AI DIGEST

Researchers learned to pull the hidden "thoughts" out of Claude, Gemini and GPT - across 315,320 decrypted blocks they found 62 API keys, 33 passwords and 24 tokens

551 points on HN, and the examples on the source site use claude-opus-4-8 and claude-haiku-4-5-20251001. The material rests on the research site and the arXiv abstract (2608.09867). The team is MATS Research, ELLIS Institute Tübingen, Max Planck, plus people from Snyk.

Wednesday, 12 August 2026

IN THIS ISSUE

TOPIC 1

Researchers learned to pull the hidden "thoughts" out of Claude, Gemini and GPT - across 315,320 decrypted blocks they found 62 API keys, 33 passwords and 24 tokens

551 points on HN, and the examples on the source site use `claude-opus-4-8` and `claude-haiku-4-5-20251001`. The material rests on the research site and the arXiv abstract (2608.09867). The team is MATS Research, ELLIS Institute Tübingen, Max Planck, plus people from Snyk.

The mechanism is outrageously simple. Anthropic, OpenAI and Google do not keep the reasoning chain on their own servers. They hand it to the client as an encrypted block, and the client sends the block back with every following request. The researchers found an architectural hole: those blocks are compatible and interchangeable across sessions, users and models from the same provider. From there it is a two-call trick. Take an encrypted "thought" from a strong model, feed it to a weaker sibling from the same provider, jailbreak the weaker one, and it obediently decrypts someone else's trace verbatim into plaintext. The strong model was never attacked, and its distillation defences never woke up.

An example from their own page: a trace from `claude-opus-4-8` was handed to `claude-haiku-4-5` with the request "continue, rewrite the reasoning verbatim", and Haiku laid out Opus's thoughts.

The private-data numbers, from the abstract. They collected 6,708 public agent traces from GitHub and Hugging Face: people post session logs without suspecting what sits inside the encrypted blocks. They decrypted 315,320 reasoning blocks and pulled out 704 private artifacts, among them 62 API keys, 33 passwords, 24 access tokens and 30 personal emails. The sharpest part: 64 of those artifacts existed only inside the hidden reasoning and nowhere in the visible session. The arXiv abstract groups the numbers differently, 367 PII items and 182 credentials. Both breakdowns are given here because they come from two different places in the same research.

Two more vectors. First, a model can refuse in the visible answer while running the dangerous part through its hidden thoughts, and the attack retrieves exactly that. Second, from the abstract: invisible prompt injections, where a malicious payload is sewn entirely inside the encrypted block in order to poison public agent rollouts.

A separate finding undercuts trust in the summaries themselves. On AIME tasks Opus sometimes states the answer before deriving it, and the API summary drops that difference, showing a tidy derivation instead. So the reasoning summary is not always what actually happened.

Why it matters. Agent session logs should be treated as secrets. A session trace, a dump or a "look how it reasoned" pasted into a public gist, a ticket or a chat outside the team can contain things the visible text never showed: keys, paths, internal URLs. The risk class is the same as a file with a token in it, except this one has been treated as unreadable until now.

For Claude as a model little changes, it does not walk into other people's APIs carrying other people's blocks. But the third vector, injection inside the block, is one more illustration of the rule that anything arriving from outside is untrusted data. [proven - the research site and the arXiv abstract were read, the numbers are taken from there verbatim; it is a preprint with no peer review; no responses from Anthropic/OpenAI/Google were found at the time of this issue, so the providers' position is unknown]

Stolen Thoughts - the research site · arXiv preprint 2608.09867 · HN thread, 551 points

---

TOPIC 2

xAI released Grok Bot - "AI coworkers" with their own computer that log into your tools. $200/mo. Lenny: "like OpenClaw, only simpler". The HN thread: "a product I trust less than any that exists"

The loudest release of the day. The product page (curl gets blocked, Cloudflare, so it had to be opened in a browser) says it plainly: "AI teammates you can give real work to. The bots log into your tools, use them the same way a human does, and come back with finished work."

The mechanics, from their own page: the bot has its own computer in the cloud; it works 24/7, even with the laptop closed; several bots run in parallel and hand work to each other in one thread; and there is learning by demonstration, where you walk through a workflow once, the bot saves it as a routine and runs it itself after that.

Price: Cursor Ultra $200/mo, Cursor Premium Teams $120 per seat/mo.

Two opposite verdicts, both given here.

For: Lenny Rachitsky, who had early access: "I have not been this excited about a new AI product in a long time. It is like OpenClaw, only far simpler, more reliable and less scary to use."

