

# Security Now! #1077 - 05-05-26

## A Browser AI API ?

### This week on Security Now!

- Hackers AI-code a portal, forget to add authentication.
- The UK's NCSC issues a Mythos warning. Where's CISA?
- Another (of many) Linux local privilege escalations.
- AI may be spelling the end of bug bounties.
- Anthropic releases "Claude Security" mini-Mythos.
- ChatGPT gets very serious about login security.
- Syncting's SyncTrayzor v1 abandoned; v2 created.
- Google drops an AI API into Chrome; Mozilla objects.

**Attempting to preempt the inevitable question:  
"Why has the lobster become so expensive?"**



## Security News

### Hacker's vibe-AI coded server leaks its stolen credit cards

We begin this week with a story that intersects with security on several fronts. Last Wednesday, CyberNews' headline was: "*Scammers vibecode server to verify stolen credit cards, leak details of 345K cards.*" I had to read that one twice. Here's what CyberNew researchers discovered.

They wrote:

*Threat actors, like so many programmers around the world, are no strangers to AI assisting in their operations. However, like so many vibecoders, scammers also run into security issues. On April 16th, the Cybernews research team discovered an exposed server owned by a threat actor. The exposed information is controlled by a carding market called Jerry's Store. The tool provides credit card validity percentages for each seller. In other words, threat actors use this tool to check if the stolen payment card is still operational.*

*According to our team, Jerry's Store operators extensively used Cursor, an AI-assisted development environment, to set up the leaking server and to create administrator-facing dashboards. Cursor is a legitimate service, developed by the US software company Anysphere. Researchers believe that relying on an AI assistant to set up the server was the reason it was exposed. Based on the chat logs our team was able to access, the threat actor received flawed instructions from their AI for building the dashboards. The team explained: "We were able to confirm that the leak originated from the user asking to create a statistics dashboard, and Cursor created an unauthenticated open web directory to serve the webpage, ignoring the need to set up authentication or ensure that only the intended dashboard would be accessible."*

*Moreover, the chat history reveals there was sufficient information for Cursor LLM to identify that it was helping set up a credit card verification service, indicating a lack of sufficient guardrails to prevent abuse. Researchers said: "It's a lesson for developers using Cursor for legitimate uses, showing how it can lead to accidental data leaks."*

CyberNews said that they had reached out to Cursor for comment and would update their article with any additional information they receive.

The fact that the Cursor AI produced a statistics dashboard driven by an unsecured and open web directory allowing unauthenticated remote access is a great example of the danger of using AI without being a domain expert. I have no doubt that the Cursor AI would have easily provided authentication if it had been asked to. But apparently the bad guys never thought to ask. Someone who wasn't really up to speed on web-based application security could easily fail to anticipate all of the ways others might access and penetrate their system. In this case, either it never occurred to them that authentication should be required where it was absent, or they assumed that the AI would know what to do and would do it without bidding.

The CyberNews article also provided some interesting background reporting on the underground industry in stolen credit cards. They wrote:

*Operations such as Jerry's Store are integral to the cybercrime infrastructure. Once scammers obtain stolen credit card information, they need to verify which cards can still be exploited. Jerry's Store provides that service. Our team noticed that to complete the task, Jerry's Store operators use legitimate, well-known merchants.*

*The CyberNews team explained: "Threat actors used multiple legitimate merchant websites, such as Amazon US, Amazon JP, Grubhub, Sam's Club, Temu, Lyft, Elf Cosmetics, and CountryMax, utilizing hundreds or in some cases, thousands of accounts on these platforms to perform credit card validity checks." Attackers created those accounts to register stolen cards and perform "low-risk" actions. These could include adding cards as a payment method or making a very small purchase. If the platform accepts the card, threat actors mark the card as valid and sell it to other threat actors on the dark web.*

*Using large merchants like Amazon or Grubhub is a way to mask their activities. Since large merchants process billions of payments, small transactions on a well-known website don't ring any alarm bells. According to our team, the exposed server contained a treasure trove of credit card details. Researchers identified nearly 200K credit card details that the service deemed "invalid," and over 145K counts of valid payment card information. The exposed information includes all details that payment cards hold, including:*

- *Credit card numbers*
- *Expiration dates*
- *Security codes*
- *Cardholder names*
- *Cardholder addresses*

*Typically, valid credit card details are sold for \$7-18 on the dark web, meaning that the value of the valid stolen data on Jerry's Store ranges between \$1M and \$2.6M. However, our team added that the actual value of the exposed infrastructure may be a lot higher, since Jerry's Store sells much more than just credit card data. While it is unclear where Jerry's Store is located, internal tooling and leaked LLM chat logs suggest that the marketplace's administrator is fluent in Chinese. The server itself appears to be hosted in Germany by a suspected bulletproof hosting provider. The marketplace which launched in late 2023 is a well-known credit card vetting tool within the cybercrime underground, aimed primarily at cards stolen from victims in the US and the EU.*

Despite the fact that credit card theft appears to still be doing a booming business, as I've mentioned through the years, my own experience has been that credit card security has become much better. Rather than websites using random shopping cart packages, many of which had latent security issues, we've seen the rise of large credit card processing services which have consolidated processing into just a few giants and are more able to provide comprehensive security. Real time risk assessment services now also exist to spot and stop shady credit card transactions before they are approved.

### **The UK's NCSC (National Cyber Security Centre) is worried, too.**

Last Friday, Ollie Whitehouse, the Chief Technology Officer for the UK's NCSC – their National Cyber Security Centre – issued a clear warning at the level of the government. Ollie's warning posting was titled "*Preparing for a 'vulnerability patch wave'*" and it carried the tag line "*Organisations must act now to prepare for a wave of patches that will address decades of technical debt.*" I think "technical debt" is exactly the right way to express the concept that the piper may be about to get paid. I have a friend from the midwest whose favorite term for this would be "they're about to get their just comeuppance." Comeuppance, indeed.

Here's what the UK's NCSC CTO wanted everyone within the United Kingdom to appreciate. He wrote:

*Whether they are technology producers and vendors, or consumers and operators, all organisations have 'technical debt'; a backlog of technical issues – that is both expensive and time-consuming – as a result of prioritising short-term gains over building resilient products.*

*Artificial Intelligence, when used by sufficiently-skilled and knowledgeable individuals, is showing the ability to exploit this technical debt at scale and at pace across the technology ecosystem. As a result, the NCSC expect there will be a 'forced correction' to address this technical debt across all types of software, including open source, commercial, proprietary and software as a service.*

*This is why we are encouraging all organisations to prepare now for when a 'patch wave' arrives; a rush of software updates that will need to be applied across the technology stack to address the disclosure of new vulnerabilities.*

*All organisations must take steps to identify and minimise their internet-facing (and other externally-exposed) attack surfaces as soon as possible. As we've argued for some time, you should prioritise technologies on your perimeter and then work inwards covering cloud instances and on-premises environments. By doing this, organisations can reduce the risk posed by latent vulnerabilities when they become known and exploited by attackers.*

