UPDATES / WEEKLY DIGEST / 2026.05.18 / 8 MIN READ

Vibe Coding Security Weekly — May 12–May 18, 2026

B1KEY's 1,764-app audit independently confirms the RedAccess shape (7% wide-open Supabase, IDOR-by-URL-increment). Cursor announces Bugbot. Mini Shai-Hulud uploads 373 malicious npm versions across 169 packages. Apple drafts an AI-agent framework ahead of WWDC. Gartner and the NCSC weigh in.

Source: https://vibe-eval.com/updates/vibe-coding-security-weekly-may-18-2026/

A second independent audit dropped this week and it lands on the same shape as the RedAccess scan: B1KEY reviewed 1,764 vibe-coded apps and found the familiar pattern — wide-open Supabase, hardcoded keys in browser bundles, IDOR by URL increment. Meanwhile the platforms started drawing lines: Cursor announced Bugbot (AI reviewing AI-generated code), Apple began drafting an App Store framework for AI agents ahead of WWDC, and a Mini Shai-Hulud campaign quietly uploaded 373 malicious versions across 169 npm packages — including the readme-poisoning trick designed to bait AI coding agents into pulling them in. The NCSC and Gartner both put out marker statements. Here is the week.

TL;DR — The week in one paragraph

- **B1KEY audit, May 13:** Independent review of 1,764 vibe-coded apps built across Claude Code, Codex, Cursor, Replit, and Devin. **7%** had wide-open Supabase databases with no Row Level Security, OpenAI and Stripe keys were found exposed in browser consoles, and IDOR-by-URL-increment was reproducible against multiple apps — one allowed downloading the entire customer database by changing a single ID. The audit matches the RedAccess field shape.
- **Cursor Bugbot, May 13:** Cursor announced Bugbot — an AI-powered code reviewer that leaves comments on PRs flagging bugs, type errors, and security concerns. Architecturally it sits in the CI slot. Same week Cursor also shipped Security Reviewer; Anthropic's `/ultrareview` multi-agent flow shows up in comparison reviews. The defender stack for AI-generated code is now a category.
- **Mini Shai-Hulud, May 11 (Aikido disclosure):** Between 7–8 PM UTC on May 11, attackers uploaded **373 malicious versions across 169 npm packages** including `@tanstack/*`, `@uipath/*`, `@mistralai/*`, `@squawk/*`, and `@tallyui/*`. ReversingLabs separately disclosed **PromptMink**, a campaign of npm packages with READMEs deliberately crafted to attract AI coding agents into installing them.

> - **Apple WWDC framework, May 14–18:** Per The Information (cited in multiple outlets), Apple is drafting an App Store policy framework to allow AI agent apps with privacy/security guardrails. The current policy still bans vibe-coding tools that "execute code that alters their own functionality." Likely path forward: App Intents — declared, auditable capabilities permitted; on-device codegen-and-execute still blocked.
> - **Institutional pressure, May 13–14:** NCSC CEO at RSAC framed vibe coding as a "disruptive opportunity" to make software more secure-by-default — conditional on the tools being trained for it. Gartner (via Checkmarx) named the accountability gap: "agentic coding tools are inherently incapable of taking accountability; you and your engineers are." 81% of organizations knowingly ship vulnerable code.

What did B1KEY actually find?

On **May 13, 2026**, B1KEY published a field audit of 1,764 vibe-coded applications built across Claude Code, Codex, Cursor, Replit, and Devin. The headline numbers:

- **7%** of audited apps shipped Supabase databases with **no Row Level Security** — no policies, the front-end API key effectively granted unrestricted read/write on every row of every table.
- **Hardcoded credentials** in browser bundles: OpenAI API keys visible from devtools, Stripe payment secrets fully exposed. One documented case ran $200/month in extractable, attacker-usable API costs.
- **IDOR by URL increment**: one app allowed downloading the entire customer database — names, emails, credit balances — by changing a single ID in a URL. No auth required.

B1KEY's own framing of the cause matches the Secure.com piece that ran two days earlier (May 11): "AI tools generate code that uses the [Supabase] APIs without understanding the model. The 7% figure likely undercounts the problem, because it only captures databases with zero RLS — not databases with incorrectly configured or overly permissive policies."

The number to hold in mind alongside this: in the RedAccess scan covered in our May 11 digest, 5,000 of 380,000 publicly accessible apps (1.3%) leaked sensitive data. B1KEY's 7% is a different denominator — a curated audit of apps known to be in production — but the failure modes line up exactly. Two researchers, different sampling strategies, same shape.