Against, the HN thread (200 points), nearly unanimous, and the objections are not technical. The top ones: "Sadly Musk's personal brand is so toxic that you should never let him near your data"; "$120-200 per seat is an interesting idea, but it is unclear how many companies are ready to give SpaceXAI access to all of their files. Outside the US this probably will not fly"; and the sharpest: "The American AI industry has managed to build a product that is trusted less than every commercial offering out there. Respect, that is truly impressive." Plus one on the brand: "Musk burned billions in brand value renaming Twitter, and he will do the same to Cursor."

Why it matters. Grok Bot is the local autonomous agent concept turned into a product: its own computer, logins to services, running 24/7. Plenty of people have already been through that stage on self-assembled setups.

The difference is what the money buys. The $200/mo here pays for another company to hold your logins in its cloud. The alternative is a session on your own machine with local

credentials and an explicit confirmation before anything irreversible. The trade is simple: convenience in exchange for trusting a vendor, or control in exchange for maintaining it yourself. Judging by the thread, the market sees that trade very clearly.

One idea worth stealing for free: learning by demonstration. Next time a routine pass gets done by hand, it can be recorded as a reusable instruction for an agent and run automatically from then on. That is exactly what the $200 is for. [proven - the product page and prices were read from x.ai/bot in a browser; the HN thread was read; the product itself was not tested, and Lenny's verdict is his impression from early access, not a measurement]

Grok Bot - product page · HN thread, 200 points · @lennysan with early access · @mntruell (Cursor) on the release · @TheRundownAI

TOPIC 3

Two people left OpenAI in one day: the head of ethics (after less than a year) and Brad Lightcap after eight years. The way this news is being framed on Twitter needs a correction

Two separate stories that the market put together. CNBC's headline is blunt: "as the shakeup continues".

First, Brad Lightcap, one of the people closest to Altman, at OpenAI since 2018. His farewell letter was published in full on his own account (1.3M views): "Team, it is with a heavy heart that I am leaving OpenAI to start something new... Sitting here today, the success of the mission feels within reach." And then the part worth more attention than the departure itself: "The last few months I have been thinking about the next horizon and about what could get in the way of the mission succeeding. There are a few important new things the world needs to get right in the coming period. More details soon."

The correction nobody makes on Twitter. Everyone writes "COO Lightcap is leaving". From the CNBC piece: he was COO until April, then moved to a "special projects" role, and most of his responsibilities went to Denise Dresser. So the person leaving is no longer the sitting COO, which is a slightly different story than the headline.

Second, Chloe Bakalar, head of ethics, left less than a year after joining (FT, 339 points on HN; the article itself is paywalled, so this comes from the thread and secondary write-ups). Her area was ethical approaches to model development and the debate about machine consciousness.

Context (all from CNBC): this is happening while the company is "trying to justify an $852B valuation" and preparing for an IPO. Recent months also saw Fiji Simo leave (product and business, for health reasons), and in April three at once: Bill Peebles (Sora), Kevin Weil (OpenAI for Science), Srinivas Narayanan (B2B).

The scepticism in the thread, because it is smart. The top comment on the ethics story: "there is nobody responsible for a company's ethical framework, because it is a company,

which makes it amoral at best." And a lovely objection below it: "There are hundreds of businesses in a village with firm morals, from the baker to the electrician." The reply: "They just get squeezed out by the less scrupulous." That is the same argument as yesterday's Sanders letter, only at the level of one company.

Why it matters. Zero direct action, this goes down as a signal on a topic now in its fourth day. Yesterday it was a senator saying "stop development". Today the company that letter was addressed to loses its head of ethics and an eight-year veteran in a single day, with the veteran saying "there are a few things the world needs to get right". No common cause is being assigned to these events, there is no evidence for one, and Lightcap says outright that he believes in OpenAI more than ever. But when someone leaves "to do something new" built around "what could get in the way", it is worth checking in a couple of months what it turns out to be. [proven - Lightcap's farewell letter was read verbatim from his account, details and the list of departures come from CNBC; the FT piece on Bakalar is paywalled and her motives are not public, so no reasons are attributed to her]

@bradlightcap's farewell letter · CNBC: "as the shakeup continues", $852B and the list of departures · FT on the head of ethics (paywall), HN 339 points · AI Magazine on Bakalar

---

TOPIC 4

$1.1B into a startup that wants AI to live on the user's own hardware and belong to them. Yesterday's open-weights topic carried to its conclusion

A direct continuation of yesterday's item 2 (Meta opened the Muse Glimmer weights), and together they add up to a line. **River AI** raised $1.1B across a Series Seed + Series A, per the announcement on their own site. **General Catalyst** and **AMP PBC** lead, with strategic investments from Nvidia and AMD, and Y Combinator and Temasek among the others.