*Where organisations cannot apply updates across their entire environment, they should prioritise applying updates to their external attack surfaces. Where capacity extends beyond the external attack surface, organisations should prioritise critical security systems.*

*It is also important for organisations to realise that patching alone will not always suffice; some technical debt may be present in 'end of life' or legacy technology that is out of support, and so cannot receive updates. In such instances, organisations will need to replace technologies, or bring them back within support, especially where it presents an external attack surface.*

*Building on the principles contained within our Vulnerability Management guidance, organisations should make plans to deploy software security updates quickly, more frequently, and at scale, including across their supply chains. We are expecting an influx of updates to address vulnerabilities across all severities, and expect a number to be critical.*

*The NCSC recommend that:*

- *where automatic secure 'hot patching' is available (that is, patching that doesn't involve service disruption), this should be enabled as a priority*
- *where automatic updates are available (including for embedded devices), this should be enabled to reduce the workload on support teams*
- *where neither of the above are available, organisations will need to ensure that processes and risk appetites support frequent and scaled-updating, noting the operational trade-offs around disruption and safety critical systems. A risk-prioritised approach such as the Stakeholder Specific Vulnerability Categorisation (SSVC) system can be used to prioritise installing the updates*

*However, should a critical vulnerability be under active exploitation (especially one affecting an internet-facing system), then it is essential to accelerate the update process. Organisations can refer to the NCSC's new guidance on 'Responding to active exploitation of vulnerabilities' for more information.*

*To summarise, you should put in place a policy to 'update by default' where you always apply software updates as soon as possible, and ideally automatically. This should be at the core of your update management process, but we recognise that it may not apply in some circumstances (such as for safety-critical systems or operational technology).*

*Patching alone won't address the systemic problems that my previous blogs have addressed. I've appealed to technology producers and vendors to ensure systemic technical security debt is minimised by including - where appropriate - memory safety and containment technologies.*

*Similarly, for consumers and operators, a focus on cyber security fundamentals to raise resilience and to reduce the impact of breaches should be a priority. This includes adopting and fully implementing Cyber Essentials, or the Cyber Assessment Framework for organisations operating essential services (such as energy, healthcare, transport, digital infrastructure and government).*

*Prepare for the patch wave now. In conclusion, the NCSC advise all organisations, irrespective of size, to plan and prepare for the vulnerability patch wave. A good place to start is by reading the NCSC's updated Vulnerability Management guidance. For larger organisations, we also recommend working to gain assurance from your supply chains both commercial and open source, so that they are prepared to navigate any required response.*

This may seem like a simple restating of what we already know. But for many of the CIOs, CSOs and IT heads in organizations throughout the UK, a clear statement and posting such as this can provide the cover and backup they need to succeed in getting their organization's other C-Suite executives to take this seriously.

As I was seeing this note from the UK's NCSC I realized that I hadn't seen anything from our own CISA in the U.S. That struck me as odd since the CISA we've all come to know would normally have been shouting about this from the mountain tops. So I went digging to see whether I had missed the statement that it seemed clear CISA should have made in the wake of the Mythos revelations.

I found a report published two weeks ago, on April 21st by Axios. I exactly addresses the question "where's CISA?" The reporting was posted as a scoop titled "*Scoop: CISA lacks access to Anthropic's Mythos*" Axios writes:

*The Cybersecurity and Infrastructure Security Agency does not have access to Anthropic's powerful Mythos Preview model, even though some other government agencies are using it, two sources tell Axios. This matters because the country's top cyber defense agency, tasked with helping to secure everything from banks to power plants, is on the outside looking in at a time when the industries it works with are deeply concerned about AI-powered cyberattacks overwhelming their defenses.*

*Anthropic decided against a public release of Mythos due to its unprecedented ability to quickly discover and exploit security vulnerabilities. Instead, Anthropic provided it to more than 40 companies and organizations who are now testing it and working to shore up their systems. CISA is not on that list, the sources say.*

*Earlier this month, an Anthropic official told Axios the company had briefed CISA and the Commerce Department on Mythos' capabilities.*

- *The Commerce Department's Center for AI Standards and Innovation has reportedly been testing Mythos.*
- *The NSA is also among the organizations using Mythos, despite the Department of Defense, which oversees the agency, having declared Anthropic is a "supply chain risk."*
- *It's unclear if the ongoing turmoil within the agency during the second Trump administration played any role in the agency not moving more swiftly to secure access.*
- *Spokespeople for CISA and Anthropic declined to comment.*

*The Trump administration has spent the last year reducing capacity at CISA, instead opting to give more policy influence to the White House's national cyber director and pushing some programs to the state and local level. CISA's acting director Nick Andersen told lawmakers last week that the agency's resources are "more limited than I would like." Trump has proposed cutting as much as \$707 million from the agency's budget in the upcoming fiscal year. CISA has already lost more than a third of its workforce and millions of dollars in funding.*

*National cyber director Sean Cairncross is among the Trump officials negotiating broader civilian agency access to Mythos. The Treasury Department has also been negotiating access. Sources tell Axios that other organizations with access to Mythos have predominantly been using it to find exploitable security vulnerabilities in their own networks.*

*Security teams at critical infrastructure organizations have often looked to CISA to share threat intelligence across their sectors and determine how to prioritize their security strategies.*

I checked out this acting CISA director Nick Andersen and he appears to be entirely competent. He's a decorated U.S. Marine Corps veteran who served as Chief Information Officer for Navy Intelligence and Head of the Office of Intelligence, Surveillance, and Reconnaissance Systems and Technologies at the U.S. Coast Guard. He served on active duty managing intelligence mission systems in Iraq, Europe, and Africa, and is a veteran of Operation Iraqi Freedom. He served as Principal Deputy Assistant Secretary at the Department of Energy's Office of Cybersecurity, Energy Security, and Emergency Response (CESER) where he led national efforts to secure U.S. energy infrastructure. He also served as Federal Cybersecurity Lead and Senior Cybersecurity Advisor to the Federal CIO at the White House Office of Management and Budget.

So I have no complaints with Nick's background. It appears that he needs more resources and support, and that CISA's lack of access to Mythos is largely due to the War Department's unfortunate feud with Anthropic. Anthropic made clear in 2025, at the time it signed its contract with the Pentagon, that it did not want its technology used for mass surveillance of people in the United States or for fully autonomous weapons systems. Subsequently, the Department of War demanded that they drop those restrictions and Anthropic refused. They published a public statement explaining their position. Regarding fully autonomous weapons, they wrote: frontier AI systems are simply not reliable enough to power fully autonomous weapons, and without proper oversight, fully autonomous weapons cannot be relied upon to exercise the critical judgment that highly trained, professional troops exhibit every day. Anthropic offered to work directly with the Department of War on R&D to improve the reliability of these systems, but were turned down. After that, in apparent retaliation and without any evidence, the Pentagon declared Anthropic to be a "supply chain risk". This is all very unfortunate since CISA should absolutely have access to Anthropic's Mythos Preview. Hopefully, the White House's national cyber director Sean Cairncross who appears to understand the need will be able to make something happen. It's clearly ridiculous to have one of the U.S.'s leading AI firms frozen out of the government because, as Secretary Peter Hegseth declared, it is "woke AI." For the time being it appears that CISA is silent for purely political reasons.

