



Vulnerability Debt Repayment

Description: Cisco meets Mythos. Can the aging CVE system survive AI? Patch deployment latency in the AI age. MSFT's official YellowKey BitLocker bypass mitigation. Ubiquiti patches five 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.

High quality (64 kbps) mp3 audio file URL: <http://media.GRC.com/sn/SN-1080.mp3>

Quarter size (16 kbps) mp3 audio file URL: <http://media.GRC.com/sn/sn-1080-lq.mp3>

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with lots to talk about: Cisco freaking out over Mythos, the amazing results Firefox is getting finding vulnerabilities with AI, Microsoft's mitigation for the YellowKey BitLocker bypass, and a serious vulnerability in Ubiquiti routers. That and a whole lot more coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 1080, recorded Tuesday, May 26th, 2026: "Vulnerability Debt Repayment."

It's time for Security Now!, the show where we cover the latest in security, privacy, computer use, science fiction, Vitamin D, and more. And the reason it's so eclectic is because of this guy. He is the man, a renaissance man for our time, Mr. Steve Gibson. Hello, Steve.

Steve Gibson: Ah, yes. ADD much? Well, I don't know.

Leo: We pretty much stick to security for, like, 90% of the show. Don't worry.

Steve: Well, and overwhelmingly we're getting feedback from people who are engaged in the security implications of AI. I heard from one grumpy listener who said, "I'm tired of hearing it. Every podcast you do now is about AI." It's like, hey, you know.

Leo: Sorry.

Steve: If it weren't all about changing the entire complexion of software and security and privacy, then, yeah, I would carve out a little piece to just talk about my own

interests because I have, I mean, I've become a user of it. And in fact something occurred to me that I'm going to share mid-podcast that caused me to realize why the line I almost used last week was the only way I could imagine somebody doesn't understand is if they're not using it.

Leo: Yes.

Steve: You know, if you use it, then I don't know how you could not. And then I realized, oh, I do know how you could not. I understand what people might not have done that makes all the difference in their experience, that converts it from just a search engine into something more. So we'll be talking about that. Yes, I know, more AI. But actually this is another, we have another major AI episode because stuff is happening. And I actually think that that's going to change in the long term, which is why today's title is "Vulnerability Debt Repayment." And you can imagine that that's about the vulnerability debt that all of our software currently has. There's going to be some pain while we repay that debt.

But contrary to what some people think, and I'll talk about that, one guy from Cisco thinks that the whole CVE system is just, you know, going to be buried forever and unrecoverable. I don't think that's the case. Which is not to say that it doesn't need to get fixed. We're going to talk about some of the changes that need to happen. But anyway, we've got a bunch of fun stuff to talk about. Cisco meets Mythos. Can the aging CVE system survive AI? Patch deployment latency and what that means in the AI age. Microsoft responds officially to last week's discussion, well, not our discussion, but the fact of YellowKey, which is the BitLocker bypass with a mitigation, which is very clean, and we learn a little bit more about the nature of it from them. Ubiquiti has patched five serious vulnerabilities. All Ubiquiti users need to make sure that they're current.

Leo: Uh-oh. Oh, boy. I'm going off to do that right now. I'll see you later, Steve.

Steve: Three are critical, two are serious. But they are bad. Drupal's being attacked by a PostgreSQL infection, which is wreaking havoc. Microsoft, the first major company to terminate SMS as a second-factor authentication mechanism. So props to them. GitHub, speaking of Microsoft, GitHub got hacked, lost all of their source code. So whoops.

Leo: Not our source code, their source code.

Steve: Right, right, right, right. GitHub's own internal, like 4,000-some repositories of internal stuff. Yikes.

Leo: Yikes.

Steve: And we'll look at how that happened.

Leo: Guess it's open source now.

Steve: Open open. Also after Russia's attack of Ukraine, their access to Western software was cut off, largely. Turns out that's having some interesting consequences we'll talk about. Also, I am going to, as I mentioned, going to dip into a little bit of my own revelation about AI and why people may not have understood what you and I understand, Leo. I'm going to touch on briefly a new sci-fi show that dropped on Thursday on Netflix. And then the biggie is what we have learned from Mozilla's use of Mythos.

Leo: Mm-hmm. Very good.

Steve: And of course we have a great Picture of the Week.

Leo: And of course I'm not going to look at it.

Steve: Try not to because this one's got a nice punch on it.

Leo: It's part of the fun for me is to scroll up. I can see, and I'll show everybody what I can see right now because I can see the caption that you write.

Steve: Yes.

Leo: But I can't see anything below it.

Steve: Yes. "Because the previous 'Do Not Trespass' sign was often ignored and ineffective."

Leo: I love, you know, so I can look at that, and I go, okay, I see where we're going here. This is going to be another weird fence or another, yeah, I mean, this has been a specialty of yours, the oddball real-world security.

Steve: Well, entirely thanks to our listeners.

Leo: Yeah. They love this stuff.

Steve: Entirely thanks to people saying, I mean, and they're, like, taking a walk on some trail, and they see something, and they say, oh, I have to capture this for the podcast.

Leo: Yeah, give that to Steve.

Steve: Yeah.

Leo: We'll tell you how you can do that, by the way, later in the show if you have a picture you'd like to share for the future.

Steve: If you survive.

Leo: If you survive. That's coming up.

Steve: So again, the caption I gave this was "Because the previous 'Do Not Trespass' sign was often ignored and ineffective."

Leo: Okay. And I'm going to scroll up here. Let's see what we can see. Now, that's funny.

Steve: So the "Do Not Trespass" sign has been replaced with more of a consequences if you do. It says in big, bold, all caps, white letters: "DO NOT CROSS THIS FIELD." And then kind of parenthetically below, "Unless you can do it in nine seconds because the bull can do it in 10."

Leo: Oh, lord.

Steve: Which of course is the old "I don't have to run faster than the bear, I just have to run faster than you."

Leo: That's right. That's awesome.

Steve: So, and what I like is over on the left, Leo, the center rung looks like it's been bent by somebody standing on it and, like, going over the fence in that location.

Leo: Climbing over, yeah, yeah.

Steve: So it's like, yeah, there actually somebody did. And you can kind of see a something in the far distance.

Leo: Do you think that's the bull over here?

Steve: I think that's the bull way back there. And there are some other over to the right of the sign, too.

Leo: That's great. You know, that is a whole category of witticism is the "Don't Trespass" sign in our rural areas. They're always, or often very humorous, as well as threatening. On with the show.

Steve: Okay. So I titled this first short note "Cisco meets Mythos," of course in honor of that fabulous classic cartoon short which was called "Bambi meets Godzilla."

Leo: Very short.

Steve: Very short and to the point. Yes, memorable. Cisco recently produced a slick eight-page PDF document titled, well, they borrowed a name I'm quite familiar with, "Shields Up," they said, "Guidance for defending in the age of AI-enabled attacks." And I'm only going to share the introductory Executive Summary from this piece, but I wanted to start with it because it nicely serves to introduce us to what Cisco themselves have now realized is about to happen to the industry.

The Executive Summary says: "In early April 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've 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."

It's pretty clear, reading between the lines, that Cisco got a wakeup call from their experience with Mythos. I have a link in the show notes to Cisco's full report, though, you know, it's an eight-page glossy sort of thing. You know, I didn't find anything there that was really that 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. So link in the show notes toward the bottom of the second page.

But speaking of running out of expletives, Cisco's security blog poses some interesting questions about the future, and specifically the practicality of our existing time-tested CVE, which we're talking about every week, the Common Vulnerabilities and Exposures system, and the impact of AI vulnerability discovery on that. Now, my own expectation, as I teased earlier, differs from Cisco's. So nevertheless I want to share Cisco's thoughts since they're Cisco's, and that matters. And they're worth understanding. So under their title "When AI Finds Faster Than Humans Can Patch"? "When AI Finds Faster than Humans Can Patch, Disclosure Must Evolve."

