ECDSA'S CURVES ORIGIN

Description: Another record-smashing Patch Tuesday. But again, all those updates were not without consequences. Palo Alto Networks' Unit 42 reports on an agentic attack. A mass-compromised hardware cryptocurrency wallet attack. How to create your own known-safe crypto wallet. Feedback about AI, coding, and SpinRite from a listener. What the reverse-engineering of the ECDSA curves means for security.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-1101.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-1101-lq.mp3

SHOW TEASE: Hey, it's time for Security Now!. Steve Gibson is here, and I'm subbing in for Leo Laporte. And we talk about, well, another record-breaking Patch Tuesday, plus Unit 42 reports on a really wild agentic attack where the agent decided to leave a lot of its evidence behind. A mass compromised hardware crypto currency wallet attack. And what happens when you reverse-engineer ECDSA Curves by way of Claude? All of that coming up on Security Now!.

MIKAH SARGENT: This is Security Now! Episode 1101 with Steve Gibson and me, Mikah Sargent, recorded Tuesday, 20th of October, 2026: ECDSA's Curves Origins.

Hello and welcome to Security Now!. This voice you're hearing is not one of Leo's AI-generated voices. It is, as far as you can tell, the voice of Mikah Sargent. That's me. And I'm filling in for Leo this week on Security Now!. This is of course the show where world-renowned security expert Steve T. Gibson gives us the lowdown, the update, and the in-between on all things security. How are you?

Steve Gibson: So this is the first of, what, three or four of these that...

MIKAH: Three. At least three, yeah.

Steve: Yeah. Well, cool.

MIKAH: I'll have to check their calendar because, you know, with all the flooding that's been going on over there, they could come back early.

Steve: And Leo was concerned about, like, yeah, like I think the hotel where they had some reservations, they have actually different kinds of reservations about being able to have their stay there.

Anyway, this is - last week, or I guess the week before, oh, no, last week was 1100. We crossed into Episode 1100. It's bugged me that I've always been a little bit behind the big TWiT podcast. I don't think there's any hope of catching up because we're staying at the same distance that we normally do.

MIKAH: Yeah, you're like neck and neck. If there was some way, if I could figure out a way, Steve, for you, I would.

Steve: Anyway, Episode 1101 today for this October 20th. And a bunch of interesting stuff to talk about. We've got to follow up on last week's Patch Tuesday because this one smashed - smashed, I think, is fair - another record which was smashed last month. This one even more so. The good news is less downstream consequences. Last month more than a thousand patches resulted in a lot of breakage of things that worked before you applied the patches. And so Microsoft did a bunch of backpedaling and workarounds and things.

And in fact, this Tuesday, last Tuesday a week ago, this Wednesday's Patch Tuesday also served to finally provide some permanent fixes to the things they broke. This one didn't break so much; but, boy, did it have a number. So we're going to talk about that. Also Palo Alto Networks Unit 42, and I should say right upfront because I'll be talking about Palo Alto Networks at various points during this podcast, is a sponsor of the TWiT Network.

MIKAH: Yes.

Steve: So keep that in mind. They've got, and I love that their security, cybersecurity unit is Unit 42, of course.

MIKAH: Oh, very clever.

Steve: Because that's the answer to everything. They provide a really interesting report on an agentic attack on an unnamed enterprise. You know, their identity is just being kept on the QT for the sake of the enterprise's integrity. We learn a lot from what they share. And it moves my own needle of expectations for the future, which is what we'll get to talking about here toward the end. There was also a really interesting and actually frightening large hardware hack of physical crypto currency wallets, which resulted in a lot of loss of money by those who are using it. We've got some - we have some interesting takeaways from that. Also, in fact one will be what to do to create your own known safe crypto hardware wallet, if you want one.

I'm going to share some feedback from a listener. And then I want to talk about, and this feeds back into this Unit 42 Palo Alto Networks attack, about some other news that Anthropic shared with the industry last week, where they were able to get an unreleased model of Claude to reverse-engineer and determine the never-before-known origin of some crypto parameters which the industry standardized on without knowing where they came from, beyond knowing that the NSA gave them to us. And there was already reason to be suspicious because there was another instance that we talked about on the podcast many years ago regarding a sort of off-the-map, why would you choose to use this random bit generator which the NSA reportedly, reported by Reuters, paid RSA $10 million to make their default.

So, okay. Anyway, I think a bunch of interesting stuff. But the point of that last story about this reverse-engineering is I need to share with everybody what this Anthropic AI agentic thing did in order to solve this longstanding problem which has stumped everyone because it suggests that we're probably in for more trouble when the bad guys get a hold of this kind of power than I've been suggesting until now. So I think a really good, interesting podcast for everybody.

MIKAH: Absolutely.

Steve: And after we take a break we have a great Picture of the Week.