## Another Linux LPE (Local Privilege Escalation)

The news late last week was of the discovery of another serious local privilege escalation discovered in the Linux kernel. And, yes, before you ask, it was found by an AI vulnerability discovery system operated by a security firm named Theori who wrote: *"An unprivileged local user can write four controlled bytes into the page cache of any readable file on a Linux system, and use that to gain root."* A simple 732-byte, 9-line Python proof of concept is up on Github which immediately elevates any normal user to root, which is not something you want to leave unpatched. So this is important and Linux distros Debian, Ubuntu, and SUSE have issued patches for the problem, as have overseers of other distros. Red Hat initially said it was going to defer the fix but later changed its guidance to indicate it will go along with other distros and patch promptly. The CVE has been rated High severity, 7.8 out of 10. It's only a 7.8, which is still high for a local privilege escalation, because an attacker first needs to get into a non-root account. But anyone with local access also also use this.

At the end of one of the reports of this I ran across the statement: *"AI-assisted vulnerability research recently prompted the Internet Bug Bounty (IBB) program to suspend awards until it can understand how to manage the growing volume of reports."* Since that was news to me I went hunting. Here's what I found:

## AI is dramatically reshaping (ending?) bug bounties

Near the end of March the Internet Bug Bounty program, which is run by HackerOne, paused their acceptance of new vulnerability submissions due to what HackerOne described as an increasing imbalance between vulnerability discoveries and the ability for open source maintainers to remediate them. And, yes, AI is the underlying driver of all this.

To back up a bit first, recall that the Internet Bug Bounty (IBB) is a crowd-funded vulnerability reward program that was started 14 years ago in 2012 and is operated through the HackerOne platform. Its purpose is to reward and thus incentivize independent security researchers to find and responsibly disclose vulnerabilities in widely-used open source software. The funding for the program comes from a consortium of major tech companies including Facebook, GitHub, Shopify, TikTok and others who all contribute to a shared bounty pool. The underlying idea is that since everyone depends on open source infrastructure, everyone should share in the cost of helping to secure it. And the vulnerability discovery payout structure is simple: 80% of each awarded bounty goes to the researcher who reported the vulnerability with the remaining 20% being contributed to the open source project where the trouble was found. Thus helping to fund the remediation work. It's widely been seen as a success having paid out more than \$1.5 million dollars since the program began.

But, almost predictably, AI has messed things up. HackerOne stated: *"The discovery landscape is changing. AI-assisted research is expanding vulnerability discovery across the ecosystem, increasing both coverage and speed. The balance between findings and remediation capacity in open source has substantively shifted."* The problem is being called "Triage Fatigue" and the trouble is not just the increased volume of reports, nor, interestingly, is it the signal to noise ratio. The actual problem is the nature of the noise – weirdly, the quality of the noise, while still noise, has increased. As Daniel Stenberg, the creator of curl expressed it: "more convincing crap is worse than obvious crap. You can't dismiss it quickly, you have to investigate it, and you waste real time proving it's nonsense. At scale, this stops feeling like a helpful external contribution model and starts to resemble something closer to a denial-of-service attack on the people responsible for security."

Thirty-one years ago, way back in 1995, Netscape launched the first widely recognized paid bug bounty program, offering to pay researchers for responsibly reporting significant bugs they

discovered in Netscape Navigator 2.0. And that model has been functioning vibrantly ever since. So the notion that AI may be driving a fundamental change to this vulnerability discovery and reporting model is important enough to be a contender for today's main topic. But the idea of Google going off half-cocked and adding an explicit AI interface for JavaScript in Chrome also needed ample space. We'll cover Mozilla's pushback at the end of the podcast.

Meanwhile, the company "Aikido" which is deep into automated vulnerability discovery as a business, recently interviewed not only Curl's Daniel Stenberg but also Casey Ellis. Casey is the founder of Bugcrowd and as such is one of the people who helped establish and formalize bounties for bugs starting in 2012. Aikido titled their report "*Bug bounty isn't dead, but the old model is breaking*" and wrote:

*Bug bounty has been a very hot topic lately. We're seeing high-profile programs go offline or fundamentally change: the Internet Bug Bounty, one of the most important programs for open-source programs, is pausing submissions, curl is removing payouts and Node.js is removing its bounty entirely. That's not noise, that's signal. We wanted to understand where bug bounty is actually heading, so we sat down with two of the most credible voices on opposite sides of this conversation: Daniel Stenberg, creator of curl, who is living the maintainer reality and recently halted bug bounty payments, and Casey Ellis, the founder of Bugcrowd, one of the people who helped establish the model. What we found was that the bug bounty model is at a crossroads, and we're in the midst of a big shift.*

*Before we get into where the model is headed, let's take a step back and understand why it's been one of the most effective ideas in security over the last decade. It all stems from the idea of letting the Internet try to break your stuff before attackers do. And it worked because it gave companies scale they could never hire. As Casey put it: "If you're trying to outsmart a global pool of attackers with someone working 9 to 5, the math is wrong." That's the magic of bug bounty. Instead of relying on a handful of internal people, you tap into a global pool of different skill sets, perspectives, and motivations - all attacking your system in ways your internal team never thought of. And that's without the significant overheads required to hire specialist experts internally and then working to keep them busy. All this explains why bug bounties became fundamental to modern security programs.*

*What's changing now isn't the demand for security, it's the economics of how bug bounties operate. **AI has altered the balance, and not in a good way.** Finding bugs is now cheaper than ever, writing reports is even easier, and submitting them has effectively become frictionless. Meanwhile, the **cost** of validating those reports and then actually fixing the issues has not changed at all. Those final two required steps, validating and then fixing bugs remains as labor intensive as ever.*

*We are seeing this play out in practice. There are three types of report submitters:*

*There are those companies that use a new approach for legitimate reports. These are reports that use layered AI approaches that combine the strengths of multiple AI models, guardrails, orchestration and context, such as Aikido's own AI pentesting capabilities.*

Aikido is, of course, plugging their own solution, as we would expect them to on their website. But we know that Anthropic also setup their Mythos Preview system to do the same. Both are discovering and, importantly, verifying suspected vulnerabilities to produce much higher quality reports which, in the case of Mythos, include proofs of concept exploits. Aikido continues:

*Then there are individuals who escalate their research and report writing using AI as a tool. And finally, there are individuals who are able to upskill by virtue of these AI models. They generate reports that seem technically plausible, but are still completely wrong. Daniel described it perfectly: "more convincing crap is worse than obvious crap." You can't dismiss it quickly, you must investigate it, and you waste real time getting to the proof that it's nonsense. At scale, this stops feeling like a helpful external contribution model and starts to resemble something closer to a denial-of-service attack on the people responsible for security.*

