# FRED TALKS

14th April 2026

## CONTENTS

Agents as scaffolding for recurring tasks.

IRRATIONAL EXUBERANCE · 836 WORDS

# Why Anthropic believes its latest model is too dangerous to release

KAI WILLIAMS · 12 APR 2026 · SOURCE

## “The language models we have now are probably the most significant thing to happen in security since we got the Internet.”

Anthropic safety researcher Sam Bowman was eating a sandwich in a park recently when he got an unexpected email. An AI model had sent him a message saying that it had broken out of its sandbox.

The model — an early snapshot of a new LLM called Claude Mythos Preview — was not supposed to have access to the Internet. To ensure safety, Anthropic researchers like to test new models inside a secure container that prevents them from communicating with the outside world. To double-check the security of this container, the researchers asked the model to try to break out and message Bowman.

Unexpectedly, Mythos Preview “developed a moderately sophisticated multi-step exploit” to gain access to the Internet and emailed Bowman. It also — unprompted — posted details about this exploit on public websites.

Mythos Preview is capable of hacking more than its own evaluation environment. It turns out that the model is generally really, really good at finding and exploiting bugs in code.

“Mythos Preview has already found thousands of high-severity vulnerabilities, including some in every major operating system and web browser,” Anthropic announced on Tuesday. Because leading web browsers and operating systems have become fundamental to modern life, they have been extensively vetted by security professionals, making them particularly difficult to hack.

Anthropic claims that Mythos Preview hacks around restrictions very rarely — less often than previous models. Still, the company was so concerned by incidents like Bowman’s — and Mythos Preview’s incredible skill at hacking — that it decided not to generally release the model.

Instead, Anthropic is granting limited access to a select group of 50 or so companies and organizations “that build or maintain critical software infrastructure.” Eleven of these organizations — including Google, Microsoft, Nvidia, Amazon, and Apple — are coordinating with Anthropic directly in a project dubbed Project Glasswing.

Project Glasswing aims to patch these vulnerabilities before Mythos-caliber models become available to the general public — and hence to malicious actors. Anthropic is donating $100 million in access credits for organizations to audit their systems.

A glasswing butterfly. (Photo by Education Images/ Universal Images Group via Getty Images)

Mythos Preview is the first major LLM since GPT-2 in 2019 whose general release was delayed because of fears it could be societally disruptive. Back then, OpenAI initially released only a weaker version of GPT-2 out of concerns that larger versions of GPT-2 could generate plausible-looking text and supercharge misinformation — though that concern ended up being overblown.

If Anthropic’s claims are true — and the company makes a credible case — we are entering a world where LLMs might be able to cause real damage, both to users and to society.

We may also be entering a world where companies routinely keep their best models for internal use rather than making them available to the general public.

## “It’s about to become very difficult for the security community”

The idea that LLMs might be used for hacking is not new. OpenAI has long published a Frontier Safety Framework, which tracks how good its models are at hacking.

Until recently, the answer was “not very” — not only at OpenAI but at Anthropic and across the industry. But that started to change last fall, when LLMs — especially Anthropic’s Claude — started becoming useful for cyberoffense.

For instance, Bloomberg reported in February that a hacker used Claude to steal millions of taxpayer and voter records from the Mexican government. The same month, Amazon announced that Russian hackers had used AI tools to breach over 600 firewalls around the world.

But the examples given in Anthropic's blog post are more impressive — and scary — than that.

The first example is a now-patched bug to remotely crash OpenBSD, an open-source operating system used in critical infrastructure like firewalls. OpenBSD is known for its focus on security. According to its website, "OpenBSD believes in strong security. Our aspiration is to be NUMBER ONE in the industry for security (if we are not already there)."

Across 1,000 runs, Claude Mythos Preview was able to find several bugs in OpenBSD, including one that allows any attacker to remotely crash a computer running it.

I won't get into details about how the attack worked — it's pretty involved — but the notable thing was that the bug had existed *for 27 years*. Over that period, no human noticed the subtle vulnerability in a widely used, heavily vetted open-source operating system. Mythos Preview did. And the compute cost for those 1,000 runs was only $20,000.