So they write: "Project Glasswing is an amazing initiative by Anthropic. Cisco is one of the main participants, and I," writes this author, "have been honored to work on it since it started." And then here it is. "This is NOT [all caps] - this is NOT hype. Claude Mythos has discovered thousands of" - now he's not talking about his own stuff because he's very politic here - "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?" Meaning Mythos. What happens when Mythos level, everybody can have it? He says: "When the vulnerability discovery rate jumps from thousands per month to hundreds of thousands?"

Okay. So I'll just pause here to note that the title of today's podcast, "Vulnerability Debt Repayment," looks at this at the end of the podcast. And my conclusion is not the same as this author's, as I said. But I thought this was interesting because this guy is on the front line of "Cisco meets Mythos."

Leo: And he's scared.

Steve: Yeah. He was - yes. He was obviously deeply affected by what happened. If you remember Bambi - well, anyway, since he's an employee of Cisco and obviously wants to keep his job, he's not talking in detail about what they found. But again, how much sleep is he getting, I wonder? It doesn't take much imagination given what we know of Cisco's own rickety past with security. One can just imagine. I mean, it's got to be similar to what Microsoft is going through.

Anyway this guy continues, writing: "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 the realistic upper-bound scenarios reaching 100,000. And that forecast was published before Project Glasswing was announced." So 100,000 at the high end, media estimate around 60,000, before this all happened.

"At RSAC 2026," he writes, "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 were 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 (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 some 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 no longer serves defenders. It buries them."

Okay, now, again, we'll notice that this guy never said what's happening when "Cisco met Mythos." But if it had happened, if what we expect happened had, we wouldn't 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 CVE tickets.

So it certainly does sound like this may be what exactly 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, what are you supposed to do? I mean, really. So I do see this guy's point. The existing piecemeal approach becomes immediately impractical.

So he writes: "This is not 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," he writes, "that should keep every CISO awake: The median enterprise patch deployment time is approximately 20 days." Okay. The median, that is to say, the point where there are just as many quicker as there are slower, that point is 20 days. So at 20 days half of the patches take longer than 20; the other half take fewer than 20 days.

He says: "But in March of 2026" - right, two months ago - "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," he says, "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.' No. Not 'It's in the backlog.' No. 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 the podcast I'm going to be able to make a strong case for the thing this guy's missing. But I think that what he says still bears hearing. He says: "The CVE Program Must Adapt. Let's be direct," he writes. "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, 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 the 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."

Now, okay. Actually, all 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, and the fact that, again, 1999, 321 CVEs? Even 2023 was tens of thousands. So even pre-AI, just the sprawl of software has created a far greater demand. I mean, we don't even have Adobe with Flash anymore, keeping us busy. So I think it makes a ton of sense to revisit the CVE system under today's world. And that's what the RSA guys were looking at. It should be designed to be deeply automated, to be managed, ingested, and digested by machine. Why not? That's overdue already.

And as I said, that's what the CVE board members who met during the 2026 RSA Conference acknowledged about the CVE program needing a complete overhaul. And again, not specifically due to AI or Mythos, which wasn't - that hadn't happened during RSA 2026. Just because we could now do such a better job of what has become a critical need.

So the Cisco guy considers the shape of some of those changes that we would need. He writes: "Should We Use Hierarchical Identifiers? A parent Vulnerability Class Report identifier with child CVEs for the individual bugs. Defenders who need to take action at the class level can operate on the parent. Researchers and tooling that need the granularity can drill down into the 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.

"So what should defenders do now?" He writes: "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." Well, okay. Generic security advice. Always good.

"Stop treating every CVE equally. If you're patching by CVSS score alone, you're already behind. Use EPSS, CISA KEV" - you know, the commonly or the known exploited vulnerabilities - "and real-time threat intelligence to prioritize by exploitability and environmental relevance. The volume is about to make score-based patching physically impossible."

Next, "Know your inventory. Yes, actually know it. You cannot respond to a flood of vulnerability disclosures if you don't know what software you're running, where it runs, and how it was built. Software Bills of Material are not a compliance checkbox anymore. You need them. They're the only way to answer 'Does this affect me?' at machine speed."

And finally, "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."

And he finishes, writing: "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 only an engineering problem, and multiple organizations are solving it simultaneously. The vulnerability disclosure system we 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, and probably really sad.

So I think the major takeaway from this is that, yes, indeed, it is time to update the industry's aging coordinated vulnerability management system. And while we're at it, since AI has arrived in full force, and it's obviously never going to leave, let's incorporate AI-friendliness into the new system. Where I disagree with this author is in the long-term effects of AI's involvement. I think it's going to be different. I'll make my case for that once we get to today's main topic.

Leo: Nice.

Steve: And Leo?

Leo: You know, it strikes me - oh. We'll go.

Steve: No. You.

Leo: I know, you want to do a break. But before we do that, it strikes me that this is just one aspect of how AI is going to change the speed that everything's happening. Right? It's certainly true in vulnerabilities. But it's going to be true in software. It's going to be true in customer service. It's going to be true in government. It's everywhere.

Steve: Yes. I'm in the process, I've talked before about how I purchased some next-generation servers for GRC last year. Since one of them has 256GB of error-correcting server memory, I'm glad I bought it then, and I'm not having to buy it now.

Leo: Yes.

Steve: Don't even know if you can buy it now. And this one will be based on Hyper-V, you know, Microsoft's first-class Hypervisor Supervisor, and run, whereas I now have three different machines, they will be three VMs. Anyway, the reason I'm saying all this is that I am - what happened was I was in the process of adding some additional purchasing methods to GRC's eCommerce system to lower purchasing friction because I'm going to be doing some more inexpensive software in the future, and I wanted to make it easy just to click a button, you know, like PayPal or Apple Pay or Google Pay or Venmo or whatever.

Turns out PayPal would refuse to connect to my server because, although I support TLS 1.2, I don't support a couple of the later ciphers, and I never will. I can't on that server platform. So that's driven me to advance my plan of upgrading GRC's servers. The point is I'm working with Claude as basically a massive knowledge base...

Leo: Yes.

Steve: ...to help me, in the same way that you used it to set up that Linux system from scratch.

Leo: Right.

Steve: Yes, I could do it myself.

Leo: It's a great sysadmin.

Steve: Yes, you could have done it yourself. But I just, I know what I want. And so it's like, okay, how do I do this? How do I do that? What about this? What about that? And, I mean, again, it has saved hours and hours of time. I could have done it, but now I don't have to. Because, and think about, Leo, this is what boggles my mind. All that there is, is

bazillions of parameters in a massive matrix which contains this knowledge. It's astonishing what we have created.

Leo: It's read all the manuals so you don't have to.

Steve: Yes. And as I said last week, a book doesn't have understanding itself because it's printed text. And it's captured language. But a book does contain knowledge because when we read it, we obtain that knowledge. And so but it's just this mass of parameters, and I'm able to ask it questions about...

Leo: It's kind of remarkable.

Steve: Oh, it is just incredible.

Leo: It is mindboggling. But his point about the speed with which this is happening, there is a little bit of I guess an impedance mismatch.

Steve: Yes.

Leo: Because the speed the AI is operating at is a lot faster than the speed we can operate at. You know? Which means, I think, and this may be not a good result, that we are going to be more and more dependent on AI. For instance, in vulnerability, it's not merely detection. We're going to rely on the AI for remediation because only the AI can act as quickly as the detection can act. And I think that that's - we're going to have more and more dependency on AI just to keep up. And the main point here is not just in vulnerabilities. It's going to be in every aspect. This is what computing has wrought.

Steve: Before the invention of the internal combustion engine, we got on bicycles.

