Tidy had built.
03. He gained unauthorized access to Tidy's project — and **modifying the code in another person's vibe-coding project was sufficient to take over the laptop**.

The structural detail: Orchids stores user code on the cloud side by default. Most AI-based vibe-coding platforms do. That means "protect my local machine" is not a sufficient security model — the attack surface includes the platform's storage, the project sharing primitives, and the developer's habit of running platform-generated code on a local machine without inspection.

xhack's own check on **May 17**: of three vibe-coding services the author was using, two were "cloud storage, including execution environment" and one was "local storage, cloud sync only." The category is not uniform on the most basic question — *where does my code live and what runs it*.

The xhack writeup also surfaces two numbers worth noting:

- An independent investigation found security flaws in approximately **10.3% of public Lovable apps** (consistent with the RedAccess 1.3% lower-bound and B1KEY 7% audit numbers when you account for sampling).
- The **Tenzai red-team** of five major coding agents found **69 issues**; **SVIBES** independently found that only **1 in 6** pieces of code that passed functional tests was actually safe.

The takeaway is operational: if your team uses a vibe-coding platform, ask the platform where the code lives, whether shared projects can mutate other users' execution environments, and whether the platform has had its own pen-test in the

last 90 days. The Orchids incident is the first widely covered case where the platform itself was the attack surface, not just the generated code.

What did the legal layer say this week?

On **May 21, 2026**, Fischer Legal published Vibe Coding Legal Compliance: What AI App Builders Need to Know Before Launch — the cleanest read so far on the regulatory exposure that vibe-coded apps inherit the moment they collect user data. Two anchors:

Pennsylvania's Breach of Personal Information Notification Act (BPINA), amended in September 2024, applies to any business that maintains personal information about Pennsylvania residents — regardless of size. If your app collects names with email addresses, account credentials, financial info, SSNs, or driver's license numbers, BPINA applies. Breaches affecting more than 500 residents require simultaneous notification to the PA AG and affected individuals, plus 12 months of free credit monitoring for breaches involving financial or government ID info. A private right of action bill passed the PA House in October 2025 and is pending in the Senate — if it passes, users can sue you directly.

The harder line is in Fischer's framing of cyber insurance: insurers are now asking applicants whether they have written security policies and a documented incident response plan, and whether a penetration test was performed before launch. Fischer's example is blunt: "Your app has a misconfigured database, the kind found in 58% of vibe-coded apps. An attacker finds it in an afternoon. Your users' data is exposed. You file a cyber insurance claim and the insurer finds your application incomplete."

The number to anchor on alongside the 45% Veracode figure: **Bain & Company's 2026 M&A Report** (cited via FrontierNews.ai) finds that **one in five strategic dealmakers walked away from a transaction in 2025–2026 because of**

anticipated AI risk on the target's codebase. AI coding due diligence is now a standalone workstream with its own buyer team and its own price impact.

What this means at the founder level:

- **The "we'll add security later" line is now a financing problem, not just a security problem.** It shows up at due diligence.
- **Penetration testing before launch produces documentation that feeds directly into your insurance application.** The audit is the insurance enabler.
- **The 58% misconfigured-database figure** (Fischer's number, methodology not fully sourced — treat as directional) is in the same family as the B1KEY 7% wide-open Supabase finding. The distinction is sampling: 7% is the lower bound on audited apps with zero RLS; 58% is the looser definition including incorrectly configured RLS policies.

What did the framing fight produce this week?

Generative Labs' Vercel-vs-Replit piece on **May 20** finally named the disagreement that has been growing under the surface since April. The two tools have **the same architecture**:

01. Deterministic scan (pattern matchers for known vulnerability shapes).
02. AI agents investigate the flagged candidates, trace data flow, and check existing mitigations.
03. Findings with severity ratings.
04. A human-review step.

The framings are opposite. **Replit Security Agent** is explicitly positioned as a tool *for vibe-coded apps* — AI-generated code is a category that needs its own security model. **Vercel Deepsec** is positioned as "the first tool that's surfaced the kind of issues we'd actually want a security engineer to flag" — a posture about

review quality, not category. Generative Labs' read: "Anyone selling you certainty on the framing right now is selling."

The reason this matters is downstream: if AI-generated code is its own category, you buy a category-specific scanner (Replit Security Agent, Bugbot, Cursor Security Reviewer). If it is "security review just got pulled left," you buy a general AppSec tool that knows how to talk to LLM-generated artifacts (Deepsec, Snyk, Veracode). The buyer decision is downstream of the framing decision, and the framing decision is not settled.

Two ancillary stories tie in:

- **Codev** (May 21) is pitching an "agentic delivery layer" — a team of agents that plan, code, review, test, document, and prepare changes for review. The pitch is explicit: "vibe coding hangover" is the cleanup cost after fast, informal AI-generated code without surrounding engineering discipline. Codev positions itself as the system that prevents the hangover rather than the scanner that catches it.
- **ServiceNow's Vibe Coding** (May 22) is the platform-vendor version: natural-language entry, but the output snaps to Flow Designer, App Engine, governed catalog items, and existing identity/entitlement boundaries. Vibe coding inside a fence the enterprise already trusts.

