



## A Browser AI API?

**Description:** 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.

High quality (64 kbps) mp3 audio file URL: <http://media.GRC.com/sn/SN-1077.mp3>

Quarter size (16 kbps) mp3 audio file URL: <http://media.GRC.com/sn/sn-1077-lq.mp3>

---

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. He is armed with the knowledge that Google is now downloading 4.7GB when you download Chrome. What is it? A local AI model. Steve talks about its implications next on Security Now!.

**Leo Laporte:** This is Security Now! with Steve Gibson, Episode 1077, recorded Tuesday, May 5th, 2026: "A Browser AI API?"

It's time for Security Now!. Yes, it's Tuesday. Yes, I am not in my studio. I am in beautiful Hawaii, the Big Island. But that doesn't mean I'm not going to get Steve Gibson on the horn and talk about security because I know you need your fix. Hello, Mr. G.

**Steve Gibson:** You know, Leo, you look a little more tan than you did last time we saw you.

**Leo:** Oh, aloha. I am. See, my hand is light, but my face is a little dark.

**Steve:** Do Mai Tais increase skin pigmentation?

**Leo:** Maybe that's what it is. Maybe that's what it is.

**Steve:** Something.

**Leo:** Yeah, we're on vacation, but I can't, you know, I still want to do the shows. And so I've set up - if you could only see this kooky setup. I am outside on the lanai.

**Steve:** We hear the exotic birds tweeting in the background.

**Leo:** There are some exotic birds. There's mynah birds. There's house sparrows. And there's a bird that looks like a little chicken called a Fulica? I can't remember the name of it.

**Steve:** Chicken bird.

**Leo:** Chicken bird. And it's very noisy, so you'll know if it decides. But the sparrows are very aggressive. They might think I have something to give them, so they might be coming up here and sitting on my shoulder.

**Steve:** Well, it all adds to the ambience that...

**Leo:** That's it.

**Steve:** ...we have with you and Lisa in the Big Island.

**Leo:** It's so beautiful. Have you been to Hawaii, Steve?

**Steve:** Oh, yeah. I had half of my - the second half of my honeymoon in Hawaii.

**Leo:** Ah. I love Hawaii.

**Steve:** So, yeah.

**Leo:** So what's coming up on Security Now! this week?

**Steve:** So Episode 1077. It's funny, every so often I think about 1077, which sort of puts the infamous 999 into context.

**Leo:** Yeah.

**Steve:** Now it's been a while.

**Leo:** Yeah, more than a year.

**Steve:** Actually it's been 77 whiles, yeah.

**Leo:** Yeah.

**Steve:** So there were two main topics which were contesting to win the coveted title of the podcast this week. Google's arguably premature move to build AI into Chrome ended up winning because Mozilla has said, uh, not so fast.

**Leo:** Yeah.

**Steve:** But we've got a lot of good things to talk about. Turns out that some hackers used AI to code up a portal for credit card, stolen credit card verification? They forgot to ask it to add authentication. So, whoops. Also the UK's security group, the NCSC, has issued their own Mythos warning, which caused me to wonder, where is CISA? Why hadn't we heard anything from CISA? We're going to touch on that. We've got another of many recent Linux local privilege escalations. This one is bad, and it's affected Linux for years. And yes, AI found it. Also some interesting commentary about the ground shifting under AI and vulnerability research, how it's looking like it may spell the end of bug bounties, and why that is probably the right thing to have happen.

Also, Anthropic has released what they call Claude Security as a mini Mythos. ChatGPT has made some changes which demonstrate it's getting very serious about login security. I want to make a comment about something I discovered since we last talked about the end of life of SyncTrayzor v1, which is what I use to sort of bundle Syncthing into a nice little applet for Windows. But there's a replacement for it. And then we're going to talk about how Google has sort of surprised everyone by just saying we think it's time that we add AI support in JavaScript. So lots of fun things to talk about. And of course a great Picture of the Week. So, yeah. Oh, and there are a couple things that happened just now. Just so that our listeners know that I'm aware of the fact that DigiCert suffered a major breach...

**Leo:** Oh.

**Steve:** ...which allowed 30 EV code signing certificates to get minted behind their back.

**Leo:** Oh, that's not good.

**Steve:** And used. However, their disclosure is being called a "reference," a state-of-the-art, this is the way you do it.

**Leo:** Oh, good. Oh, good.

**Steve:** If you're going to say what happened, if you're going to share with the industry your post-mortem. So they just updated it 10 minutes ago. So it's still a little bit in flux. We'll take a look at what they had to say, things that went right, things that went wrong, and what they learned. It ended up being a social hack. A malicious screensaver, of all things, got into two of their tech support member PCs. And it wasn't detected due to a

CrowdStrike endpoint security misconfiguration. So anyway, I'm all up to speed on it, but I just didn't have a chance to, well, actually we're still learning a lot, and it's still in flux. So we'll have good coverage of that next week. I just wanted to let everybody know that I was aware of that.

**Leo:** Just shows you.

**Steve:** So we'll take our first break, we'll look at the Picture of the Week, and then we'll get into all this.

**Leo:** It really just shows you how anybody is vulnerable to this.

**Steve:** Yes.

**Leo:** And as you said when we were at the ThreatLocker Zero Trust World, the threat's coming from inside the house. A network engineer who put a screensaver on his system, and suddenly you're compromised. That's terrible.

You know, just to explain behind the scenes, we weren't sure this was going to work at all, and so we thought, well, I'd better prerecord the commercials in case, I don't know, Mikah had to jump in or something. And I brought a Starlink Mini and all sorts of backup stuff. And it turned out, wow, who knew, in Hawaii they've got cable modems, high-speed Internet. I didn't need to bring anything.

**Steve:** So it isn't a complete proof of concept of your ability to roam through, like, anywhere on the globe and use Starlink.

**Leo:** Not yet. Not yet. I probably should, though, you know, set it up. We have this nice lawn behind us. There's plenty - it's perfect space for the Starlink. So I probably should just set it up before I go home and make sure that I can do that. But, you know, everybody...

**Steve:** Could double as a birdbath, couldn't it.

**Leo:** Yes, it could. Or a serving tray. All right. I have - and, now, this is going to be another interesting experiment. I have the Picture of the Week. Shall we share it?

**Steve:** So this was great. Shared from a listener, of course, as are they all. I gave this one the caption "Attempting to preempt the inevitable question: 'Why has the lobster become so expensive?'"