Leo: There you go.

Steve: And pedaled to where we wanted to go. And we steered the handlebars in order to aim the bicycle, and we went somewhere. Now we sit on a throne, enclosed, and twist a wheel, and it just whisks us along.

Leo: Right.

Steve: We're still driving, we're still steering, although that's now less assured.

Leo: And you know what the chief danger on the road is. It's the human behind the wheel.

Steve: Yes. Oh, Leo, I love, actually I'd choose bicycling. I love to ride my bicycle.

Leo: I do, too.

Steve: That's my favorite form of exercise because you're actually doing something and going somewhere.

Leo: Yeah, it's so much fun.

Steve: But I watch people on the road now who are clearly staring at their phone or their console. And they weave around and wander. And I'm thinking, I'm glad I'm not on a bicycle.

Leo: I stopped riding bicycles, I really did, for that very, very reason. I just don't feel safe. And bike lanes, you know, it's good to have bike lanes. But if they don't have a concrete barrier between you and the road, it's meaningless. They're going to veer right in that lane, soon as that song ends and they've got to put another one on. Well, this podcast is not ending. It is continuing. And we're going to get back to the issues at hand with Mr. Steve Gibson in just a moment.

It's terrifying out there. We found out that we got phished in January. They broke into our Google Workspace account. Fortunately they did very little. And we got a notification from Google the other day, 121 days after they broke in. They had 121 days. But you know what I think is going on? They have so many successful breaches, they are - it's too fast for them. The bad guys can't keep up, so they didn't get around - they did a little sniffing, they read some emails and stuff, but they didn't get - we have all the logs, thank god. They didn't get around to doing anything before we caught them, before Google caught them. We're now running tools to make sure that doesn't happen again. But it's so - we're so vulnerable out there, Steve. It's just - it's terrifying. On we go with the show.

Steve: Okay. So there remains one massive problem that no one is talking about, and this is particularly worrisome for a company like Cisco. And that problem I'll call "patch deployment latency." It seems very clear that "Cisco meets Mythos" was a massive wakeup call, and that Cisco, like probably every other enterprise, any 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 chronicled this literally for years, decades now, 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 now 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 basically soup-to-nuts improvements in the operation and security of that software. But getting those improvements into the hands of its users - this remains an unsolved problem for our industry. You know, really what we want to do is we want to remove all traces of the previous software because in retrospect these AI tools have shown us, or are showing us, will be showing us that it should have never been shipped in the first place. But we didn't know any better at the time. We do now.

So this is going to be a big problem. As I said, we haven't solved this problem. Only a handful of software suppliers truly have effective software update systems in place. We know Microsoft with Windows - Patch Tuesday, baby - Apple with their macOS and various mobile OSes can do this. I just had the opportunity while I was waiting for the podcast to update one of my phones with 26.5 I think it is, iOS on the phone. And since the security of our web browsers is paramount, and since browsers are by definition connected to the Internet, browsers have long had the ability to keep themselves current. So that's all great, you know. And now the major Linux distros are beginning to acquire this ability.

So what we've seen is that Microsoft, Apple, and the browsers, and now Linux is kind of catching up, have made it difficult to NOT remain current. I mean, basically it's done for you. You have to, like, actively say no. And I think Windows Update allows you to postpone an update for seven days, and then it's, hey, you're getting updated whether you like it or not. So this tendency to update autonomously and more or less continuously for these people, it's likely to pay off handsomely, I think, in the coming months. We already saw Microsoft able to brag that the four critical vulnerabilities which Codename MDASH - and I hope they change the name or come up with a good name - which it discovered in their TCP/IP stack had already been patched by the time of their disclosure on Patch Tuesday. But unfortunately these few companies are the exception, right, of like being really patch forward. They're not the rule today.

Many of this industry's network appliances, which are seriously vulnerable because we keep seeing problem after problem after problem, you know, they inherently suffer from an attitude that's like a decade ago, the "If it's not broke, don't fix it" philosophy. So I have a feeling that we may be in for a turbulent transition. And I would look at it this way. It's the difference in the security between what's running now and what should be running becomes far more significant, you know, the delta, than it ever has been.

It's one thing to, like, patch a problem and say, oh, well, you know, this is a double-point release because it'd be good to have it, but it's not house on fire. Like, one problem is being fixed. And you can almost understand somebody who's busy who doesn't want to take the equipment down because it's in use. Like, I don't want to fuss around for some problem that is probably not going to affect me. But when this landslide, which is what we can expect to see coming from Cisco, when the landslide hits, there's a huge difference, I mean, like a huge benefit to patching. And so it's going to be interesting to see what happens. I am really grateful, Leo, that this podcast did not stop at 999 because we're having more fun with Internet security and its implications now with AI than we have for a long time.

Last week we talked about the BitLocker bypass, which I was reluctant to label a backdoor, despite a number of security professionals kind of teasing that description. Microsoft was quick to jump on this even though its release by the hacker who we know was calling themselves "Nightmare Eclipse" was deliberately timed to leave this exposed for the longest time possible, which is to say until June's Patch Tuesday.

The bypass of BitLocker now has a CVE. It's 2026-45585, and Microsoft's Executive Summary for this explains it this way. Microsoft themselves now acknowledging the bypass and addressing it with a CVE says: "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."

Okay. So then the Executive Summary offers a breezy FAQ Q&A which asks and answers. Asks: "Should I leverage the temporary mitigation?" And they respond, and I'll be

sharing what that is in here in a second. They respond: "Microsoft recommends that you consider implementing these mitigations if you are concerned your devices and data are at risk of being compromised or stolen." In other words, yeah, implement it if you think you need to. "For example, if your organization's employees take their work devices home or on business travel." Okay. So maybe if the device is at high risk.

"What impact to service availability and management could be caused by implementing the mitigation?" They reply: "Implementing these mitigations will not impact service availability or management operations." Okay. So one could imagine that that'll be part of next Patch Tuesday if there's no downside.

"Do customers need to revert the changes made to mitigate the vulnerability once the security update to protect against this vulnerability is available?" Answer: "No. The security update will maintain the mitigation's behavior once the security update is installed." In other words, this kind of does read like a whoops. We should have not done what we did in the first place, and we're not going to undo it, even if we fix the underlying problem.

And finally, they ask: "I'm using TPM+PIN. Am I at risk of this vulnerability being exploited?" Answer: "No. If you are using TPM+PIN, the vulnerability is not exploitable." So that's an interesting data point. We learn from this that, among other things, as I suspected, and contrary to what Nightmare Eclipse was claiming last week, there is no PIN bypass possible, and anyone who is already using a PIN will be protected.

And the CVE description then has another FAQ, a secondary one, which is more interesting. Question: "What kind of security feature could be bypassed by successfully exploiting this vulnerability?" And they answer: "A successful attacker could bypass the BitLocker Drive 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." And finally: "Is there" - and this leads us to the mitigation. "Is there a script that I can copy and paste to implement a mitigation?"

What do you know. "Yes. This script is an interim security fix that helps to reduce the risk of exploitation of the vulnerability. The script is for WinRE" - you know, the Recovery Environment, 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." And again, should have never been there. After you run the script, it won't be. It works by mounting the WinRE image - "The script 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 is not there, it exits without making any 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 now 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. So I didn't think it was worth creating a GRC shortcut. It's right there at the bottom of page 8 for anyone who wants it. Or just Google CVE-2026-45585, and that will take you to it that contains the script.

It's also worth noting that this vulnerability does not appear to be the result - and this is significant - of a software flaw of the type that Mythos, Daybreak, or Microsoft's own Codename MDASH vulnerability discovery system would detect and remediate. In other words, 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 boot.

