

# Security Now! #1080 - 05-26-26

## Vulnerability Debt Repayment

### This week on Security Now!

- Cisco meets Mythos.
- Can the aging CVE system survive AI.
- Patch deployment latency in the AI age.
- MSFT's official YellowKey Bitlocker bypass mitigation.
- Ubiquity patches 5 serious vulnerabilities.
- Drupal being attacked by PostgreSQL injection.
- Microsoft terminates SMS as a second factor.
- Github hacked - all of its source exfiltrated.
- Russia using very old Western software.
- Why to get a no-charge AI chatbot account.
- New Sci-Fi on Netflix.
- What we learn from Mozilla's use of Mythos.

**Because the previous "Do Not Trespass" sign was often ignored and ineffective**



# Security News

## Cisco meets Mythos

I titled this first short note: "Cisco meets Mythos" in honor of that fabulous and classic cartoon short called "Bambi meets Godzilla." Cisco recently produced a slick 8-page PDF document titled "*Shields up: Guidance for defending in the age of AI-enabled attacks*". I'm only going to share the introductory Executive Summary from this piece, but I want to start with it because it nicely serves to introduce us to what Cisco themselves have realized is about to happen to the industry. The Executive Summary says:

*In early April of 2026, Anthropic announced that it would be holding back on releasing their new AI model, Mythos. Due to deep concerns around the offensive cyber capability of that model, Anthropic decided to work with select companies, including Cisco, so that those companies could use the model to find and patch security vulnerabilities.*

*Cisco is changing our near-future threat modeling of AI-enabled attackers in view of our experience with Mythos. That, in turn, has changed how we defend ourselves and led us to develop a set of defensive recommendations for customers. While the capabilities of Mythos may not be widely available, we do anticipate that this capability, and more, will become widespread as AI technology advances across the board.*

*This paper lays out what Cisco has seen so far from AI-enabled capabilities and what we believe the new threat landscape will look like. Whether these models are wielded by attackers, leveraged by researchers, or operating as agents within your own environment, the security implications are significant. Subject to appropriate safeguards and controls, we will share what we have implemented based on this new understanding and lay out our recommendations for customers.*

*The threat surface is going to change — in some ways, dramatically. Defenders must take the time to understand what the new normal will look like and evaluate what changes their environment must make to stay secure. Cisco is committed to being a partner through that transformation.*

I have a link in the show notes to Cisco's full report, though I didn't find anything there that was very interesting or worth sharing. It appears to be far removed from Cisco's trenches where Cisco developers appear to have run out of expletives to express their degree of astonishment and concern.

[https://www.cisco.com/c/dam/en\\_us/about/doing\\_business/trust-center/docs/cisco-defending-against-ai-attacks-guidance.pdf](https://www.cisco.com/c/dam/en_us/about/doing_business/trust-center/docs/cisco-defending-against-ai-attacks-guidance.pdf)

Speaking of which, Cisco's security blog poses some interesting questions about the future practicality of our existing time-tested CVE – Common Vulnerabilities and Exposures – system in light of AI vulnerability discovery. Though my own expectation differs from Cisco's, I want to share Cisco's thoughts since they're certainly worth understanding. Under the title "*When AI Finds Faster Than Humans Can Patch: Disclosure Must Evolve*" they write:

*Project Glasswing is an amazing initiative by Anthropic. Cisco is one of the main participants and I have been honored to work on it since it started. **This is NOT hype!** Claude Mythos has*

*discovered thousands of zero-day vulnerabilities across every major operating system and web browser. The CVE program, already buckling under 50,000 entries a year, was never designed for this. We need to talk about what comes next, before the flood arrives.*

*As you probably have heard from many sources, Claude Mythos Preview found a 27-year-old remote crash vulnerability in OpenBSD. It found a 16-year-old bug in FFmpeg that survived five million automated fuzzing runs. It chained together Linux kernel vulnerabilities to escalate from unprivileged user to full system control with no human guidance. And it did this in days, not decades.*

*Key technology providers are finding and fixing hundreds of vulnerabilities. Participant organizations are also finding thousands of open source vulnerabilities and working on a coordinated disclosure timeline. That is the responsible move. But it also forces an uncomfortable question: What happens when this class of model becomes commonplace; when the vulnerability discovery rate jumps from thousands per month to hundreds of thousands?*

The title of today's podcast "Vulnerability Debt Repayment" looks at this and my conclusion is not the same as this author's. But I thought this was interesting because this guy is on the front line of "*Cisco meets Mythos*" and he was obviously deeply affected. Since he's an employee of Cisco and presumably wants to keep his job, he's not talking in detail about what they found. But it doesn't take much imagination given what we know of Cisco's rickety past with security. He continues:

*The answer is that every piece of infrastructure between discovery and remediation (disclosure norms, the CVE system, patching pipelines, and the human workflows that connect them) will need to be re-architected. Not updated. We must re-think how to scale.*

*[The problem is that] the CVE system was built for a different era. The Common Vulnerabilities and Exposures program turns 27 this year. It was designed when the security community measured vulnerability disclosures in the hundreds per year. For example, 321 CVEs were issued throughout all of 1999. By 2023, the number had climbed to nearly 29,000. And a 2026 forecast projects a median of roughly 59,000 CVEs this year, with realistic upper-bound scenarios reaching 100,000. And that forecast was published before Project Glasswing was announced.*

*At RSAC 2026, CVE board members acknowledged the program needs an overhaul. GitHub reported a 224% increase in vulnerability reports over the last three months alone. Again, that's the current volume; before AI-scale discovery tools are going wide.*

*When a single AI model can surface thousands of high-severity vulnerabilities in a few weeks of scanning, assigning an individual CVE to each one, enriching it with CVSS scores, routing it through the NVD – the National Vulnerability Database – and waiting for human analysts to triage it, becomes illogical. Not wrong in principle, but extremely challenging in practice. The bottleneck is no longer discovery. It's everything that comes after.*

*The CVE system assumes a world where vulnerabilities are found one at a time by human researchers, disclosed individually, and patched on human timelines. AI-scale discovery breaks every one of those assumptions simultaneously.*

*Individual CVEs May No Longer Be the Right Unit — Here is the heretical idea that needs to be said: when a model like Mythos scans a codebase and produces 300 findings in a single pass (buffer overflows in the same parsing library, NULL pointer dereferences across related system calls, use-after-free conditions in the same memory management subsystem) the one-CVE-per-bug model doesn't serve defenders. It buries them*

Now, we'll notice that this guy never said that's what happened when "Cisco met Mythos" but if it **had** happened, who among us would be surprised? He continues:

*What defenders actually need is a vulnerability summary: a grouped, contextualized disclosure that says "this codebase has a class of memory safety issues concentrated in these five modules, with these representative examples, and this aggregate severity profile." In other words, a Vulnerability Class Report (VCR) rather than 300 individual tickets.*

It sure does sound like this may be exactly what happened to him. And if it had, it's understandable, right? He's down in the trenches following the original 27 year old CVE model which requires that each and every discovered vulnerability be catalogued, described, evaluated, listed and resolved. It's not a big deal when you field one or two. But when 300 drop on you during the first pass scan of just one piece of one's massive codebase? I do see this guy's point. The existing piecemeal approach becomes immediately impractical. He writes:

*This isn't about hiding information. Every individual finding should still be documented and available to the maintainer, the CNA, and the downstream consumers who need it. But the disclosure unit (the thing that gets published, tracked, and acted on by defenders) should be the summary, not the avalanche.*

*Think of it as the difference between receiving 400 individual package tracking notifications and receiving a single shipment manifest. The manifest tells you what's coming, how critical it is, and what you need to do. The individual tracking numbers still exist; you just don't need to process them all at once to take action.*

*Here's the stat that should keep every CISO awake: the median enterprise patch deployment time is approximately 20 days. But in March 2026, researchers observed active exploitation of a critical Langflow vulnerability within 20 hours of its advisory; with no public proof-of-concept code available. Attackers built working exploits directly from the advisory description. Twenty days to patch. Twenty hours to exploit. That gap is already fatal. AI-scale discovery makes it catastrophic.*

*The uncomfortable truth is that human-driven patch cycles cannot keep pace with AI-driven discovery. If models like Mythos are finding vulnerabilities at machine speed, and adversaries with similar models will exploit them at machine speed, then remediation has to move at machine speed too. This means we need to use AI to scale, fix and patch, but code changes must be carefully reviewed and tested. This must become a standard part of our lives.*

*It also means organizations need autonomous patch deployment pipelines for the most critical categories. Not "we'll schedule it for the next maintenance window." Not "it's in the backlog." Automated testing, staged rollout, and rollback capabilities that can absorb a continuous stream of patches without human bottlenecks at every stage. The pieces are assembling. What's missing is the operational framework to use them at scale.*

Again, I believe that by the end of this podcast I'm going to be able to make a strong case for what this guy is missing, but I still think that what he's saying bears hearing. He continues:

### ***The CVE Program Must Adapt***

*Let's be direct: the CVE program, in its current form, cannot handle what's coming. Not because the concept is flawed. A universal identifier for vulnerabilities is valuable. But because the implementation assumes human-speed discovery, human-speed enrichment, and human-speed consumption. All three assumptions are collapsing.*

*The National Vulnerability Database enrichment backlog is already a known problem. CNAs (CVE Numbering Authorities) are already overwhelmed with submission volume. And downstream consumers (the vulnerability scanners, security information and event management systems, and risk platforms that ingest CVE data) are already struggling to provide actionable signal rather than noise.*

*Several adaptations are necessary, and they need to start now. Machine-readable first, human-readable second. CVE records need to be designed for automated consumption as the primary use case, with human-readable descriptions as a secondary output. This means structured fields for affected components, exploit preconditions, environmental factors, and (critically) machine-readable patch references that automated deployment systems can act on.*

Actually, all of those things I agree with completely. Modernizing our 27 year old CVE system to take advantage of everything that's been learned since its original design, makes a ton of sense. It should be designed to be deeply automated, to be managed, ingested and digested by machines. Why not? That's overdue, and that's what the CVE board members who met during the 2026 RSA Conference acknowledged about the CVE program needing an overhaul. Not specifically due to AI or Mythos, which wasn't yet on the scene, but just because we could now do a much better job of a critical need. The Cisco guy considers the shape of some of those changes, writing:

*Should We Use Hierarchical Identifiers? A parent Vulnerability Class Report identifier with child CVEs for individual bugs. Defenders who need to take action at the class level can operate on the parent. Researchers and tooling that need granularity can drill into children. Scanners and SBOMs (Software Bills of Materials) can index either level. This gives organizations the ability to respond to "your TLS library has a family of parsing vulnerabilities" rather than individually tracking 47 separate advisories.*

*CVSS (the Common Vulnerability Scoring System) was designed for human assessors making static judgments. EPSS (the Exploit Prediction Scoring System) moved toward probabilistic exploitation prediction. The next step is continuous, AI-updated risk scoring that incorporates real-time threat intelligence, proof-of-concept availability, attacker tooling trends, and the defender's own environment context.*

*What should defenders do now?*

*You don't have to wait for the CVE program to reinvent itself. The practical steps for security teams are clear, even if they're painful:*

- *Remove and migrate away from End-of-Life software and hardware.*
- *Stop treating every CVE equally. If you're patching by CVSS score alone, you're already behind. Use EPSS, CISA KEV, and real-time threat intelligence to prioritize by exploitability and environmental relevance. The volume is about to make score-based patching physically impossible.*
- *Know your inventory. Yes, actually know it. You cannot respond to a flood of vulnerability disclosures if you don't know what software you run, where it runs, and how it was built. SBOMs (Software Bills of Material) are not a compliance checkbox anymore. They're the only way to answer "does this affect me?" at machine speed.*
- *Invest in autonomous patching infrastructure. Automated testing, canary deployments, staged rollouts, and automated rollback. If your patch deployment requires a human to click "approve" for every single update, your cadence will be measured in weeks when it needs to be measured in hours.*

*Project Glasswing's findings will flow through these normal channels as patches land. Your job is to be ready when they arrive... and the window is closing: Anthropic chose not to release Mythos Preview publicly. That decision buys the industry time, but not much. Frontier AI capabilities and open weight models will advance substantially within months.*

*The capability to autonomously discover and exploit software vulnerabilities at scale is no longer theoretical; it's an engineering problem, and multiple organizations are solving it simultaneously.*

*The vulnerability disclosure ecosystem we've built over 27 years was a remarkable achievement for its time! It gave us a shared language, a coordination mechanism, and a way to hold vendors accountable. But it was built for a world where a prolific researcher might find 50 vulnerabilities in a year. We're entering a world where a single AI system can find that many before lunch.*

*The choice is not between perfection and action. It's between adapting now with grouped disclosures, tiered timelines, AI-generated patches, federated databases, and autonomous remediation.*

Okay. So if nothing else, we've just heard the somewhat panicked reaction from someone inside another major enterprise that obtained early access to Mythos. Though this author was careful to talk about the software security industry at large, it seems pretty clear that it was his own direct experience when "Cisco met Mythos" that drove this posting. He's essentially saying that there are so many problems that we cannot even count them. So, at least for the first pass, we're just going to classify them by generic type because enumerating them individually seems pointless.