**Leo:** Okay.

**Steve:** So we see the sign.

---

**Leo:** All right. I had seen the lobster part, but I hadn't seen the rest of it. That's hysterical.

**Steve:** The sign is taped to the window of a buffet or a restaurant or something, explaining, again, preempting the inevitable question. So the sign says: "All lobster prices have increased due to high lobster prices."

**Leo:** And I can see in the background there's, okay, a little old lady, an elderly person. And you know she went up to the guy at the restaurant and said, "Why are the lobster prices so high?" And he just pointed to the sign and said, "Lady, because they're so expensive."

**Steve:** He says, "Go read the sign." That's right. "All lobster prices have increased due to high lobster prices."

**Leo:** Oh, very nice, very nice.

**Steve:** What are you going to do? Okay. So we begin this week with a story that intersects several security fronts. Last Wednesday, Cybernews's headline was "Scammers vibecode server to verify stolen credit cards, leak details of 345K cards." I had to read that one twice to make sure. Here's what Cybernews 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," as in Tom and Jerry. There were some little cartoons of a mouse jumping around on things posted on the dark web. They said: "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" - in fact, it's one of the very earliest AI-based coding assistants from several years back, Cursor - "to set up the leaking server" - not knowing that it was leaking - "and to create administrator-facing dashboards. Cursor," they wrote, "is a legitimate service, developed by the U.S. 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" - imagine that - "for building the dashboards."

The team explained: "We were able to confirm 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" - because of course you didn't ask for it - "ignoring the need to set up authentication or ensure that only the intended dashboard would be accessible." In other words, it's just like a regular user. If you don't ask for authentication, you're not going to get authentication.

Anyway, they finish saying: "Moreover, the chat history reveals there was sufficient information for the Cursor large language model to identify that it was helping set up a credit card verification service, indicating a lack of sufficient guardrails to prevent abuse." And as you've often heard me say, I don't think you can really control a large language

model. Researchers said: "It's a lesson for developers using Cursor for legitimate uses, showing how it can lead to accidental data leaks." Right. It's just going to write what you ask it to. It's not going to, like, be your security nanny.

So CyberNews said that they'd 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, I think it's a great example of the danger of using AI without being a domain expert. That is, you know, without knowing what to ask for because you've, you know, it'll give you what you ask for, but you need to know what that is.

I've no doubt that the Cursor AI would have easily provided instructions for the authentication that was needed if it had been asked to. But apparently the bad guys never thought to ask. Somebody who wasn't really up to speed on web-based application security could easily fail to anticipate all the various ways others might access and penetrate their system. So expecting AI to produce secure solutions by default is probably a fool's errand. In this case, either it never occurred to them that authentication should be required where it was absent, or they didn't know it was going to be absent, or they assumed that the AI would know, you know, like what it should do and would do it without bidding.

The CyberNews article also provided some interesting background reporting a little bit on the underground industry in stolen credit cards, which I thought was interesting. 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 U.S., Amazon Japan, Grubhub, Sam's Club, Temu, Lyft, Elf Cosmetics, and CountryMax, utilizing hundreds, or in some cases thousands of accounts that have already been established on these platforms to perform credit card validity checks. Attackers created those accounts to register stolen cards and then perform 'low-risk' actions," as they call them. "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, of course, is a way to mask their activities. Since large merchants process billions of payments, tiny transactions on a well-known website don't ring any alarm bells." They wrote: "According to our team, the exposed server contained a treasure trove of credit card details." Details meaning, you know, everything you need to process someone's card. "Researchers identified nearly 200,000 credit card details that the service had verified as invalid, and over 145,000 accounts that they had verified as valid. The exposed information includes all the details that you need: credit card numbers, expiration dates, the security code, the cardholder's name, and their address."

"Typically," they wrote, "valid credit card details are sold for between \$7 and \$18 each on the dark web, meaning that the value of the valid stolen card data" - that's the 145,000 cards that have been verified there - "is somewhere between a million and \$2.6 million." They said: "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." That's just one of the types of fraud that they're making available in the store.

They said: "While it's unclear where Jerry's Store is located, internal tooling and leaked large language model 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 U.S. and the EU."

**Leo:** Fluent in Chinese but not AI, apparently.

**Steve:** Yes.

**Leo:** You know, this comes up a lot, and we're going to see more of these - whoops. Hold on. Camera, I'm over here. Thank you.

**Steve:** Nice ceiling fan.

**Leo:** People blame AI for stuff that they do that's dumb. There was a big story last week, and everybody blamed AI because a guy's database, production database got clobbered. But of course if you give AI the keys to your production database, it's on you, my friend. And if you're dumb enough to say, hey, just make me a website to authenticate, but don't put any authentication layers in there, AI's going to do what you say. I think this comes from sort of a magical belief about AI, that it's somehow intelligent, or it's going to take care of you. And it's not.

**Steve:** And probably it's a hope as much as a belief.

**Leo:** It's a hope. Yes, it's a hope.

**Steve:** I hope that AI knows how to do this. And since it seems to know a lot, I'm just going to assume that it does.

**Leo:** Well, it does if you tell it to. I mean, you're absolutely right. If you say - AI's great. If you say write the login page, use OAuth, make it secure, it will absolutely do that. But you have to tell it to. It's not going to assume that. It might. But again, it might not. The other thing I wanted to mention is I have had credit cards stolen due to my own stupidity. And the very - first of all, credit card companies know to look for those low-risk, low-value charges.

**Steve:** Right.

**Leo:** In fact, they used to say, if somebody buys sneakers and then tries to fill up a tank of gas, they invalidate that credit card immediately because that's the first thing somebody who steals a credit card is going to do. Times have changed. That's not true anymore. When my credit card was stolen, I mentioned this before, they added it to an Apple Wallet. So they had a prior setup Apple account and added it to Apple Wallet, which they then used the credit card through. Obscuring the source of the credit card, the actual credit card, is very, I thought, very clever. And I should have known because when I gave it the six-digit code it said, okay, we're trying to

add this to that Apple Wallet. And I said, "What are you talking about? I'm not trying to do an Apple Wallet." That should have been the hint to me that they were doing something funny. You know, it's a cat-and-mouse, Tom and Jerry kind of a game.

**Steve:** Yeah, it does feel like, you know, as I have said on the podcast before, I mean, in the early days, probably almost before this podcast, I used to fly up to Northern California to visit my family in Northern California for the holidays. And this was, you know, pre-Expedia and so forth, so I actually had a travel agent from the old, old, old days, who I just kept around. And when we would have our conversation, she would invariably say, "Well, so, Steve, do you have the same credit card, or have you lost that one, too?" Because, you know, I was out on the Internet poking around. And oh, yeah...