The common thread: the market split between "vibe coding is a category" and "vibe coding is a workflow step inside the existing category" is now visible in three places — scanners, agent platforms, and low-code suites. Each side is hiring.

The week's smaller stories

- **Google I/O on May 19.** Pichai's keynote pitch: "now anyone can be a builder," with 8.5 million developers coding with Gemini and broader agentic Android creation tools rolling out. Implicit policy: where Apple draws lines, Google opens doors.

- **fizzgig published a Lovable/Bolt audit guide on May 21.** fizzgig.ai made the actual workflow explicit: Lovable/Bolt sync to GitHub → open in Cursor or Claude Code → run the audit there. The piece's headline finding: Supabase's whole authorization model is RLS, and on a default Lovable build "if RLS is disabled, or a policy is `using (true)`, then your app's anon key — which ships to every browser — can read and write every row in the table."
- **Qwen3-Coder-Next, May 20.** Alibaba released Qwen3-Coder-Next, an open-source ultra-sparse coding model claiming 10x throughput for repo-scale tasks. Pattern: every major lab now has a vibe-coding-positioned model.
- **"Vibe coding is dead" coverage, May 22.** Multiple syndications (positioniseverything, freedom251) ran the same "agentic swarm coding is the new enterprise moat" framing. The argument: prompt-and-generate doesn't scale to enterprise; coordinated specialized agents (plan/code/review/test/document/secure) do. Karpathy's acknowledgement last week that "the era is ending" is now the working assumption in enterprise coverage.
- **B1KEY's credibility-cost follow-up, May 19.** B1KEY's hidden-credibility-cost piece is the companion to last week's 1,764-app audit: cost of fixing security issues in production is roughly 6x catching them in development (B1KEY's number across 30+ projects, methodology not published — treat as directional). The qualitative claim — "AI-generated code that works is not the same as AI-generated code that is reviewed" — is the part that holds.
- **Master of Code 138-tool count, May 20.** Master of Code put the current count of "vibe coding tools flooding the market" at 138. A quarter of Y Combinator's W25 batch shipped codebases that were ~95% AI-generated; the same writeup pegs developer confidence in AI-generated output at 60% (down from 77% in 2023, per Hashnode's State of Vibe Coding 2026).

Why this week's stories rhyme

Three responses to the same maturation, all visible in the same week:

- **The platforms picked sides.** Apple removed Anything under 2.5.2; Google's I/O message was "now anyone can be a builder." iOS is closed to generate-and-execute, Android is open. The split is now product-defining.
- **The model layer reorganized.** Cursor shipped its own model (Composer 2). OpenAI is reportedly buying the editor (Windsurf ~$3B). Amazon is drafting both (AWS-native AI coding service). Three distinct directions of vertical integration, one shared bet: the editor is the distribution channel.
- **The number became canon.** Veracode's 45% is no longer one citation — it is the stat that legal, M&A, and security writeups now all reach for. Combined with B1KEY's 7%, Lovable's 10.3% (xhack), and Fischer's 58% misconfigured-database figure, the field has converged on a range. The number is no longer the open question. The fix is.

The Orchids/BBC incident is the live attack example that ties the strands: the platform was the attack surface, the journalist was the target, the platform's default storage architecture was the vulnerability, and the broader question — "is this its own security category?" — is unsettled even at the vendor level (Replit yes, Vercel no). What is settled: this is no longer fringe.

Manual checklist — 10 things to verify yourself

01. **List every vibe-coding platform your team uses, and ask each one where the code lives.** Cloud storage with shared execution environment is the attack surface the Orchids incident exploited. Local-storage / cloud-sync only is a meaningfully smaller surface.
02. **If you ship to iOS, audit any feature that "downloads, installs, or executes code" against Guideline 2.5.2.** Apple removed Anything for exactly this pattern. The category will be enforced more, not less, after WWDC.

03. **If you ship to Google Play, assume your iOS-removed competitors will land beside you in days.** Anything was on Google Play in 30 seconds. Plan for category density.
04. **Check whether the model that touches your codebase is your model or somebody else's.** Cursor with Composer 2 is a different data-handling story from Cursor with Claude or GPT routed through Cursor. Update your AI-BOM.
05. **For every Lovable/Bolt/Replit app in production, run the fizzgig audit flow: sync to GitHub, open in Cursor or Claude Code, grep every Supabase policy.** RLS disabled or `using (true)` = anon-key-readable.
06. **If you're raising capital this year, pre-empt the M&A diligence question.** Bain says 1 in 5 strategic dealmakers walked from a deal in 2025–2026 because of AI risk on the target's code. A pen-test report dated before you open the round is cheaper than a re-priced term sheet.
07. **If you collect PII from US users, confirm BPINA applies (Pennsylvania residents) and verify your incident-response plan covers simultaneous AG + affected-individual notification.** The private-right-of-action bill is pending in the PA Senate; the cost line moves if it passes.
08. **For any cyber insurance application open or renewing this year, line up the documentation:** written security policies, IR plan, pen-test report, audit log retention. The insurer's first question is now whether you did this *before* the breach.
09. **Decide your framing position deliberately: is AI-generated code its own security category, or a workflow step inside your existing AppSec posture?** The tool you buy next quarter is downstream of that answer. Generative Labs is right that the market hasn't settled it.
10. **Read the Orchids/BBC writeup end to end** and use it as a tabletop. The question to answer: if our vibe-coding platform had Mohsin's vulnerability, who at our company would be the next Joe Tidy?