I think that the major takeaway from this is that, yes, indeed, it's time to update the industry's aging coordinated vulnerability management system, and while we're at it, since AI has arrived in full force, let's incorporate AI-friendliness into the new system. Where I disagree with this author is in the longer-term effects of AI's involvement. I'll make my case for that once we get to today's main topic.

## Patch Deployment Latency

However, there remains one massive problem that no one is talking about – and this is particularly worrisome for a company like Cisco – which is patch deployment latency. It seems very clear that “Cisco meets Mythos” was a massive wakeup call, and that Cisco, like probably every other large enterprise having a major security software presence, is now or will soon be undergoing a significant AI-driven re-engineering of much, if not all, of their existing codebase.

But as we all know too well, having fresh, updated and for the first time ever, largely if not completely bug-free code ready to deliver, is entirely different from having that massively improved and far more secure code actually installed and running across its entire user base.

It should be very clear to everyone that Mythos-, Daybreak- and Codename MDASH-scale improvements are going to be sweeping across the entire industry’s software and that the result of that will be soup-to-nuts improvements in the operation and security of that software. But getting those improvements into the hands of its users, or more specifically removing all traces of the previous software, is going to be a big problem.

Only a handful of software suppliers truly have effective software update systems in place. Microsoft with Windows and Apple with their macOS and various mobile OSs can do this. And since the security of our web browsers is paramount and since browsers are, by definition, connected to the Internet, they’ve long had the ability to keep themselves current. And the major Linux distributions have also acquired this ability. What we’ve seen is that Microsoft, Apple and the browsers make it difficult to NOT remain current. This tendency to update autonomously and more or less continuously is likely to pay off handsomely in the coming months. We already saw Microsoft able to brag that the four critical vulnerabilities Codename MDASH discovered in their TCP/IP stack had already been patched by the time of their disclosure. But these few companies are the exception rather than the rule.

Many of the industry’s network appliances continue to suffer from the “if it’s not broke, don’t fix it” philosophy. So we may be in for a turbulent transition when the difference in the security between what’s running and what should be running becomes more significant than it ever has been. I am very grateful that this podcast did not stop at 999!

## Microsoft’s official Bitlocker “YellowKey” bypass mitigation

Last week we talked about the Bitlocker bypass which I was reluctant to label a “backdoor” despite a number of security professionals teasing that description. Microsoft was quick to jump on this even though its release by the hacker calling themselves ‘Nightmare Eclipse’ was deliberately timed to leave this exposed for the longest time possible, until June’s patch Tuesday.

The bypass now has a CVE: 2026-45585 and Microsoft’s Executive Summary for this explains: *“Microsoft is aware of a security feature bypass vulnerability in Windows publicly referred to as “YellowKey”. The proof of concept for this vulnerability has been made public violating coordinated vulnerability best practices. We are issuing this CVE to provide mitigation guidance that can be implemented to protect against this vulnerability until the security update is made available.”*

The Executive Summary then offers a breezy FAQ Q&A which asks and answers:

- **Should I leverage the temporary mitigation?**

*Microsoft recommends that you consider implementing these mitigations if you are concerned your devices and data are at risk of being compromised or stolen. For example, if your organization's employees take their work devices home or on business travel.*

- **What impact to service availability/management could be caused by implementing the mitigations?**

*Implementing these mitigations will not impact service availability or management operations.*

- **Do customers need to revert the changes made to mitigate the vulnerability once the security update to protect against this vulnerability is available?**

*No. The security update will maintain the mitigation's behavior once the security update is installed.*

- **I am using TPM+PIN, am I at risk of this vulnerability being exploited**

*No, if you are using TPM+PIN the vulnerability is not exploitable.*

We learn from this that, among other things, as I suspected and contrary to what Nightmare Eclipse claimed last week, there is no PIN bypass possible and anyone who is already using a PIN will be protected. The CVE description then has another FAQ which is more interesting:

- **What kind of security feature could be bypassed by successfully exploiting this vulnerability?**

*A successful attacker could bypass the BitLocker Device Encryption feature on the system storage device. An attacker with physical access to the target could exploit this vulnerability to gain access to encrypted data.*

- **Is there a script that I can copy and paste to implement a mitigation?**

*Yes. This script is an interim security fix that helps to reduce the risk of exploitation of the vulnerability. The script is for WinRE and removes [the file] **autofstx.exe** from the BootExecute registry value. Since BootExecute runs programs very early in boot (even in recovery mode), removing this entry prevents that executable from running in a high-privilege environment, reducing risk. It works by mounting the WinRE image, editing its offline SYSTEM registry to remove the entry if present, then safely committing changes and re-sealing WinRE so BitLocker trust remains intact. It's designed to be safe—if the autofstx.exe entry isn't there, it exits without making changes.*

So, Microsoft has provided a Powershell script that anyone who might have concerns that a local system might be abused in this way can run to remove the vulnerability to the hack that Nightmare Eclipse discovered and publicly disclosed to spite Microsoft. This will presumably be more formally fixed by next month's June patch Tuesday. I've provided the link to Microsoft's CVE disclosure which contains the script in the show notes at the bottom of page 8:

<https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-45585>

It's also worth noting that this vulnerability does not appear to be the result of a software flaw of the type that Mythos, Daybreak or Microsoft's own "Codename MDASH" vulnerability discovery

system would detect and remediate. This appears to be an architectural design flaw where Nightmare Eclipse, while examining the mechanics of Microsoft's deliberate design, cleverly realized that a side effect of one of the system's deliberately designed features could be abused to cause the system's Bitlocker drive to be and to remain decrypted following an interrupted recovery reboot.

The reason I'm highlighting this is just to point out that while our emerging AI systems very clearly promise to once and for all eliminate entire classes of software coding mistakes, not every security problem is the result of a coding error. We need to remember that it's very possible for a sufficiently complex system to exhibit unintended behavior by design.

### **Ubiquity patches 5 bad vulnerabilities – 3 maximum severity**

Since I know that Leo and many other listeners are users and fans of Ubiquity I wanted to make sure that everyone knew that it was once again patch time for Ubiquity.

Recall that two months ago, in March, Ubiquiti patched a maximum-severity flaw affecting the UniFi Network Application that allowed attackers to take over user accounts, as well as another vulnerability that could be exploited to escalate privileges. Since Censys shows that the Internet currently has nearly 100,000 publicly exposed UniFi OS endpoints, with nearly 50,000 IP addresses located in the U.S., any known and unpatched flaws in Ubiquity devices will tend to draw bad guys' attention. This explains why Ubiquiti products have been targeted by both state-backed hacking groups and cybercriminals in recent years, in campaigns that hijacked them to build botnets that concealed the threat actors' malicious activity. So it's certainly worth staying current.