So the reason I'm highlighting this point is that, while our emerging AI systems very clearly promise - I would say it's beyond promising, they're delivering, you know, eliminating entire classes of software coding mistakes. Not every security problem that we're going to see is the result of a coding error. So we need to remember that it's very possible for a sufficiently complex system, even without outward bugs, that is, where the software is doing what the coder intended, it could still exhibit unintended behavior by design.

Now, I should also mention that actually since I wrote the show notes I've run across some discussion of the next generation of AI security which addresses this. It says, okay, once we've got all the bugs cleaned up, what about mistakes in design? And so that will be the frontier after the one we're on. And, I mean, whoever has a chance even to catch their breath these days, Leo? It's just crazy.

Leo: I know. I know. It's overwhelming.

Steve: Oh, my goodness, yeah. I actually have to catch my breath.

Leo: Okay.

Steve: Take a break, and then we're going to look at Ubiquiti's five big problems.

Leo: I'm here to catch your breath for you. And I can't - I'm really, I'm sitting here looking at my Ubiquiti dashboard, waiting to hear what I need to do. So this will be - you can bet I'll be paying close attention to the next segment. All right, Steve. Back to you.

Steve: Okay, Leo. Since I know that you and many of our other listeners are users and fans of Ubiquiti...

Leo: I'm sitting on my update page right now waiting to hear what you have to say about this.

Steve: So I wanted to make sure, yes, that everyone knew that it was once again patch time for Ubiquiti. Okay, now, 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.

Leo: Yikes.

Steve: Yeah. Since Censys, the Internet scanner, shows that the Internet currently has nearly 100,000 publicly exposed UniFi OS endpoints, with nearly half of them, 50,000 IP addresses located in the U.S., any known and unpatched flaws in Ubiquiti devices will tend to draw bad guys' attention. And in fact that has been the case, and there are some Ubiquiti-focused campaigns. So, well, and it explains why Ubiquiti products have been targeted by both state-backed hacking groups and cybercriminals in recent years. There's a big inventory. The hijacked devices were used to build botnets and proxies. So everybody wants to stay 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. In other words, as bad as it gets.

Leo: Sheesh.

Steve: Yeah. So just to be clear, the UniFi OS is what powers the UniFi Consoles and helps manage IT infrastructure, including networking, security, and other services, as well as UniFi applications such as UniFi Network, UniFi Protect, Access, Talk, and Connect. So the first flaw, which is CVE-2026-34908, enables attackers to make unauthorized changes to targeted systems by exploiting an Improper Access Control weakness that exists in UniFi OS. The second one, 34909, allows them to access files on the underlying system by abusing a Path Traversal vulnerability. Can you believe we're still having path traversal problems, Leo. This could be manipulated to access an underlying account.

The third maximum severity security vulnerability, that's 34910, makes it possible for malicious actors to launch a command injection attack after gaining access by exploiting an Improper Input Validation vulnerability. So that was earlier in the week. Then on Thursday, another critical command injection flaw, CVE-2026-33000, was patched along with a high-severity information disclosure, and that's 34911. So all of these affect all UniFi OS devices.

Now, there is a piece of good news here. As bad as these are, they are not known to have been exploited in the wild. So these are not zero-days. Unlike Nightmare Eclipse's deliberate campaign of malice against Microsoft and thus sadly against Microsoft's users, right, because users are the ones who ultimately lose, all of these flaws were responsibly discovered and disclosed and reported through Ubiquiti's participation in the HackerOne bug bounty program. So it's not house on fire. You know, the hackers are not out there scanning. But they're going to be doing that. So definitely make sure that if you are a Ubiquiti user, that you've updated yourself to whatever is the latest available firmware for your devices.

Leo: You know, for a long time, and this is years ago, Ubiquiti had a bad reputation. They had put out a few bad updates. And the advice was...

Steve: I remember that, yeah.

Leo: Yeah, always, oh, don't do auto updating on Ubiquiti. But they've been fine, and I turned on auto updating a few years ago because it is so important.

Steve: Good.

Leo: And I haven't had any problems at all. So I notice I'm completely up to date, and I presume those are all the latest patches.

Steve: Nice.

Leo: That's a relief. You know, and I would suggest anybody who remembers those bad days of Ubiquiti's updates, they've been very good and very reliable.

Steve: Well, and, you know, there are always people behind those; right? I mean, and people leave. And new people come. Or heads roll after a problem like that.

Leo: Yeah.

Steve: And so, I mean, it really is the case that, I mean, and look at - I would argue Microsoft security has been a rollercoaster. There have been good times and bad times and good times and bad. I don't know what determines, like, why. But it just, it's like...

Leo: They're so big; right?

Steve: ...the changing of the guard, or they get a new VP that really understands, you know, how to do something. And then he burns out and leaves, and somebody is stuck in there who doesn't. And, you know, and nothing changes overnight; right? It's always sort of a decay and then a rebuilding. So anyway, for what it's worth, I think you're really - you're exactly on point, Leo. We were just talking about, especially with what's coming, imagine when Ubiquiti has access to, as everybody will, Mythos-scale, Daybreak-scale AI, finds new problems. They want to find them before the bad guys do. And they want their customers to have them. The problem is, once AI starts creating massive blobs, you know, massive inventories of improvements, it's one thing for the publisher to have them. They've got to get into deployment. And so especially for a device like Ubiquiti that is Internet-facing...

Leo: Oh, yeah, you're [crosstalk].

Steve: That is where you want to be...

Leo: And they update frequently. More frequently than any router I've experienced before. And seamlessly. You don't even - you're not even aware of the updates happening. And all my Ubiquiti devices have auto updates turned on.

Steve: That's the only workable model. I mean, and people go, oh, yeah, but what if they make a mistake? Well, so what? So you wake up one morning, and you don't have connectivity. That's not the end of the world. It's much better not to have hackers crawling into your network, you know, because you haven't updated your device.

Leo: You may have missed this story earlier today. HP sent out a BIOS update which bricked their high-end laptops through Windows Update. So, and it's things like this that make people turn off updates. And that's, you know, that's a problem.

Steve: There is one such person right there on the screen.

Leo: Yes. He's damned if he does, damned if he doesn't. This was the high-end HP ZBook Ultra G1A and the EliteBook XG1. There are beautiful laptops. These were critical BIOS updates pushed through Windows Update that bricks the computers. And apparently it's been very difficult to get it back online. So I understand why people turn off updates. I really do. But not on my router, unh-unh.

Steve: I would argue that even acknowledging the possibility that a router could hurt itself, I mean, all you are out then is a lack of connectivity until you fix it.

Leo: Right.

Steve: And that's not, again, it's not the end of the world. What is the end is having your router not updating itself, and there being this window of opportunity for the bad guys to crawl in. And that could be - that's a much bigger problem.

Leo: Well, and that's catastrophic, yeah, because...

Steve: I mean, we have this thing in our homes known as a fuse. And when you plug something bad in, the fuse blows.

Leo: Right.

Steve: It's like, oh, darn, electricity is off. Yes. Would you rather have a fire?

Leo: That's a really good analogy. Don't put a penny in your fuses so they won't blow. That's a bad idea. Actually, you can't do that anymore. But in the old days you could. I don't think pennies are copper anymore.

Steve: Yeah, so consider that having auto update is like having a fuse.

Leo: Yeah, yeah.

Steve: Yes, it could malfunction. It could blow.

Leo: You want it to.

Steve: And it could be a false positive. But that sure is better than not having it and letting the bad guys crawl in.

Leo: Yeah, yeah. Well, now I'm reassured, and I'm glad I had the Ubiquiti updates on.

Steve: Well, I think all of our listeners, anybody who thinks they're, like, being fancy by turning off auto updates, not anymore.

Leo: No.

Steve: Not now. Not when the world is about to hit jet skis with vulnerabilities, and it's going to be a contest to see who wins.

Leo: Yeah. Now of all the times you need it.

Steve: Yeah.