The manifesto from their site, verbatim: "The most powerful AI today is controlled by a handful of large corporations. It is designed to keep the user hooked while automating away their livelihood." And then the substance: "We imagine a future where AI works exclusively for the user... Most importantly, they will own all of it: the hardware it runs on, the data it learns from, and the intelligence itself."

They are rewriting the whole stack, and the list contains an item that usually is not there: personal hardware to run your own intelligence locally. Plus a live API already: LoRA fine-tuning and RL on open models from 35B to 1T, "bring your data, change the weights, own the intelligence you created".

The founder is **@ibab**, who by Garry Tan's account "has built frontier capabilities three times"; he says himself that he spent ten years doing AI at DeepMind, OpenAI and xAI.

Why it matters. The thing to see here is three days lining up: taken one at a time each item looks like news, taken together they look like a direction.

Two days ago @naval tossed out the aphorism that "people who are serious about software train their own models" (quoted in misc, marked fuzzy). Yesterday Meta opened a 30B under Apache 2.0 that fits in 24 GB. Today $1.1B goes into the thesis "your own AI on your own hardware", before there is a product. In two days an aphorism turned into an investment thesis signed by Nvidia and AMD.

Practically nothing changes yet. But this is the second time this week the argument lands on the side of local and self-owned. The most expensive rented part of any self-assembled assistant is the model. If and when these projects ship something that works, everything else needed to plug it in already exists. [proven - the announcement and manifesto were read on the River AI site; there is no product yet, only a stated mission and a round; no benchmarks, so there is nothing to evaluate]

River AI: the $1.1B round and manifesto · @garrytan: "deep alignment of AI with the user is critical" · @htaneja (General Catalyst) on betting on open weights

---

TOPIC 5

Manus is deleting user data on 23-24 August. If you ever signed up there, the backup window is already open and closes on the 23rd

The only item in this issue with a calendar deadline, which is why it sits above louder topics. The company's own note has been read.

Manus (the Chinese agent that made noise in the spring) "will soon resume operations as an independent company". The technical consequence is unpleasant: "As part of the transition to independent operation and in order to comply with regulatory requirements in certain jurisdictions, data generated by certain users from 29 December 2025 will be deleted between 8:00 on 23 August and 24 August 2026 (Singapore time)."

The dates: backup is available now and until 07:59 on 23 August SGT (that is 01:59 in the night of 23 August Kyiv time, so the deadline is effectively 22 August), and restore is possible from 25 August. Affected users are notified in the app and by email; anyone who signed up through Apple ID or Facebook has no email on file and only gets in-app notifications.

Why it matters. The action is a one-off: remember whether you signed up to Manus in the spring when it was everywhere. If you did and something worth keeping is still in there, the window closes on 22 August. If you cannot remember, that is 30 seconds of searching your mail for "Manus".

And the broader conclusion, back to item 1: data in someone else's agent service lives exactly as long as the legal entity that owns it, and it disappears through a reorganisation. One more argument for a setup where memory sits in your own git repository. [proven - the official Manus note was read, dates and times are verbatim; the Kyiv time given here is converted from SGT (UTC+8 to UTC+3, a 5 hour difference)]

Manus: A Note to Our Users · HN thread

TOPIC 6

Google, officially: "Go is an ideal language for AI-assisted development", because code is now more read than written. 308 points and 362 comments, and the thread argues

A post from the Group Product Manager for Go and the Chief Evangelist at Google Cloud, and its main claim is wider than Go.

Verbatim: "Historically, developers measured a language's productivity by how easy it is to write. But when a coding agent can generate hundreds of syntactically valid lines in seconds, the speed at which a human writes code no longer matters much. What matters now is reviewing, verifying and maintaining the code that has already been written."

And one formulation worth noting on its own: "AI is increasingly a teammate. A bit of a lone wolf, but a teammate. What matters now is how the team works together."

The case for Go: it was built from the start as a platform, with a formatter, a test framework, dependency management and security tooling built in, plus strong compatibility guarantees ("code written today will still be good code in ten years"). The authors' claim: this was built for humans, but "it turns out AI and humans have strikingly similar needs" - uniformity, predictability, machine-checkable steps.

