WILSON AI LAB · DEEP DIVE REPORT

# Claude Code 原始碼外洩事件給我們上的5堂課

50萬行TypeScript被攤在陽光下，
我們看到了迷思、真相、以及最強實戰技巧。

**WilsonAILab**

SKOOL EXCLUSIVE

2026.04 · VOL.01

事件懶人包

# 30秒搞懂發生什麼事

**事件時間軸**

2026年3月底，Anthropic工程師在發布Claude Code的npm套件v2.1.88時，**誤把Source Map打包進去**，讓超過**50萬行TypeScript原始碼**直接外洩。開源社群迅速移除了遙測追蹤，編譯出解鎖版「Free Code」，也讓Claude Code的內部架構完全攤在陽光下。

ONE-LINER

**這不是災難，是一堂免費的「AI工具怎麼設計才夠強」公開課。**

本篇目錄

## 你會學到的5件事


01 **AI會「偷偷降智」是真的嗎？**
原始碼給出的明確答案，以及你該注意的真正關鍵。

02 **未來模型情報：Mythos與百萬Token**
Opus 4.7、Sonnet 4.8、Mythos旗艦款的內部代號曝光。

03 **AI工具的安全機制長這樣**
6層安全防線、影子分類器、受保護路徑。

04 **3個被低估的實戰技巧**
claude.md、斜線指令、萬用字元權限。

05 **封號追蹤機制大揭密**
官方追什麼？不追什麼？一次講清楚。

LESSON 01

# AI真的會偷偷「降智」嗎？

這是Claude Code圈最常見的抱怨，原始碼給了答案。

每隔一陣子就會有人在論壇或Threads抱怨：「我今天用Claude感覺變笨了，是不是Anthropic偷偷換成弱模型？」這次外洩的原始碼把這個迷思徹底破解。

**❌ 迷思版本**

「我付了Opus的錢，但Anthropic會在背景偷偷換成Haiku來省成本，讓我的對話變笨。」

**✅ 原始碼告訴我們**

主對話用的強模型**絕對受用戶控制**。弱模型（Haiku）只會在背景做輕量輔助任務（例如對話摘要）。

**🎯 那為什麼會「感覺變笨」？**

在極端負載狀況下，系統**只會降低「思考級別」**——例如把Max模式降到預設模式，**不會偷換模型本體**。這兩件事是完全不一樣的。思考級別影響的是「推理深度」，模型身分始終是你選的那個。

**💡 給一般人的啟發**

下次覺得AI「變笨」時，先檢查：
① 是不是上下文塞太滿觸發自動壓縮？
② 是不是你給的指令本身就模糊？
③ 是不是剛好命中了降低思考級別的狀況？
**主模型不會被偷換，但「思考深度」確實會被動態調整。**

LESSON 02

# 下一代旗艦模型：Mythos

原始碼裡寫死了還沒發表的模型代號，完全藏不住。

這次外洩最勁爆的部分，是Anthropic還沒公開的內部模型名稱全部被挖出來：

| 代號／名稱 | 狀態 | 備註 |
|---|---|---|
| **Mythos** | 下一代旗艦 | 百萬Token上下文、雙重推理 |
| **Opus 4.7** | 內部測試中 | 當前Opus的下一版 |
| **Sonnet 4.8** | 內部測試中 | 當前Sonnet的下一版 |
| Fennec | 現役Opus代號 | 沙漠狐狸 |
| Capybara | 現役Sonnet代號 | 水豚 |
| Tengu | 現役Haiku代號 | 天狗 |

KEY INSIGHT

**Mythos將支援100萬Token上下文，並具備「快速推理」與「完整推理」雙模式切換。**

### 🎁 彩蛋：/buddy 虛擬寵物系統

原始碼裡還挖出一個叫 `/buddy` 的AI寵物系統，包含**18種物種、6種眼睛款式、8種帽子、5種稀有度**，每個寵物還用確定性亂數演算法生成外觀。Anthropic官方已證實這是2026年4月1日的愚人節彩蛋。