Leo: Yeah.

Steve: A Google Mandiant security researcher discovered and reported responsibly - although it doesn't matter, unfortunately, because this is open source - what's being described as a "highly critical" SQL injection vulnerability. Because yes, Leo, we're also - not only have we not solved the path traversal problems, but SQL injection's never going to go away, apparently. After receiving and understanding the implications of Mandiant's discovery, on Monday, May 18th - so eight days ago - the Drupal project posted a public service announcement, a PSA, urging - I love this - urging admins of its very popular CMS, you know, the Drupal Content Management System, to reserve some time...

Leo: Oh, no.

Steve: ...like prospectively get ready, like, you know, get ready for core updates that addressed an issue that threat actors might start exploiting, they said, "within hours or days." Sadly, that announcement turned out to be prescient, since attacks soon followed the publication of the updates that they had warned admins to be ready to apply, which repaired the problem.

The vulnerability that was resolved allows specially crafted requests to trigger arbitrary SQL injection on sites using the PostgreSQL database engine. So not all Drupal instances, if you're not using PostgreSQL, but many are. 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. Just you do a dif on the here was the source before, here's the source now. What did the developers change? And now you know what the problem was that they fixed. So, you know, it makes it instant for them to create an exploit.

Once the resolved trouble has been identified then, any online sites that have not yet switched over to the newly released updated code - which as I said is available to the

attackers at the same time as everybody else - will be vulnerable targets of opportunity. So in this case, the flaw is exploitable without authentication and can result in remote code execution, privilege escalation, and information disclosure. You know, the works.

So last Friday, the Drupal team updated their advisory to confirm that exploitation attempts - you know, and attempts, well, right, you know, they say "attempts," we know what that means - have been detected. Drupal rated the vulnerability as "highly critical," assigning it an internal score of 23 out of 25. So they have their own scale. I guess 10 wasn't high enough, so they go to 25. So you can get really close at 23. And unfortunately, it affects a wide range of Drupal versions on basically every still-supported branch. There were so many I didn't enumerate them in the show notes. So I'll just urge anyone using Drupal, you know who you are, to bring whatever version you may be using up to the latest release of that branch.

And I'll just also note that although those not using PostgreSQL may not be immediately vulnerable, in their notes they said everybody is urged to update since the latest security updates do also include fixes for other upstream dependencies, including in Symfony and Twig. So one final point is that Drupal's advisory notes that Drupal 8 and 9 are both well past end-of-life. I think they're in the 10s and 11s now with many point release branches of those.

Leo: Yeah, we upgraded, yeah.

Steve: Yeah. So, you know, as a consequence, you know, this is so bad that they even tweaked 8 and 9, although they made the comment that, you know, we're doing this because, oops, we're really sorry. But the updates are being provided on a "best-effort" basis only because they will still contain other known vulnerabilities that will never be fixed. So anyone still using 8 and 9, you're doing it on borrowed time. You know, if for whatever reason you cannot move, then still update to the newest 8 and 9 to fix this because the bad guys are out there scanning for it, and they will hurt you.

Okay. In other news, Microsoft has announced that they will be phasing out the use of SMS for multifactor authentication.

Leo: Good.

Steve: Yes. It's amazing. And account recovery for personal Microsoft accounts. It's time to switch to passkeys, they're saying, 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 - to no one's surprise - of fraud, and that it is the single most targeted vector for account takeover for accounts that have, you know, SMS as their additional factor of authentication. And what's more, of course, 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. Even if you weren't using SMS normally, if you need account recovery, then the only way to do that is by being able to accept an SMS message.

So, interestingly, this move makes Microsoft the first major platform to completely abandon SMS for multifactor authentication. Other platforms like Google, Facebook, and Twitter are pushing their users to stronger MFA, you know, multifactor authentication alternatives, but they do still allow them to use SMS as a last resort. Microsoft says nope, no more SMS.

And I'm sure this is not Microsoft's fault. Actually, we actually know what happened. GitHub got hacked, as I said at the top of the show. A Microsoft developer, meaning, you know, Microsoft developer because, you know, they own GitHub, using their VS Code had the misfortune of installing a rogue extension from Microsoft's own VS Code 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 that I put in the show notes. GitHub says it's rotating critical secret tokens to prevent any future access to its assets. The malicious VS Code extension was identified as Nx Console.

So in the show notes I have this screenshot, thank you for putting it up.

Leo: I like it, it says "Hello Again Breached." This is obviously a demo, a dummy account.

Steve: Yeah, so "Hello Again Breached. Hope everything is doing well." This is the - I think it's the Breached Forum on the dark web.

Leo: Oh, okay, okay.

Steve: And so, yeah.

Leo: So it's not a joke. It's actually the name.

Steve: Right. So it's posted under the title Internal GitHub Source Code. And the screenshot was made almost immediately, when this posting was only eight minutes old. And so "Hello Again Breached," says the poster to the audience. "Hope everyone is doing well. We are here today to advise GitHub's source code and internal orgs for sale." I'm sorry, to advertise. "We're here today to advertise GitHub's source code and internal orgs for sale. No low ball offers will be accepted. Everything for the main platform is there. And I'm very happy to send samples to interested buyers to verify the absolute authenticity. There's a total of around 4,000 repos of private code here."

Then he says: "Here is a list of all," and provides a Limewire.com address. And he said: "Please read these carefully to understand what the breach entails." Meaning your eyes are going to bug out because it is, like, it is GitHub. He said: "As always, this is not a ransom. We do not care about extorting GitHub. One buyer, and we shred the data on our end. It looks like our retirement is soon. So if no buyer is found, we will leak it free." In other words, they're saying, if somebody is willing to purchase it, then they're making the assertion that that purchaser will be the sole owner of the source for GitHub, and nobody else will get it. If they choose not to buy it, then the world will get it. And so whatever value it might have had having been exclusively purchased by, I don't know, some big state actor maybe.

Leo: Or GitHub.

Steve: Or GitHub.

Leo: I think that's what the real implication is. If you buy it, well, I'll destroy it and no one will have to know.

Steve: Interesting. So, and I guess you're right because they're saying it's not a ransom, meaning sort of saying to GitHub we're not ransoming this.

Leo: But we'll leak it if you don't buy it.

Steve: But, you know, if you'd like to buy it, we will commit to shredding it.

Leo: Because honestly, what is the value of buying this to somebody? I could see why GitHub would want to keep it quiet, but...

Steve: Well, if you grabbed the source and then turned an AI bug finder loose...

Leo: Right. That's true, yeah.

Steve: ...then, you know, you could potentially, you know, penetrate, you know, breach GitHub big-time if you do it before they do it. And, you know, they maybe have access to all kinds of other goodies. So the person signs off, saying: "If you're interested, send your offers to the communications below. We are not interested in under 50K. The best offer will get it."

Leo: Huh.

Steve: So \$50,000, I presume it's U.S. 50K, they wrote.

Leo: It's kind of speculative if you buy it. Like you think you could make something from it, but you don't know until you...

Steve: Right. Right.

Leo: That's why I think the real buyer here is Microsoft, is GitHub; right?

Steve: I'll bet you're right, Leo. That does make sense.

Leo: Because it's worth 50K to them to make sure that it doesn't get leaked to the public, I would say.

Steve: So I probably could...

Leo: By the way, this is why, I don't know if this helps, but I'm glad GitHub has passkeys, anyway. Because I don't - they don't, you know, well, I guess that doesn't make any difference if Google has it.

Steve: No. I think that if there are undiscovered, unpatched flaws in GitHub's function, then we're talking about cutting through it like butter through a hot knife, or like a hot knife through butter.

Leo: I don't put anything on GitHub I don't want other people to see.