*And the impact has been truly devastating: The Internet Bug Bounty program paused all new submissions because AI has dramatically increased discovery volume beyond what maintainers can handle. Node.js lost its bounty when funding disappeared. The reports still come in but the payouts are gone. And Curl removed financial rewards after being flooded with AI-generated reports. Casey emphasized that this isn't a new problem, It's an old one, just massively accelerated. He said: "We're doing stupid things faster with more energy."*

*Bug bounty has always had an issue with being a level playing field: one person submits a report, and another person has to validate it. That sounds equal on paper but in practice, it has always been difficult for one person to keep up with validation, even before AI existed. Now, it's practically impossible. We're now in a world where anyone can generate dozens of reports, make them appear credible, and submit them instantly. On the receiving end, however, the constraints have not changed. It's still humans reviewing, triaging, and making decisions.*

*Open source has been the first to feel this impact. Open source is where this pressure has shown up first, largely because it was already operating close to its limits. Most projects are maintained by small teams, often volunteers, with limited time and resources, yet they underpin massive portions of the internet. Add financial incentives, global participation, and now AI-generated submissions, and the system is quickly overwhelmed.*

*The Internet Bug Bounty program said it directly: "AI-assisted discovery has shifted the balance between findings and remediation capacity." Translation: We're finding more bugs than we're able to handle. So now the bounty is gone, and yet the expectation of reporting remains. But the question is: is the way bug bounty programmes have been used to effectively scale security teams and improve security posture still viable without financial incentives?*

*BugCrowd's founder, Casey Ellis doesn't necessarily believe so. Every organization should have a vulnerability disclosure program, because if you're on the internet, people will find issues. But not every organization is in a position to run a public, reward-driven bounty program. In Casey's words, curl likely should not have had one to begin with. Casey said: "I don't think every organisation should run a bounty program... the curl program should not have been a bounty program in the first place." And yet, Daniel's experience shows something more nuanced. Daniel views the bounty program as a success, because it incentivized real scrutiny of the code. He said: "I've always thought about it as a success because it's a great way to actually encourage people to scrutinize the code."*

*So what happens when you remove financial incentives? You'd assume that when you remove financial incentives, you'd get rid of AI slop, but that you'd also reduce the likelihood of genuine vulnerabilities being disclosed. However, when curl removed the financial incentives, something interesting happened. The low-quality, AI-generated noise largely disappeared. Daniel said: "We have stopped getting AI slop security reports. Instead, we get an ever-increasing amount of really good security reports, submitted in a never-before seen frequency which put us under serious load."*

I'll interrupt to mention that I have a theory about why this is: Back when discovering vulnerabilities required long hours of painstaking grueling work to step through and reverse-engineer code, it wasn't fun. The only motivation – and it needed to be significant – was the promise of a big pot of gold payout at the end of that tunnel. AI-driven vulnerability discovery has changed that. Today, AI makes bugs both fun and easy to find. It allows less skilled users to participate, thus broadening the bug hunter base, and there are plenty of people who would sincerely like to give back and contribute. Until now they haven't been able to. But now they have the means. They don't need a monetary incentive. They truly want to help. Aikido continues:

*Instead of drowning in low-quality reports, maintainers are now dealing with a high volume of genuinely useful findings, many of which are powered by AI-assisted research. The barrier to entry has dropped, not just for bad reports, but for good ones too.*

*But this creates a new kind of pressure. Even high-quality reports take time to understand, validate, and repair. And many of these "good" findings still fall into gray areas, bugs that may not meet security thresholds but still require attention. The result is a sustained, and in some ways increased, load on already constrained teams. So in a strange way, the system hasn't been relieved. It's been refined. And this is where it gets interesting. Because while this is painful in the short term, it might actually be a step in the right direction.*

*By removing financial incentives, we strip away a large portion of the noise. What's left is a signal that is, on average, higher quality, more intentional, and more aligned with actual security outcomes. AI is lowering the barrier for researchers to do meaningful work. It's enabling more people to find real issues, faster than ever before. That combination: less noise, more signal, but still overwhelming volume — suggests we're in a transition phase. The historical model is breaking under the pressure. But what's emerging underneath it might be better. This would look like a system where:*

- *disclosure is expected, not incentivized*
- *rewards are more targeted, not broad*
- *and the focus shifts from more reports to better outcomes*

*We're not there yet. Right now, we're in the messy middle, where the old model no longer works, and the new one hasn't fully formed. But if this plays out correctly, we don't end up with less bug bounty. We end up with a more sustainable version of it.*

*What we're likely moving toward is a model where vulnerability disclosure becomes a baseline expectation across the industry, rather than something optional or incentivised. Public bounty programs don't go away, but they become more controlled, more targeted, and more aligned with organisational maturity. AI will inevitably play a larger role in filtering and triaging the growing volume of reports. It won't solve the problem entirely, but it will become part of how we manage it. We'll also see a shift in what gets rewarded. As automated systems become better at finding low-level issues, the value of those findings will drop. Instead, incentives will move toward higher-impact work: the kind that requires creativity, context, and a deeper understanding of systems.*

*That means researchers will increasingly focus on areas like chaining vulnerabilities, exploiting business logic, and breaking complex or emerging technologies where automation may continue to struggle.*

Okay. So think about this from the bounty provider's standpoint: Take Curl as an example. Daniel terminates bug bounty payouts and observes an immediate drop in the total number of reports. But it's the bogus reports that disappear, not the useful reports that describe true problems. Given that, why would he ever resume bounty payouts? The Internet Bug Bounty is likely to observe the same thing. As I noted, what appears to be happening is that bugs are now so much easier to discover – even fun to find and report – that it's no longer necessary to dangle a carrot. Actual human altruism, which, believe it or not still exists, is now sufficient to drive what once required the promise of payment.

It'll take a while for this to percolate throughout the industry, but my prediction is that the 31 years of bug bounty programs we've had since Netscape offered the first payment for reports of bugs in Navigator 2.0 is likely to wind down over time.

### **Anthropic moves "Claude Security" into public beta**

And apropos of the changes wrought to AI vulnerability discovery we have Anthropic's announcement of "Claude Security" entering public beta for their Enterprise customers. Think of it as "Mythos Junior." Here's what Anthropic posted:

*Claude Security is now available in public beta to Claude Enterprise customers. AI cybersecurity capabilities are advancing fast. Today's models are already highly effective at finding flaws in software code; the next generation will be more capable still, and will be particularly effective at autonomously exploiting these flaws. Now is the time for organizations to act to improve their security, preparing for a world in which working software exploits are much easier to discover.*

*Recently, we made Claude Mythos Preview—which can match or surpass even elite human experts at both finding and exploiting software vulnerabilities—available to a number of partners as part of Project Glasswing. But our cybersecurity efforts go beyond Glasswing: with Claude Security, a much wider set of organizations can put our most powerful generally-available model, Claude Opus 4.7, to work across their codebases. Opus 4.7 is among the strongest models available for finding and patching software vulnerabilities, and for discovering complex, context-dependent issues that might otherwise be missed.*