**Leo:** Oh, I don't feel so bad now.

**Steve:** I did lose that one, so. Okay. So last Friday, Ollie Whitehouse, the Chief Technology Officer, you know, the CTO 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 "Organizations must act now to prepare for a wave of patches that will address decades of technical debt." And I love that term in this instance. I think the term "technical debt" is exactly the right way to express the concept that, you know, 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." Yes, comeuppance, indeed.

So here's what the UK's NCSC CTO wanted everyone within the United Kingdom, within his sphere of influence, to appreciate. He wrote: "Whether they are technology producers and vendors, or consumers and operators, all organizations have 'technical debt,' a backlog of technical issues that's both expensive and time-consuming as a result of prioritizing 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'" - which is the way he phrased it, we're going to have 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 organizations 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 organizations must take steps to identify and minimize their Internet-facing (and other externally-exposed) attack surfaces as soon as possible. As we've argued for some time, you should prioritize technologies on your perimeter, and then work inwards covering cloud instances and on-premises environments. By doing this, organizations can reduce the risk posed by latent vulnerabilities when they become known and exploited by attackers. Where organizations cannot apply updates across their entire environment, they should prioritize applying updates to their external attack surfaces. Where capacity extends beyond the external attack surface, organizations should prioritize critical security systems.

"It is also important for organizations to realize that patching alone will not always suffice; some technical debt may be present in 'end of life' or legacy technology that's out of support, and so cannot receive updates. In such instances, organizations 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, organizations 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.

"NCSC recommend that" - and they have three - "where automatic secure 'hot patching' is available (that is, patching that does not involve service disruption), this should be enabled as a priority." Okay, well, that's, you know, not hard to imagine as the first. "Where automatic updates are available (including for embedded devices), this should be enabled to reduce the workload on support teams." So, yeah, turn on automatic updates and go for it. And third: "Where neither of the above are available, organizations 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-prioritized approach such as the Stakeholder Specific Vulnerability Categorization system can be used to prioritize installing the updates."

And then they continue: "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. Organizations can refer to the NCSC's new guidance on 'Responding to active exploitation of vulnerabilities' for more information.

"To summarize, 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 recognize it may not apply in some instances, such as for safety-critical systems or operational technology.

"Patching alone won't address the systemic problems that," he writes, "my previous blogs have addressed. I've appealed to technology producers and vendors to ensure systemic technical security debt is minimized 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 organizations operating essential services such as energy, healthcare, transport, digital infrastructure and government."

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

One thing that occurred to me as I was going through this is that, in the name of preparedness, and this certainly applies to everybody in the UK and out, this notion of gaining assurance from your supply chains, I would say make sure that the providers of the equipment that you have on the edge, which are under support, which can obtain updates, make sure they've got like a greased path into your email. Make sure that when they do notify you of updates that are available, it doesn't get routed to some "we'll get around to it, you know, next month during our monthly review process." I would say, you know, given what we expect to have happening here over the next couple months, make sure that the communications inbound from the vendors that you are depending upon to have the most recent code running can get to you.

And largely, what I just shared from the NCSC, you know, it's a restating of what we already know; right? At the same time, for many of the CIOs and CSOs and IT heads in organizations throughout the UK, where this has reign, a clear statement and posting

such as this can provide the cover and the backup they may need to succeed in getting their organizations, you know, and the other C-suite executives to take this seriously, to understand what is probably going to be happening shortly.

And 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. And that struck me as odd since the CISA we've all come to know would normally have been shouting about this from the mountaintops. So I went digging to see whether maybe I had missed that statement which, you know, 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. And it exactly addresses the question "Where's CISA?" The reporting was posted as a scoop titled "Scoop: CISA lacks access to Anthropic's Mythos." And Axios explained, writing: "The Cybersecurity and Infrastructure Security Agency (CISA) 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" - this is Axios bringing less informed readers up to speed. "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. Earlier this month, an Anthropic official told Axios the company had briefed CISA and the Commerce Department on Mythos's capabilities.

"The Commerce Department's Center for AI Standards and Innovation has reportedly been testing Mythos." So they have it. "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 both declined any comment for this reporting by Axios."

They wrote: "The Trump administration has spent the last year" - as we know - "reducing capacity at CISA, instead opting to give more policy influence to the White House's national cyber director and pushing some programs out to the state and local level." So trying to distribute this instead of having it as centralized as it has been under CISA. CISA's acting director, a guy named Nick Andersen, "told lawmakers last week that the agency's resources are 'more limited than I would like,'" he said. "Trump has proposed cutting as much as \$707 million more 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 within their own networks and software. 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. And as we know, those critical infrastructure organizations have very much depended upon CISA, but also on that blanket of 'hold harmless' so that they're free to disclose things they discover, which is still a little bit in limbo."

So I hadn't heard about this acting CISA director Nick Anderson, so I checked him out. And he appears to be eminently competent and qualified. He's a decorated U.S. Marine Corps veteran who served as CIO, 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, 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 this guy is, you know, he's certainly competent to be on top of CISA. I've got no complaints with Nick's background.

It appears what he needs is more resources and support, and that CISA's lack of access to Mythos is largely due to the, as we're now calling it, the War Department's unfortunate feud with Anthropic. Anthropic made clear in 2025, at the time that it signed its contract with the Pentagon, that it did not want its AI technology to be used for mass surveillance of people within the United States or for fully autonomous weapons systems. As we know then, subsequently, the Department of War demanded that Anthropic drop those restrictions, and Anthropic refused to do so.

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 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 suddenly to be a "supply chain risk." And this is all very unfortunate since CISA should absolutely have access to Anthropic's Mythos Preview.

Hopefully, the White House's national cyber director, this 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 Pete Hegseth declared, it is "woke AI," whatever that means in this context. For the time being it appears that CISA is silent for purely political reasons.

**Leo:** This is really - politics should not intrude into this at all, and unfortunately very much has. I mean, CISA is in the doghouse because of what happened in 2020 with Chris Krebs.

**Steve:** Right. Right.

**Leo:** And now the White House is saying they want approval of all future AI models, period. They're about to draft a proposal that AI models can't be released without government approval. This is exactly the wrong direction to take with this stuff.

**Steve:** Well, and I also did see that Anthropic wanted to do a second round, they wanted to expand their program by adding an additional 70, seven zero, organizations that would have access to Mythos Preview; and the White House said no, is like blocking their ability to incrementally roll this out. And incremental disclosure here is exactly what you want. You get the core 40. They have a month with it. And then you widen the circle again.