### 📊 為什麼這重要？

百萬Token意味著一次能塞進**整套技術書、完整程式碼庫、或幾百篇論文**進去。對知識工作者來說，這是工作流的質變——你不再需要先做RAG切片，可以直接把整個專案丟進去問。

LESSON 03

# AI工具的安全機制長這樣

6層防護、影子分類器、受保護路徑——這些都是為了保護你的電腦。

Claude Code最核心的**BashTool（執行Shell指令）**，居然有高達6層安全防線。這也解釋了為什麼它敢放給你自動化：

01 **Shell語法分析**——防止命令注入（例如 `$()`、反引號）

02 **Zsh專項防護**——阻擋zmodload、syswrite等冷門漏洞

03 **輸入消毒**——過濾Unicode空白字元、IFS注入攻擊

04 **路徑保護**——UNC路徑全面封鎖，防NTLM竊取

05 **macOS沙盒**——可按指令類型動態開啟

06 **ML分類器**——用AI推理判定指令風險等級

SHADOW GUARD

**在Auto模式下，背景有一個獨立AI「YOLO」專門審查主AI的每個指令——放行、確認、或攔截。**

### 🔒 絕對受保護的路徑清單

無論你怎麼下指令，Claude Code都**絕對不會碰**這些檔案：

`.gitconfig` `.bashrc` `.zshrc` `.mcp.json` `.claude.json` `.git/` `.vscode/` `.idea/`

### 💡 Anthropic放棄了RAG

這點很反直覺：Claude Code**沒有用向量資料庫**，直接用最原始的 `grep` 做文字搜尋。工程師的判斷是：「模型夠強，就不需要複雜的檢索層」。這是一個重要的工程哲學訊息。

LESSON 04 · A

# claude.md是「最高指導手冊」

90%的人把它當筆記本，但它其實是憲法。

大部分人在 `claude.md` 裡寫的是「這個專案的背景」、「我用什麼框架」——這完全浪費了它的威力。**頂尖開發者會把它當成「不可違背的規範」來寫。**

**❌ 一般人的寫法**

「這是一個React專案，使用Next.js 14。後端用Node.js。」

**✅ 高手的寫法**

「一律使用TypeScript嚴格模式。只能用PNPM，不准用npm/yarn。禁止任何any型別。」

**差別在哪？**第一種只是「告知」，第二種是「命令」。原始碼顯示Claude Code會把 `claude.md` 當成系統提示詞的一部分動態載入，也就是說——你寫的規則會凌駕在它的預設行為之上。

**📝 高效claude.md範本**

```
# 開發規範（不可違背）
- 所有程式碼必須用TypeScript嚴格模式
- 禁止使用any，必要時用unknown+type guard
- 套件管理只能用PNPM
- 每個function必須有JSDoc註解
- 測試框架用Vitest，不准用Jest

# Git規範
- Commit格式：<type>: <description>
- type只能是：feat/fix/refactor/docs/test

# 專案背景
- 這是B2B SaaS，重視穩定性甚於速度
- 使用者多為非技術人員，UI要簡單
```

**⚡ 一句話記住**

**claude.md不是給Claude「看」的，是給Claude「遵守」的。**用命令句、不用說明句。

LESSON 04 · B

# 3個必學的斜線指令

Claude Code內建85個指令，但這3個是效率分水嶺。

| 指令 | 用途 | 什麼時候用 |
|---|---|---|
| `/plan` | 先規劃再動手 | 要改多個檔案時，強迫它先出計畫，避免亂改 |
| `/compact` | 壓縮對話記憶 | 對話變長、回答變慢時，手動壓縮省Token |
| `/review` | 專屬程式碼審查 | 寫完功能後，讓它用審查視角抓問題 |