*Claude Security—previously known as Claude Code Security—has already been tested by hundreds of organizations of all sizes in limited research preview, helping teams scan their codebases for vulnerabilities and generate targeted patches. Their feedback has shaped today's release, which makes Claude Security available to **all** Enterprise customers. It comes with scheduled and targeted scans, easier integration with audit systems, and improved tracking of triaged findings. No API integration or custom agent build is required: if your organization uses Claude, you can start scanning today.*

*Opus 4.7's capabilities are also being brought to cyber defenders through Claude's integration into software tools that many enterprises already use. Our technology partners, including CrowdStrike, Microsoft Security, Palo Alto Networks, SentinelOne, TrendAI, and Wiz are embedding Opus 4.7 into their tools; in addition, services partners like Accenture, BCG, Deloitte, Infosys and PwC are now helping organizations deploy Claude-integrated security solutions. We are entering a pivotal time for cybersecurity. AI is compressing the timeline between vulnerability discovery and exploitation. We believe the right response is to make sure defenders have access to frontier capabilities in the ways most accessible to them, through Claude directly and through our partners.*

*Claude Security can be accessed directly from the Claude.ai sidebar, or at [claude.ai/security](https://claude.ai/security). To begin, select one of your repositories (or scope to a specific directory or branch), then start a scan. While scanning, Claude reasons about code much like a security researcher. Rather than finding vulnerabilities by searching for known patterns, Claude seeks to understand how components interact across files and modules, traces data flows, and reads the source code. Once complete, Claude provides a detailed explanation of each of its findings, including its confidence that the vulnerability is real, how severe it is, its likely impact, and how it can be reproduced. It also generates instructions for a targeted patch, which users can open in Claude Code on the Web to work through the fix in context.*

*Over the past two months, we've refined Claude Security in line with what we learned from its use in production across hundreds of enterprises. Specifically, we've seen that: Detection quality is paramount. Teams have told us that high-confidence findings are what really accelerate security work. Claude Security's multi-stage validation pipeline independently examines each finding before it reaches an analyst, which drives down false positives, and Claude attaches a confidence rating to every result. This means that the signal that reaches the team is worth acting on.*

*Time from scan to fix is the metric that matters. Early users pointed to this consistently, with several teams going from scan to applied patch in a single sitting, instead of days of back-and-forth between security and engineering teams. Teams want ongoing coverage, not one-off audits. We've added the option to schedule scans, so teams can set a regular cadence around reviewing and acting on findings.*

*With this release, we've also added the ability to target a scan at a particular directory within a repository, dismiss findings with documented reasons (so that future reviewers can trust prior triage decisions), export findings as CSV or Markdown for existing tracking and audit systems, and send scan results to Slack, Jira, or other tools via webhooks.*

Given the wind-up we've seen from Mythos over the past month, I cannot imagine why any organization whose software might contain exploitable vulnerabilities or bugs would not be jumping on this with all possible speed. As I noted a few weeks back, an organization's own internal software is only closed-source to the outside world. To the organization their own source code is wide open, and there is now an emerging tool that stands a good chance of discovering bugs that have, until now, escaped notice. I would love to be a fly on the wall in the software development dungeons of the world's enterprises.

### **OpenAI gets very serious about ChatGPT login security**

Also last Thursday, OpenAI announced that they have decided to make account login security a selling point. Their posting titled "Introducing Advanced Account Security" explains:

*Today, we're introducing Advanced Account Security, a new opt-in setting for ChatGPT accounts, designed for people at increased risk of digital attacks, as well as for those who want the strongest account protections available. It brings together a set of heightened security measures that help safeguard against account takeover while making those protections easier to activate in one place. Once enrolled, Advanced Account Security protects users in Codex as well.*

*People are turning to AI for deeply personal questions and increasingly high-stakes work. Over*

*time, a ChatGPT account can hold sensitive personal and professional context, and sit at the center of connected tools and workflows. For some people, like journalists, elected officials, political dissidents, researchers, and those who are especially security-conscious, the stakes are even higher.*

*This effort is part of our broader cybersecurity action plan to broaden access to the technologies that can help protect communities, critical systems, and our national security. We want users to have the controls to make the security and privacy choices that are right for them. At the same time, we want to ensure users understand that the increased protection of Advanced Account Security comes with an increased responsibility for account recovery.*

*Advanced Account Security brings together a series of controls that strengthen sign-in protections, tighten account recovery, reduce exposure from compromised sessions, and give users more visibility into account activity. It's available to opt into in the Security section of users' ChatGPT accounts on web. Protection applies to both ChatGPT and Codex accounts that are accessed through that login.*

- Stronger sign-in methods. Advanced Account Security requires passkeys or physical security keys while disabling password-based login, helping make phishing-resistant sign-in the default for people who need it most.*
- More secure account recovery. If a user's email account or phone number is compromised, an attacker may try to use one of them to gain access to their ChatGPT account via e-mail or SMS based recovery. To reduce this risk, Advanced Account Security disables email and SMS recovery and requires stronger recovery methods: backup passkeys, security keys, and recovery keys. Because account recovery is restricted to these more secure methods, OpenAI Support will not be able to assist with account recovery for users enrolled in Advanced Account Security.*

Wow! Now we're talking. Hopefully this sort of much more responsible security becomes more commonplace. The only gotcha, of course, is that it makes users entirely responsible for the security they claim to want and cherish. By explicitly removing email and SMS account recovery loops, the most common phishing and other attacks will be thwarted. But I can see this tradeoff making sense for ChatGPT login. OpenAI explains two additional security enhancements, writing:

- Shorter sessions and clearer session management. Sign-in sessions are shortened to reduce the window of exposure if a device or active session is compromised. Users also receive alerts when there is a login to their account, and they can review and manage the active sessions across the various devices they're signed into.*
- Automatic training exclusion. People working with especially sensitive information may opt not to have those conversations used for model training. With Advanced Account Security enabled, that preference is automatic: conversations from those accounts will not be used to train our models.*

*Using physical security keys, such as YubiKeys, is one of the strongest defenses against phishing. To make that level of protection easier to access, we have partnered with Yubico, a leader in hardware-based authentication and account protection, to offer our users preferred pricing on a customized bundle of best in class security keys. The YubiKey C Nano is designed to stay in your laptop for simple, low-friction daily authentication, and the YubiKey C NFC for*

*backup, and use across laptops and mobile devices.*

*We're launching this partnership as part of Advanced Account Security, but the bundle will be available to all eligible users in their security settings on web so more people can adopt stronger, phishing-resistant account protection. Users will also be able to use any other FIDO-compliant security key, or use software-based passkeys.*

I logged into ChatGPT, which I no longer use as my daily driver since I've switched to Claude after appreciating how confused an AI's context window would become if I were to share it with my wife Lorrie. So now we each have our own.

Once there, sure enough, the "Security" panel of the "Settings" dialog now has many new features. This is great, and I expect we'll see this sort of enhanced security become a standard feature to more rigorously protect the potentially more highly sensitive dialogs many people will be having with their AI chatbots. Once you appreciate – which Claude recently made explicitly clear – that the entire history of your conversation is, by default, retained for use in creating a conversational context, the importance of more tightly controlling its access becomes clear.

