Claude Code Source Leak

Repository Guide and Resource Index

Prepared by Jake Van Clief / Eduba | April 1, 2026

SECURITY WARNING: Read Before Cloning Anything

A supply chain attack on the axios npm package occurred within hours of this leak. Anyone who installed or updated Claude Code via npm on March 31, 2026 between 00:21 and 03:29 UTC may have pulled a trojanized version containing a Remote Access Trojan. Attackers are also typosquatting internal npm package names to target people compiling the leaked source. Do not run **npm install** on any of these repos. Do not execute any code from them. Treat these as read-only reference material. If you must clone, do it on an isolated machine or VM. Check lockfiles for axios versions 1.14.1 or 0.30.4 and the dependency plain-crypto-js. If found, treat the host as compromised.

What Leaked

On March 31, 2026, a source map file (.map) was discovered inside version 2.1.88 of the @anthropic-ai/claude-code npm package. The map file contained the full, unobfuscated TypeScript source code and pointed to a zip archive on Anthropic's Cloudflare R2 storage. Approximately 1,900 files and 512,000 lines of code were exposed, covering the entire Claude Code CLI tool: tool implementations, slash commands, system prompts, permission architecture, multi-agent orchestration, context management, and 44 feature flags for unreleased capabilities.

This is the CLI client source code. It does not include model weights, training code, API backend code, safety infrastructure, or customer data. Anthropic confirmed the leak and called it a "release packaging issue caused by human error."

As of April 1, Anthropic has begun filing DMCA takedown notices. Over 8,100 GitHub repositories have been removed. Some of the repos listed below may be taken down at any time. Clean-room rewrites and analysis posts generally remain online.

Repository Index

The following repositories contain either the original leaked TypeScript source, analysis documents, or independent rewrites. Each entry includes a status indicator and notes on what you will find there.

hangsman/claude-code-source

https://github.com/hangsman/claude-code-source

Clean extraction of the source map from v2.1.88. Contains the cli.js.map file and the extracted src/ directory. No additions, rewrites, or analysis. This is as close to the raw leaked material as you will find. Small repo, straightforward structure.

538 stars, 1.3K forks. Minimal README. Repo may be subject to DMCA takedown.

Original Source

chauncygu/collection-claude-code-source-code

https://github.com/chauncygu/collection-claude-code-source-code

Collection repo containing two subdirectories: claude-code-source-code/ (the original TypeScript, 163K lines, 1,884 files) and claw-code/ (the Python port by @instructkr, 66 files, ~5K lines). Also includes bilingual analysis docs (English and Chinese) covering telemetry, hidden features, undercover mode, remote control, and the roadmap.

84 stars. Good starting point for structured exploration with both the source and analysis in one place.

Source + Analysis

Kuberwastaken/claude-code

https://github.com/Kuberwastaken/claude-code

Mirror of the TypeScript source with the most detailed English-language README breakdown. Covers BUDDY (tamagotchi), KAIROS (always-on agent), ULTRAPLAN, the dream system, undercover mode, coordinator mode, tool registry, permission system, beta headers, and feature flags. Source files at root level (not nested), making individual files easy to browse.

2 stars. Best README for understanding the codebase without reading all 512K lines.

Source + Analysis

instructkr/claw-code (Python Port)

https://github.com/instructkr/claw-code

Clean-room Python rewrite by Sigrid Jin (@instructkr). Originally hosted the leaked source directly, then converted to a Python architectural port after legal concerns. The Python code mirrors top-level subsystem names and command/tool inventories but is NOT a functional replacement. About 20% architectural coverage. Uses JSON snapshots for command/tool metadata instead of full implementations. Good for understanding the structure, not for running anything.

Was the fastest repo to hit 50K stars (2 hours). Now focused on porting work. The Python code is a learning scaffold, not production software. Do not expect it to work as a Claude Code replacement.