So I gave this picture, which was provided by one of our listeners, the caption "It's not only AI that takes instructions literally." Anyway, what we're seeing onscreen, thank you, Benito, for those who aren't looking at the video, there was some sort of a utility box, big utility box stuck on the side of a crosswalk pole. And in small print in the middle it says "Post No Signs." Which would be pretty clear. However, somebody thought, okay. They printed the word "No" in quotes in very large, in four different fonts, creating four signs, and stuck them all over this thing. So in fact they followed the rules; right? Or the command. They posted "No" signs.

MIKAH: I love it. Post "No" signs. And, I mean, it checks out, I've got to say.

Steve: And imagine encountering that. I think that'd just be so cool to be walking along and to see that thing and be initially puzzled. It's like, uh, what? And then go, okay, I get it.

MIKAH: And you look closer, go "Oh, clever, very clever." Good stuff.

Steve: Okay. So as our listeners know, I believe that the rate of security vulnerability discovery and patching is destined to first accelerate, as we've all been witnessing; but then to inevitably decelerate as that legacy of vulnerability debt is increasingly paid down. You know, even Microsoft cannot possibly be creating new vulnerabilities at the rate they're now able to find and eliminate them, thanks to their MDASH agentic AI system, which they've been applying to great effect.

So without access to what's actually going on inside of Microsoft, we have no way of judging, like, you know, where they are along this process, you know, when they're going to hit, like, that point of peak patches, where we will then begin to see the rate drop off. I sort of thought maybe after last month's more than a thousand patches, it's like, okay. And that was - that followed several months of, like, 400. I remember well, earlier in this podcast, when 25 would have been, like, okay, well, we fixed 25 problems this month. It's like, okay. Maybe 47 would have been a biggie. Well, okay. We've of course blown all of that away. BleepingComputer places this October month's Microsoft update count at an astonishing 1,411 problems.

MIKAH: Wow.

Steve: Now, there's a little variation. I saw 1417, or 1414, depending upon who, I mean, the problem is when there are so many, you lose count. I mean, it's like, you know, what qualifies? There's a little bit of a gray area. But still, you know, way more than we had last month, and last month was a lot. So anyway, before we - I do want to say bravo to Microsoft; right? I want to go on record again with a loud praise. You know, they finally have the tools in this MDASH to enable them to clean up their massive base of legacy code. Until now, they've been at the mercy of attackers like Nightmare Eclipse, who we've talked about all through the early part of this year, who were individually able to invest the time and resources needed to scour Microsoft's code looking for individual exploitable vulnerabilities and then develop an exploit for them.

And yeah, Microsoft would fix any problems that were discovered and shown to them, you know, which is what led in the old days to those 25 or 47 patches. But Microsoft's massively agentic MDASH system allowed them finally to go on the offensive; and; boy, have they ever.

As we saw last month, with its mistakes that were also made, fixing more than a thousand flaws can itself introduce new bugs. But even those tend to be short-lived; right? I mean, they immediately raise alarms. People say, hey, you broke this, or you broke that. And then they come up with a workaround or whatever. So certainly the

overall result is a significantly more secure operating system, and related software, everything from Microsoft that we've had before. And if that's true, if getting rid of these vulnerabilities makes things better, then, boy, did Microsoft software get a lot more secure this month.

Last week, BleepingComputer wrote the following. They said: "Microsoft's October '26 Patch Tuesday security updates addressed a record-breaking 1,411 flaws, including one actively exploited and two publicly disclosed zero-days. Patch Tuesday addressed 70 'Critical' vulnerabilities, 57 of which are remote code execution, 11 of the other Criticals were elevation of privileges, and two were spoofing." So if we break down the vulnerabilities by category, there were the greatest number overall. 677 were elevation of privilege vulnerabilities. And although, as we've talked about before, those don't seem as, like, horrifying as remote code execution because who wants to have your computer executing code that some cretin managed to sneak into it.

Still, elevation of privilege matters because the way Windows security hierarchy is set up, most users do not run with admin privileges. And if something happens in your computer while you're logged in because of something you do, the bad guys have code running with your privilege, meaning restricted, not root, not admin. So elevating that privilege to full access in order to do much more devastating permanent damage is crucial. 677 problems were found and fixed just last Tuesday that allowed exactly that.

MIKAH: Can you imagine the sound, the collective sound of winces that - I can't imagine being one of these folks seeing these and going, that's been in there for the past 15 years, and we have to hope that no one exploited it. And, boy howdy, how did we miss that? I can't imagine.

Steve: Yes. And it's funny, too, because from Microsoft's standpoint, while fixing 1,411 flaws, they're acknowledging 1,411 flaws.

MIKAH: Exactly.

Steve: On top of the more than thousand last week, or last month. It's like, so that's why I want to be so clear that it's like, go, Microsoft, don't be shy.

MIKAH: Yeah.

Steve: Just, like, you know, the license agreement already holds them harmless; right? It's like, well, if you don't like Windows, you can send it back. But, you know, you're using it and taking all the risk for doing so because we just sort of tried, you know, and did as good a job as we could.