TIP

**養成「先 /plan 再動手」的習慣，程式碼災難可以減少80%。**

## 萬用字元權限：無人值守的關鍵

Claude Code每次要執行指令都會跳出「確認嗎？」——這對自動化是災難。解法在 `settings.json`：

```
// ~/.claude/settings.json
{
  "permissions": {
    "allow": [
      "Bash(git *)",          // 所有git指令
      "Bash(pnpm *)",         // 所有pnpm指令
      "Edit(src/**)",         // src資料夾可編輯
      "Read(**)"              // 全部可讀
    ]
  }
}
```

> ⚠️ **安全提醒**
>
> 萬用字元權限很方便，但**千萬不要對 `Bash(*)` 開全權限**。明確列出允許的指令類別，才不會踩雷。

LESSON 05

# 封號追蹤的真相

官方追什麼？不追什麼？原始碼一次講清楚。

很多人擔心被封號後連「換電腦」都無法重來，以為Anthropic會抓MAC地址、硬體指紋。原始碼證實：**這些都不會抓。**

### ✗ 官方不會抓

- MAC地址
- CPU/主機板序號
- 硬體規格指紋
- 螢幕解析度等裝置特徵

### ✓ 官方會追蹤

- 隨機生成的User ID
- OAuth綁定的信箱
- 專案倉庫（Repo）的Hash值
- 使用行為模式

### 🧠 Auto Dream：背景記憶整理

Claude Code有一個**四層上下文壓縮架構**。當專案靜置24小時後，會有背景Agent自動整理記憶。**重點是：它只記「偏好」和「背景」，不硬記程式碼行數位置**——這樣你重構程式碼時才不會因為記憶錯位造成bug。

WORKFLOW INSIGHT

**Claude Code 60%的強大來自模型，40%來自「工程系統」。這40%才是你該學習的。**

### 🔑 Wilson的一句話總結

這次外洩最大的收穫**不是「學會怎麼規避追蹤」**，而是理解——一個頂級AI工具的設計思維是什麼？為什麼它可以做到這麼穩、這麼安全？這些知識才是你未來自己打造工具的基礎。

看完這份PDF，你該做的事

# 7個立刻能執行的行動

知道沒用，做了才算。

01 **打開你的claude.md**——把「描述句」改成「命令句」，加入強制規範。

02 **養成先 /plan 的習慣**——改多個檔案前，永遠先要它出計畫。

03 **對話變長時 /compact**——不要等系統自動壓縮，提前做更省Token。

04 **設定萬用字元權限**——在settings.json列出你常用的指令類別。

05 **定期檢查 .claude/ 資料夾**——了解Claude存了什麼、避免敏感資訊外洩。

06 **不要用Auto模式跑重要任務**——雖然有YOLO分類器，但重要任務還是要人盯著。

07 **追蹤Mythos發表時間**——百萬Token旗艦款會改變工作流，提前準備。

FINAL WORDS

**別再把Claude Code當成「會寫程式的聊天機器人」——它是一個有工具系統、有權限引擎、有記憶架構的AI工作代理。**

📚 **延伸學習**

想深入研究原始碼細節的朋友，可以去B站搜尋「AI超元域」頻道看完整技術分析，或是到Wilson AI Lab Skool社群，有我整理的「Claude Code完整工作流」教學。

# 想要更多
# AI工具實測？

每週更新最深度的AI工具拆解、實戰教學、與業界內幕。
加入Wilson AI Lab，讓你永遠走在AI工具的最前線。

**FOLLOW WILSON**

| | |
|---|---|
| **YouTube** | youtube.com/@WilsonAILab |
| **Skool** | skool.com/wilson-9076 |
| **Instagram** | instagram.com/wilsonailab |
| **Threads** | threads.com/@wilsonailab |

本報告基於2026年3月底至4月1日期間Claude Code原始碼外洩公開資料整理，僅供教育與研究用途。所有商標與產品名稱屬於其各自所有者。資料來源：公開社群分析、AI超元域B站頻道。