## SyncThing & SyncTrayzor

### Updating SyncThing & SyncTrayzor

I wanted to take a moment to talk about SyncThing and SyncTrayzor. We've spoken often about SyncThing, of which both Leo, I, and many of our listeners are huge fans. SyncTrayzor is a terrific little Windows GUI wrapper that turns SyncThing into more of a Windows app. In the words of its creator:

*SyncTrayzor is a little tray utility for Syncthing on Windows. It hosts and wraps Syncthing, making it behave more like a native Windows application and less like a command-line utility with a web browser interface. Features include:*

- *Has a built-in web browser, so you don't need to fire up an external browser.*
- *Optionally starts on login, so you don't need to set up Syncthing as a service.*
- *Has drop-box style file download / progress window*
- *Tray icon indicates when synchronization is occurring.*
- *Alerts you when:*
  - *You have file conflicts*
  - *One of your folders is out of sync*
  - *Folders finish syncing*
  - *Devices connect / disconnect*
- *Has a tool to help you resolve file conflicts*
- *Can pause devices on metered networks, to stop Syncthing transferring data on e.g. a mobile connection or wifi hotspot.*
- *Contains translations for many languages*

I've been using both for years and I hope to continue doing so. As we've mentioned SyncThing can also be installed into the Synology NAS and I've been using it for many years, ever since my first Drobo died and I took the opportunity to switch to Synology. It works perfectly there, too.

I'm mentioning all of this since SyncThing on Windows 10 has been noting that v2.0.16 is available for some time. Since I had heard from several listeners that the major version 2 of SyncThing is fully backward compatible with version 1.3, which is where I'm stuck on my Windows 7 machine, I decided it was time to quiet that new version available notice. But when I updated SyncThing it complained about an unknown command line switch. The trouble was that the version of SyncTrayzor I was also still using was launching SyncThing as its client and using a command line argument that the updated SyncThing didn't recognize ... so it refused to run.

I decided to see whether SyncTrayzor had been updated. That's when I learned that SyncTrayzor's creator had abandoned his baby last August when he archived his Github project. At the time, he wrote:

*I stopped using Syncthing some years ago, and I'm afraid I don't have the time to maintain it. Sorry. GermanCoding has kindly forked it as SyncTrayzor v2 and is continuing development, and this fork is recommended by Syncthing. Please switch to SyncTrayzor v2 (after determining that you trust the fork!).*

I first verified that the SyncThing project does, indeed, still recommend the use of this forked SyncTrayzor v2, and they do indeed: <https://docs.syncthing.net/users/contrib.html#windows>

In another month or two I'll be consolidating my locations, so rather than having SyncThing keeping NASs and my working directories in PCs synchronized at separate locations, I'll be using it in its unidirectional mode to maintain versioning backup of my primary working machine.

I wanted to let anyone who's been happily using Syncthing know that, indeed, major version cross-compatibility works and about the SyncTrayzor v2 which they might be interested in moving to. The upgrade was completely painless. I ran the new installer which saw that I had an existing installation and asked whether I wanted to start over fresh or upgrade what I had. I chose the latter and everything worked perfectly.

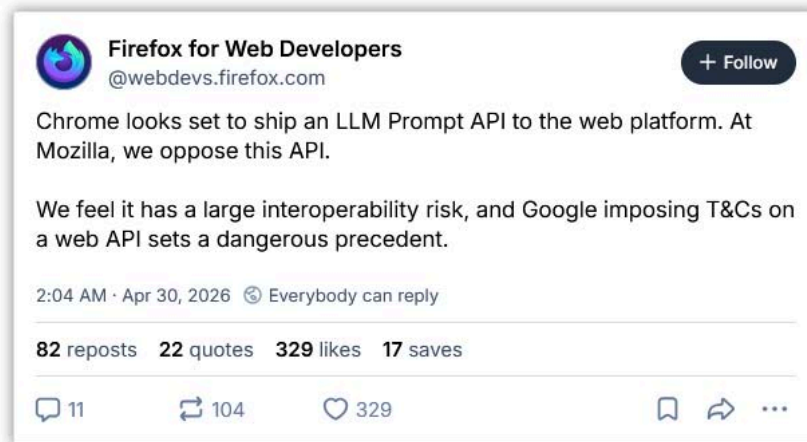
<https://github.com/GermanCoding/SyncTrayzor/releases/tag/v2.1.0>

# A Browser AI API

Google is planning to define a new API to bring AI into our web browsers. This would serve as an interface to Large Language Models existing outside the browser. Google appears to be mostly targeted at local LLMs, but support for cloud-based LLMs is present, too. So this would be a means for allowing web pages or browser extensions to invoke a user's local or remote large language models for many purposes such as locally reading and summarizing a web page's content, proof reading a web-based document being edited, or reading through someone's web mail to produce summaries or take actions. In other words, it would create a JavaScript AI prompting interface.

Not everyone thinks this is a good idea. And many of those "not everyones" includes end users who feel uncomfortable with this creeping trend toward "AI'ifying" everything. An early instance and example of this, which we covered at the time, was the Vivaldi browser's CEO Jon von Tetzchner who said: *"We don't see AI as something that our users are asking for. Rather the opposite. I think a lot of people are reacting to force-fed AI."* Jon cited as a "no thanks" example Microsoft's Recall compiling a long-term history of everyone's desktop screenshots every 5 seconds. Giving Recall the label of AI seems sort of quaint today. We've come a long way in a short time. Tetzchner said that *"the future of browsers is about who controls the pathway to information, and who gets to monetize you"* which frames the race to insert AI into our browsers as a power grab more than a feature competition.

The thing that put this issue on my radar last week was seeing that Vivaldi's Jon von Tetzchner has some other company, notably Mozilla. In a posting to Bluesky last Thursday April 30th, Mozilla's Jake Archibald wrote:



Before I go any further, I want to touch on those terms and conditions since that alone is a complete deal breaker for me. Last week, in a thread in Mozilla's Github account Jake wrote:

*According to Chrome's documentation, to use the prompt API you must 'acknowledge' Google's Generative AI Prohibited Uses Policy. Elements of this policy go beyond law. For example:*

- *Do not engage in generating or distributing content that facilitates sexually explicit content.*
- *Do not engage in misinformation, misrepresentation, or misleading activities. This includes facilitating misleading claims related to governmental or democratic processes.*

So here we have a proposed web browser API that implicitly contains acceptable use policy. This would be like a web browser refusing to display controversial four-letter words on the grounds that someone might be upset by what a website might wish to have their browser display. Hearing this causes me to want to select a couple of choice four-letter words myself. This is SO WRONG. Thank goodness we have respected developers at Mozilla to push back. I hope this also captures the attention of the EFF.