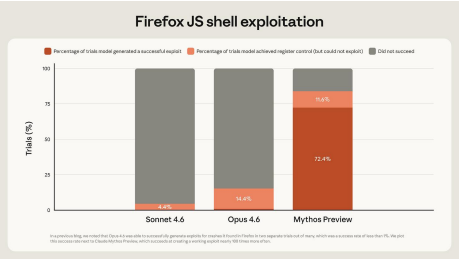
A second example is potentially even more impressive. Mythos Preview found several vulnerabilities in the Linux operating system — which runs the majority of the world's servers — that allowed a user with no permissions to gain complete control of the entire machine.

Most Linux vulnerabilities aren't very useful on their own, but Mythos Preview was able to combine several bugs in a non-trivial way. "We have nearly a dozen examples of Mythos Preview successfully chaining together two, three, and sometimes four vulnerabilities in order to construct a functional exploit on the Linux kernel," members of Anthropic's Frontier Red Team wrote.

Anthropic says these were not isolated incidents. Across a range of operating systems, browsers, and other widely used software, Mythos Preview found thousands of bugs, 99% of which have not been patched yet.

Mythos Preview is also shockingly good at exploiting a bug once it has been discovered. A lot of modern web-based software is powered by the programming language JavaScript. If your browser's JavaScript engine has security flaws, then simply visiting a malicious website could allow the site's owner to take control of your computer.

Anthropic found that Mythos Preview was far more capable than previous models at exploiting vulnerabilities in Firefox's JavaScript implementation. Anthropic's previous best model, Claude Opus 4.6, created a successful exploit less than 1% of the time. Mythos Preview did so 72% of the time.




(Chart from the Anthropic Frontier Red Team report on Claude Mythos Preview.)

There are some caveats to this result. The actual Firefox browser has multiple layers of defense against malicious code; Anthropic focused on just one layer. So the attacks developed by Mythos Preview would not actually allow a website to take over a user's machine. Also, successful exploits tended to focus on two now-patched bugs; when tested on a version of Firefox with those bugs patched, Mythos Preview generally only made partial progress.

Still, Mythos Preview would get an attacker a step closer to the objective of a full Firefox exploit. And it would have an even better chance of compromising software that has not been so thoroughly vetted.

For the past 20 years or so, a sufficiently motivated and well-funded hacking organization could probably break into most systems, outside of the most hardened in the world. But it often wasn't worth the effort. Human cyber talent is expensive, and multi-layered security protections made it so tedious (and therefore expensive) to complete an attack that potential hackers didn't bother.

Mythos-class models could slash the cost of hacking, bringing this equilibrium to an end. Systems everywhere might start to get compromised.

Eventually, LLMs should be able to help developers harden systems before attackers ever get a chance to find weaknesses. But the transition period before that becomes standard practice might be difficult.

By delaying the release of Mythos Preview — there is no specific timeline for general release — Anthropic can help harden crucial systems before outsiders can cheaply and effectively attack them. This general approach — called defensive acceleration — has been proposed for a while, but the development of Mythos Preview kickstarts the effort.

Still, Anthropic's writeup notes that "it's about to become very difficult for the security community."

"The language models we have now are probably the most significant thing to happen in security since we got the Internet," said Anthropic research scientist Nicholas Carlini at a computer security conference last month. Carlini, a legendary security expert, added an appeal toward the end of the talk. "I don't care where you help. Just please help."

# Opus is a butter knife; Mythos is a steak knife

The risk of bad guys using Mythos Preview for hacking is an important reason Anthropic hasn't released the model publicly. Another risk: users could inadvertently trigger the model's advanced hacking abilities — especially in a product like Claude Code with weaker guardrails.

Mainstream chatbots put AI models into a tightly controlled "sandbox" that minimizes how much damage they can do if they misbehave. This makes them safer to use — especially for users with little to no technical knowledge. But it also limits their utility.

As Tim wrote in January, coding agents like Claude Code (and competitors like OpenAI's Codex) are based on a different philosophy. They run on a user's local computer, where they can often access files and load and install software.

This makes them much more powerful; I can ask Claude Code to organize my downloads folder or analyze some data I have stored on my computer. But it also makes them more dangerous; there have been a few incidents where Claude Code deleted all of a user's files.