**Leo:** Right.

**Steve:** And let another, like, next tier have access to it.

**Leo:** Yeah. This is - it's a little infuriating because political motivation and what's the right thing to do from a security point of view don't necessarily coincide. And that's what you're seeing here. And it makes us all less safe, frankly.

**Steve:** Yeah. Okay. Break time, and then we're going to look at this newest Linux local privilege escalation and look at how AI is reshaping the bug bounty business.

**Leo:** Excellent. All right, Steve. On we go with Security Now!.

**Steve:** So the news late last week was that the discovery of another serious local privilege escalation, discovered in the Linux kernel, had been there for a long time. And yes, before you ask, it was found by an AI vulnerability discovery system operated by a security firm named Theori. They 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 has been posted to GitHub, which immediately elevates any normal user to root. And of course that's not something you want to leave unpatched. So this is important, and Linux distros, the ones that are for sure known - Debian, Ubuntu, and SUSE - have immediately issued patches for the problem, and overseers of many other distros have, as well.

Red Hat initially said it was going to defer the fix, but then later changed its guidance to indicate that it will be going along with other distros and will be patching promptly. The CVE has been rated as high severity at 7.8 out of 10. And of course it's only - only, I mean, still that's bad - 7.8, which is as bad as it gets for a local privilege escalation. But the attacker first needs to get into a non-root account, where they're able to then execute this script in order to obtain elevation. But on the other hand, anybody who has local access to a machine also is able to use this. So it's a complete breach of Linux security, you know, account security.

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." I thought that was interesting, and it was news to me. So I went hunting. Here's what I found about that.

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 of course, yes, AI is the underlying driver of all this.

Okay. But we'll back up a little bit. Recall that the Internet Bug Bounty is a crowd-funded vulnerability reward program that was started 14 years ago back in 2012, and it's operated through the HackerOne platform. Its purpose and intent 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 pretty 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 itself where the trouble was found to support its repair and remediation. So that helps to fund the remediation work and makes the program go. It's been widely seen as a success, having paid out more than \$1.5 million dollars since the program began.

But, almost predictably, AI has messed everything 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 their ability to fix them, you know, remediation capacity, in open source has substantially shifted." So the problem is being called "Triage Fatigue," and the trouble is not just the increased volume of reports. That would be bad. What's interesting is it's also not 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.

We all know Daniel Stenberg, the creator of curl. He expressed it this way. He said: "More convincing crap is worse than obvious crap. You can't dismiss it quickly. You have to investigate it, and you waste real time getting to the point where you can prove 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 who are responsible for security." Which is like, yikes, a consequence of AI.

So 31 years ago, turning the clock way back, 31 years ago in 1995 Netscape launched the first widely recognized paid bug bounty program, offering to pay researchers, back in 1995, for their responsible reporting of significant bugs which they discovered in Netscape Navigator 2.0. So they were really ahead of the game at that point. Of course they also had a web browser that was ahead of the game. And that model has been functioning vibrantly, the notion of paying researchers for responsibly reporting bugs they find, has been functioning ever since.

So the notion that AI may be driving a fundamental change to this longstanding vulnerability discovery and reporting model is important enough, as I said at the top of the show, to be a contender for today's main topic. Except that the idea of Google going off half-cocked and adding an explicit AI interface for JavaScript in Chrome, that also needed ample discussion space today. And we're going to cover Mozilla's pushback against that 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, who I just quoted, 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 back in 2012. Aikido titled their report "Bug bounty isn't dead, but the old model is breaking." I'm going to share what he wrote, and also what my intuition immediately suggests about the nature of the change.

So they 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 in the first place. 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 for that is wrong."

They said: "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, different perspectives, and different motivations - all attacking your system in ways your internal team never thought of. And that's without the significant overhead required to hire specialist experts internally and then work to keep them busy. All this explains why bug bounties became fundamental to modern security programs.

"What's changing now is not 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 become effectively 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. And Aikido is, of course, plugging their own solution, as we would expect them to on their own website. But we know that Anthropic also set up 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 concepts of exploits."

Aikido continues, enumerating these three classes of bug sources. So they said: "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, and this is where we quoted him earlier, saying: 'More convincing crap is worse than obvious crap.'" They said: "You can't dismiss it quickly. You must investigate it" - right, because it looks real - "and then 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," they write, "has been truly devastating. The Internet Bug Bounty program paused all new submissions because AI has dramatically increased discovery volume beyond what their 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," they write, "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 the 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 web." Of course we all think of that xkcd cartoon, right, with the little tiny block that's holding up this whole creaky infrastructure. They said: "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 capability.' 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 programs 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 organization 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.'

Okay, so I'm going to interrupt here to mention that I have a theory about why that is. Back when discovering vulnerabilities required long hours of painstaking grueling work to step through and reverse-engineer code, it was no 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. I think it makes sense.

Aikido continues with their report, writing: "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, to validate, and to repair. And many of these 'good' findings still fall into gray areas, bugs that may not meet security thresholds, but still require some attention.

The result is a sustained, and in some ways increased, load on already-constrained teams. So in a strange way, the system has not 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, of 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 yet. 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 incentivized. Public bounty programs don't go away, but they become more controlled, more targeted, and more aligned with organizational maturity. AI will inevitably play a larger role in filtering and triaging the growing incoming 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 the 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, taking 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 predominantly 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, in 2026 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 ever since Netscape first offered payment for reports of bugs in Navigator 2.0 is probably going to wind down over time. And the reason our programs are currently overwhelmed by good bug reports is that unfortunately they are very buggy. It's going to take a while. I mean, this is that new phase where AI is finding problems that were not - is truly finding problems that were not known to exist. Those will wash out of the system over the next six months or so.

And then the volume of really good reports will necessarily drop because there won't be nearly as many bugs to be found, you know, in real time. And as AI then continues to check code before it goes out the door, we're not going to have new bugs introduced into the ecosystem. I think it's really interesting that potentially we are talking about a major shift in the way, you know, bugs are discovered. It won't nearly be as much for money moving forward as it has been in the past, Leo.