Related coverage

- Vibe Coding Security Weekly — May 18, 2026 — B1KEY 1,764-app audit, Cursor Bugbot, Mini Shai-Hulud, Apple WWDC framework
- Vibe Coding Security Weekly — May 11, 2026 — RedAccess 380K-app scan, Replit Security Agent, Vercel Deepsec
- Vibe Coding Security Weekly — May 5, 2026 — prior weekly
- Apple vs Vibe Coding — the March enforcement wave that the 2.5.2 removal of Anything sits inside
- The Integration Layer Is the Real Security Gap — why per-tool scanners miss the Orchids-style platform attack surface
- Lovable Security Report — May 2026 — platform-level view of the Supabase / RLS / anon-key default failures fizzgig's audit guide is built around
- Your CLAUDE.md Is Attack Surface — prior art for the platform-as-attack-surface threat model

Sources

- Scout — Apple removes 'Anything' vibe-coding app from App Store — May 21, 2026
- freedom251 — Cursor's new coding model Composer 2 is here — May 22, 2026
- positioniseverything — Windsurf: OpenAI's potential $3B bet to drive the 'vibe coding' movement — May 22, 2026
- positioniseverything — Amazon is reportedly developing an AI Coding Service to compete with Cursor and Windsurf — May 22, 2026
- Master of Code — AI Vibe Coding Startups: 45% Ship Security Flaws — May 20, 2026
- FrontierNews.ai — Why 45% of AI-Generated Code Ships With Security Flaws — May 21, 2026

- xhack.net — A BBC Reporter's Laptop Was Taken Over: The Orchids Incident — May 18, 2026
- Generative Labs — AI Code Security in 2026: Vercel and Replit Don't Agree on the Problem — May 20, 2026
- Fischer Legal — Vibe Coding Legal Compliance: What AI App Builders Need to Know Before Launch — May 21, 2026
- fizzgig.ai — Built it on Lovable or Bolt? Audit it before you ship. — May 21, 2026
- B1KEY — AI Posts and Vibe Coding: The Hidden Credibility Cost — May 19, 2026
- androidexperto — Codev lets enterprises avoid vibe coding hangovers — May 21, 2026
- freedom251 — ServiceNow brings Vibe Coding to enterprise workflows — May 22, 2026
- freedom251 — GitHub leads the enterprise, Claude leads the pack — Cursor's speed can't close — May 22, 2026
- positioniseverything — Vibe coding is dead: Agentic swarm coding is the new enterprise moat — May 22, 2026
- geekchamp — Qwen3-Coder-Next: open-source ultra-sparse model — May 20, 2026
- Google — Sundar Pichai I/O 2026 momentum — May 19, 2026

This digest is compiled from public reporting. VibeEval is not affiliated with Apple, Google, Cursor, OpenAI, Windsurf, Amazon, Orchids, the BBC, Veracode, Bain & Company, Master of Code, B1KEY, fizzgig, Generative Labs, Replit, Vercel, Codev, ServiceNow, Fischer Legal, or any other organization cited. Numbers from vendor or marketing-shaped writeups (the 45% from Veracode, the 58% misconfigured-database figure from Fischer, the 6x cost-of-fixing-in-production from B1KEY, the 10.3% Lovable figure cited via xhack) are attributed and directional. Questions? Contact our team.

/ MORE UPDATES

Keep reading

- 2026.05.18 VIBE CODING SECURITY WEEKLY — MAY 12 - MAY 18, 2026
- 2026.05.13 2026 AI CODING SECURITY REPORT: THE DATA BEHIND THE VIBE-CODING BREACH WAVE
- 2026.05.11 VIBE CODING SECURITY WEEKLY — APR 29 - MAY 11, 2026
- 2026.05.11 LOVABLE SECURITY REPORT MAY 2026: THE DEFENDER STACK REORGANIZES AROUND VIBE CODING

/ ALTERNATIVES

Snyk Alternative

Burp Suite Alternatives

All alternatives →

LATEST

Weekly Digest (May 25)

Weekly Digest (May 18)

Lovable May 2026 Report

Lovable Apr 2026 Report

Lovable BOLA Vulnerability

Vercel/Context.ai Breach

Agent Skills Security

All updates →

LEGAL

Trust & Security

Privacy

Terms

DPA

Refunds

PROGRAMS

Get paid

VIBE-EVAL.COM