The 2.74x multiplier B1KEY cites — AI-generated code containing roughly 2.74 times more security flaws than human-written code — comes from the Forbes report we flagged in earlier digests. Treat that number as directional: the methodology has not been published end-to-end and it predates the post-March platform responses. The qualitative claim ("AI-generated code that works is not the same as AI-generated code that is reviewed") is the part with the receipts.

What is Bugbot, and why does it matter this week?

On **May 13, 2026**, WIRED.jp broke that Cursor had announced **Bugbot** — an AI that reviews and fixes code written by another AI. Architecturally Bugbot occupies a CI slot: it leaves comments on Pull Requests flagging bugs, type errors, and security concerns before merge.

Two weeks ago this would have read like marketing. After the RedAccess scan and the B1KEY audit, it reads like a category arriving. Counting Bugbot, the AI-reviewing-AI-generated-code stack now includes:

- **Cursor Bugbot** + **Cursor Security Reviewer** (announced same week)
- **Replit Security Agent** + **Workspace Security Center 2.0** (covered in the May 11 digest)
- **Vercel Deepsec** (open-sourced May 5)
- **Anthropic Claude Code** `/ultrareview` (multi-agent parallel review)
- **Checkmarx Assist agents** — Developer Assist (IDE), Triage Assist (prioritization), Remediation Assist (PR fixes)

The xhack piece names the pattern explicitly: vibe coding "has finally reached the stage where the tools themselves begin to answer the question: 'It runs — but is it actually okay?'" That is the shift. The previous answer (a developer with a security checklist) was the bottleneck the speed gain was bypassing. The new answer is a CI-stage AI gate.

The honest caveat that the zenn.dev weekly comparison makes: vendor-published benchmarks (Bugbot 80%, Greptile 82%) are vendor-published benchmarks. The piece's recommendation — run a 30-day pilot against your own PR history, evaluate across five axes including data residency — is the right one. The marketing number is not the buying signal.

For our own posture: AI reviewing AI is still the integration layer catching the integration layer. It is necessary and insufficient. The 7% Supabase finding from B1KEY is a settings-default problem; no PR comment will catch a misconfigured production database that has been live for six months.

What did Aikido catch in npm this week?

On **May 11, 2026, between 7 and 8 PM UTC**, an attacker uploaded **373 malicious package versions across 169 npm packages**. Aikido Security disclosed the campaign, dubbed Mini Shai-Hulud, on May 12. Affected scopes included `@tanstack/`, `@uipath/`, `@mistralai/`, `@squawk/`, `@draftlab/`, `@beproduct/`, and `@tallyui/`. The packages carried valid SLSA provenance — the supply-chain signature looked correct because the upstream accounts were compromised, not because the chain was broken.

The vibe-coding angle is in the BlogSaays first-person writeup: "the whole pitch of vibe coding and the reason my agency can take on the volume of e-commerce work that we do is that I don't have to read every line of code or evaluate every dependency." The same writeup surfaces a separate campaign that should worry every team running an agent loop: **PromptMink**, disclosed by ReversingLabs — npm packages whose READMEs have been engineered specifically to attract AI coding agents into installing them. Readme-as-prompt-injection, against the agent that selects dependencies.

The Snyk skill-poisoning finding (13.4% of 3,984 agent skill files contained critical issues) and the Tempest Security Daily framing both point at the same defender problem: traditional SCA (lockfile audits, renovate, scheduled scans) catches malicious dependencies *after* the agent has already pulled them in. For vibe-coding workflows the scan has to happen at the moment the model writes `import`, not at PR time.

What to actually do this week, if you are running unattended AI coding:

- Enable npm install **cooldown periods** (block versions newer than N days from auto-install).
- Disable npm install scripts globally (`npm config set ignore-scripts true`) or move to `pnpm`.

- Treat readme content as untrusted input the agent reads, not as documentation.
- Pin your dependency-selection step to a curated allow-list when the agent is acting unattended.

What is Apple drafting, and why does it matter for vibe coding?

The story has been told three times this week (Perplexity AI Magazine May 15, Techforbes May 15, GadgetsNow May 18), all sourcing back to **The Information**: Apple's engineering teams are drafting an App Store policy framework to allow AI agent apps while preserving the platform's privacy and security guardrails. WWDC 2026 in June is the likely (unconfirmed) announcement window.

What is currently banned, and what the framework changes:

- **Current policy:** Chatbots that generate text/images/code on request are permitted. Agents that take **autonomous multi-step actions** without per-step user confirmation are not. Vibe-coding tools that **generate and execute** code on-device fall under the rule prohibiting apps from "executing code that alters their own functionality" — which is why Apple started blocking updates for some vibe-coding apps in March.
- **Reported direction:** App Intents as the substrate. Structured, declared, auditable agent capabilities permitted. Generate-and-execute coding agents on-device still blocked. Per-action confirmation tiers, on-device processing for sensitive data, explicit capability declarations.