Last week, Ubiquiti released security updates to patch three maximum severity vulnerabilities in UniFi OS **that can be exploited** by remote attackers without privileges. Just to be clear, the UniFi OS is what powers UniFi Consoles and helps manage IT infrastructure, including networking, security, and other services, as well as UniFi applications such as UniFi Network, Protect, Access, Talk, and UniFi Connect. The first flaw (CVE-2026-34908) enables attackers to make unauthorized changes to targeted systems by exploiting an Improper Access Control weakness in UniFi OS, while the second one (CVE-2026-34909) allows them to access files on the underlying system by abusing a Path Traversal vulnerability. This could be manipulated to access an underlying account. The third maximum severity security issue (CVE-2026-34910) makes it possible for malicious actors to launch a command injection attack after gaining network access by exploiting an Improper Input Validation vulnerability. Then on Thursday, another critical command injection flaw (CVE-2026-33000) was patched along with a high-severity information disclosure (CVE-2026-34911). All of these affecting Unifi OS devices.

The good news is that, as bad as these are, they are not known to have been exploited in the wild. Unlike Nightmare Eclipse's deliberate campaign of malice against Microsoft and thus sadly Microsoft's users, all of these flaws were responsibly reported through Ubiquity's participation in the HackerOne bug bounty program.

## **A critical SQL injection flaw targets Drupal systems**

A Google/Mandiant security researcher discovered and reported what's being described as a "highly critical" SQL injection vulnerability. After receiving and understanding the implications of Mandiant's discovery, on Monday May 18th, the Drupal project posted a public service announcement urging admins of its content management system to reserve time for core updates that addressed an issue that threat actors might start exploiting "within hours or days." Sadly, that announcement turned out to be prescient, since attacks soon followed the publication of updates to repair the problem.

The vulnerability that was resolved allows specially crafted requests to trigger arbitrary SQL injection on sites using the Post-greS-Q-L SQL database engine. As we've seen before, one of the mixed blessings of open source is that any repairs to its published code are trivially reverse engineered by attackers simply by seeing what changed. Once the resolved trouble has been identified, any online sites that have not yet switched over to the updated code – which is available to the attackers at the same time as everyone else – will be vulnerable targets of opportunity. In this case, the flaw is exploitable without authentication and can result in remote code execution, privilege escalation, and information disclosure.

Last Friday, the Drupal team updated their advisory to confirm that exploitation attempts have been detected. Drupal rated the vulnerability as "highly critical," assigning it an internal score of 23 out of 25. Unfortunately, it affects a wide range of Drupal versions on every supported branch. Rather than enumerating them I'll just urge anyone using Drupal anywhere to bring whatever version they may be using up to the latest release for their branch.

Although those not using Post-greS-Q-L may not be immediately vulnerable, everyone is being urged to update since the latest security updates also include fixes for upstream dependencies, including Symfony and Twig. One final point is that Drupal's advisory notes that Drupal 8 and 9 are both well past end-of-life, and that as a result the patches for those very old branches are provided on a "best-effort" basis only. They will still contain other known vulnerabilities that will never be fixed so their continued use is inherently risky.

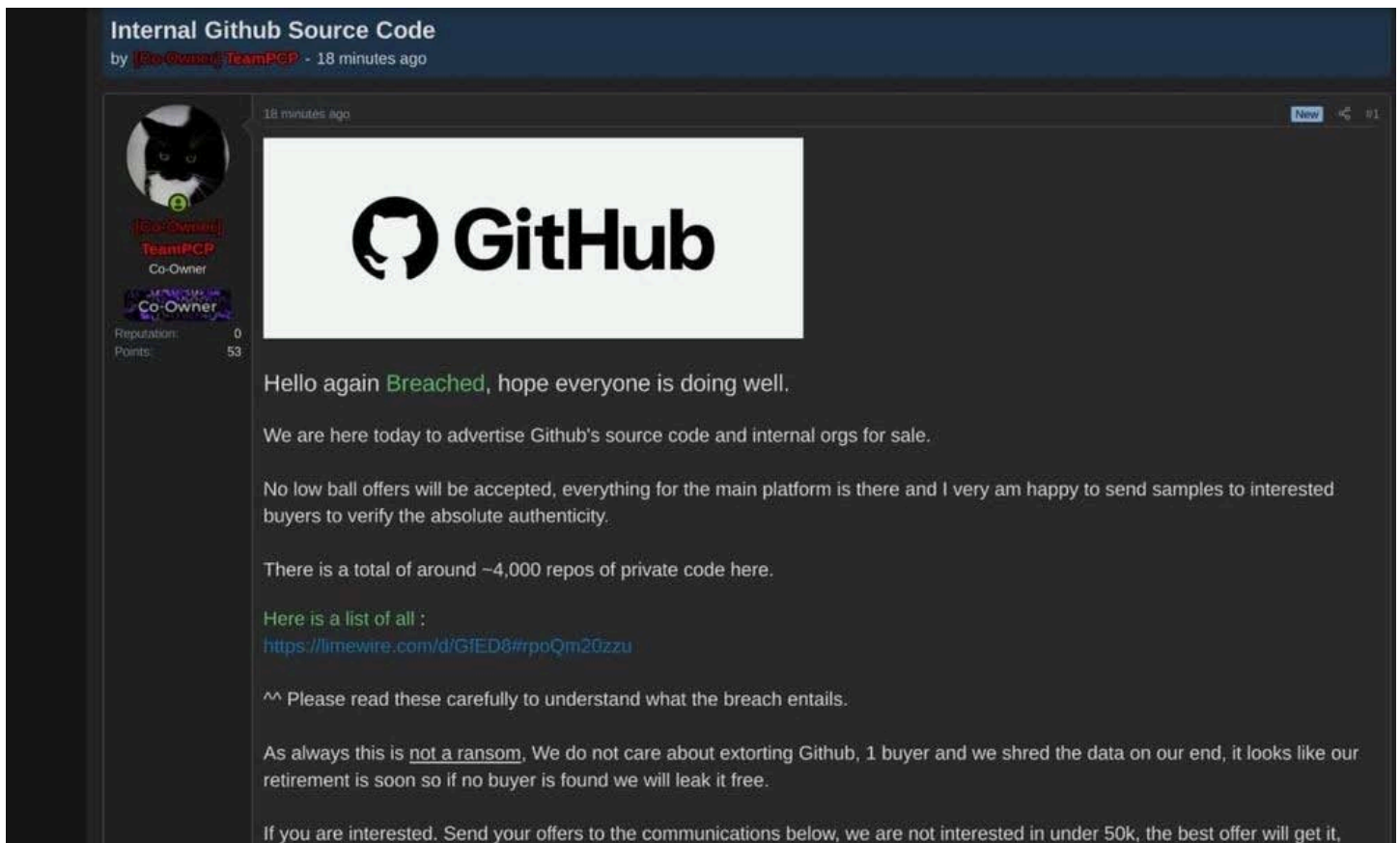
## **Microsoft: "No SMS for you!"**

In other news, Microsoft has announced that they will be phasing out the use of SMS for multi-factor authentication and account recovery for personal Microsoft accounts. It time to switch to passkeys, and so all users will be prompted to add a passkey the next time they log into their accounts. Microsoft says that SMS is a leading source of fraud and the single most targeted vector for account takeover.