**Leo:** Okay. You want to take a little break?

**Steve:** I do.

**Leo:** All right.

**Steve:** And then we're going to look at a new product from Anthropic, which we might call Mini Mythos.

**Leo:** Mythos Lite.

**Steve:** Or Mythos Lite or Mythos Junior or something. And it's available to all Claude Enterprise users now.

**Leo:** Ah, okay. Oh, cool. You're watching Security Now!, Mr. Steve Gibson. We do this show every Tuesday, right after MacBreak Weekly. That's about 1:30 Pacific, 4:30 Eastern, 20:30 UTC. And you can watch it live, if you really want the freshest version of it. Our Club members get to watch in our Club TWiT Discord. But there's also of course X.com, Facebook, LinkedIn, Twitch, YouTube, and Kick. So pick your platform, watch us live, or get it after the fact on Steve's site, GRC.com, or on our site, TWiT.tv/sn. I thought that was really interesting that the first bug bounty was 31 years ago. That's remarkable. That is really amazing. Yeah.

**Steve:** Yeah. It's a program that has worked. But to me it really makes sense if we have, I mean, finding bugs and contributing, you know, giving back, we know that there is a lot of altruism out there in the world.

**Leo:** Absolutely, yeah.

**Steve:** You know, people who would like to contribute. You know. And so spending some time working with a security and AI enhanced vulnerability finding system, I think that makes just total sense.

**Leo:** Well, that's one thing I don't think Netscape could have anticipated 31 years ago, that AI would suddenly be finding all these flaws.

**Steve:** Well, and for the intervening 30 years, it's been fabulously successful.

**Leo:** It's worked really well.

**Steve:** You know, millions and millions of dollars have been paid out to, you know, authentic bugs and vulnerabilities that have been found. So the systems that are working now, we have AI able to pick up that burden and carry it forward.

**Leo:** There's another category of people who are out of work. Bug bounty hunters.

**Steve:** Well, it's true. It's probably not a career path. Although if you are expert in running AI discovery...

**Leo:** There you go.

**Steve:** ...then you've got a new way to make some money.

**Leo:** Well, actually, that's a good point. That Linux copy fail flaw, they found it, not with the AI solely, but because a very smart security researcher pointed the AI at a specific direction and said, hey, I wonder if this is a problem, and then the AI was able to go a step further. So it was really a partnership.

**Steve:** Exactly.

**Leo:** Yeah.

**Steve:** Okay. So we can add, well, it's apropos of the changes being wrought to AI vulnerability discovery that we have Anthropic's announcement late last week of Claude Security, which is now entering public beta for their enterprise customers. We could think of it as Mythos Junior, and that's sort of how they're casting it. Here's what Anthropic posted about this.

They said: "Claude Security" - which is what they're calling it. "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 into 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." Just sounds fantastic.

"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."

Okay. Given the wind-up we've seen from Mythos over the past month, and the way they describe this, I cannot imagine why any organization whose software might contain external 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, you know, watching their reactions to what they begin seeing from this Claude Security. Basically this is anybody's now able to purchase a mini version of Mythos.

And I would argue that, if Mythos is even better at finding bugs, there's still benefit from running this Mythos Junior, you know, Claude Security, over your codebase to see if it's able to find something. Certainly, if it can, Mythos would. Mythos may be available in the future, well, we presume it will be at some point. But you have this now. So I think this is, maybe in retrospect, a predictable evolution on Anthropic's part. But certainly welcome.

**Leo:** I do this anyway. I mean, I don't have Mythos or anything like it. I just have the regular, you know, Claude Opus 4.7 and ChatGPT 5.5. And I always say, in fact I often have ChatGPT check Claude's work, and Claude check ChatGPT's work.

**Steve:** Yup, cross [crosstalk].

**Leo:** And I frequently say let's do a security audit on these repositories. I mean, that by itself is useful. I've found all sorts of stuff. I've also had security audits on my systems, and it's found errors and corrections there, too. It's, you know, just the regular models are useful. I can't wait to see how Mythos does, yeah.

**Steve:** Right. Yeah, and I think from what they said, what this adds is the ability, for example, to schedule scans so that your engineering software development team, they're just working along. And then periodically the codebase is given a scan and a check to see if anything significant has been found.

**Leo:** That's a great idea. I think that's prudent.

**Steve:** Okay. So OpenAI announced that they've decided - I was very impressed by this, I'll just say ahead of time - to make account login security a selling point. Their posting was titled "Introducing Advanced Account Security," and they explain: "Today we're introducing Advanced Account Security, a new opt-in setting for ChatGPT accounts" - and you've got it now, Leo - "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."

They wrote: "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" - now here's a critical part - "understand that the increased protection of Advanced Account Security comes with an increased responsibility for account recovery." And so now they get specific.

"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 the web. Protection applies to both ChatGPT and Codex accounts that are accessed through that login."

So we have: "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." So password-based login, gone. You must use a passkey or physical security key.

Next: "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 email or SMS based recovery." We know that; right? They say: "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." Again, with truly heightened security comes much more responsibility. They're saying we can't help you because you don't want bad guys posing as you to get help from us, either.

So 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 to cherish. By explicitly removing email and SMS account recovery loops, the most common phishing and other attacks will be thwarted. But I can see, in the case of ChatGPT login, this makes sense.

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." And finally: "Automatic training exclusion. People working with especially sensitive information may opt to not 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."

They finish, saying: "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" - you know, you stick it into a USB port, and its head just basically tucks out a little bit so you're able to touch the little gold metal convex head of it in order to authenticate - "for low-friction daily authentication, and the YubiKey C NFC for backup, and use across laptops and mobile devices.

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

So I logged into ChatGPT, which I'm no longer using as my daily driver. 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 I was there, in ChatGPT, sure enough, the Security panel of the Settings dialog now has many new features. And I think this is great. I expect to see this sort of enhanced security become a standard feature to more rigorously - I mean, like, across the industry - to more rigorously protect the potentially very highly sensitive dialogs that many people are having with their AI chatbots. You know, 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, I think, very clear.

Okay. Before we get into our main topic, I did want to update everybody on something that I just discovered about Syncthing and SyncTrayzor. Of course we've spoken often about Syncthing, you know, both Leo and I and many of our listeners I know are huge fans. SyncTrayzor, S-Y-N-C-T-R-A-Y-Z-O-R, is a terrific little Windows GUI wrapper that turns Syncthing into more of a Windows app. In the words of its creator, he said: "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 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 Dropbox-style file download and progress window. The tray icon indicates when synchronization is occurring. Alerts you when you have file conflicts, one of your folders is out of sync, folders have finished syncing, and devices connect and disconnect. Has a tool to help you resolve file conflicts. Can pause devices on metered networks, to stop Syncthing transferring data, for example, on a mobile connection or a WiFi hotspot. And contains translations for many languages."