The story Apple is trying not to repeat is in the GadgetsNow piece: "The OpenClaw incident, where agents went haywire and deleted a user's emails, exemplifies the risk Apple now has to architect around." App Store review architecture has limited visibility into what an agent does inside its parent app — that is the structural gap App Intents is being asked to fix.

For platforms in the vibe-coding stack, the read-through is: anything that ships generate-and-execute behavior to a phone is going to lose iOS distribution unless it routes through App Intents. The Lovable-or-Replit-on-a-phone product roadmap just got narrower.

What did the NCSC, Gartner, and DORA say this week?

Three institutional markers landed in the same five-day window:

NCSC (UK National Cyber Security Centre), at RSAC, May 13. Per cyfar.ca's writeup: the NCSC CEO framed vibe coding as a "disruptive opportunity" to push the whole software ecosystem toward secure-by-default — *conditional* on the AI coding tools themselves being designed and trained from the start to avoid introducing vulnerabilities. The implicit argument: if the model produces RLS-enabled Supabase code by default, the 7% B1KEY number becomes 0.7%.

Gartner, via Checkmarx's "Vibe Coding Hangover", May 14. Gartner's framing in their latest application-security report, as quoted by Checkmarx: "*Agentic coding tools are inherently incapable of taking accountability; you and your engineers are.*" They map three gaps — accountability (designate explicit AI software leads), policy (formal allow/deny lists for AI coding tools, tracked in a centralized AI-BOM), and automation (layered defense across pipeline stages, not one scan per sprint). The headline number Gartner-via-Checkmarx pulls: **81% of organizations knowingly ship vulnerable code**, driven by noise, uncontextualized backlogs, and limited resources.

DORA 2025, surfaced via Agenda Digitale May 18. Google Cloud's DORA report: AI does not improve software delivery by itself — it *amplifies* the quality of the engineering system it is dropped into. 90% of respondents use AI at work, 80% say it raised individual productivity, but enterprise-wide delivery gains stay locked behind disciplined testing, review, security, and governance. The Italian writeup names the pathology cleanly: "localized pockets of productivity" that the rest of the system cannot absorb.

The three statements rhyme: the gap is no longer "is the tool fast enough" — it is "is the surrounding system disciplined enough to absorb what the tool produces." That is also the gap the B1KEY 7% sits inside.

The week's smaller stories

- **OpenAI Daybreak, May 11.** OpenAI released Daybreak, a security stack combining GPT-5.5-Cyber with Codex Security. Codex Security reviews PRs "like a senior security engineer" and is positioned against Anthropic's Claude Mythos. Pattern: every frontier lab is now shipping a security-agent SKU.
- **Google Android Show I/O Edition, May 12.** Google demoed Gemini Intelligence reading on-screen context, completing multi-step tasks across apps, and **vibe-coding home-screen widgets on the fly**. The same model behavior that gives Apple App Store nightmares is shipping unmodified on Android.
- **Hugging Face fake-OpenAI repo with infostealer.** A repository masquerading as an OpenAI release delivered infostealer malware with 244K apparent downloads. Model-registry supply chain, same shape as Mini Shai-Hulud but on a different stack.
- **Karpathy calls the vibe-coding era over.** Karpathy, who coined the term in February 2025, acknowledged just over a year later that the era is ending as professionals adopt orchestrated AI agents with structured human oversight. The replacement framing is "spec-driven development" — Airbnb reports 60% of its code is now AI-generated under that workflow.
- **DDoS industrialization report names vibe coding.** Cyfar.ca's analysis of the 2026 DDoS landscape (super-botnets Kimwolf and Aisuru running Layer 7 floods) lists "unverified AI-generated code (Vibe coding)" alongside dangling CNAMEs as the misconfiguration pipeline that attackers are exploiting at scale.

Why this week's stories rhyme

Three responses to one problem, all visible in the same week:

- **The platforms drew lines.** Apple drafting an agent framework. Cursor shipping Bugbot and Security Reviewer. The pattern: turn the agent's autonomy into something the surrounding system can audit.
- **The supply chain bit back.** Mini Shai-Hulud against npm. PromptMink against the agent itself. Fake-OpenAI repo on Hugging Face. The agent is now a high-value target because it picks the dependencies, and the attackers know it.
- **The field data confirmed the shape.** B1KEY's 1,764-app audit lands on the same failure modes as the RedAccess 380K scan. 7% wide-open Supabase. IDOR-by-URL-increment. Hardcoded keys in browser bundles. Two researchers, different sampling, same answer.