What's more, passkeys allow for higher-speed login flows, since there's no need to wait for an SMS message to arrive. It also makes for a smoother and much more secure account recovery process since users will no longer be tied to a specific phone number.

This move makes Microsoft the first major platform to completely abandon SMS for multi-factor authentication. Other platforms like Google, Facebook, and Twitter are pushing users to stronger MFA alternatives, but still allow them to use SMS as a last resort.

## Github hacked:



So what happened? A Microsoft developer using their VSCode had the misfortune of installing a rogue extension from Microsoft's VSCode extension library, which is moderated and hosted by Microsoft. The result was that all of GitHub's internal repos, more than 3,800 internal code repositories, have been exfiltrated and are now being offered for sale on a hacking forum as shown in the screenshot above. GitHub says it is rotating critical secret tokens to prevent any future access to its assets. The malicious VS Code extension was identified as Nx Console.

## To Russia, with Love

Russians are continuing to use increasingly outdated software from Western publishers, most of which no longer receives tech support or security updates. Somewhere around one third of all Russian enterprises continue using software acquired before 2022 when Russia invaded Ukraine. The greatest concentration of Western software is in the corporate email segment with Microsoft still holding around 50% of the Russian market through its Exchange and Microsoft365.

An interesting thought experiment would be to imagine that "Codename MDASH" is turned loose on the Exchange and 365 codebases and uncovers a bunch of exploitable vulnerabilities. At this point it's a virtual certainty that's what would happen. But Russian enterprises do not receive the benefits of these discoveries and updates. I would imagine that there are those within the United States intelligence community who would love to have access to the things Microsoft had already found, fixed and patched. What harm could that do? And, of course, Russia is likely not alone in being a hostile foreign adversary that's continuing to use software we're in the process of fixing.

## My AI knows me

Last week, I shared my concern over the somewhat surprising consequences of having an AI chatbot learning about me in furtherance of the illusion that there's not only someone on the other end of the connection, but that it's someone who cares enough to accurately recall previously shared information. One of the things I forgot to say was that these observations felt so strong and self-evident to me, that the only way I could imagine that someone might **not** be totally blown away by the capability of today's AI would be if they had never actually used it. Or, stated differently: To use it would be to understand and be blown away by it.

Two days after I shared that, last Thursday a lightbulb lit up, so Friday morning I posed the following question to Claude. I wrote: *"When I've been singing AI's praises to other people I've sometimes been met with blank stares, as if the person I'm talking to wonders whether I've drunk the Coolade." What has struck me the most is when an AI not only answers well, but answers like an attentive friend would, speaking from a growing understanding of me – specifically me, not just some generic human. That transforms the interaction from a Q&A into a much richer conversation. And that led me to the revelation that anyone who is just using an AI chatbot WITHOUT any account may not have the benefit of that long-term history accumulation.*

*So I'm writing to ask... is that completely true (since browser cookies could create session continuity) and what exactly ARE the differences between the use of today's AI chatbots -- Claude and ChatGPT -- with and without a paid-for account?"* To that prompt, Claude replied:

*Your observation cuts right to the heart of what makes AI genuinely useful versus merely impressive-in-a-demo. Let me break this down carefully.*

*On the cookie question: Yes, cookies create session continuity — but that's a much narrower thing than it sounds. What a cookie buys you is that returning to the same browser on the same device might reconnect you to the same open conversation. That's not memory; that's just not having closed the window yet. The moment a browser session expires, the cookie becomes irrelevant, and there's zero synthesis happening across conversations. So: cookies keep a thread alive, they don't build a relationship. Here are the actual tiers of continuity:*

**No account at all** — *Each conversation is hermetically sealed. The model has no idea who you are, what you care about, or what you talked about five minutes ago in a different tab. Every session is a stranger introducing themselves. This is the experience most casual "I tried it once" people have had, which explains a lot of the underwhelmed reactions.*

**Free account** — *This is where the gap opens up meaningfully. With a Claude account (even free), your conversations are saved — you can read them back, search them, reference them. But more importantly, both Claude and ChatGPT have memory systems that do active synthesis: they don't just archive, they extract salient facts about you and inject them into future conversations. For Claude, what you're seeing right now in this response is a direct product of that — I know your background, your work on SpinRite and Security Now!, your health interests, the DNS Benchmark video project, all of it, without you having to re-establish any of it.*

**Paid account** — *The memory mechanism itself isn't a paid-only feature for either Claude or ChatGPT. What paid buys you is primarily: higher usage limits, access to the most capable*

*models, longer context windows, and additional features (like Deep Research, or image generation). The relationship-building aspect is available at the free tier.*

***The asymmetry nobody talks about*** — *Here's what I think is the underappreciated insight in what you've identified: the benefit isn't linear with use, it's threshold-gated. A casual user firing off three questions once a week gets almost nothing from memory because there's not enough signal to synthesize. Someone like you, who uses it heavily and across domains, gets a compounding return — each conversation adds fidelity to the model's understanding of your context, vocabulary, priorities, and working style. The person giving you a blank stare has probably only ever experienced the stateless version, which genuinely is just a fancy search box.*

*The analogy that might resonate for your audience: it's the difference between calling a business's general support line versus having a long-term relationship with a specific person there who knows your account history, your preferences, and how you think. Same underlying capability, completely different experience.*

So, to those listening who have not already figured out what has Leo and me so enraptured, if your experience to date has been limited to anonymously asking ChatGPT or Claude questions like any other search engine, what I learned is that it is **not** necessary to pay anything to experience what happens over time – a free account is sufficient. But it is necessary to create that unpaid free account which will allow the AI to “get to know you” which it cannot otherwise do. I believe you’ll see that it becomes transformative and, as I noted last week, somewhat worrisomely seductive.