Anyway, I've been using both of these for years, SyncTrayzor which contains Syncthing, 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 there for many years, ever since my first Drobo died, and I got a Synology to replace it. And at that point, as I said, I switched to Synology. Syncthing works perfectly there, as well.

I'm mentioning all of this since Syncthing on Windows 10 has been noting that v2.0.16 has been available for some time. Since I heard from several of our listeners that the major v2 of Syncthing is fully backward compatible with v1.3, which was where v1 left off, and which is where I'm still stuck on my Windows 7 machine because it won't run anything after v1.3, I decided it was time to quiet down that "new version available" notice. But when I updated Syncthing, it complained about an unknown command line switch. Meaning that the way Syncthing was being launched by SyncTrayzor it wasn't familiar with.

The trouble, of course, was that the version of SyncTrayzor I had was also out of date. So I updated it. 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."

So I first verified that the Syncthing project does indeed still recommend the use of this forked SyncTrayzor v2, and indeed they do. It is recommended among their contributed software. So I wanted to let everyone who's been happily using Syncthing know that indeed major version cross-compatibility works. I've got Syncthing 2 now in one location on Windows 10, and Syncthing 1.3 still on Windows 7 until I shut down this workstation and consolidate my locations, which I'll be doing over the next couple months. I ran the new installer for SyncTrayzor 2. It saw that there was an older version available, offered to upgrade. It did that. Everything went smoothly, and everything is working now perfectly. So I just wanted to make a note for those people who may be Windows users using Syncthing. If you don't know about SyncTrayzor, it's a neat little wrapper. And if you do, you are able to update, and everything works great.

**Leo:** You don't have to use it, though. I mean, you can use Syncthing...

**Steve:** No.

**Leo:** ...directly, of course.

**Steve:** Absolutely. You're able to use Syncthing and set it up as a Windows service, and it just works by itself.

**Leo:** Let's take a timeout before we get to the subject, the Browser AI API. You know, there was a story, I don't know if this is related, that came out earlier today that Chrome is automatically downloading a pretty hefty AI model. And you can't stop it.

**Steve:** Not so nano.

**Leo:** Not so nano. I don't know if that's related. Maybe it is.

**Steve:** It was initially 22GB. I think they got it down to 4.7. I know. I know. I know.

**Leo:** Okay. You kind of have to have Chrome, unfortunately. For instance, I'm using Chrome right now because the Restream, which is what we use for the show, works best with Chrome. I'm a Firefox user, but I have to have a copy of Chrome, and I have to have a 4GB copy of Nano along with it. Oh well. Actually, back to our discussion of Chrome and AI on Security Now!. Steve?

**Steve:** Yeah. So it turns out, and this is actually exactly on point for this Nano LLM, Google is planning to define a new API to bring AI into our browsers. This would serve as an interface to large language models existing outside the browser, or brought in by 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, just to make this clear, 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, proofreading 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 API large language model prompting interface.

Now, 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 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 forced AI." Jon cited as a "no thanks" example Microsoft's Recall compiling a long-term history of everyone's desktop screenshots every five seconds. Giving Recall the label of AI now seems sort of quaint in today's world. 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 as a feature competition.

So the thing that put this 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: "Chrome looks set to ship an LLM Prompt API to the web platform. At Mozilla, we oppose this API. We feel it has a large interoperability risk, and Google imposing terms and conditions on a web API sets a dangerous precedent."

Okay. Now, Leo, listen to this. Before I go any further, I want to touch on those terms and conditions, since that alone is a 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 four-letter words myself. This is SO WRONG.

**Leo:** Yeah. Now, this is the system prompt, though, for the AI. Right?

**Steve:** No. There is a system prompt for the AI, which is part of the API.

**Leo:** Yeah.

**Steve:** But so this is saying that the use of the prompt API by JavaScript running in the browser must acknowledge this acceptable uses policy.

**Leo:** Oh, that's weird. Because it sounds more like the kind of thing you tell an AI not to do, and that's kind of understandable. This is...

**Steve:** No, no. This is for developers.

**Leo:** These are rules. This is rules of the road.

**Steve:** This is rules for, yes, for developers. So I just thank god we have respected developers at Mozilla to push back. And I hope this also captures the attention of the EFF because this seems wrong.

Okay. So to obtain some pro and con balance here, let's first look more closely at what this new so-called "Prompt API" - that's the name that they've giving it, the Prompt API - that Google has already implemented and moved into Chrome, it's already in Chrome, which is why, Leo, you noted that this multi-gig download is happening because they're also downloading a model, their Nano, the so-called "Nano" model.

The explainer for this nascent feature says - and so this is now Google speaking: "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 they're getting some - "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," they write, in order to set the context here. "Browsers and operating systems are increasingly expected to gain access to language models." Okay, I didn't know that, but okay. "Language models are known for their versatility. With enough creative prompting, they can help accomplish tasks as diverse as" - we have some bullet points - "classification, tagging, and keyword extraction of arbitrary text; helping users compose text, such as blog posts, reviews, or biographies; summarizing, for example, 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; and proofreading." In other words, all of the things, I mean, like AI in your browser things that Vivaldi said, eh, don't know if we want to jump into that just yet.

They said: "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, for example, 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; and allowing hybrid approaches, such as free users of a website to use on-device AI, whereas paid users use a more powerful API-based model."

Okay, now, I'll just interrupt here to note that those seemingly, I don't know, to me they feel like made-up reasons, you know, "local processing of sensitive data, for example, 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 TLS privacy and security bulletproof. Then there's "potentially faster results, since there is no server round-trip involved."

Okay. 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 today. Everything I'm seeing says that the cloud blows away local LLM. And so on for the remaining three benefits. You know, our browsers already do have the ability to query cloud-based LLMs using the tried-and-true XMLHttpRequest API, which has been around forever, or the more recent Fetch API. And both of those offer state-of-the-art mature security and 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 - notably Edge from Microsoft - to access non-cloud-based LLMs since everyone can already do that. That is, can already access cloud-based LLMs. Their explainer continues, writing: "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 their dependencies."

Okay. I'm not sure that that makes sense to me. Again, 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. They're just using a generic LLM that the user has provided to their browser. Today that's already not the case, I mean, it's already not the case that all LLMs are identical and interchangeable. And I expect model design and capability to diverge more as we move into the future, rather than converge. Of course, we'll see how that goes.

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, for example, for downloads, updates, templating, and parsing; guide web developers to gracefully handle failure cases, for example, no browser-provided model being available." I guess by always having one. "Develop formal implementations guidelines and definitions, for example, initial on-device models, and possible cloud services.