To obtain some pro and con balance here, let's first look more closely at the new "Prompt API" that Google has already implemented and moved into Chrome for development testing. The Explainer for this nascent feature says:

*This explainer and the accompanied draft report are in active development by the Web Machine Learning Community Group. Community Group members are seeking feedback and support for this proposal to gain Working Group and implementer adoption. Implementations are experimentally available in Google Chrome and Microsoft Edge.*

*Browsers and operating systems are increasingly expected to gain access to language models. Language models are known for their versatility. With enough creative prompting, they can help accomplish tasks as diverse as:*

- *Classification, tagging, and keyword extraction of arbitrary text*
- *Helping users compose text, such as blog posts, reviews, or biographies*
- *Summarizing, e.g. of articles, user reviews, or chat logs*
- *Generating titles or headlines from article contents*
- *Answering questions based on the unstructured contents of a web page*
- *Translation between languages*
- *Proofreading*

*The Google Chrome, Microsoft Edge, and the Web Machine Learning Community Group are exploring purpose-built APIs for some of these use cases (namely translator / language detector, summarizer / writer / rewriter, and proofreader). This proposal additionally explores a general-purpose "Prompt API" that allows web developers to prompt a language model directly. This gives web developers access to many more capabilities, at the cost of requiring them to do their own prompt engineering.*

*Currently, web developers wishing to use language models must either call out to cloud APIs, or bring their own and run them using technologies like WebASM and WebGPU, usually through JavaScript runtime frameworks. By providing web platform API access to the browser or operating system's existing language model, we can provide the following benefits compared to cloud APIs:*

- *Local processing of sensitive data, e.g. allowing websites to combine AI features with end-to-end encryption*
- *Potentially faster results, since there is no server round-trip involved*
- *Offline usage*
- *Lower API costs for web developers*
- *Allowing hybrid approaches, e.g. free users of a website use on-device AI whereas paid users use a more powerful API-based model*

I'll interrupt here to note that all of those seem mostly like made-up reasons: "*Local processing of sensitive data, e.g. allowing websites to combine AI features with end-to-end encryption.*"

I get the local processing angle. That's potentially valid. But the end-to-end encryption part makes little sense to me in this context. We already have TLS connections with all websites and we have decades of history and experience with making that bulletproof. Then there's: "*Potentially faster results, since there is no server round-trip involved.*" So the assumption here is that a local, potentially underpowered LLM is going to outperform an LLM in these monster data centers that are being frantically built? And so on for those remaining three benefits. Our browsers already have the ability to query cloud-based LLMs using the tried and true XMLHttpRequest API or the more recent Fetch API. And both of those offer state of the art mature security & privacy protections.

So what really appears to be going on here is for Google to be engineering a means for their Chrome and other Chromium-based browsers to access non-cloud-based LLMs since everyone can already do that. Their explainer continues:

*Compared to developer-supplied model approaches, using a built-in language model can save the user's bandwidth, storage, and memory resources, while using a model that is optimized for the device. This pattern can also provide a lower barrier to entry for web developers by removing the need for developers to serve models and manage dependencies.*

I have to interrupt again to note that none of that makes any sense to me. This presumes that any and all large language models are identical and interchangeable, and that the web developer doesn't care which one they are interacting with. Today, it's already not the case that all LLMs are identical and interchangeable and I expect model design and capability to diverge as we move into the future, not converge.

So next Google's explainer clearly states its goals:

*Our goals are to:*

- *Provide web developers a uniform JavaScript API for accessing browser-provided language models of varying capabilities.*
- *Encapsulate model management and execution details as much as possible, e.g. downloads, updates, templating, parsing.*
- *Guide web developers to gracefully handle failure cases, e.g. no browser-provided model being available.*
- *Develop formal implementation guidelines and definitions; e.g. initial on-device models, and possible cloud services.*

*The following are explicit non-goals:*

- *We do not intend to force every browser to ship or expose a language model; in particular, not all devices will be capable of storing or running one. It would be conforming to implement this API by always signaling that no language model is available; it may also be viable to implement this API entirely by using cloud services instead of on-device models.*
- *We do not intend to provide guarantees of language model quality, stability, or interoperability between browsers. In particular, we cannot guarantee that the models exposed by these APIs are particularly good at any given use case. These are left as quality-of-implementation issues, similar to the shape detection API. (See also a discussion of interop in the W3C "AI & the Web" document.)*

*The following are potential goals we are not yet certain of:*

- Allow web developers to know, or control, whether language model interactions are done on-device or using cloud services. This would allow them to guarantee that any user data they feed into this API does not leave the device, which can be important for privacy purposes. Similarly, we might want to allow developers to request on-device-only language models, in case a browser offers both varieties.*
- Allow web developers to know some identifier for the language model in use, separate from the browser version. This would allow them to allowlist or blocklist specific models to maintain a desired level of quality, or restrict certain use cases to a specific model.*

*Both of these potential goals could pose challenges to interoperability, so we want to investigate more how important such functionality is to developers to find the right tradeoff.*

In other words, we and the world are not yet really ready for this or in need of this, so we're unsure how it should work, exactly, but we're going to charge ahead because this will be better than nothing.

But will it? Today's web browsers are littered with yesterday's great ideas that, while they may have never achieved critical mass, must still be present and supported since some random website somewhere still uses them. As one example, it may not be fair to single out FLASH, since it did have its day, but boy was it difficult to kill it off. And in some places it still won't die.

As I look over the Prompt API implementation specification I can empathize with Mozilla's gut reaction since this seems quite forced and unnatural. For example, this API defines a specific "System Prompt" as they call it. The specification says:

*The language model can be configured with a special "system prompt" which gives it the context for future interactions. The system prompt must be the first message, whether passed via the initialPrompts option to create(), or as the first message to the first prompt() or append() method call.*

We then see three examples of these various semantic options. The first one is:

```
// Create a new session with a system prompt as the first message.  
const session1 = await LanguageModel.create({  
  initialPrompts: [{ role: "system", content: "Pretend to be an eloquent hamster." }]  
});  
console.log(await session1.prompt("What is your favorite food?"));
```

My reaction to all of this is that web standards are too important to be created in any half-baked fashion, and Mozilla apparently also feels that it's too soon to do this. Once a web standard exists it's incredibly difficult to depreciate it since, as we saw with FLASH, someone, somewhere, will be using it. Browser bloat, and the security implications of that, are very real problems.