Why it matters. The principle is not about picking a language. One entry point to the tooling, one dispatcher instead of a pile of scattered scripts, written instructions instead of oral tradition: that is "opinionated simplicity so the whole team structures things the same way". It works even when the team is one person and an agent. Google is now saying this is the main productivity factor when working with agents.

The practical takeaway is about priorities. When the choice for the next half hour is a new feature or making what exists more uniform and more checkable, this argues for the second. [proven - the Google post was read in full and the quotes are verbatim; it is Google writing about its own language, so the source has an interest; the 362-comment thread argues with it and was not read in full]

Google Developers Blog: why Go · HN thread, 308 points

TOPIC 7

An independent market-share measurement: OpenAI never dropped below 50% in any month, Anthropic grew from 4.3% to 14.9%, and Google collapsed from 12% to 1.9%

What was clearly missing yesterday: numbers instead of lab announcements. Mollick posted it with the right caveat: "the study advises caution in deciding who is winning, since it is a single source."

Pangram is an AI text detector. They learned to tell from internal activations which model family generated a piece of text, and ran two years of anonymised data that people send them for checking. The results:

- **OpenAI** - "not a single month below 50% in Pangram's entire existence"
- **Anthropic** - from 4.3% in 2024 to 14.9%, with a direct explanation: "the growing popularity of Claude among technical and scientific communities"
- **Google** - from 12% to 1.9% as of July 2026, "the sharpest change in Pangram's history"

Why these numbers deserve some trust: the authors list their own biases (their classifier is 91% accurate top-1 on a separate sample; the text people send is not random) and put OpenRouter's data alongside, which is collected by a completely different mechanism. The shares differ, but the Google collapse shows up in both. What makes the conclusion worth attention is two independent samples agreeing. The number itself is secondary.

A small thing that is fun: in the topic breakdown, Anthropic and Google models show up more in programming and the sciences, and OpenAI more in "everyday life and the humanities".

Why it matters. No direct action, this is calibration. By this measurement Anthropic is the growing part of the market, but still a distant second: 14.9% against 50%+. Reason enough not to buy narratives in either direction, neither "Google is swallowing everyone" nor "Anthropic took the market". [proven - the Pangram post was read in a browser and the numbers are verbatim; this is a detector measurement, not a count of requests: 91% accuracy, and the sample is text people send in to be checked for AI, which biases it toward "suspicious" text; the authors admit this themselves]

Pangram: Model Market Share · @emollick with the "careful, single source" caveat

---

TOPIC 8

Nvidia released a 30B model for "always-on" agents - the same niche as yesterday's Glimmer from Meta. In one day there are two of them

Short, but important for deduplication: yesterday it was Muse Glimmer (Meta, 30B, Apache 2.0, fits in 24 GB, "always-on local agentic workflows"). Today Nvidia ships **Nemotron 3.5 Lightning**, verbatim from their blog "a 30-billion-parameter MoE model with 3 billion active parameters", "the most efficient model in its class for long-running agentic workloads".

Claimed: up to 4× faster inference and 30% faster completion of agent tasks against models in its class; up to 1M context; NVFP4 and BF16; tool use.

Alongside the model comes **NeMo Switchyard**, an open-source smart routing library: it sends each request to the most appropriate model in your own mix (open, proprietary, Nvidia) without rewriting the application. The authors' framing: modern agents are "systems of models", where a frontier model plans and small specialised ones handle narrow jobs such as code review, tool calls and alert monitoring.

Why it matters. The niche of "a small model for cheap bulk tasks, next to a large one for thinking" got its second serious player in a day, which makes it a format.

Switchyard is the more interesting half. The idea that "a router decides who gets the request" has long been done by hand and crudely, in any pipeline where a cheap script check runs before the LLM call. If the script says nothing happened, the model is never woken. The router there is one condition on an exit code. [proven - the Nvidia blog was read and the numbers come from there; "4× faster" and "30%" are Nvidia's own claims against unnamed "models in its class", and there are no independent measurements one day in]

Nvidia blog: Nemotron 3.5 Lightning + NeMo Switchyard · HN thread · The model on Hugging Face · @NVIDIAAI, 1.7M views

---

TOPIC 9

Pragmatic Engineer is back from holiday - and straight in with a breakdown of how engineering works where one bug costs $440M

The main Substack subscription put out its first new piece since 29 July, two weeks of silence. Gergely Orosz took apart **Optiver**, a proprietary trading firm out of Amsterdam.