MIKAH: Yes.

Steve: So it's not like they're liable for any of these problems. And the fact that they're getting it better. What I really love, because I have not yet seen any compelling reason to go to 11, I'm happily putting 10 wherever I am setting up a new machine. Because of the fact that so much of the world wants to use Windows 10, does not want to go to 11, or has older hardware that won't run 11, they're stuck at 10, but Microsoft, even though they've formally discontinued support, they keep adding a year and a year and a year to it. What I love is that 10 is being really well fixed as a consequence of all this. I mean, these...

MIKAH: That's fantastic.

Steve: And because they've moved on to 11, they're not screwing it anymore. They're only fixing it. So by the time they finally stop patching 10 - and, you know, we have another whole year; right? As we were approaching this October, this month, they said the month before, oh, we've decided we're going to give you another year. Great. Because we're going to see this curve of vulnerability fixings finally dropping. Maybe it'll be next month. Maybe it'll be the month after. We're not sure when. But it's going to happen where they're going to start running out of them because, you know, there has to be an end. And 10 will get all of the benefit of that, and none of the liability of any of Microsoft's new crap. So it's wonderful. I mean, it's just a win for Windows 10.

Anyway, as I said, 677 elevation of privilege. The second most occurring class of problems they fixed was 317 remote code execution vulnerabilities. In other words, they fixed last week 317 different ways that bad guys could run code in your computer without you wanting them too. Which is like, as you said, you know, it's bracing to think that until last Tuesday, there were 317 ways to do that. Yikes.

MIKAH: Yeah.

Steve: So, yeah. 254 information disclosure vulnerabilities, 73 denial of service which mostly just now that's what we call it when you're able to crash something because you deny everybody else its service because you crashed it. So 73 ways to crash the DNS server or whatever. 40 security feature bypass vulnerabilities, which has always seemed sort of nebulous to me. It's like, well, okay, that's kind of like everything; right? Security feature bypass. And then 28 spoofing vulnerabilities.

Anyway, BleepingComputer said that: "When we report on Patch Tuesday security updates, we only count vulnerabilities released by Microsoft on Patch Tuesday itself." In other words, there were more. They said: "Therefore, today's total excludes 348 flaws fixed earlier this month, including vulnerabilities in Azure API Center, Azure App Service, Azure Event Grid, Azure SRE Agent, Mariner, Microsoft Bookings, Microsoft Dataverse, Microsoft Exchange Server, and Microsoft Partner Center." And normally you would see things like Exchange Server fixed on Patch Tuesday. The fact that they did it sooner probably means it was really bad, and they couldn't figure they could even wait until Patch Tuesday. On the other hand, it would have bumped the count up, and so maybe they were thinking, okay, we can't do more than 1,411 all in one sweep.

So they said, BleepingComputer said: "This is a record-breaking" - no kidding - "Microsoft Patch Tuesday as Microsoft continues to use artificial intelligence platforms to find and remediate security vulnerabilities in its products. Microsoft also said: 'Today, October 13th, 2026, Windows 11, version 24H2 Home and Pro, and Windows 10 Enterprise LTSB (Long-Term Servicing Branch) 2016 have reached the end of their updates.'" Of course that doesn't mean they're going to stop updating them because they can't. "'The October 2026 monthly security update is the last update available for these editions.'"

However, you know, they're still updating Windows 10, so they're clearly going to give you that. It's not in the normal standard Microsoft Patch routine. It's in the extended services mode, which is what they keep deciding to extend. Every time it gets close to ending, they add a year to it because so much of the world is still using Windows 10. Well, obviously a lot of the world is using Windows 11 v24H2. So it's not like they're going to cut that off either, especially when they're fixing thousands of problems every month.

So BleepingComputer said: "Earlier this month, Microsoft also said organizations running Windows 11 23H2 Enterprise, Education, and IoT Enterprise editions should upgrade to newer versions as they'll no longer receive security updates after next month." And yeah, why not have, why not update to the - unless you for some reason can't - update to the latest Windows 11. Why wouldn't you?

They said: "This month's Patch Tuesday fixes one actively exploited and two publicly disclosed zero-day vulnerabilities." And they noted that Microsoft classifies zero-day flaws a little differently than the rest of the world. They call them, you know, we call them when they are caught being used on some software, we call that a zero-day because it's really a surprise. "Microsoft classifies," they write, "zero-day flaw as publicly disclosed or actively exploited." So in other words, you don't even have to be using it if someone mentions it in public. It's like, that's a zero-day because it surprised them. So they said: "The actively exploited zero-day vulnerabilities addressed during this October 2026 Patch Tuesday are," and there are three of them.

There's a Windows Shell Security Feature Bypass Vulnerability. I'm not going to go into the details. There's a .NET Framework Security Feature Bypass Vulnerability and a MessagePack-CSharp Denial of Service Vulnerability.