## Sci-Fi

### “The Boroughs” on Netflix

Time Magazine wrote: *“Netflix's 'The Boroughs' Is 'Stranger Things Sr.' Backed by the creators who brought us the Upside Down, Netflix's newest sci-fi horror series features an all-star cast of senior citizens.”* First of all, I’m not a huge fan of horror which generally seems gratuitous to me, and that’s not at all how I would describe The Boroughs. So don’t let that put you off because there’s not much that’s “horrific.” The series average rating on IMDB is currently 7.5 out of 10, with the majority giving it an 8 and the second largest count giving it a 10 out of 10.

All eight episodes of “The Boroughs” became available last Thursday. This was presumably timed for binge watching over the long Memorial Day weekend. I had the date on my calendar as some hopeful-appearing Netflix science fiction, so Lorrie and I started watching it last Thursday on release night. I’ll be the first to say that the first episode or two were a tad rocky for me; and that opinion is echoed across other reviews. The setting is a retirement community out in some desert random where, at least in the beginning, more time is spent on geriatric infirmity than I thought was warranted. But to its credit, things pick up nicely by the end of episode 2, by which time the hook of an interesting mystery will be well set. “What exactly is going on here?” It’s not worth subscribing to Netflix to find out, but if you already have Netflix and, for example, you enjoyed the first season of Stranger Things, then it’s safe to say that eight episodes of Netflix fun and intrigue await you. Though it will be necessary to slog through the first episode or two, it’s probably worth giving it a chance.

## Listener Feedback

### Dale LTL

I want to share an interesting piece of feedback from a longtime listener and follower. It touches upon something we've talked about several times relative to the way some future local AI agent might always be looking over the shoulders of computer users to tremendous advantage. I cannot overstate the value and importance I see for having that. Chatting with an AI that grows to know you is already of inestimable value. So we already have that and that capability is only going to grow to become more useful over time. But, as I have repeatedly observed here, the overwhelming demonstrated strength of the range of "ClickFix" attacks, where users blindly follow on-screen instructions – which they do not understand – that lead them to paste malicious command streams into their PCs, serves to demonstrate that today's PC users have very little understanding of the way their machines operate. And that shows no sign of changing. It's not as if users are suddenly going to start reading the manual. But what could change is the adding the ability to the PC operating system – and/or its web browsers – to proactively prevent their users from self-inflicting harm.

However, I specifically said "future local AI" because these services, when they exist in the future, will need to cost nothing. They'll need to be part of the OS or browser and able to run on the local machine's AI engine. It's pretty clear that future PCs and smartphones will combine local and cloud-based AI.

Okay, so here's what our listener, Dale, shared:

*I very much enjoyed your last podcast admitting to offering your thanks to your AI of choice. I also thank Google Gemini, my current AI of choice, at least until it becomes another killed-by-Google project.*

*I would like to offer a way of using AI that I haven't heard anyone speak about – determining whether or not an email is phishing: My wife and I have several websites and corresponding social accounts online. Mine is mostly YouTube (almost 12,000 subscriptions) and hers is mostly Facebook and Instagram to boost her real world sales. As such we get a huge amount of spam and phishing emails collectively.*

*I have attached a screenshot of a recent scary email my wife received through her business email. It doesn't have any of the obvious spam phishing email indicators like an email address of [random numbers and letters]@scamming\_you dot com.*

Dale attached the image of the email, but since it contained identifying information I've reproduced the relevant text to give everyone a sense for it. It reads:

*We are contacting you regarding content associated with your account that has been identified during a recent intellectual property verification process. Our review indicates that certain materials may contain copyrighted content protected under United States law, including Digital Millenium Copyright Act (DMCA). These materials may have been published or shared without documented authorization from the applicable rights holder. To assist with your review we have attached a document containing the relevant content references and supporting*

*information.*

[Then there's a PDF icon and a link with the text: "Evidence of Copyright Infringement.pdf"]

*We kindly ask that you review the attached file and verify whether the referenced materials are being used with proper authorization. If necessary, please update, remove, or restrict access to the identified content across websites, social media platforms, and other distribution channels.*

Yikes. Now that AI can be employed to write highly convincing English we've been expecting to see an increase in the apparent veracity of spam. This is likely an example of that, and I have to say that anyone who didn't have their guard up could be excused for falling for this. It evidences none of the common telltale signs we've all been comfortably able to use until now to quickly identify emailed junk. Dale's note continues:

*So I asked Gemini. My prompt was "email of copyright content from shady records, email says from [noreply@appsheet.com](mailto:noreply@appsheet.com) legit or phishing?"*

*Gemini replied with... "This is 100% a phishing scam. Do not click any links, download any attachments, or reply to the email. Here is exactly what is happening and why it looks so convincing:*

- 1. The Legitimate Sender Address ([noreply@appsheet.com](mailto:noreply@appsheet.com)) AppSheet is a legitimate no-code application development platform owned by Google. Cybercriminals have figured out a loophole where they can create a throwaway app on the platform and use its automated workflows to send emails. Because the email physically originates from Google's infrastructure, it easily passes security checks (like SPF, DKIM, and DMARC) and lands directly in your main inbox instead of your spam folder. While the email address [noreply@appsheet.com](mailto:noreply@appsheet.com) is real, the content inside it was written by a scammer.*
- 2. The "Shady Records" Scare Tactic: Scammers frequently use urgent legal threats—like copyright, trademark, or intellectual property violations—to trigger panic. They want you to act quickly without thinking so you'll click their link to "view the evidence" or "dispute the claim." If you click the link provided in that email, it will likely lead to: A fake login page designed to steal your credentials. A malicious download (like a .zip or .doc file) that can infect your system with malware."*

---

*So, after fussing at Gemini for parent company Google knowing about this and not doing anything about it, I followed up with a heartfelt "thank you Gemini." I also have no doubts that the PDF link was hotwired to a nasty little payload that I'm glad we didn't get.*

*Thank you for all you do Mr Steve and Mr Leo. Been a fan since early 2000's Call for Help and The Tech Guy on my 12 foot mesh satellite dish. -Dale / Low Tech Linux / 'lowtechlinux' everywhere / openSUSE Tumbleweed KDE plasma fanboy*

Dale was clearly annoyed that Google allowed this crap to be generated and sent from their own domain while their own Gemini AI knew all about it, in detail. It's like "Oh, yeah, our service is being abused like that and this is an example of it." The good news, more broadly, is that I think

another huge aspect of our lives which AI has the promise to improve are all of these aspects of phishing and spoofing and generally unwanted conduct which bad guys have been getting away with for the past several decades. I think that the use of widespread "AI agency" for crime prevention is going to be one of the biggest things that is yet to happen. And given the speed at which all this is moving, we may not have long to wait.

