SECURITY INCIDENT

Claude Code Source Map Leak

Key Findings, Unreleased Features & Lessons for Agentic Development

Mohamed Noordeen • AWS GenAI Innovation Center • 2025

THE INCIDENT

What Happened?

An accidental npm publish exposed Claude Code's entire readable source via a source map file.

The Leak

Anthropic accidentally shipped a .js.map source map file in npm package version 0.0.88 of Claude Code.

The .npmignore file — which should have excluded this file — was misconfigured during a rapid release cycle.

Source maps reverse obfuscated code back to its original readable form, exposing internal logic, unreleased features, and architecture.

Anthropic's Response

Confirmed it was a human error, not an AI-generated mistake.

Stated they blame the system for lacking safeguards — not the individual engineer.

Version 0.0.88 was pulled from npm, but the source map had already been downloaded and analyzed by the community.

TECHNICAL BREAKDOWN

How It Happened

From TypeScript source to public exposure — a four-step failure chain.

1

TypeScript Source

Claude Code is written in TypeScript — readable, structured code.

2

Obfuscation & Minification

Before publishing, the code is minified and obfuscated to protect IP.

3

Source Map Generated

A .js.map file is created to reverse obfuscation — for internal debugging only.

4

.npmignore Failure

The ignore file failed to exclude the .map file. npm publish shipped it publicly.

Analogy: Similar to accidentally pushing your .env file (API keys) to GitHub — except this exposed the entire readable codebase.

THE REBUILD

The Open-Source Rebuild

A developer rebuilt Claude Code as open source in under 24 hours using AI-powered coding agents.

What Was Built

A developer ("UltraWorkers") used the leaked source map to rebuild Claude Code as a fully open-source project.

Initially built in Python, then ported to Rust within hours.

Gained 100K+ GitHub stars in under 24 hours, surpassing established projects like OpenCode (~135K stars).

Built using OpenAI Codex with GPT models, leveraging team-mode parallelism and the RALF Loop iterative plugin.

130K+

GitHub Stars in ~24 Hours

How It Was Done

- Team Mode — parallel agents on separate features
- RALF Loop — iterative auto-fix cycles
- Est. $50K+ in API tokens burned

UNRELEASED FEATURES

Features Discovered in the Leak

Internal codenames and capabilities found in the leaked source map.

kiron Background Agent

Persistent agent that runs across sessions — monitors logs, triggers proactive actions, and continuously tests code while the developer is away.

Dream Mode Idle Sub-Agent

Runs while the user is idle. Merges observations, resolves contradictions, converts vague insights into concrete facts. Read-only bash access.

Mythos Ultra Plan / Remote Sessions

Offloads complex planning to remote cloud sessions for more sophisticated multi-step reasoning and orchestration.

Undercover Attribution Removal

Removes traces of AI involvement from commits and contributor lists, making code appear fully human-authored.

COMPETITIVE LANDSCAPE

Agentic Coding Landscape

Claude Code in context — terminal-based coding agents and alternatives.

Tool	Type	Key Strength	Stars
Claude Code	Proprietary (Anthropic)	Deep Anthropic model integration	N/A
OpenCode	Open Source	Most popular OSS coding agent	~135K
Codex (OpenAI)	Open Source	Lightweight, extensible	~72K
Conductor	Commercial	Parallel agent execution, better UI	N/A

Key Insight: According to TerminalBench, Claude Code ranked ~39th despite being the most popular agent. Opus 4.6 leads the benchmark — suggesting agent design, not just model quality, is a key differentiator.

TAKEAWAYS

Key Takeaways

What this leak teaches us about security, speed, and the future of agentic development.

Security	Automate your guardrails. Never rely on manual ignore files alone — add CI/CD checks that block source maps, .env, and credentials from being published.
Speed	AI-powered development is real. A single developer rebuilt a complex product in ~24 hours using team-mode agents and iterative loops.
Transparency	Leaks cannot be unleaked. Once code ships publicly, competitors can analyze architecture, features, and strategy.
Direction	Background AI agents are coming. Features like Dream Mode and Chiron signal a shift toward agents that work autonomously — continuous testing, QA, and improvement.