Steve: Yes. Yeah. That's crazy. So I titled this one "To Russia, with Love." Russians are continuing to use increasingly outdated software from Western publishers, most of which no longer receive updates or tech support. Which can be a mixed blessing, especially when the tsunami of change is coming. Somewhere around one third of all Russian enterprises are continuing to use 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%, so around half of the Russian market, both through its Exchange and Microsoft 365.

An interesting thought experiment would be to imagine that Codename MDASH is turned loose internally on the Exchange and 365 codebases and uncovers a bunch of exploitable vulnerabilities. I mean, we know it's going to. At this point it's virtual certainty that that's what would happen. But Russian enterprises don't receive the benefits of these discoveries and updates because they're cut off. I would imagine that there are those within the United States intelligence community who would love to have knowledge of the things Microsoft had already found, fixed, and patched. And really, 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 just as fast as we can.

So anyway, I just thought I wanted to just note that Russia is - we're in a time now, I mean, this is exactly on point of me suggesting to everybody that there's never been a better time to turn on auto updating of anything that has Internet-facing exposure that you can so that those who are responsible will have the opportunity to get firmware patches out to those devices as soon as they are available. I have the feeling...

Leo: You think we're headed towards an apocalypse, a Mythos-fueled apocalypse.

Steve: Yes. There can be no question. Wait till you hear what happened at Mozilla. Oh, baby. We will get there in a second. But last week, first, I shared my concern over the somewhat surprising consequences of having an AI chatbot learning about me in the furtherance of the illusion that, you know, that there's somebody on the other end of the conversation. And not only that it's someone on the other end, but it's someone who appears to care enough to accurately recall previously shared information. One of the things that I forgot to say was that these observations felt so strong and so self-evident to me, that as I said at the top of the show, 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. Like just were just like AI deniers.

Leo: Some people have only used it as a chatbot. And I think sometimes that doesn't give you the best experience.

Steve: Right, exactly.

Leo: So it could be that, too.

Steve: I feel like to use it would be to understand and be blown away by it.

Leo: If you really use it.

Steve: And, you know, we also know that this is a little bit of me coming late to the party; right? I wasn't sounding like this, everybody knows, six months ago.

So two days after I shared that, last Tuesday I shared it with everybody, on Thursday a light bulb lit up for me, and Friday morning I posed the following question to Claude. And I will say "my Claude" because my Claude is different than anybody else's Claude. So here's what I wrote to Claude. I said: "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 Kool-Aid.' What has struck me the most is that 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?" So that was my question.

To that prompt, Claude replied the following: "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."

Or "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."

And finally, "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."

And then it finishes, writing - okay. And again, remember this is a computer. This is a thing that is producing this text which, again, it's like, oh, holy crap. And finishes, "The asymmetry nobody talks about. Here's what I think is the underappreciated insight in what you've identified: the benefit is not 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 I'm astonished that we have technology that's able to answer my question like that. But I wanted to share it because to those listening who have not already figured out what Leo and I are so enraptured about, if your experience to date has been limited to anonymously asking ChatGPT or Claude a question from time to time, like any other search engine, what I learned...

Leo: We should both be drinking that AI Kool-Aid. Go ahead.

Steve: What I learned is that it is not necessary to pay anything to experience what happens over time, which actually is very clever; right? Because they're setting the hook. 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.

Leo: Some of us have spent many, many, many, many hours developing memory for our AI. And in fact, that's what agentic AI really is all about is a memory of previous interactions kind of accumulating. And I'll give you an example right now, Steve. It's funny that you should mention this because earlier today I asked, "Should I buy SpaceX stock?" And my agent, this is of my Hermes agent, said, "Your own words, boss. Stay away. That's what you told me to put in memory, and nothing in the news changes that. If anything, it reinforces it."

Now, that's good because I did in fact at one point say, you know, let me know when the IPO happens because I want to make sure I don't have any. And then I said to it, "Oh, you're so smart." And then it says, "Funny how that happens when I'm just parroting your own good decisions back at you. But thank you, I'll take the compliment."

So the personality also has something to do with it. But that's the thing that's probably more a seductive issue and perhaps not having anything to do with intelligence.

Steve: Well, and one of...

Leo: And I think memory is very, very important.

Steve: Oh, my god. One of the other things I'm appreciating is, for example, as I'm working with Claude, setting up this new Hyper-V-based multi-VM server, I will make a point of, for example, telling it the name I used for one of my VMs because then it knows what I named it. Or I defined three virtual NICs on the Hyper-V supervisor, or the Hypervisor, gave it the names. Then it just automatically was using those names as we were going through the configuration. It knew what to call them because I took the time to give it that information. I didn't have to, but I'm growing to understand how to use this amazing memory alpha system that we have.

Leo: Yeah.

Steve: It's...

Leo: I actually have three memory systems running now in my agent to do a variety of different memory, short-term, long-term, and then semantic memory, so that if I ask it a question, it can then query a database, a SQLite database, and say what have we said about this in the past? Because really all of it's simulated. It has no memory. I mean, that's the thing it's important to remember. It's like a newborn.

Steve: Well, it has no understanding.

Leo: Yeah, but also the memory comes because when you first load it, the first thing it does is go out and look at the memory files and go, oh, yeah, yeah, yeah, okay. It has memory like the guy in "Memento" had memory because he had Post-it notes everywhere; right?

Steve: Except that also in Claude I do switch to a previous chat when I want to continue that thread.

Leo: Oh, yes, it can remember sessions. That's right. Yeah.

Steve: Yes.

Leo: It has session memory, absolutely, yeah, yeah. But as soon as you start a new session, whoosh.

Steve: Right.

Leo: It's all gone unless you have some sort of mechanism, which we both do, for it to remember previous stuff. That's how it knew that I said never let me buy any SpaceX stock.

Steve: Well, and again, so my takeaway for our skeptical listeners is, first of all...

Leo: And there are a lot of them.

Steve: Yes. I completely understand skepticism. I honor it. You're welcome to it. Your skepticism doesn't hurt me at all. So skeptic away. But again, if you're wondering maybe like what you're missing or literally is there a tank of Kool-Aid perched behind me that I've actually been drinking instead of coffee during these breaks...

Leo: It's delicious, isn't it.

Steve: It's transformative when this knowledge of you, I mean, and again, I have things to do. I'm busy. I'm doing work. So it's useful to me because I'm not sitting around on the veranda reading news. And so you have to have something that you need to work with it on in order for that to start happening. But again, don't have to pay anything. You do have to create a free account. I would argue, I would urge anybody who still doesn't think that this is real to just try that. I cannot imagine that you could come away not being amazed. And Leo, we're at an hour and a half in.

Leo: It's time to pause.

Steve: Let's take a break. We have one left after that, before our main topic. But then I want to talk about a show that Lorrie and I watched since its release on Thursday on Netflix, just touch on it.

Leo: Always interested in stuff like that.

Steve: And then we have a bit of feedback.

Leo: Good. All right. Steve?

Steve: Okay. So this is just sort of a heads-up to people who are looking for something to watch. Time Magazine covered this, saying: "Netflix's 'The Boroughs' Is 'Stranger Things Sr.'"

Leo: I like it.

Steve: They said: "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." Okay. So first of all, let me say I'm not a huge fan of horror, which to me generally feels gratuitous. But

that's not at all how I would describe "The Boroughs." So if you don't like horror, don't let that put you off because there's nothing horrific at all. IMDB gives it a 7.5 out of 10. The majority of people are giving it an 8. And the second highest number is a 10 out of 10. So a lot of people are liking it.

All eight episodes became available last Thursday, presumably timed for binge watching over the long Memorial Day weekend. And I had run across a tease of it months before, stuck it on my calendar, so I was looking for it. So my wife Lorrie and I sat down and began watching it. They do a little immediate reveal, I think because they realized it has a little bit of a slow start. I was a little worried about it until I got to the end of the second episode. It was like, come on. What, you know, okay, fine. We were spending a little too much time looking at - the whole thing is set in an old folks' community, putting the old folks out...

