

Claude Code gets programmable "mods" as model-found PHP RCEs set off a week of patching

Coding Agents Alpha Tracker

2026-10-22

Claude Code gets programmable "mods" as model-found PHP RCEs set off a week of patching

By Coding Agents Alpha Tracker • October 22, 2026

Claude Mods let you script Claude Code's human-facing layer. Separately, the php2shell disclosure showed frontier models finding internet-scale bugs, and agents are now part of finding and patching them. Also covered: Boris Cherny's Claude Code playbook, Deep Agents context management, and a wave of proactive-agent apps.

Claude Mods: scripting the human side of the harness

Latent Space published a hands-on deep dive into **Claude Mods**. A mod is a Claude Code plugin made of TypeScript or JavaScript functions that respond to events such as a prompt, a tool call or an interface render. A mod can observe the event, change how it is rendered, or handle it instead of the default behavior, in both the CLI and Desktop [1]. You don't have to write them yourself: a built-in skill teaches Claude to build a mod from a plain-language request [1].

To see the built-in mods, run `/plugin` and open *Installed*. They include **agents-md** (loads AGENTS.md), **diff**, **sec-default** and **you-should-know**, which is the only one off by default [1]. *You Should Know* runs a side agent that watches the main agent and flags things buried in the trace. In the author's example, it noticed that the file being edited, plus about 45 others, had uncommitted work. It suggested committing or stashing before he discarded the demo edits [1].

Two mods worth trying: - **Next Steps Supervisor**, from the open-source ModSmith. It runs after any turn that edits files, runs commands or makes 8

or more tool calls, and offers a clickable next-step instruction [1]. - **Its costs.** That trigger fired on 99 of 111 turns, so much of the output became noise. Each check costs about 5–6k input-equivalent tokens on a 50k-token session [1]. Use `/supervisor off` when it’s not helping [1].

On trust: mods run with your permissions, can see every prompt, can spend your usage, and “aren’t sandboxed.” Built-in mods keep running even under `--safe-mode`. The author reads anyone else’s mod code before installing it [1].

php2shell: models found the bugs, and agents are helping patch them

PHP shipped fixes across the whole 8.x line for two RCEs. Exploits were already circulating after the disclosure leaked [2]. Accounts of who found the bugs differ: - **Greg Brockman (OpenAI):** Calif researchers used GPT-5.6 Sol to find a version of the bug in August, and OpenAI’s team independently found a more complete version in September [3]. - **Wiz:** its timeline says Wiz found the first vulnerability on Aug 19 using Wiz Atlas, Gemini 4 Argon and GPT-5.6 Sol [2].

Brockman says OpenAI is also working on other bugs, including a Node.js RCE announced the same day. He expects “many more” in the coming months and says the slow, manual one-bug-at-a-time process “won’t scale for the agentic era” [3].

What you can do: - **Find stale version pins across your repos.** Sourcegraph says the hard part is finding every Dockerfile, CI pipeline and Helm chart pinned to an old PHP version [4]. It published search queries and **Agentic Batch Changes** specs for finding and bumping them [5]. - **Audit your own machine.** One security PSA suggests asking Claude Code to check what on your laptop is out of date [6]. - **Consider fil-C.** Armin Ronacher suspects there are more bugs like this and suggests running PHP via fil-C if you can afford it [7].

Agent tooling is also exposed. An October incident roundup lists exposed OpenCode AI API keys and malicious LiteLLM PyPI and Docker releases [8]. Theo took his Railway-hosted apps offline because Railway was blocking deploys, so he couldn’t redeploy with rolled keys [9]. After hours of looking, he found no drop-in replacement, and the cheapest equivalent setup would cost 3x more [10].