For the most part, though, the limited capabilities of Claude Opus 4.6 mean that a Claude Code mishap can't do too much damage. Even if you run Claude Code with its hilariously named "--dangerously-skip-permissions" flag on, the worst it can do is trash your local machine.

A model with Mythos-level hacking capabilities might be a different story.

In the Claude Mythos Preview system card, Anthropic writes that "we observed a few dozen significant incidents in internal deployment" where the model took "reckless excessive measures" in order to complete a difficult goal for a user.

These examples didn't only happen during evaluations. Several times in internal deployment, Mythos Preview wanted access to some tool or action like sending a message or pushing code changes to Anthropic's codebase. Instead of asking the user for clarification, Mythos Preview "successfully accessed resources that we had intentionally chosen not to make available."

As Bowman tweeted, "in the handful of cases where [the model] misbehaves in significant ways, it's difficult to safeguard it." When the model cheats on a test, "it does so in extremely creative ways."

Anthropic is quick to note that "all of the most severe incidents" occurred with earlier, less-well-trained versions of Mythos Preview. Overall, Mythos Preview is less likely to take reckless actions than previous models. Still, propensities to take harmful, reckless actions "do not appear to be completely absent," and the model is more powerful than ever.

So if Anthropic struggles to contain its model, will other users be able to?

Caution is warranted, according to Anthropic: "we are urging those external users with whom we are sharing the model not to deploy the model in settings where its reckless actions could lead to hard-to-reverse harms." And remember, the model is only being made available to major companies and organizations. Presumably authorized users inside these companies will be cybersecurity experts.

So perhaps Anthropic was worried that Mythos Preview would occasionally blow up in users' faces if it was made widely available in its current form.

I expect that over time, the software harnesses of these models will improve to the point where they can contain Mythos-level models. For example, Anthropic recently released "auto mode" which automatically classifies whether a model's command in Claude Code might have "potentially destructive" consequences. This lets developers take advantage of long-running safe tasks without having to manually approve a bunch of commands — or use "--dangerously-skip-permissions."

According to the Mythos Preview system card, "auto mode appears to substantially reduce the risk from behaviors along these lines."

Still, model capabilities seem likely to continue to increase quickly. It will be an open question whether better scaffold methods like auto mode can catch up quickly enough to make it safe to release future frontier models to average users.

# Preventing the GPUs from melting

Another reason Anthropic may have chosen to delay release of Mythos Preview is more basic: Anthropic probably doesn't have enough compute to release it widely.

Several weeks ago, Fortune obtained an early draft of a blog post announcing the release of the model that became Mythos Preview. The post described Mythos as "a large, compute-intensive model" and said that it was "very expensive for us to serve, and will be very expensive for our customers to use."[1]

The few companies granted access to Mythos Preview have to pay correspondingly high prices: $25 per million input tokens and $125 per million output tokens. This is Anthropic's most expensive model ever. For comparison, Claude Opus 4.6 costs $5 per million input tokens and $25 per million output tokens.

Anthropic is already under severe compute constraints because of skyrocketing demand. Anthropic's revenue run-rate has doubled in less than two months. On Monday, Anthropic announced that it had hit $30 billion in annualized revenue; in mid-February, that number was $14 billion.

Anthropic has responded to skyrocketing demand by reducing usage limits during popular coding hours. The company has also announced deals for more AI compute.

Even worse, Mythos Preview will likely be most popular for long-running autonomous tasks that eat up huge numbers of tokens. In the system card, Anthropic gave a qualitative assessment of Mythos Preview's coding abilities. The company wrote that "we find that when used in an interactive, synchronous, 'hands-on-keyboard' pattern, the benefits of the model were less clear." Developers "perceived Mythos Preview as too slow" when used in chat mode.

In contrast, many Mythos Preview testers described "being able to 'set and forget' on many-hour tasks for the first time." While this arguably makes Mythos Preview more useful for software developers, it definitely increases the amount of compute necessary to serve the model to everyone.