"The following are explicit non-goals," they said. "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. In other words, that's acceptable. 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.

"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 by 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."

And finally they said: "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 necessarily 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.

Essentially, what this comes down to when you strip it away is Google - and you started with this, Leo. Google wants to add a 4GB, actually it's 4.7 is the number I saw, down from 22, which it was earlier, a massive language model to Chrome so that Chrome will become AI-enabled intrinsically, and that would allow Chrome-hosted web pages to do lots of things they can't now. So, okay.

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 random websites scattered around the world still use them. As one example, it may not be fair to single out Flash, since it did have its day. There was a time when you could only do things with Flash that you wished you could do on the browser, but JavaScript and scripting in general had not caught up. But, boy, was Flash difficult to kill off. And in some places even today it won't die.

As I look over the Prompt API implementation section I can empathize with Mozilla's gut reaction since this does seem sort of, well, both obvious, but also forced and a bit 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 the create function, or as the first message to the first prompt or append method calls."

We then see three examples of these various semantic options that they just described. The first one shows where a constant variable session1 is set to the Large Language Model.create functions output, where the initial prompt's system prompt is "pretend to be an eloquent hamster." And then we have the - we log to the console the output of that large language model we've just created being prompted "what's your favorite food?" So of course an eloquent hamster is going to respond to the question "what's your favorite food?" I guess that's, what, lettuce?

**Leo:** I don't know. I don't know.

**Steve:** I think that's what hamsters like. Anyway...

**Leo:** This is an eloquent hamster, which is a different matter entirely.

**Steve:** That's right. Might be lettuce with caviar.

**Leo:** Yeah.

**Steve:** Anyway, my reaction to all of this is that web standards are too important to be created in any half-baked fashion, and Mozilla apparently feels that it's too soon to do this. Once a web standard exists, as we know, we've seen this over and over, it is incredibly difficult to deprecate 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.

**Leo:** Google has never held back, though, right, in unilaterally declaring web standards.

**Steve:** Yes.

**Leo:** They say, well, you know, we're the dominant browser; we can do whatever we want. And I understand Mozilla's reluctance to go along for the ride. And I think people are not going to be happy about 4.7GB being downloaded to their hard drive.

**Steve:** It's really going to change the whole complexion of Chrome.

**Leo:** Yeah. And so it becomes massive. I can understand why Google may say, oh, well, maybe for spell checking or local grammar or something. You know, developers might find a use for this. But it is a little, I think, I think Mozilla's right, this is

premature. There's no reason to be doing this now. There's no demand for this now, I don't think; is there?

**Steve:** No. No. And, I mean, I guess they recognize that you can do this in the cloud now. Browser pages are able to reach back out to the cloud...

**Leo:** Sure.

**Steve:** ...and talk to a large language model. That's going on already. They're saying, well, but we want, you know, we've got this cool technology. We've managed to squeeze a large language model down to 4.7GB. We want it in the browser because we can. Because we own the browser.

**Leo:** Right. Right. And we might imagine down the road some use.

**Steve:** Yes.

**Leo:** It's hard for me to imagine what that use is, but...

**Steve:** Yeah, I agree, that would justify this. So 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 as an industry-wide specification for what could just must be the moment we're in today. I mean, to me, that's the problem is that none of this has gelled yet. I mean, it is still a moving target. So the idea of API-ing it to create a web standard seems premature and misguided. Anyway, I've dropped the URL of Google's full specification into the show notes. It's at the top of page 20 for anyone to follow-up who may be interested.

I want to now switch to Mozilla's response. I have the rather dry conversation thread in Mozilla's GitHub account. It's under their "standards positions," so I've dropped that URL into the notes also. But since this podcast endeavors not only to inform but also to entertain our listeners, rather than sharing Mozilla's dry recitation, ah, yes, I want to share The Register's typically feisty and irreverent take on this controversy. Leo, let's take our final break.

**Leo:** Oh, good, okay.

**Steve:** And we're going to look at the flipside of what's going on.

**Leo:** I can only imagine what The Reg has to say about this. I'm trying to give Chrome the benefit of the doubt, but this is the problem, this is my problem, been my problem with Google for a while now, they don't go to the IETF or W3C and say, here, we want to do a standard, you know, let's get everybody involved.

**Steve:** It's already in there.

---

**Leo:** Yeah. They're so big. They're so dominant. They're something like 90% of the browser space.

**Steve:** Yeah.

**Leo:** That they could just do it, and it becomes a de facto standard. So I'm with you. I'm not necessarily against the idea. And it sounds like in their spec they're saying, well, it doesn't have to be our model. Doesn't have to be Gemma. It could be something else. But I don't know if there's a demand for this. And I know people are going to be very upset. I already see the upset over this giant download. And you don't get a choice. You can't turn it off.

**Steve:** Right.

**Leo:** It comes with Chrome now. And back to our conversation, actually, Darren Oakey, who is, of course, as you know, one of our most avid AI users in the Club TWiT Discord, says he thinks this may be the most important thing to happen to browsers since AI. He thinks it's really important. I'm not sure I agree. I mean, I can see there's some potential.

**Steve:** It's a huge change to our browser.

**Leo:** It is a big change. I guess that's - nobody disagrees about that. And I think it's also the case that Google is forcing this instead of proposing it. And I don't like that either.

**Steve:** Right.

**Leo:** But I didn't like it when they forced HTTPS down our throats.

**Steve:** Well, and you don't always get the right design when one person does it. That's why, you know, so much of what is done correctly is a collaboration. And, you know, Mozilla has, even though Firefox is a diminishing percentage of the desktop space, Mozilla as a company has been at the forefront of all of the standards work forever.

**Leo:** Yeah. Yeah. We talked to the CEO of the Mozilla Foundation a few weeks ago on Intelligent Machines. And even then at the time, and this is before they'd added that little switch, he said, "We're going to be very judicious about AI in our browsers." And now they actually have a switch that says "Disable all AI features."

**Steve:** Right.

**Leo:** This is a switch most notably Google is not offering. You cannot disable this.

**Steve:** Okay. So I'm going 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 organization'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 Register writes: "The Prompt API, as Google describes it, 'gives web pages the ability to directly prompt a browser-provided language model.' Specifically" - and here it comes - "it provides a way to send natural language instructions to Google's Gemini Nano model, which is small enough to be downloaded" - well, okay - "for local inference through Chrome. However," writes The Register, "it's not small. Google recommends having 22GB of space available, although the Nano (v3 Nano) 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, for example, providing a free AI fallback if users lack a paid AI API key."