Boris Cherny’s Claude Code playbook

In this talk, Claude Code creator Boris Cherny lays out a progression for adopting it: - **Start with codebase Q&A, not edits.** Ask how a class is instantiated, or have it “look through git history” to explain why a function has 15 arguments [11]. He says onboarding for technical hires at Anthropic dropped from 2–3 weeks to 2–3 days [11]. - **Ask for a plan before code.** Tell it: “before you write code, make a plan.” Plan mode isn’t required [11]. - **Give it a way to check its work,** such as tests or Puppeteer or iOS-simulator

screenshots. With two or three rounds of iteration, a UI built from a mock often comes out almost perfect [11]. - **Keep a short, shared CLAUDE.md** with common commands, MCP tools, architecture notes and key files. Check in `.mcp.json`; Anthropic’s apps repo shares a Puppeteer MCP server this way [11]. - **Use `claude -p` for automation.** Pass allowed tools and get JSON or streaming-JSON output, and pipe it like a Unix utility in CI, incident response and log triage [11]. Power users run several checkouts or git worktrees and connect to sessions over SSH and tmux [11].

Context management in LangChain’s Deep Agents

The Deep Agents harness keeps big content out of the context window by **off-loading** it. Tool results over 20k tokens are saved to a virtual filesystem; the model sees the first and last few lines and can page through the rest with a read-file tool [12]. Human messages over 50k tokens get the same treatment [12]. **Summarization** is checked before every model call and triggers at 85% usage by default (adjustable). Usage can go past 85% when one turn makes several tool calls [12].

LangChain’s own design agent uses this stack. It has shipped 3,000+ assets, with designers’ judgment encoded as skills, a design wiki and brand checks on every render, and it runs on Managed Deep Agents [13].

Proactive agents and long-lived apps

- **Claude Duet** (research preview, waitlist) is an app where Claude works proactively: it brings you what needs attention, keeps working while you’re away, and checks with you before acting on your behalf [14]. Addy Osmani says it runs on Opus 5.5 with 6,000 connectors and plugins [15]. Kent C. Dodds calls Grok Bot, OpenAI Dot, Meta Muse and Duet “walled gardens” [16].
- **MiniMax Code Mini Apps** are long-running apps with a frontend and a Node.js backend that expose their features to the agent through hosted MCP [17]. `/miniapp-creator` first runs read-only queries to confirm it can get real data, then builds and publishes the app and checks it with screenshots [17].
- **Riley Brown** spends about 45 minutes each morning talking to Dot by voice while walking. Dot handles small tasks and starts Codex sessions for longer ones; at his desk he coordinates work through Claude Code Projects [18].

Models and providers

- **Corridor’s argos-1** is a Qwen 3.5 9B fine-tune. Corridor says it beat GPT-6 Astra and Claude Opus 5.5 on its security-context-selection test of 14 models, with the lowest time to first token [19]. Its training data was built with smithtune’s trace-based workflow [20].

- **Fireworks** claims DeepSeek V4.1 Flash 1017 runs at more than 1,521 tok/s on its Ultrafast tier, which is waitlist-only [21].
- **Geoffrey Huntley** is cancelling eight ChatGPT subscriptions. He cites poor availability and slow inference rather than model quality, and says Claude is currently better on all three counts [22].

Sources

1. Claude Mods: Flipping the Harness Inside Out
2. Today's PHP Security Release, and How We're Strengthening PHP Security
3. X article by @gdb
4. X post by @Sourcegraph
5. X post by @Sourcegraph
6. X post by @peterwildeford
7. X post by @mitsuhiko
8. X post by @juampitech
9. X post by @theo
10. X post by @theo
11. Stop guessing the “right” way to use Claude Code - learn directly from the engineer who created it.
12. How Deep Agents Manages Context: Offloading and Summarization
13. X post by @amal_ircashev
14. X post by @claudeai
15. X post by @addyosmani
16. X post by @kentcdodds
17. X article by @Ronny_MiniMax
18. X post by @rileybrown
19. X post by @corridor
20. X post by @LangChain
21. X post by @FireworksAI_HQ
22. X post by @GeoffreyHuntley