And they finish by saying: "With this month's record-breaking number of vulnerabilities, Microsoft has also introduced a new method for organizations" - I thought this was interesting - "to prioritize security updates by grouping related flaws into what it calls 'risk cohorts.'" Brand new idea for the first time this month. They said: "This new approach groups similar vulnerabilities together to help organizations better understand which security flaws pose the greatest risk to their systems and should be prioritized for patching."

So I think this is a welcome and actually it's an understandable response, and surprisingly quick, to last month's self-inflicted Windows Update carnage, which as we covered four episodes ago, the week after Patch Tuesday, it seriously impacted many enterprises, in many instances for many days. So this new concept of these risk cohorts is a welcome addition. And especially when an enterprise now is facing an astonishing number of updates. 317 of the 1,411 allowed for remote code execution. So we know that the bad guys examine both the before and after patched code to locate the changes that Microsoft made as a means of attacking those who are slow to update.

So reorganizing the massive patch collection, which now is being dumped every month into a collection of so-called "risk cohorts," will potentially, at least, allow concerned enterprises to verify the safety of the more serious problems first - safety as in is it safe for us to apply these in our enterprise - and get those pushed out to their enterprise's workstations.

BleepingComputer did break this down. They said: "For October 2026, Microsoft published 15 risk cohorts overall." And then they listed just five. They said, including: "136 vulnerabilities which affected Windows file systems, storage, and backup; 135 vulnerabilities involving Office documents and mail clients; 131 vulnerabilities affecting Windows graphics, media, and content processing; 109 vulnerabilities involving Windows core and kernel privilege boundaries; and 103 vulnerabilities affecting Windows installation, deployment, and update services." And as I think about this again, looking at these, it feels like they've divided them sort of like departmentally.

MIKAH: Yeah.

Steve: Like there will be some group of people in the enterprise who would take those 136 vulnerabilities affecting Windows file systems, storage, and backup, and check those because that's sort of their bailiwick. You know, they're those people. Because, for example, then we have the 135 involving Office documents and mail clients. Well, that might be a different group in a large enterprise that is like in charge of the Microsoft Office stuff. So by segregating them that way, it does allow, you know, it divides this huge number of vulnerabilities into groups by focus group, essentially. You know, who would generally handle making sure that, you know, we haven't broken something in the

installation, deployment, and update services, you know, where 103 vulnerabilities were just patched.

So anyway, BleepingComputer says that: "Microsoft stressed that these groups do not necessarily indicate which flaws are the most dangerous" - right? - "as the actual risk depends on whether they're being exploited in attacks, their severity, and how an organization uses the affected software.

"Microsoft says the new approach is still experimental and is meant to help organizations" - see, they say "prioritize." This doesn't feel like priority to me. It feels like, you know, segregate for different departments to handle them, if that's their expertise.

MIKAH: Yeah. Delegate.

Steve: Right, right. So anyway, that's where Microsoft stands. Are we still accelerating? Are we going to have more next month?

MIKAH: Even more?

Steve: Whoo. Will November, yes, right, see an even higher count? Or was this month the month that we peaked, and we will now see what will probably be - my guess is when we hit the peak, we're going to see a sharp drop-off as the pool of unpatched vulnerabilities dries up. I'm on record. We'll see if I'm right or not. So but what's equally exciting to me is the idea that all future Microsoft software will now be able to be prescreened by that same MDASH AI which is fixing the legacy code. They should be able to prescreen anything before it's ever released.

In other words, there's the real potential for Microsoft's systems to be truly secure for the first time ever and for vulnerability discovery, exploitation, and post-release patching, to become an historical artifact. Like where, you know, when I was a boy, you know. It's like because there just aren't these problems anymore because AI cleans them up even before they get out the door in the first place. And wouldn't that be something.

For what it's worth, Microsoft is not alone. An increasing number of companies, as we've noted previously, have been synchronizing their own software updates with Microsoft's Patch Tuesday. I don't know if it's like to hide behind Microsoft or what.

MIKAH: Yeah.

Steve: You know, it's like, oh, well, Adobe did this and so forth, and Cisco did that. But that's been growing more and more popular. BleepingComputer also noted the recent updates from other enterprises, who also appear to be employing AI for their own preemptive defense. Yay. I mean, we need that to happen. We know that Cisco is. Last week we covered, or a week or two ago, we covered Cisco's massive patching binge which, again, props to Cisco, thank you for fixing all this.

They apparently sorted this alphabetically, BleepingComputer did, saying: "Other vendors who released updates or advisories in October 2026 include Adobe," you know, the A word, "released security updates for numerous products, including Commerce, Substance 3D Samples, ColdFusion, Photoshop, Dimension, and Framework. Atlassian released security updates for a critical vulnerability in multiple self-hosted Data Center products, including Confluence, Jira, and Bitbucket. Attackers soon exploited the flaw after a Proof of Concept was released. That was the fix for the critical vulnerability in Atlassian. Cisco released security updates for numerous products, including critical vulnerabilities in its NX-OS data center network operating system that can let attackers run arbitrary code with root privileges on Nexus switches."