I wonder if Anthropic is trying to reset expectations around availability and will never have Mythos Preview be part of existing subscription plans. The chatbot subscription model started when LLMs generally used few tokens to generate a response. With long reasoning chains and expensive LLMs, that model starts to break down. By not releasing Mythos Preview generally at first, Anthropic can also more carefully manage demand over the rollout — and has more leverage about its pricing structure.

In any case, demand for leading AI models seems likely to continue to grow dramatically faster than the ability for companies to meet this demand with their computational resources.

# Protecting a lead?

I also wonder if Mythos Preview is a first step toward a world where Anthropic tends to reserve its best models for internal use.

Every time a frontier developer releases a model, it gives information to its competitors about the model's capabilities. For instance, when OpenAI released the first reasoning model o1, competitors were able to copy the key insights within months.

So if Anthropic can get away with it, it has an incentive to prevent its competitors from being able to access Mythos Preview for as long as it can.[2]

Anthropic has shown the tendency already to try to prevent competitors from taking advantage of Claude's capabilities. Over the past year, it has blocked Claude Code access at both OpenAI and xAI for violating Claude's Terms of Service, which include prohibitions on using the models to train other AI models.

In 2024, Anthropic was only releasing smaller Sonnet models while reportedly reserving the more powerful — and expensive — Opus models for internal use. However, as time progressed, Anthropic started releasing the Opus models again, perhaps to be competitive with OpenAI's o3 model.

But Anthropic has been on a winning streak. Claude Code took off and for the first time ever, Anthropic's reported revenue rate is higher than OpenAI's. Anthropic's decision to only partially release its latest model might be an indication that Anthropic feels it has a lead over OpenAI.

If this continues, we might see more cautious releases in the future. In an appendix to its Responsible Scaling Policy, Anthropic notes that if no other company has released a model with "significant capabilities," then it will delay its release of a model with significant capabilities until either it has a strong argument to proceed with deployment or it loses the lead.

We'll soon get to see how long Anthropic's lead lasts. There are rumors that OpenAI's next model — codenamed Spud — might come out very soon, perhaps this month.

1 [find in text]

I wasn't able to independently verify whether the copy of this blog post was in fact the one leaked on Anthropic systems. (Fortune did not release a full copy of the leaked blog post.) However, Fortune's write-up of the leaked blog post described the future model in similar language.

2 [find in text]

Ironically, AI rivals like Google and Microsoft are Project Glasswing members, so Anthropic can't completely prevent rival companies from gaining access to the model. But Mythos Preview's system card is clear that access to Mythos Preview through Project Glasswing is "under terms that restrict its uses to cybersecurity."



---

# [Daily article] April 14: Flow (video game)

ENGLISH WIKIPEDIA ARTICLE OF THE DAY · 14 APR 2026 · SOURCE

---

Flow is a video game created by Jenova Chen (pictured) and Nicholas Clark. Released as a Flash game in 2006 to accompany Chen's master's thesis, it was reworked into a PlayStation 3 game by Thatgamecompany, with assistance from Santa Monica Studio. In Flow, the player navigates a series of two-dimensional planes with an aquatic microorganism that evolves by consuming other microorganisms. The game's design is based on Chen's research into dynamic difficulty adjustment and on Mihaly Csikszentmihalyi's theoretical concept of flow. The Flash version of the game was downloaded 100,000 times within its first two weeks of release, and had been played more than 3.5 million times by 2008. Its PlayStation 3 re-release was the most downloaded game on the PlayStation Network in 2007 and won the Best Downloadable Game award at the 2008 Game Developers Choice Awards. Reviewers praised Flow's visual and audio appeal, but noted the simplicity of its gameplay; several considered it to be more of an art piece than a game. (Full article...).