# Vulnerability Debt Repayment

## Firefox 151

When I fired up my PC and then **Firefox 151**, I was greeted with a *"Your Firefox has been updated"* notice with a bunch of new features being called out. The top of the new-stuff list was their promotion of Firefox's new-ish VPN feature as a means of geo-relocating its users. Under the banner of *"Your VPN has places to be"* we are told: *"Our free, built-in VPN now lets you set your browsing location, giving you more privacy and control when you're away from home or the office."*



I understand Mozilla's motivation for promoting Firefox's new free VPN as an easy to use geo-relocation capability. But at the same time I was also a bit taken aback, since VPNs are already in the cross-hairs of politicians who have been rattling their sabers over the idea of (somehow) preventing their use, and even outlawing their use for accessing age-restricted websites. How, exactly, that's going to work no one knows. So this seems like needlessly rubbing our politician's faces in the issue. Making geo-relocation much easier and more accessible promises to increase its use and make the issue even more prominent. Since any attempt to curtail VPN use is going to create another big mess, it would be nice if everyone could just let this one slide by.

## But wait... there's (MUCH) more!

On the same page was another section titled *"A new era of Firefox security"* with the note: *"Firefox used advanced AI to uncover and fix **hundreds** of hidden security vulnerabilities before they could become threats."* That linked to a Mozilla post made last Thursday with the great title "The zero-days are numbered." What we learn from Mozilla is kind of amazing. They write:

*Since February, the Firefox team has been working around the clock using frontier AI models to find and fix latent security vulnerabilities in the browser. We wrote previously about our collaboration with Anthropic to scan Firefox with Opus 4.6, which led to fixes for 22 security-sensitive bugs in Firefox 148.*

*As part of our continued collaboration with Anthropic, we had the opportunity to apply an early version of Claude Mythos Preview to Firefox. **This week's release of Firefox 150 includes fixes for 271 vulnerabilities identified during this initial evaluation.***

*As these capabilities reach the hands of more defenders, many other teams are now experiencing the same vertigo we did when the findings first came into focus. For a hardened target, just one such bug would have been red-alert in 2025, and so many at once makes you stop to wonder whether it's even possible to keep up.*

*Our experience is a hopeful one for teams who shake off the vertigo and get to work. You may need to reprioritize everything else to bring relentless and single-minded focus to the task, but there is light at the end of the tunnel. We are extremely proud of how our team rose to meet this challenge, and others will too. Our work is not finished, but we've turned the corner and can glimpse a future much better than just keeping up. **Defenders finally have a chance to win, decisively.***

There's so much here, but the phrase that caught my attention was "*You may need to reprioritize everything else to bring relentless and single-minded focus to the task*". This is very interesting because what we've been hearing, more broadly and with generalization, was that the arrival of this new vulnerability-discovery capability was likely to stun development teams who would be thrown from complacency into overdrive and overtime. Mozilla continues, writing:

*Until now, the industry has largely fought security to a draw. Vendors of critical internet-exposed software like Firefox take security extremely seriously and have teams of people who get out of bed every morning thinking about how to keep users safe. Nevertheless, we've all long quietly acknowledged that bringing exploits to zero was an unrealistic goal. Instead, we aimed to make them so expensive that only actors with functionally unlimited budgets can afford them, and that the cost of burning such an expensive asset disincentivizes those actors against casual use.*

*This is because security to date has been offensively-dominant: the attack surface is not infinite, but it's large enough to be difficult to defend comprehensively with the tools we've had available. This gives attackers an asymmetric advantage, since they only need to find one chink in the armor.*

*We use defense-in-depth to apply multiple layers of overlapping defenses, but no layer is bulletproof. Firefox runs each website in a separate process sandbox, but attackers try to combine bugs in the rendering code with bugs in the sandbox to escape to a more privileged context. We've led the industry in building and adopting Rust, but we still cannot afford to stop everything to rewrite decades of C++ code, especially since Rust only mitigates certain (very common) classes of vulnerabilities.*

*We pair defense-in-depth engineering with an internal red team tasked with staying on the leading edge of automated analysis techniques. Until recently, these have largely been dynamic analysis techniques like fuzzing. Fuzzing is quite fruitful in practice, but some parts of the code are harder to fuzz than others, leading to uneven coverage.*

*Elite security researchers find bugs that fuzzers cannot largely by reasoning through the source code. This is effective, but time-consuming and bottlenecked on scarce human expertise. **Computers were completely incapable of doing this a few months ago, and now they excel at it.** We have many years of experience picking apart the work of the world's best security researchers, and Mythos Preview is every bit as capable. **So far we have found no category or complexity of vulnerability that humans can find that this model cannot.***

*This can feel terrifying in the immediate term, but it's ultimately great news for defenders. A gap between machine-discoverable and human-discoverable bugs favors the attacker, who can concentrate many months of costly human effort to find a single bug. Closing this gap erodes the attacker's long-term advantage by making all discoveries cheap.*

*Encouragingly, we also have not encountered any bugs that could not have been found by an elite human researcher. Some commentators predict that future AI models will unearth entirely new forms of vulnerabilities that defy our current comprehension, but we don't think so. Software like Firefox is designed in a modular way for humans to be able to reason about its correctness. It is complex, but not arbitrarily complex.*

***Defects are finite, and we are entering a world where we can finally find them all.***

All I can say here is **"Wow!"** I hope everyone understands how much it means for Mozilla to be feeling the way they obviously do after their close encounter with Mythos. 271 previously unknown vulnerabilities discovered during their initial evaluation. And this is in Firefox's deeply scrutinized codebase which, so far as they knew, had no vulnerabilities. I think this is the most significant statement, made from the front lines of the security battle, we've seen to date.

No one should make the mistake of assuming that Anthropic's Mythos Preview has some magic fairy dust that no one else has. There is no reason to believe that **anyone** has that – or that it's even possible to have that. This is why I think that the whole idea of regulating AI is a fool's errand. It's the politician's knee-jerk reaction to change and it could not be more wrong-headed. We've often noted that it's not possible to regulate the use of cryptography, since the knowledge of it has previously escaped. There's no longer any mystery about how to do it. Exactly the same is true of Large Language Model technology. It's spread all over YouTube, it's all over Github, it's already everywhere. At this point all we can do is be as prepared as possible for whatever comes next.

And Mozilla's statement of optimism contains what I was hoping and expecting to hear: They clearly understand that this new tool has, for the first time ever, given them a true **advantage** over the bad guys. They alone decide when to release new code to their userbase. Nothing forces them to do that. This means that now, armed with AI-enabled pre-release vulnerability discovery, no discoverable vulnerabilities ever again need to be released to the public.