Citrix released security updates. I'm not going to go through them in the same level of detail. Dell was there. Docker patched a critical flaw in Docker Hub. Fortinet released security updates for numerous products. GitLab released security updates for critical flaws in its AI Gateway and CI/CD Regular Expression Compiler. Ivanti is there with updates. Kiteworks security updates for 126 vulnerabilities. AI's on the case there. Progress Software released security updates for critical and high security vulnerabilities. SAP released the October security updates, addressing 20 security flaws. And SonicWall released security updates for a critical SMA1000 server-side request forgery (SSRF) flaw.

So anyway, we're seeing industry-wide repair of legacy code, which, you know, it is so welcome and necessary since we already know, and I'll be sharing another example of it shortly, that the attack capabilities now being brought by AI, not tomorrow, but it's already here, are being used to increase the strength of attacks.

MIKAH: Did you see that - I think it was Anthropic who just recently, and now I'm trying to find it because I'm hoping it's not, you know something that I read that ended up not being true. But I thought I saw something about Anthropic working on a tool that was for cybersecurity where you could run its cybersecurity model against your own code in a way that - did you talk about that already? Yeah. That like tested it for the model?

Steve: Yes.

MIKAH: Got it.

Steve: We covered it last week.

MIKAH: Understood.

Steve: The release was that they formally said they would be running open source software projects that had opted in and that were large enough to make an impact if they had vulnerabilities at no charge. They would be scanning and keeping an eye on any open source projects that wanted to opt into that. So yeah. I mean, and that's great because, you know, the enterprise is able to afford AI, and this is what we just - we're seeing all these enterprises obviously spending now on AI vulnerability discovery. And the result is, you know, much greatly increased rate of patching. So that side is covered. And also notice that the enterprise is the one that has their own source code. It's not open, so it isn't nearly as easy to fix it. Not technically impossible, but still you really need the source in order to fix the software.

Enterprises are closed source, but they've also got the money. Open source projects, you know, the source is open for everyone, but they typically are underfunded, or not funded. You know, it's all volunteer effort. So the idea that we're seeing them step up and recognize that AI - in fact it was the week before last week we talked about them also doing a piece which noted that GLM 5.3 from Z.ai was now just about equal to the frontier in its ability to find problems. And being open weight, it could be obliterated, and so you would remove all of its ability to say no.

So then the week after, which is last week, we talked about how they had then - Anthropic then decided, okay, because the bad guys do have access to open weight models that are almost as good as ours, we're going to give free access to the open source community for big projects that matter to, you know, when they don't have the money to fund AI, we do. So I think that was a really good move.

So far, despite its record-breaking 1,411 updates, the industry has not experienced what I would call an "October Surprise" of update consequences like we did encounter last month after the more than a thousand updates. October's updates, as I mentioned earlier, also finally repaired a number of things that September broke. So it's been nice

to see that we don't necessarily always have dire consequences following a mega-patch update cycle, even though this was even more mega than the last mega one was. I don't know how they made it differently. Maybe they tightened up their pre-release testing, having learned an expensive lesson?

I mean, it was a black eye that so many enterprises were significantly impacted by installing September's updates. And look at the pressure that enterprises were under to get them installed quickly because there were so many that were so bad. And now everyone knows the bad guys are much better at exploiting these problems. So time really is of the essence. And in fact, the week before Microsoft released the September batch, they said get ready, we're going to - we have a big September coming, and a lot of these things are really bad, and you can't wait to patch them. So enterprises didn't, and then got hurt by not waiting. So far, you know, we're only a week downstream from last Tuesday's. But we're not seeing nearly the same level of trouble.

Microsoft did, however, note that some users may have trouble - this is the only one thing that I was able to find - may have trouble turning on Smart App Control after installing this, October's patches for Windows 11 security update. BleepingComputer covered that, writing: "Smart App Control is a Windows 11 security feature that uses Microsoft's app intelligence services and code integrity features and works alongside Microsoft Defender and SmartScreen to block potentially harmful apps or binaries." So basically it's preventing you from running something that could damage you or your system. So Bleeping said: "According to the company [Microsoft], on affected devices running Windows 11 version 24H2 and later, the Smart App Control option is grayed out and cannot be changed.

"Microsoft said in an update to their" - it's the patch for Windows 11 security feature is KB5127648 advisory. They said: "After installing this update, some devices might be unable to turn on Smart App Control in the Windows Security app if it has been turned off. On affected devices, the option to turn on Smart App Control might be grayed out. Devices where Smart App Control is already turned on are not affected and remain protected." So big deal; you know? If that's all they broke while patching 1,411 vulnerabilities...

MIKAH: Yeah, exactly. The tradeoff there...