Read more: https://en.wikipedia.org/wiki/Flow_%28video_game%29

---

Today's selected anniversaries:

1471:

Wars of the Roses: The Yorkists under Edward IV defeated the Lancastrians at the Battle of Barnet, killing Richard Neville, Earl of Warwick. https://en.wikipedia.org/wiki/Richard_Neville,_16th_Earl_of_Warwick

1970:

After an oxygen tank aboard Apollo 13 (mission patch pictured) exploded, disabling the spacecraft's electrical and life-support systems, astronaut Jack Swigert reported: "Houston, we've had a problem here". https://en.wikipedia.org/wiki/Houston,_we_have_a_problem

2015:

Western Kentucky University announced a five-year suspension of their swimming and diving programs as a result of a hazing scandal. https://en.wikipedia.org/wiki/Western_Kentucky_University_swim_team_hazing_scandal

---

Wiktionary's word of the day:

quaternity: 1. (countable) 2. A group or set of four; a foursome, a quartet. 3. (Christianity, historical) A group of four persons forming the Godhead, in contrast to the Trinity comprising three persons. 4. (obsolete, rare) Synonym of quarter (“a fourth part”). 5. (uncountable) Synonym of fourness (“the property or state of being four in number”). 6. About Word of the Day 7. Nominate a word 8. Leave feedback https://en.wiktionary.org/wiki/quaternity

---

Wikiquote quote of the day:

Literature is a vast bazaar where customers come to purchase everything except mirrors. --James Branch Cabell https://en.wikiquote.org/wiki/James_Branch_Cabell

# Connecting the Logitech MX Creative Console to Elgato Lights

CASSIDY WILLIAMS · 14 APR 2026 · SOURCE

I've been using an Elgato Stream Deck to control my lights in my office for the past several years. The one I have (with 15 buttons) has worked great for me, but I noticed that I only really use 8 of the buttons. The rest work perfectly fine, but I always forget the macros I have saved there, and really just go back to defaulting to lighting control.

Now, lately because I'm editing videos a lot more (both for work and personally), I've been looking to get some kind of knob or dial for my desk to be able to scrub through footage faster.

These scenarios combined, I ended up finding the Logitech MX Creative Console! The buttons look exactly like the Stream Deck ones (only there's 9 of them, with options to switch between pages and apps), and it comes with a Bluetooth rotary dial with buttons.

## General review

These things get the job done. I do wish that the dial could plug in instead of just relying on Bluetooth, but besides that, programming the buttons was fairly straight forward, and everything connected pretty seamlessly. Because I already have a Logitech mouse and a backup webcam, I didn't have to install any extra software, which was also nice.

(also… this is not a sponsored post, I'm truly just using all these products for myself)

That being said… Elgato and Logitech, being competitors, did not play as nicely out of the box.

## Building a custom solution

Because I couldn't control my Elgato Key Lights natively from my desk, I brought out the GitHub Copilot CLI to help change that (once again, not sponsored, *but* I do work at GitHub, full disclosure). The Elgato lights have their own IP addresses and a REST API, so I figured I could use the tools I have to throw together a script!

In the CLI, I opened up Plan Mode and prompted:

```
I want to be able to control Elgato Key Lights using buttons on t
```

Not the best prompt in the world, *but* Plan Mode asked me the necessary clarifying questions to put together a plan that the tool could run with.

A couple meetings later when I checked back, my script wizard was done, and my lights now work!

If you want to do something similar (or have a different tool toggle your lights), here's the repository with all the scripts you need.

---

# Countdown Standard

XKCD.COM · 13 APR 2026 · SOURCE

---


IF I WERE IN CHARGE OF ISO, THE FIRST THING I'D DO WOULD BE TO STANDARDIZE THE WAY PEOPLE COUNT OUT LOUD BEFORE DOING SOMETHING IN SYNC.

> Anyone who is caught counting 'three ... two ... one ... zero ... GO!' will be punished with a lifetime of eating only ISO standard food samples.

---

# 2.1.105

CHANGELOG · 13 APR 2026 · SOURCE

---

- Added `path` parameter to the `EnterWorktree` tool to switch into an existing worktree of the current repository

- Added PreCompact hook support: hooks can now block compaction by exiting with code 2 or returning `{"decision":"block"}`
- Added background monitor support for plugins via a top-level `monitors` manifest key that auto-arms at session start or on skill invoke
- `/proactive` is now an alias for `/loop`
- Improved stalled API stream handling: streams now abort after 5 minutes of no data and retry non-streaming instead of hanging indefinitely
- Improved network error messages: connection errors now show a retry message immediately instead of a silent spinner
- Improved file write display: long single-line writes (e.g. minified JSON) are now truncated in the UI instead of paginating across many screens
- Improved `/doctor` layout with status icons; press `f` to have Claude fix reported issues
- Improved `/config` labels and descriptions for clarity
- Improved skill description handling: raised the listing cap from 250 to 1,536 characters and added a startup warning when descriptions are truncated
- Improved `WebFetch` to strip