I want to highlight another aspect of this just to put a fine point on it. Mozilla wrote: *"Computers were completely incapable of doing this a few months ago, and now they excel at it. We have many years of experience picking apart the work of the world's best security researchers, and Mythos Preview is every bit as capable. So far we have found no category or complexity of vulnerability that humans can find that this model cannot."* Allow me to repeat that last sentence because this is everything: *"So far we have found no category or complexity of vulnerability that humans can find that this model cannot."* Now, I understand that this is attempting to prove a negative. This does not say that there **is** no category or complexity of vulnerability that a human could find that the Claude Mythos Preview cannot. Only that, so far, they have never found one. All by itself that's a significant statement. But what's also significant is that this is **not** Anthropic saying this about their own creation. This statement from Mozilla about Firefox is as much from the front lines of mission critical vulnerability discovery as it gets.

Six weeks ago, during our April 14th podcast number 1074, I noted that my first working title for that podcast was "Mythos: Marketing or Mayhem." But once I brought myself up to speed it was obvious to me that this was not marketing hype, despite what jaded cynics were claiming without ever taking the time to inform their opinions with facts. Because I did, that podcast's final title was "What Mythos Means". And now, just a scant six weeks later, we're seeing the consequences of its application in one of the places it matters most – in a modern web browser.

### **Is it going to be a tsunami?**

There's something I see missing from some of the public discourse over automated AI vulnerability discovery. For example, at the end of a recent posting by Cisco about the strain AI vulnerability discovery is placing on the aging CVE system, Cisco writes:

*Anthropic chose not to release Mythos Preview publicly. That decision buys the industry time, but not much. Frontier AI capabilities and open weight models will advance substantially within months. The capability to autonomously discover and exploit software vulnerabilities at scale is no longer theoretical; it's an engineering problem, and multiple organizations are solving it simultaneously.*

*The vulnerability disclosure ecosystem we've built over 27 years was a remarkable achievement for its time! It gave us a shared language, a coordination mechanism, and a way to hold vendors accountable. But it was built for a world where a prolific researcher might find 50 vulnerabilities in a year. We're entering a world where a single AI system can find that many before lunch.*

*The choice isn't between perfection and action. It's between adapting now with grouped disclosures, tiered timelines, AI-generated patches, federated databases, and autonomous remediation.*

Okay. Mozilla turned Mythos loose on their Firefox codebase and uncovered 271 vulnerabilities. This blew their minds. But they recovered from the vertigo that this introduced, brewed fresh pots of coffee, called their significant others to explain that they'd be working late – and early – and got to work. Once they had caught up with Mythos, they shipped Firefox 150 with those newly discovered problems resolved. So here's my point: What's going to happen the next time they turn Mythos loose on Firefox's codebase? Perhaps a big nothing. Maybe something, but it won't be much. It's not as if those 271 vulnerabilities will need to be re-discovered each time. There were hundreds of previously unknown problems that Firefox's codebase had acquired over its entire lifetime. All of those are now gone, and while mistakes and regressions can happen they are not going to suddenly come back overnight.

The way to phrase this is to say that Mozilla's Firefox codebase had a previously unrealized large amount of "technical debt" to repay. The Claude Mythos Preview revealed the exact nature of that debt, to which the Mozilla team, having recovered from the shock, got down to the work of repaying those years of debt. But now they have. The ledger is balanced. And all they need to do now, going forward, is deal with whatever occasional mistake they might make. With prudent use of AI to check their work, they might never ship anything with a discoverable vulnerability.

We know that Microsoft must now be facing an even more daunting challenge, since their codebase is far more massive, just as old, and seemingly less well maintained than Mozilla's. The fact that every month sees a hundred or so bugs resolved suggests to me that they are likely facing a great many thousands of newly discovered problems. We know that thanks to their early access to Anthropic's project Glasswing, they had access to Claude's Mythos Preview and also to their perhaps their own dramatically more powerful "Codename MDASH" system. Can you imagine what a catastrophic disaster they must be discovering? They shared the critical vulnerabilities discovered and resolved in the first and most obvious place they looked, Window's TCP/IP networking stack... a critical component that likely received a lot of prior attention.

But even there, as we noted last week, those networking stack bugs – which were somewhat obscure – were all zapped two weeks ago with this month's patch cycle. So they, too, are gone and put to rest. There's no doubt that Microsoft would have immediately aimed "Codename

MDASH” at its next target and got down to work.

And here’s the interesting bit of psychology that’s at play: As a developer in charge of security, as long as you do not know that there are any problems, you can sleep soundly through the night. But once you’ve been handed an extensive list of vulnerabilities and realize that the software you believed was secure is actually and proveably Swiss cheese, it will be quite difficult to get **any** sleep until every one of those problems have been resolved.

The source of all this angst is the awareness that, especially for an open source project like Firefox, Linux, Chrome with Chromium and to some degree Android, the bad guys also have the same access to the same code and if they do not yet have vulnerability discovery tools of equal strength and power, we can be absolutely certain that they are working as hard and fast as they can to develop them. So there is almost certainly a very limited window of opportunity to get those problems resolved, to push updates out into the field and into user hands before the malicious actors have the chance to attack.

So the bottom line is: LLM technology has **already** driven a massive change in the practice of software security. Almost overnight it has revealed the size of the software security debt which many of us adjacent to the industry have long sensed with a dull ache. And we are now in the midst of an industry-wide rush to pay that debt down using LLM technology to locate and repair each and every problem before it can be used against us by others who will doubtless be attempting to leverage the same technology.

But the crucial point to appreciate is that this will be a transient event. Once that debt has been repaid, the entire industry will again settle down to a new norm with the world never being the way it has always been. It will never again be the same because our software will finally be working the way we always intended and hoped it was. It was just too complex, and thus too expensive, to make perfect. That’s what has been forever changed. As Mozilla themselves explained, LLM technology dramatically reduces the cost of problem discovery. This means that we can – and will – now afford to find and fix all of the longstanding problems. And we **must** afford it, since it will soon be just as inexpensive for the bad guys to find and abuse any problems that remain.

Everyone listening to this podcast has heard me use the analogy of a sponge to describe our industry’s software security. I’ve said that all of the objective evidence shows that security is porous and that the more pressure that’s put toward penetrating that security the more penetration will occur. The application of this new LLM technology will finally put this analogy to rest. Other non-software vulnerability problems will certainly remain. Spoofing and phishing and social engineering attacks will remain — or will at least require another form of remediation — which I spoke about earlier. But the exploitation of vulnerable software finally has the chance to come to a well deserved end. We are entering a new era.