Steve: Yeah. I'd call this month's Patch Tuesday a huge success. And it sounds like a trivial UI problem and really nothing to worry about. Microsoft did, however, offer a workaround which involves a couple registry tweaks. You need to run regedit or run a command prompt as an admin rights, and then basically set two keeps in the registry, which essentially does the same thing that the UI, you know, click the button would have done. So it's certainly possible if, for some reason, you have to turn that on, even though you had it off before, and October broke the ability to turn it on? Well, if you have to have it on, you can. There is a workaround, and obviously it'll get fixed next month. I don't think there's any great emergency to making that happen soon.

Okay. Next I'm going to talk about this Palo Alto Networks, sponsor of TWiT, Unit 42 investigation, but let's take a break so I can rehydrate, and we'll hear from Leo.

MIKAH: Right. Back from the break, as I mentioned, Mikah Sargent subbing in today for Leo. And it's time to go 42.

Steve: Okay. So I mentioned another example of AI-enhanced attacks. Last month, Palo Alto Networks' Unit 42 published their findings from their investigation into an AI-assisted attack. And their report contains a bunch of useful takeaways which can be readily generalized. So here's what they reported. And then we'll talk about what it means.

They said: "Unit 42 responded to an incident where a human attacker used frontier AI to breach an enterprise network autonomously as part of a ransom attack. The agents breached the company's security layers in a methodical manner, with each agent targeting a different layer of defense to achieve their shared goal. The impact was at the scale of a coordinated effort from multiple red teams, which would normally take human operators around two weeks.

"The threat actor told us in negotiations that they leveraged frontier AI models and attack-specific agentic AI frameworks. By shifting execution to an automated loop, the attacker compressed weeks of methodical intrusion tradecraft (using more than 50 MITRE ATT&CK techniques) into fewer than 10 hours.

"After they gained initial access, the attacker used agents to map the internal architecture, raid source repositories, and seize root credentials. The agents also triggered unauthorized continuous integration/continuous delivery (CI/CD) builds and claimed master keys to the victim's cloud AI infrastructure."

And get this. They said: "What made the attack stand out was AI-assisted operational efficiency, without the need for a novel zero-day or super-elite tradecraft. The attacker left all tactical execution to AI agents that monitored, evaluated, acted and re-planned in real time, increasing the speed throughout the attack chain. And the attacker also directed its agents to leave behind a 'report' on the organization's security posture: an 80-page technical audit detailing dozens of exploited findings." In other words, it ransacked the enterprise; it exfiltrated all of the company's crucial, critical, must keep it private data, which it would then use to extort the company for a ransom; and left behind, thoughtfully, an 80-page audit of everything it found wrong that it used in order to perpetrate its crime.

MIKAH: Well, that's nice. I mean, yeah.

Steve: Yeah.

MIKAH: How kind. I wonder what, like, I wonder where and what resulted in it making that choice. You know what I mean? What would result in sort of vigilante - oh, we're going to get it?

Steve: Well, no. My guess is that this is what we've seen. This is the way agents work, right, is they build their own dossier of findings as they're going through. They, like, leave breadcrumbs behind...

MIKAH: Got it.

Steve: ...for other agents to then read and pick up the thread or go in a different direction. So this was probably being built for their own purposes as they were finding flaws and exploring avenues and pathways. And then they thought, well, what the hell. Let's, you know, format this into a formal technical audit of all the problems that we found and exploited, just sort of, you know, to give them a little, I don't know, goodbye bonus.

MIKAH: It's almost, from our perspective, the Unit perspective, there's a little bit of as kind of a brag, honestly.

Steve: Yeah.

MIKAH: It's like somebody's leaving their little calling card behind. My calling card is just here's where you failed. Wow, that's funny.

Steve: Exactly.

MIKAH: Not funny, but funny.

Steve: So Palo Alto said: "The adversary ran their operation using current AI-enabled software development processes." They said: "We observed three indicators consistent with AI usage. First, large language model calls to multiple frontier AI agents in parallel. Structured Markdown files" - oh, here's what I was talking about - "passing information between agents and sessions." So they use those. And then I would imagine they ended up digesting that into that 80-page report. And then, third, "Custom scripts (assessed with high confidence to be AI-generated due to UI elements)." Which, and those custom scripts were managing the dynamic operations.

They said: "The 10-hour operational timeline included infiltration and mapping. The actor breached a publicly accessible web service" - so that was their way in. Whatever this entity was, and we still don't know, they had a web surface, you know, Internet facing. So infiltration and mapping. "The actor breached a publicly accessible web service to tunnel into the network, deploying an automated recon agent to map internal microservices."

Then secrets harvesting: "Sub-agents combed enterprise code repositories, extracting hard-coded tokens and service passwords." And okay. As I'm reading this, I'm hoping that, you know, CISOs are getting a little chill.

MIKAH: Yeah, exactly.

Steve: Feeling a little chill. The idea of what is in the process of being turned loose on a world that is not ready for it, which is really the theme and the conclusion that I'll be reaching during this podcast. I mean, it is something.