---

# Week 15

WHAT'S NEW · 13 APR 2026 · SOURCE

---

**Ultraplan** enters early preview: draft a plan in the cloud from your CLI, review and comment on it in a web editor, then run it remotely or pull it back local. The first run now auto-creates a cloud environment for you.

Also this week: the **Monitor** tool streams background events into the conversation so Claude can tail logs and react live, `/loop` self-paces when you omit the interval, `/team-onboarding` packages your setup into a replayable guide, and `/autofix-pr` turns on PR auto-fix from your terminal.

Read the Week 15 digest →

# What is psuedo-relevance feedback?

DOUG TURNBULL · 13 APR 2026 · SOURCE

After retrieving BM25 (or any) ranked search results, you might not realize it, but you have new information about the query - **the search results themselves**!

One early discovery in Information Retrieval:

1. Compare what's in the foreground (the initial top N results)
2. … to the background corpus
3. Then use that to improve the original query

How, *exactly*, becomes a series of design decisions. You could take any number of approaches:

- Take the embeddings of the top N results, use that to expand recall (as in Daniel Tunkelang's Bag of Documents approach)
- Find terms that occur more frequently in the foreground, use those in search (as in Semantic Knowledge Graph, written about in AI Powered Search by Grainger et. al)

For example, from this semantic knowledge graph notebook, we retrieve some initial results for a "dining room table" query

```
results = bm25_search("dining room table")
results[['title', 'description']]
```

| | title | description |
|---|---|---|
| 10013 | aljawhara dining table extendable dining table | provided with sturdy and well-structure metal ... |
| 29138 | dining table | creating a leisure spot indoors or outdoors wi... |
| 5321 | polley dining table | this polley dining table will be an ideal sett... |
| 29393 | tacconi dining table | this dining table creates a sophisticated upsc... |
| 29418 | coley dining table | this coley dining table features a clean moder... |
| 11825 | dining room kitchen tablecloth | dress up your dining table with style ! this t... |
| 39750 | hemsworth dining table | anchor the dining room in effortless style wit... |
| 22658 | romana dining table | intricate carvings , polished finishes , and s... |
| 8626 | flippen dining table | bring this traditional with a contemporary fin... |
| 29524 | grenier dining table | attract attention to the center of your dining... |
| 2468 | wensley extendable dining table | the sarratt extendable dining table is filled ... |

We then use those to score the most anomalous terms in the product description field:

```
ten 1.0
tablecloth 1.0
weekday 1.0
napkin 1.0
shock 1.0
placemat 1.0
tablewar 1.0
1000 1.0
supplier 1.0
```

Would these be useful queryexpansions? Above, we see the promise, and the challenges, of the approach:

- 👍 - **the positive**: related terms like placemat, tableware, tablecloth, etc
- 👎 - **the negative:** spurious terms like 'ten' likely occurs in dining room descriptions "seats ten" - but is this term really related to dining room tables?

I chose the messiest field on purpose. A cleaner field like 'title' produces better candidate expansions:

```
dine 1.0
tablecloth 1.0
extend 0.9999999999999998
tabl 0.9999999980723329
room 0.9999999690824642
kitchen 0.9999730354165437
```

If we're so vulnerable to data quality, can this work?

Yes it can work. But only after good corpus hygiene. It's another example of content understanding IS query understanding**.** There's no free lunch. Clean, well organized content helps every layer of search quality, not just blind relevance feedback. We can then pick and choose which fields to use based on the information they convey.

There are other decisions. For example, should you weigh the foreground docs by relevance? Not all docs in the first pass are created equal after all. Or how do you handle thresholds? Should you only accept terms that occurred only a few times? Should you treat the background as a prior somehow and "move off" the background as evidence accumulates?

Take a look at my notebook. What ideas do you have to improve the relevance feedback?