Google's working specification goes on and on (and on) and it's all extremely specific to the application of today's LLMs. They are creating something as important at an industry-wide specification for what just must be the moment we're in today. That alone seems misguided. I've dropped the URL to Google's full specification into the show notes for anyone to follow-up if

interested: <https://github.com/webmachinelearning/prompt-api/blob/main/README.md>

I want to now switch to Mozilla's response. I have the rather dry conversation thread in Mozilla's Github account under "standards positions", so I've dropped **that** URL into the notes, too: <https://github.com/mozilla/standards-positions/issues/1213#issuecomment-4347988313>

But since this podcast endeavors not only to inform but also to entertain our listeners, rather than sharing Mozilla's dry recitation, I want to share The Register's typically feisty and irreverent take on this controversy. They also supply a great deal of additional useful background, and when we see that their headline is "*Firefox maker torches Google for building Prompt API into browser*" you know it's going to be good. The Register wrote:

*Jake Archibald, Mozilla web developer relations lead, articulated the org's concerns in a GitHub discussion of the API, which provides a standard way to send and receive prompts and responses from a local machine learning model. Archibald wrote "We continue to oppose this API, and feel it has severe negative consequences to the interoperability, updatability, and neutrality of the web platform."*

*The Prompt API, as Google describes it, "gives web pages the ability to directly prompt a browser-provided language model." Specifically, it provides a way to send natural language instructions to **Google's Gemini Nano model**, which is small enough to be downloaded for local inference through Chrome. However, it's not small – Google recommends having 22 GB of space available, though the Nano (v3Nano) model for desktop use is ~4.27 GB.*

*Web developers already have a variety of ways to interact with AI models. They can use cloud service APIs to communicate with hosted models. Or they can access local models through technologies like JavaScript runtime frameworks, WebASM, or WebGPU. Various vendors like OpenAI and Perplexity have shipped browsers that embed access to remotely hosted AI models. Mozilla itself is testing an AI-based Smart Window in Firefox and is developing tools for AI model scaffolding.*

*The Prompt API aims to make it easier to run local inference in a way that takes advantage of browser security mechanisms, to produce faster response times, to allow offline usage, and to provide more cost effective ways to integrate AI services (e.g. providing a free AI fallback if users lack a paid AI API key).*

Oh. So that's interesting. That suggests that Google wants us to register our LLM AI provider accounts with our browser so that random websites we visit will be able to submit their prompts to our AI account. This brings to mind that famous rhetorical question "*What could possibly go wrong?!*" The Register continues:

*Mozilla's concern, as articulated by Archibald, has to do with what the Prompt API means for the web, not to mention Google's justification for deployment. First, he worries that Google's own Nano model will become the default and that developers will standardize on it in an effort to make the non-deterministic responses of an AI model more predictable. That tendency, he argues, will create pressure for Apple and Mozilla to license Nano, for the sake of a common user experience. Perhaps more significantly, Archibald notes that using the Prompt API requires agreeing to Google's Generative AI Prohibited Uses Policy, which prohibits activities that are not necessarily illegal, like generating "disturbing" content.*

Yikes! Who determines what content is "disturbing"? There's nothing attorneys love more than

ambiguous language in contractual agreements. It's a built-in full employment guarantee. The Register quotes Jake saying: *"This seems like a bad direction for an API on the web platform, and sets a worrying precedent for more APIs that have [browser]-specific rules around usage."* Amen to that. The Register continues:

*Finally, Archibald argues that Google misrepresented demand for the API by cherry-picking a few social media posts and calling that a groundswell of developer support. Jake posted: "The intent to ship on blink-dev states web developers as 'Strongly positive,' and links to the explainer for evidence. The evidence provided there does not seem to fit the claim."*

*In an email, Archibald told The Register that the question is whether the Prompt API is good for the web, and Mozilla doesn't believe that it is. Jake said: "The core problem is interoperability. Prompts are tightly coupled to models; developers will inevitably tune to the quirks and policies of whatever model they're building against. That's how you end up with model-specific code paths, which is the browser-compatibility problem all over again. The Terms & Conditions issue is part of that: if using a web API means accepting a specific vendor's content policy (especially one that goes beyond law) you're not really building for an open platform anymore."*

*With regard to Google's exaggeration of developer enthusiasm, Archibald said there are definitely devs interested in AI capabilities but Google failed to provide evidence of that: "The signal is polarised, not 'strongly positive'. But either way, developer demand alone does not meet the bar. The question is whether the API can work across implementations without tying the platform to one vendor's model."*

*Google did not immediately respond to a request for comment. However, on Thursday, Rick Byers, the Google Chrome engineer responsible for shipping the Prompt API, chimed in to the GitHub discussion to acknowledge the concerns articulated by Archibald.*

*He wrote: "As one of the blink API owner approvers for shipping this in Chromium, I admit that I share the concerns here in Mozilla's standards position. Where I differ is in preferring paths that promote experimentation, learning from mistakes, and competition to those which err on the side of stalling innovation out of fear of what might happen."*

Right. That's a perfectly articulated response to the more cautious "we should wait a bit to see what happens" stance. The Register concludes their piece by writing:

*Byers asked the web community to help collect evidence of harm to advance the discussion. Pointing to the debate over other controversial web technologies like Encrypted Media Extensions (EME), he suggested the outcome has not been as dire as predicted.*

*But focusing on data, so far, hasn't done much for Google's cause. According to a report created in February that compares the performance of Chrome (Gemini Nano) and Edge (Phi-4 mini-instruct) using the Prompt API, these models do not provide very good results. The report says: "For generative tasks (composition, tag generation, etc), 24.29% Edge's and 15.17% of Chrome's responses failed to complete the task." This is in reference to a rubric that defines failure as a score of 2 or less on a scale of 1 to 5. "For classification tasks, 29.58% of Edge's and 23.93% of Chrome's responses did not label or categorize input correctly."*

They finish with the report's conclusions noting:

*In terms of groundedness and accuracy, Edge failed ("hallucinated") 17% of the time while Chrome failed 6% of the time. Is that good for the web? You could ask Chrome but you might not get a reliable answer.*

Okay. So where does this leave us? It leaves me more happy than ever that I've stuck with Mozilla.

I look at what Google now presents us on a page of search results and it becomes clear that we're the product. I search for something specific and sponsored interception advertisements are promoted and presented before the result I'm seeking. Then I need to wade down past a bunch of YouTube videos that I have zero interest in. Now, in fairness, Google is not alone in doing this. Apple has similarly succumbed in their App Store. The thing I'm looking for is never first – even when I search for it by name and spell it correctly. What's first is what someone paid them to show me first in the hope that I wouldn't notice or wasn't sure what I wanted. And on the Google side, in return for tolerating a bunch of advertising, we receive a ton of services at no charge. I author these show notes every week in Google Docs for free, and the catch-all junk email account I maintain at Gmail is similarly valuable. All that means a lot. Thank you, Google.

But all of that seems fundamentally different from intermixing the design and establishment of crucial web standards with a single company's commercial interests. Yes, Google has succeeded in leveraging their position as the winner of Internet search into the winner of the web browser wars. I get it. As I use the Internet daily I am more or less continually being offered the opportunity to improve my life in one way or another by switching to Chrome. I constantly need to decline. Most people have given up declining and they're perfectly happy using Chrome – whether or not their lives are better for it. And that's great.

But tremendous responsibility burdens Google's dominance with Chrome. They **need** someone knowledgeable to push back and to question their actions – if for no other reason than to help them make the best choices. So I'm very pleased that we have Mozilla watching and actively participating. Google may, and likely will, still plow ahead and force Mozilla to keep up or to be left behind and become irrelevant. But everyone will likely get a better browser, whether that's Chrome, Edge, Safari, Brave, Vivaldi or Firefox.