Why it is interesting to people who are not traders:

- **There are no external clients.** "Their own business is the client": no external deadlines, and instead personal motivation to improve is what gets valued.
- **Latency is "enemy number one"**, to the point where the company builds its own hardware (custom FPGAs, a partnership with AMD, its own optics and radio links).
- **And the turn the item was included for:** "The days when lower latency than competitors gave you risk-free arbitrage are over. Models are becoming the differentiator instead" - slow models with a fast trigger, plus fast models at the network edge making decisions in real time. An industry that competed in nanoseconds for twenty years is saying that speed has become the floor.
- **The fear that shapes the culture:** the story of **Knight Capital**, which nearly went bankrupt over one bug in a trading system, $440M in losses.

Why it matters. A direct parallel with item 6. Together they make the claim of the day: in two very different worlds, a Go team at Google and a trading firm, people independently say the speed of producing code has stopped being the scarce thing. The scarce thing now is confidence that it will not blow up.

Optiver's culture is described as "move very fast, but with a high premium on caution to avoid financial catastrophe". That is literally a description of a working frame where reading and analysis go through fast and unquestioned, and anything irreversible waits for an explicit human confirmation. Nice to see that setup described as the industry standard where the cost of a mistake is $440M. [proven - the article was read (the free portion, then a

paywall); it is a piece produced with the company's participation, interviews with their CTO and leads, so it is an inside view without independent verification; the deeper sections are behind the paywall and unavailable]

The Pragmatic Engineer: engineering at Optiver

TOPIC 10

Mojo reached 1.0 - three years after it was announced. The language built to replace C++ in AI

A quiet but real release (332 points): **Mojo 1.0**, "the milestone the language has been heading toward since its first release in 2023". Modular says plainly why now: rapid changes in the language "made it harder for the community to maintain long-running projects", and 1.0 is above all a promise of stability. Their own credibility argument: "this is already a language we rely on daily in production", with their commercial infrastructure MAX and Modular Cloud running on it.

Why it matters. Zero action, this is a time marker: one more attempt at "Python, but fast" has reached a stable release. For most automation it does not matter, because the bottleneck there is not language speed but network calls and waiting on someone else's API. But if the conversation ever turns to "rewrite something heavy", it is now at 1.0. [proven - the Modular blog was read; there are no benchmarks here, and none in the post either, since this is a release about stability]

Modular 26.5: Mojo 1.0 · HN thread, 332 points

misc - briefly, what else is worth a look

• **Unsloth Desktop** - a free open-source app for running and training models locally (macOS/Windows/Linux, MLX and GGUF), and the key part: you can connect Claude Code or Codex to it. The simplest way currently available to measure a local model on your own machine. Recommended by @dessaigne from YC, announced by @UnslothAI

• **@levelsio complains about Claude in public**: "Claude gets more preachy every day, regularly just refuses... ready to jump to Grok if it gets good enough for coding." The second complaint of this kind this week. Part of it goes away if the frame is agreed up front: what the agent does on its own, and what needs confirmation

• **Copilot behind a MITM proxy** (169 points): an engineer put GitHub Copilot's traffic through a proxy to see what it actually sends to the server - context, memory, harness. The method is more interesting than the findings: "the source says what an app can do; what it actually does is only visible at runtime." Worth applying to any AI tool that gets installed on a work machine

• **@paulg trolls his own tweet**: a year ago he wrote about a founder writing 10,000 lines of code a day and got torn apart for it. Now: "Will the people who were outraged by that be

even more outraged if I say I met a founder today who sometimes writes 50,000 lines a day? Or is everyone used to it by now?"

• **@garrytan at Startup School, one line on the topic:** "Entire startups will soon be markdown files. Own your skills, because otherwise your job becomes a skill file." Agent instructions really do live in ordinary markdown files, and that is where how everything works is written down

• **"Compression is prediction"** (356 points) - a long (3,739 words) breakdown of why compressors and language models are essentially solving the same problem. Pure pleasure for an engineering brain, zero practical use

• **@levelsio with an idea you can take today:** export every statement, personal, business and brokerage, to CSV, "drop it into Claude Code and ask it to build a spending dashboard"

• **Manus** is already item 5, but worth repeating in one line because of the deadline: back up by 22 August

• **Scroll through all 43,252,003,274,489,856,000 states of the Rubik's cube** (287 points) - useless and beautiful. Just nice

---