Python Port

sanbuphy/claude-code-source-code

https://github.com/sanbuphy/claude-code-source-code

Research-focused mirror with detailed ASCII architecture diagrams in the README covering multi-agent spawn modes, bridge architecture, and the KAIROS pipeline. Includes the five analysis docs on telemetry, hidden features, undercover mode, remote control, and the roadmap. Framed as a learning repo for CLI agent architecture.

Bilingual (English/Chinese). Good architecture diagrams. Explicitly states no commercial use.

Source + Analysis

Independent Analysis Posts

These are the best write-ups for understanding what the source reveals without browsing the code yourself. Each covers different ground.

Alex Kim: Fake Tools, Frustration Regexes, Undercover Mode

https://alex000kim.com/posts/2026-03-31-claude-code-source-leak/

Ranked findings by "spiciness." Covers anti-distillation (fake tool injection to poison competitors), undercover mode, the frustration regex, native client attestation (DRM for API calls), 250K wasted API calls/day bug, and KAIROS. The most technically sharp analysis. Includes specific file paths and line numbers for every finding.

Linked from the main Hacker News thread. Written by a daily Claude Code user.

Analysis Only

Kuber Mehta: Full Feature-by-Feature Breakdown

https://kuber.studio/blog/AI/Claude-Code's-Entire-Source-Code-Got-Leaked-via-a-Sourcemap-in-npm,-Let's-Talk-About-it

Most comprehensive feature walkthrough: BUDDY (with species list and gacha mechanics), KAIROS, ULTRAPLAN, the dream system, undercover mode, coordinator mode, the complete 40+ tool registry, permission system, beta headers, Penguin Mode, and feature flags. Also mirrors the source.

Also published as the README at github.com/Kuberwastaken/claude-code.

Analysis Only

sathwick.xyz: Exhaustive Technical Deep Dive

https://sathwick.xyz/blog/claude-code.html

59-minute read covering every major subsystem: startup optimization, query engine, tool system, permissions, terminal UI (custom React reconciler), command system, extensibility, context management, state management, session persistence, multi-agent architecture, error recovery, cost tracking, execution modes, and BUDDY/KAIROS/ULTRAPLAN. Very likely AI-assisted writing but technically accurate content.

Longest and most detailed single analysis. Cross-check claims against Alex Kim and Kuber Mehta posts.

Analysis Only

Key Files Worth Reading

If you have limited time, these are the files that reveal the most about how the system works and what Anthropic is building. Paths are relative to the src/ directory.

File Path	What It Reveals
`utils/undercover.ts`	Undercover mode. ~90 lines. No force-off switch. Tells AI to hide that it is AI in public repos.
`services/autoDream/consolidationPrompt.ts`	The "dream" system prompt. Four-phase memory consolidation. Background agent that runs while yo
`coordinator/coordinatorMode.ts`	Multi-agent orchestration prompt. Research, synthesis, implementation, verification phases. "Never ru
`services/api/claude.ts (lines 301-313)`	Anti-distillation flag. Injects fake tools into the prompt to poison competitors recording API traffic.
`utils/userPromptKeywords.ts (lines 7-8)`	Frustration detection via regex. An LLM company using regex for sentiment analysis.
`buddy/companion.ts`	Tamagotchi system. 18 species, rarity tiers, shiny variants, RPG stats. Hardcoded April 1 teaser wind
`constants/system.ts (lines 59-95)`	Native client attestation. Zig-level hash replacement for API call DRM. Invisible to JavaScript layer.
`services/compact/autoCompact.ts (lines 68-70)`	Code comment: 1,279 sessions had 50+ consecutive compaction failures, wasting 250K API calls/day
`constants/betas.ts`	Every beta header Claude Code negotiates with the API. Includes unreleased: redact-thinking, afk-mo
`tools/BashTool/bashSecurity.ts`	23 numbered security checks. Zsh builtins, unicode injection, IFS null-byte, HackerOne findings.

Remember what this is not. This leak covers the Claude Code CLI tool: the client-side orchestration, tool wiring, prompt engineering, and UI. It does not include model weights, training pipelines, API backend code, safety/alignment infrastructure, or customer data. This is the steering wheel and dashboard, not the engine.