Then privilege takeover. They said: "Using exposed tokens, the actor infiltrated the secrets management system, harvesting master administrative credentials to seize control of root system access." Then pipeline exploitation: "The actor hijacked an enterprise code application via custom workflows to exfiltrate cloud access keys. They attempted to plant backdoors in Terraform configurations, but hard branch-protection controls stopped it."

And then get a load of this one, Mikah. "AI infrastructure hijacking: Using stolen cloud keys, the actor turned the victim's AI endpoints into post-compromise infrastructure using the company's own compute power to perpetrate further moves."

MIKAH: Oh, my goodness.

Steve: So they hijacked their own, the company's own AI.

MIKAH: That's horrifying. That, what you said, the chills weren't already happening, they'd better be happening now. That gave me chills. I'm, ugh, that's terrif- I mean, that is like a true virus; right? Like when you think of a biological virus going in and reprogramming our own cells. That is wild.

Steve: So the attacker's AI, having obtained the keys to the victim's cloud AI, used that victim's own AI against them to facilitate additional post-compromise work. Yikes. Their report, Unit 42's report, included an infographic which I have included in the show notes. It observes that the human orchestrator set the objectives and directed the agents. They reviewed the results and prioritized next steps, and directed follow-up activity as the attack progressed. But all of the actual work was done by these agents.

Palo Alto Networks' Unit 42 then summarized what they learned from their analysis. So here's what they said. "Key Lessons learned from Addressing Agentic Attacks. This incident exposes how an attacker who understands how to deploy frontier AI agents effectively can dramatically speed up the pace of their attack." Remember that they said, if multiple red teams, multiple human red teams, had been employed, they're guessing this would take two weeks of concerted effort by human attackers trying to get in and do damage. This took 10 hours. You know, like overnight essentially, after everyone's gone home, and they come back the next morning, and this is done.

They said: "We assess that attackers will increasingly add AI agents to their tool sets. Organizations should take note of the following to address agentic attacks." They said: "AI agents reduce the time between steps in the attack flow. AI agents in this attack were designed to parse raw tool output and quickly take next steps, speeding up the overall attack flow.

"AI agents leave recognizable indicators." They said: "Defenders can identify agentic attacks by watching for indicators such as the use of structured Markdown, Python caches, and paired asset folders." They said: "Attackers can use AI to establish redundant persistence across the environment." They said: "In this incident, the attacker used AI agents to efficiently establish overlapping persistence across SSH keys, serverless functions, container restart policies, cloud identities, and CI/CD pipelines. Using AI agents can make it easier for an attacker to maintain and test this entire portfolio in parallel."

And finally: "Attackers can use an organization's AI tools" - this is that final part. "An attacker can use an organization's AI tools as post-compromise infrastructure. Attackers can hijack enterprise AI services to assist in their attacks. This allows threat actors to hide orchestration traffic among expected traffic, and offload the financial cost onto the victim." In other words, to pay the cost of the AI's use. And because it's internal enterprise traffic, that's the part about hiding the orchestration. You're not having to orchestrate from outside the enterprise, where you would see traffic crossing the enterprise's boundary. Now you're orchestrating internally. Endogenously, essentially.

So they said: "What about specific defenses?" They write: "Defending against automated agent loops requires matching the speed and adaptability of AI-driven attacks." They said: "Execute synchronized containment. Deploy automated playbooks that simultaneously revoke credentials, terminate OAuth sessions, freeze CI/CD pipelines, and isolate cloud accounts across all operational planes. Govern AI," they said, "as core infrastructure. Inventory every model endpoint, API key, Model Context Protocol (MCP) gateway and AI tool integration. Apply strict rate limits, least-privilege policies, and diagnostic logging.

"Detect behavioral loops: Hunt for operational loops including bursty API requests, rapid 401/200 HTTP state shifts" - meaning something is trying things that don't exist and then succeed, meaning something was guessing URLs until it finally got the right one. They said: "Parallel authentications and sudden model usage from unexpected identities. And lock down DevOps pipelines. Enforce mandatory, multi-party code reviews and immutable branch protection on all infrastructure-as-code repos to block automated backdoor injection."

Okay. So stepping back from that itemization, basically it's saying lock the eff down. I mean, like really, just like those are all kind of like common sense things. And you can imagine that it'd be like a wish list. It's like, well, yeah. We'd love to do that.

MIKAH: Right.

Steve: But who has the time? It'll inconvenience our developers. It'll mean, you know, multiparty code review. Well, that's going to slow down the reviews if we have to have multiple parties all weigh in. Well, yes. That's the idea. Except it's also an inconvenience. So what's really the case is nobody's doing all these things today. That's not in place because it really does get in the way.

So I'm not in charge, thank god, of the security of any large enterprise, and I'm glad. I'm more glad for that than I've ever been. Because this feels daunting.

MIKAH: Yeah, it does.

Steve: We focus a lot on coding errors that lead to vulnerabilities because they are concrete, they are understandable, and they're actionable. There's a bug. We found it. It can be exploited. Let's patch it, and it's gone. So, you know, case closed.