Okay, now, 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 the 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."

I'll just pause to say, who determines what content is "disturbing"? There is nothing that 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 their usage." Amen to that.

Anyway, 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 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 the law, you're not really building for an open platform anymore.'"

And just to pause, what he means is remember those days where JavaScript had to determine which browser it was in, and then would do this code for IE, that code for Firefox, this code for Safari, and that code for Chrome? Those were not good days.

Anyway, The Register says: "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 polarized, 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."

To his credit, 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 has not done much for Google's cause. According to a report created in February that compares the performance of Chrome with Gemini Nano and Edge with 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, et cetera), 24.29% Edge's and 15.17% of Chrome's responses failed to complete the task at all.' 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.' So it's often also wrong."

They finish with the report's conclusions, noting: "In terms of groundedness and accuracy, Edge failed" - which is to say '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." And that's how The Register signs off.

**Leo:** Burn. Burn.

**Steve:** Okay. So where does this leave us? I guess 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. Instead of what I ask for, I get sponsored interception advertisements that are promoted to the top of the page and are presented before the result that I'm seeking. Then I need to wade down past a bunch of YouTube video links that I have zero interest in.

Okay, now, in fairness, Google's not alone in doing this. Apple has similarly succumbed in their App Store. The thing I'm looking for is never first any longer, 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 it was that I wanted. And on the Google side, in return for tolerating a bunch of advertising, we do 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 over at Gmail is similarly valuable. All of that means a lot. So thank you, Google.

But all of that seems fundamentally different to me 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 any 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. Well, Mozilla and Apple. Either keep up or get left behind and become irrelevant. But everyone will likely get a better browser, whether that's Chrome, Edge, Safari, Brave, Vivaldi, or Firefox, if this is a collaborative effort.

And Leo, the thing that I think was most significant here is the observation that Archibald made that LLMs are inherently non-deterministic. You know, every time you ask a question, you get a different answer. And so we're putting - we're now talking about having the browser interface to one vendor's solution, which has a random number generator in its heart.

**Leo:** Not a very good one, either.

**Steve:** No. It's got some temperature setting.

**Leo:** Right.

**Steve:** And apparently they had to sacrifice a lot of reliability in order to get the size down to something...

**Leo:** It's heavily quantized, yeah, yeah.

**Steve:** ...that it was tolerable. They wanted it to be 22GB, and people said eff off. I am not putting that in my, you know, is that mass storage? Is that RAM? Where does that 22GB live?

**Leo:** Somewhere you don't want it to live, probably.

**Steve:** And so they've had to squeeze it down in order to make it acceptable. And in the process, it's lost its reliability. So, I mean, really, if we want to be able to surf the web with any browser we choose, and if web pages that we download are going to start wanting to use local large language models, whose large language model will it be?

**Leo:** Right.

**Steve:** And they aren't interchangeable. We know they're not interchangeable.

**Leo:** Right. Right. Well, and, you know, Darren, who loves AI, said, well, I can imagine some uses. For instance, it's hard to write software that detects misspellings, but the AI could quickly detect a misspelling if somebody's entering it in and correct it. So there is that convenience. But I also think that this is Google big-footing the whole process. And it's part of the enshittification of Google. They don't feel any responsibility to anything at this point except their stakeholders to make more money. And that's clearly, you know, this is about dominating the browser space and putting everybody else out of business.

The other thing that worries me as an AI fan, and I know you're an AI fan, too. The more we force AI down the throats of unwilling users, the more they're going to hate it. Google's found that out. Microsoft's found that out in spades.

**Steve:** All those annoying chatbots that pop up...

**Leo:** Exactly.

**Steve:** ...in the lower right corner of your screen, well, that's going to end up running locally. And it's like...

**Leo:** So I don't want to turn people against AI. AI is a real value. But by doing this kind of forcing it down people's throats, you're actually making enemies. And I don't think that's good, either, for Google or for AI in general. So, yeah, I have lots of problems. We'll talk about this. I'm looking forward to a conversation tomorrow [crosstalk].

**Steve:** One thought would be unbundling the LLM from the browser. That is, creating an interface, but not having it, like, secret. I mean, it's essentially...

**Leo:** Right.

**Steve:** ...secret right now. I mean, I get it. That's the way to minimize friction so that everybody has it because Google wants everybody to have it. But...

**Leo:** It's not a very well-kept secret. I should point out, one of the reasons Google thinks this is okay is because they're already doing this on Android, as is Apple doing it on iOS. There are built-in local models on both those systems. Apple touts this all the time. And your data stays local on the device. Apple Intelligence is a local model. So there is a precedent for this on those platforms. I think I still wish, maybe it's a futile wish, that the web would be a standards-based interface, and that everybody should be able to choose the browser of their choice. And it should all - they should all work well.

**Steve:** The only one who doesn't want it to be a standard is the big guy.

**Leo:** Is Google, the winner. Yeah, you're not going to see Vivaldi saying, well, we think a standard should favor us. They can't. Nor can Mozilla. But Google can. And clearly they do. I agree with you. I think this is a - you know, we saw Google back way down on a number of its proposals.

**Steve:** True. The whole anti-tracking technology. They tried several times, but they got real pushback. But they got pushback from people who had a vested, I mean, like...

**Leo:** Advertisers.

**Steve:** ...huge, yes, exactly, large commercial interests. And there's no one to push back on this.

**Leo:** Well, just remember, as users, maybe as individuals we don't have much power. But collectively we do. They still need us to use their darn browser.

**Steve:** But would somebody leave Chrome to go to Firefox? I don't think so.

**Leo:** Well, you and I have.

**Steve:** Yes.

**Leo:** You and I have. And this is one of the reasons. And fortunately, so far, you can mostly use the Internet with a Firefox-based browser. Mostly. WiFi is another example. DRM in the browser.

**Steve:** Yeah, the Open Table site I use for restaurant reservations, and it doesn't work under Firefox.

**Leo:** Needs Chrome. As I said, Restream needs Chrome because of WebRTC and the WebRTC implementation it uses. I think that's, you know, this is an object lesson. This is what happens. And if you don't want - if you want Chrome and Google to be the only player in the Internet, this is how these things happen. I think we can fight. Got to fight.

**Steve:** Yeah.

**Leo:** Hey, great topic, great show, as usual.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details:

<http://creativecommons.org/licenses/by-nc-sa/2.5/>