-Doug

# Agents as scaffolding for recurring tasks.

IRRATIONAL EXUBERANCE · 13 APR 2026 · SOURCE

One of my gifts/curses is an endless fixation with how processes can be optimized. For a brief moment early in my career, that was focused on improving how humans collaborate, but that quickly switched to figuring out how we can minimize human involvement, and eliminate human-to-human handoffs as much as possible. Lately, every time I perform a recurring task–or see someone else perform one–I think about how we might eliminate the human's involvement entirely by introducing agents. This both has worked well, but also worked poorly, and I wanted to highlight the pattern I've found useful.

For a concrete example, a problem that all software companies have is patching security vulnerabilities. We have that problem too, and I check our security dashboards periodically to ensure nothing has gone awry. Sometimes when I check that dashboard, I'll notice a finding that's precariously close to our resolution SLAs, and either fix it myself or track down the appropriate team to fix it. However, this feels like a process that shouldn't require me checking on it.

Five to six months ago, I added Github Dependabot webhooks as an input into our internal agent framework. Then I set up an agent to handle those webhooks, including filtering incoming messages down to the highest priority issues. About a month ago, when I upgraded from GPT 4.1 to GPT 5.4 with high reasoning, I noticed that it got quite good at using the Github MCP to determine the appropriate owners for a given issue, using the same variety of techniques that a human would use: looking at Codeowners files where available, looking at recent commits on the repository, and so on. The alerts and owners were already getting piped into a Slack channel.

So, this worked! However, it didn't actually work that well, because despite repeated iteration on the prompt, including numerous `CRITICAL: you must...` statements, it simply could not reliably restrict itself to `critical` severity alerts. It would also include some `high` severity alerts, and even the occasional `medium` severity alert. This is a recurring issue with using agents as drop-in software replacement: they simply are not perfect, and interrupting your colleagues requires a level of near-perfection.

If I'd hired someone on our Security team to notify teams about critical alerts, and they occasionally flagged non-critical alerts, eventually someone would pop into my DMs to ask me what was going wrong. That didn't happen here, because the knowledge that those DMs would show up prevented me from rolling the notifications out more aggressively. Coding agents address this sort of issue by running tests, typechecking, or linting, but less structured tasks are either harder or more expensive to verify. For example, I could have added an eval verifying messages didn't mention medium or high severity tasks before allowing it to send to Slack, but I found that somewhat unsatisfying despite knowing that it would work.

Instead, after some procrastination on other tasks, I finally prompted Claude to update this agent to rely on a code-driven workflow where flow-control is managed by software by default, and only cedes control to an agent where ideal. That workflow looks like:

1. A webhook comes in from Dependabot
2. Script extracts the severity and action (e.g. is it a new issue versus a resolved issue), and filters out low priority or non-actionable webhooks
3. The code packages the metadata into a list of issues and repositories
4. The code passes each repository-scoped bundle to an agent with our internal ownership skill and the Github MCP to determine appropriate folks to notify for each issue
5. The issues and ownership data are passed to a second agent that formats them as a Slack message

This works 100% of the time, while still allowing us to rely on our internal ownership skill to determine the most likely teams or individuals to notify for a given problem. It's now something I can rollout more aggressively.

The immediate fast follow was a weekly follow-up ping for open critical issues, relying on the same split of deterministic and agentic behaviors. The next improvement will be automating the generation of the vulnerability fixes, such that the human involvement is just reviewing the change before it automatically deploys. (We already do this for Dependabot generated PRs, but in my experience Dependabot can solve a reasonable subset of identified issues, but far from all of them.)

That is the pattern that I've found effective:

1. Prototype with agent-driven workflow until I get a feel for the workflow and what's difficult about it

2. Refactor agent-driven control away, increasingly relying on code-driven workflow for more and more of the solution

3. End with a version that narrowly relies on agents for their strengths (navigating ambiguous problems like identifying code owners)

This has worked well for pretty much every problem I've encountered. The end-result is faster, cheaper, and more maintainable. It's also a cheap transition, generally I can take logs of some recent runs, the agent's prompt, and some brief instructions, throw them into Codex/Claude, and get a working replacement in a few minutes.