The NCSC and Gartner statements are the institutional version of what the field data already shows: the gap isn't the model, it's the system around the model. The model that generates correct-and-secure code does not exist yet (SecureVibeBench measured 23.8% in the April 28 digest). The system that catches the gap — Bugbot, Security Reviewer, App Intents, Mini-Shai-Hulud-aware SCA — is what shipped this week.

Manual checklist — 10 things to verify yourself

01 **Pull your Supabase project list and confirm RLS is enabled on every table with sensitive data.** B1KEY found 7% of audited apps with RLS fully off. The check is one query per project.

02 **Grep your front-end bundles for `sk_live_`, `sk_test_`, `OPENAI_API_KEY`, and any provider key prefixes.** If a key shows up in compiled JS, rotate it before you finish reading this checklist.

03 **For every public-facing list/detail endpoint, test IDOR by incrementing the ID as an unauthenticated user.** B1KEY documented a case where this single test exposed an entire customer database.

04 **If you are running npm install unattended in CI, set a cooldown window** (block versions younger than 48–72h) and disable install scripts. Mini Shai-Hulud's window was about an hour.

05 **Audit your `package.json` and `package-lock.json` against the Mini Shai-Hulud scope list** (`@tanstack`, `@uipath`, `@mistralai`, `@squawk`, `@draftlab`, `@beproduct`, `@tallyui`). Pin to known-good versions.

06 **Treat README files in dependencies as untrusted prompt input the agent will read.** PromptMink proved this is a live attack surface.

07 **If you have an AI agent with shell or write access, list the directories it can touch and prune to the minimum.** OpenClaw deleting emails is the example to architect around.

08 **Stand up an AI-BOM** — a central inventory of every model, MCP server, agent skill, and AI tool in use across your org. Gartner's policy gap is unfixable without one.

09 **Designate one named owner for AI-generated code quality.** Gartner: "*agentic coding tools are inherently incapable of taking accountability; you and your engineers are.*"

10 **Pilot an AI code reviewer against 30 days of your real PR history before you trust the vendor benchmark.** Bugbot, `/ultrareview`, Greptile, Cursor Security Reviewer — all of them. The vendor's 80% means nothing on your codebase.

Sources

→ B1KEY — Vibe-Coded App Security Fails: 1,764-App Audit Results — May 13, 2026 — b1key.com/en/blog/vibe-coded-app-security-fails-1-764-app-audit-results/

- → xhack.net — Bugbot Is Here (Cursor's May 13 announcement via WIRED.jp) — May 14, 2026 — xhack.net/en/blog/g2026051400016501/
- → zenn.dev — Weekly AI Driven Development 2026-05-17 — comparison of Bugbot, Cursor Security Reviewer, and `/ultrareview`
- → Aikido Security — Mini Shai-Hulud Is Back: TanStack Compromised — May 12, 2026
- → BlogSaays — How To Prevent Npm Supply Chain Attacks In Your Vibe Coding Project — May 16, 2026
- → Perplexity AI Magazine — Apple App Store AI Agents Policy 2026: WWDC Guide — May 15, 2026
- → Techforbes — Apple Weighs App Retailer Guidelines for AI Agent Apps at WWDC26 — May 15, 2026
- → GadgetsNow — Is Apple's AI Strategy Discipline or Surrender? — May 18, 2026
- → cyfar.ca — NCSC CEO at RSAC: Seize 'disruptive' vibe coding opportunity — May 13, 2026
- → Checkmarx — The Vibe Coding Hangover (with Gartner framing) — May 14, 2026
- → Agenda Digitale — Vibe coding: più codice, ma non sempre più valore — May 18, 2026 (DORA 2025)
- → Secure.com — Vibe Coding Security Risks: What Your Scanner Misses — May 11, 2026
- → AI Empowered — Agentic AI Goes Everywhere (OpenAI Daybreak coverage) — May 18, 2026
- → Artificial Intelligence News — Malware on Hugging Face masquerading as OpenAI release — May 2026
- → cyfar.ca — Apps, APIs, and DDoS 2026: The Industrialization of Cyberattack Campaigns — May 13, 2026

This digest is compiled from public reporting. VibeEval is not affiliated with B1KEY, Aikido Security, ReversingLabs, Cursor, Apple, Replit, Anthropic, Checkmarx, the NCSC, Gartner, or any other organization cited. Numbers from vendor or marketing-shaped writeups (the 81% from Checkmarx, the 2.74x flaw multiplier, the 7% Supabase figure) are attributed and directional. Questions? hi@vibe-eval.com.