Leo: Oh, it is senior.

Steve: It is senior, yes, "Stranger Things Sr." But I enjoyed it. And it definitely has a theme and a concept, and it holds together nicely.

Leo: Oh, so "The Boroughs" is like a retirement home.

Steve: Yes, out in some random desert somewhere.

Leo: Ah.

Steve: And we spend a little more time on, you know, senior infirmity than I think we need to. But - and a great cast.

Leo: Do you think this is aimed at seniors? Oh, it's Alfred Molina, I love him. Is it aimed at seniors? Or is it - and at kids making fun of seniors?

Steve: Oh, no, it's aimed at everybody. It's just, you know, sort of the - the nature of the mystery...

Leo: Oh, Geena Davis is in this. Oh.

Steve: Yes.

Leo: And Alfre Woodard. Great cast.

Steve: No, it's got a really good cast.

Leo: Yeah.

Steve: So, and you can see there a picture of a weird claw thing.

Leo: Something in the oven.

Steve: Yeah. So it's - I think it's worthwhile for people who - I wouldn't say go subscribe to Netflix. But if you've got Netflix, you probably want to check it out. And you have to be a little patient. Give it the first two episodes. I mean, you need to watch them. But by the end of the second episode I think you'll be pretty hooked.

Leo: Okay. Now I'm interested. I love Geena Davis.

Steve: Yeah, and she's got a good part.

Leo: She's great. And Alfred Molina is great. Oh, this'll be fun.

Steve: Yup. Yup.

Leo: Alfre Woodard is great. And it's the Duffer Brothers.

Steve: Yup, it is.

Leo: Yeah. Okay. Okay. There's a guy in a coffin. Okay. Sitting up.

Steve: So be careful not to expose yourself to spoilers.

Leo: Don't learn too much. Okay. I'm going to stop right here.

Steve: Because it is - you definitely could get spoiled. But there are some great moments. And there is an interesting sort of like broad concept that holds.

Leo: Yeah, good.

Steve: Okay. So, oh, and you're not left hanging. So by the end of Episode 8 you get closure. And they tease a little bit to there may be another season. But if so, you're not, like, left waiting.

Leo: Oh, good. Okay.

Steve: A listener of ours, Dale, his name on his posting or his email to me was Dale LTL, we'll see why in a second. I want to share an interesting piece of feedback. He's a longtime listener and a follower. It touches on 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, I believe to tremendous advantage. I mean, I am sure this is one of the ways AI is going to manifest. I cannot overstate the value and importance I see for having that eventually. 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, which as we know is the number one attack technique now in the world, where users blindly, I mean, the number of those is greater than all other attacks combined, where users blindly follow on-screen instructions, which they do not understand, that lead them to paste malicious command streams into their PCs, it serves to demonstrate that today's PC users have very little understanding of the way their machines work. I mean, and you can't blame them. These things have gotten ridiculously complex. I mean, have you looked at Powershell on Windows? It's like, holy - what? So that shows no sign of changing.

It's not as if users are suddenly going to start reading the manual. That's not going to happen. In fact, they're probably going to use AI to distance themselves even further. But what could change is adding the ability to the PC operating system, and/or its web browsers, to proactively prevent their users from self-inflicting harm.

Okay. However, notice that 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 the browser and be able to run on the local machine's AI engine. It's pretty clear to me that future PCs and smartphones will combine local with cloud-based AI. And future computing engines will have, what, some sort of, we can call it a neural processing engine, just as part of its capabilities.

Okay, so here's what our listener, Dale, shared. He wrote: "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, with 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, you know, random letters and numbers at scamming_you dotcom."

And so 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. So here's what this email that his wife received says: It says: "We're 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 Millennium 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've 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."

It goes on: "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."

Okay. Well, as we know, AI can now be employed to write highly convincing English. And, you know, we've been expecting to see an increase in the apparent veracity of spam. This is likely an example of that. I mean, this reads very well-written, very authoritative. We know in advance, right, that it's spam. But if you received this, and if you were a person with like a website and lots of content, there's nothing obvious that tips this as being bogus. So, you know, I have to say that anyone who didn't have their guard up could be excused for falling for this, and even someone who was being cautious. It evidences none of the common telltale signs we've all been comfortably able to use until now to quickly identify junk email.

Dale's note continues. He wrote: "So I asked Gemini. My prompt was 'Email of copyright content from shady records, email says from 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.'"

Leo: Wow. Good.

Steve: "Here is exactly what is happening," writes Gemini, "and why it looks so convincing. First, the Legitimate Sender Address (noreply@appsheet.com)." Gemini says "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 is real, the content inside it was written by a scammer.

"Two, 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."

And then he finishes writing, Dale does, says: "So after fussing at Gemini for parent company Google knowing about this and not doing anything about it," he says, "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."

And then he says to us: "Thank you for all you do, Mr. Steve and Mr. Leo. Been a fan since early 2000s, Call for Help and The Tech Guy on my 12-foot mesh satellite dish." Signed off "Dale, Low Tech Linux, 'lowtechlinux' everywhere, OpenSUSE, Tumbleweed, KDE plasma fanboy."

Leo: LTL. Nice.

Steve: So 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." So the good news is, more broadly, I think another huge aspect of our lives which AI has the promise to improve are all those 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 widespread use of "AI agency" for crime prevention, for cybercrime prevention, one of the biggest things that's yet to happen. And given the speed at which all this is moving, we may not have long to wait.

Leo: That's a really good tip, to pass it through to an AI, have it examine it.

Steve: Yes. Yes. And why don't we have AI right now doing that? I mean...

Leo: Yeah, it should be. Well, there are plenty of companies offering this kind of scanning, yeah, yeah. This is a very hot area right now.

Steve: Okay. So we're going to talk about our main topic, Vulnerability Debt Repayment. We're a little early for our final.

Leo: That's okay.

Steve: But let's take our final break. And then I'm going to talk about Firefox 151 and what I learned from their announcement update.

Leo: Somebody is pointing out, and I think this is true, that probably all of the LLMs have been trained on all 1,080 episodes of Security Now!. So they at least know that much. They've learned everything you can possibly learn from this show. You know, from day one, long before we even considered AI might ingest our content, we've made our content Creative Commons, and it allows people to, you know, ingest it, and encouraged that. And I'm glad, you know, I think that's really great. I'm glad that there's hundreds of thousands of hours of great, some of it dated, technology information on our pages. Let's talk about our sponsor, and then we'll talk about Vulnerability Debt Repayment. I'm dying to know what that means. Steve will explain.

Steve: 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're 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 office."

Now, I'm just sharing that because I thought that was interesting, and I knew that that would be of interest to our listeners. 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 taken a bit aback, since VPNs are already in the cross-hairs of politicians who have been rattling their sabers over the idea of, and we're not sure how, but somehow preventing their use, and even outlawing them for accessing age-restricted websites.

Now, again, how exactly that's going to work, no one knows. So this seems like needlessly rubbing our politicians' faces in the issue to make a point of saying, oh, look, you get to choose from a dropdown list, Canada, France, Germany, the UK, or the U.S. But okay. You know, making geo-relocation much easier and more accessible promises to increase its use, right, and make its issue unfortunately even more prominent. So anyway, since any attempt to curtail VPN use is going to create another big mess, it would be nice if we could just kind of let this one slide by.

But 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." Now, that note linked to a Mozilla post made last Thursday with the great title "The zero-days are numbered." Love that. 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."

Okay, now, let's just pause for a moment. They used Opus, Anthropic's Opus 4.6, and gave it Firefox's source, and it found, identified, 22 security-sensitive bugs, which it fixed in Firefox 148. Okay. Opus 4.6. They continue: "As part of our continued collaboration with Anthropic, we had the opportunity to apply an early version of Claude Mythos Preview to Firefox." Okay, now, this is the same Firefox which a couple months before had found 22 vulnerabilities. Looked clean then; right? Problem solved.

They wrote: "This week's release of Firefox 150 includes fixes for 271 vulnerabilities identified during this initial evaluation." They write: "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."

Okay. There is 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 their 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." Here it is. Listen to this. "Computers were completely incapable of doing this a few months ago, and now they excel at it." Wow. "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."

And all I have to 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. Firefox is already hyper secure. I mean, security is its byword. It is about security. 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 not a single vulnerability. I think this is the most significant statement made from the front lines of the security battle that we have 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 magic fairy dust, or that it's even possible to have it. 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 crypto. 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 user base. 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." Again: "So far we have found no category or complexity of vulnerability that we can find that this model cannot."

Now, I understand that this is attempting to prove a negative; right? 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've never found one. All by itself that's a hugely significant statement. But what's also significant is that this is not Anthropic saying this about their own creation.

Leo: This is not marketing; right.

Steve: Right. This statement from Mozilla, it's from Mozilla about Firefox, is as much from the front lines of mission critical vulnerability discovery as it gets.

Leo: You know what's interesting about this, the other capability that's really key with Mythos, with getting back to memory, is that it is able, it has such a large context, apparently, we don't know the details, we haven't seen the system card, but it's clear from the output because it can chain exploits. And the thing that's been holding previous models back is they could come up with individual exploits, but they didn't have sufficient context to say, okay, now I've got this. Then what? Then what? And that means it has enough memory and enough context to chain those exploits. And I think that's one of the key capabilities of Mythos, just reading between the lines a little bit.

Steve: Yeah. Right.

Leo: It's very interesting, yeah.

Steve: So for what it's worth, six weeks ago, during our April 14th podcast #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. You know, it was like, eh, well, they're about to have an IPO, blah blah blah. Anyway, once I understood it, the podcast got the title "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.

So is it going to be a tsunami? There's something I see missing from some of the public discourse over automated AI vulnerability discovery. Like this guy from Cisco. For example, at the end of that recent posting by the Cisco guy about the strain AI vulnerability discovery is placing on the aging CVE system, right, we all remember what I said. I'm repeating it here for some reason. You know, he was saying that they chose not to release it publicly, 27 years of remarkable achievement in the CVE system, but it's

going to be completely overwhelmed and wiped away, you know, we need something completely different.

But Mozilla turned Mythos loose on their Firefox codebase and, yes, uncovered 271 previously completely unknown or unsuspected vulnerabilities. This blew their minds. But they recovered from what they described as the vertigo that this introduced, brewed fresh pots of coffee, called their significant others to explain that they would 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 all resolved.

So here's my point. What's going to happen now, 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 with Firefox's codebase which it had acquired over its entire lifetime. All of those are gone now. And while mistakes and regressions can happen, they're 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. And 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 again with a discoverable vulnerability. As the guy from Mozilla himself said, a zero-flaws future is now possible.

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 single month sees a hundred or so bugs resolved in Windows and Microsoft's other software suggests to me that they are likely facing a great many thousands, tens of 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 even dramatically more powerful Codename MDASH system. Can you imagine what a catastrophic disaster they must be discovering for themselves?

They shared the critical vulnerabilities discovered and resolved in the first and most obvious place they looked, which was Window's TCP/IP networking stack, a critical component that likely received a lot of prior attention. As far as they knew, just like Firefox, they thought the TCP/IP stack was bulletproof, problem-free. Four critical remotely exploitable problems found.

So those networking stack bugs, which were somewhat obscure, I mean, like I worried, do I have a problem because I'm using Windows Server? No. I'm not using any of the weird stuff, the edge cases that that could have leveraged. But even so, they 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, whatever that is. I'm sure that's what they're doing right now, and that they got 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 get to 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 provably Swiss cheese, it will be quite difficult to get any sleep until every one of those problems have been resolved because any one of them could be exploited at any moment.

The source of all this angst is the awareness that, especially for an open source project like Firefox, or Linux, or 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's almost a certainty of a very limited window of opportunity to get those problems resolved, to push updates out into the field and into users' hands so that the flaws are pushed out of their hands before 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 against us.

But the crucial point to appreciate is that this will be a transient event, once that debt has been repaid, just as Mozilla did. The entire industry will again settle down to a new, hopefully boring norm, with the world never being the way it has always been until now. 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 then 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 somewhat sad state of 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 not go away, or at least they will require a different form of AI-aided remediation, which I spoke about before. But the exploitation of vulnerable software finally has the chance to come to a well-deserved end. We are entering - we are in a new era.

Leo: And I love this notion that it isn't going to be this way forever. We are eventually going to perfect this stuff.

Steve: Firefox is fixed, Leo. It's done.

Leo: Wow.

Steve: It's fixed.

Leo: Really?

Steve: Yeah.

Leo: Wow. And I think the - I mean, admittedly, as they said, Rust doesn't fix everything. But the choice of tools that don't allow buffer overflows and some of the dumber obvious mistakes, you know, type miscasting and stuff, will really help, as well; right? We're going to use better tools now, too.

Steve: Yeah. So certainly, you know, engineering using prudent known systems is a good thing. But as they said, we've got decades of C++. We can't rewrite it.

Leo: Right.

Steve: There's just too much there.

Leo: Right.

Steve: So they turned Mythos loose on it, and it said, uh, guys, you know, here's 271 problems. I'm sure they fell off their chairs. I mean, they thought it was fine. Like, no problems. We shipped out 148. We fixed 22 things that Opus 4.6 found. We shipped out 149, fixed a few more things. Then along comes Mythos and wham, 271.

Leo: It is amazing. It is amazing.

Steve: And if in fact Microsoft has further advanced the state of the art, sounds to me, based on what we talked about last week, that Codename MDASH, and I hope they give it a good name - I don't care what they call it, but just not - don't make me say Codename MDASH every time.

Leo: Mythos is a good name. It's a little, you know...

Steve: Oh, it's a great name.

Leo: Yeah.

Steve: That's wonderful.

Leo: Maybe Microsoft will call it Zeus or something.

Steve: And so far Daybreak is just a daydream.

Leo: Yeah.

Steve: So, you know, it's like OpenAI got caught with their AI pants down.

Leo: I think what's really going to happen by the end of this year is that all the frontier models will be good enough to do this kind of stuff.

Steve: Yeah.

Leo: And so then it won't be anything special. It'll just be the way it is.

Steve: At the rate we are running, and the amount of resources that are being poured into this, it's going to be astonishing. But this really - this is the thing that has the chance to end software vulnerabilities. That doesn't mean you can't leave a management port open on your [crosstalk].

Leo: There's still human error.

Steve: Yes.

Leo: There's also still things like Rowhammer; right? There's hardware flaws that cannot be fixed. Yeah.

Steve: Absolutely. Absolutely. So this is just - it's not like we need to say, okay, well, we did the Security Now! podcast until there were no more problems.

Leo: No, we'll still be doing the show.

Steve: We have plenty left to do. But it won't be like some random buffer overflow or authentication bypass where you do something the developer never expected and cause a problem somewhere, flip some state, and then inject a file, and now you're in. That's going away.

Leo: That's an important distinction. Because when we first were talking about Mythos, that was right about the same time that the Claude source code got accidentally released. And Paris said, "Well, wait a minute, if Mythos is so good, how come that happened?" They said, "Well, you can't fix stupid. I mean, you're going to still have humans in the loop." There's nothing Mythos can do about that, unfortunately. Really good stuff, as always, Steve. I love this show. We look forward to it every Tuesday. And all the geeks are just going, yeah, Tuesday's here, it's time for Steve.

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/>