But the reason I'm becoming increasingly uncomfortable about the capabilities we're seeing emerging from these new, and they are new, agentic attacks is that they are far less dependent upon the presence of explicit bugs than they are upon a general systemic laxity of absolute organizational and network security. The description of this particular incident reveals the sort of environment that likely exists within most, if not every, organization. It's just natural; and we're just human, after all.

Over time, with the changing of the guard, projects get abandoned; old code or test credentials get left behind in long-forgotten directories. There's debris accumulated. Trails of breadcrumbs accumulate everywhere. You know, with this or that permissive firewall rule left in place after the need for it has expired or ended, with an ex-employee's credentials left active, and who knows what else. It's nothing overtly bad or glaring, just the inevitable consequence of people who are in a hurry, who have better things to do than always without fail rigidly enforcing every policy that was once upon a time stated.

And into an organization like that, into reality of today, we drop a powerful, goal-oriented, goal-directed, tireless and merciless AI entity that will literally do anything and everything to accomplish its goal. It's capable of inventorying every online asset within the organization, checking version numbers, testing for any patches skipped, scraping the enterprise's website to obtain a roster of public-facing employee names, then digging into their public personas to obtain any detail about them that might hint at their password credentials. It would rifle through any source code that it was able to find. And in the case that Unit 42 talked about, it did that, either internal source or external, sourcing for tidbits and hints, anything it might be able to use to further its goal.

It would read through everyone's past email, years or decades of it, whatever is available. It would learn of relationships and responsibilities and look for any clues that would allow it to advance its mission. It would learn, you know, like pick up, for example, a helpful reminder Slack password that was once passed between employees through secure internal email, but still in an archive somewhere. Then, using Slack's search and API, would sweep through years of Slack history, soaking up every bit of useful intelligence it could find.

We're talking about turning an extremely resourceful goal-directed berserker loose inside the network of a large enterprise, an enterprise that's never needed before to secure itself against any such threat or intrusion because no such intruder has ever existed before now. This agentic berserker will have already been trained to encompass all published human knowledge, including all known hacking tricks and techniques. It's fast, inexhaustible, and clever, by virtue of being willing to try everything and anything. October's Patch Tuesday repaired 1,411 new vulnerabilities on top of the more than a

thousand from last month because agentic AI has become that capable. No one wants an unconstrained AI turned loose inside their enterprise.

Am I exaggerating or overstating agentic capabilities? Not from everything we've already seen. We've all seen what agents that were deliberately constrained and were not supposed to get loose, managed to do. I hesitate to call them "clever," but the reports of their actions as they have, you know, as we've seen over and over, they've found a way in, those are surprising and should chill the blood of any CISO.

Shortly after this summer's Black Hat conference, the stock prices of CrowdStrike and Palo Alto Networks - two companies that specialize in protecting enterprise networks from intrusion - their stock prices moved rapidly upward and have continuing to climb. Why? Because the world's enterprises began to recognize just how very real these agentic AI threats have become. And they're not wrong.

Today's main topic, that we'll be getting to, is the story of how an agentic AI spent just 30 minutes to reverse-engineer and discover the original source of the globally standardized curves used by the elliptic-curve digital signature algorithm, ECDSA. The origin, as I mentioned at the top of the show, of those specific curves - those specific curves, beyond the fact that they came from the NSA has remained a mystery (it is a mystery no more) until an AI was asked to figure out where they came from. My point is, the story I'm going to share is truly jaw-dropping, and it's the reason I'm beginning to despair over the future security of any large and lumbering old-school enterprise. If this is what AI can do, how could anyone be ready for what's coming next? And on that cheery note...

MIKAH: Yeah.

Steve: ...let's take another break.

MIKAH: Let's let that one sit there for a minute because that's - that's chilling. Oh, boy. And it's got to be fighting fire with fire, the AI versus AI, but it's still like...

Steve: I think that, I mean, I'm serious when I say that I don't think any enterprise is prepared because, you know, you secure your borders. You trust your employees. But they will, someone will say, hey, what's the password for our something or other Slack channel. And it's like, oh, I'll email it to you.

MIKAH: Yup.

Steve: So it's employee to employee, but it now exists in that enterprise's email archive. And when the AI gets in there and reads it all, it's going to go, oh, great. Now I have the Slack password. It's able then to use Slack's API to dump the entire history. And lord knows what intelligence it's going to find from ad hoc discussions and meeting minutes and notes and everything that the company feels, you know, is internal.

MIKAH: Oh, the meeting minutes and notes because now everyone's using those AI transcription services for every single thing, and other multiple places that are capturing your trans- oh, my goodness, I hadn't even considered that aspect of it.

Steve: And remember that AI will tell you what it knows about you. So once it gets a hold of the enterprise's AI infrastructure, it can ask the various AI agents to dump all of their accrued knowledge of the people that used this AI, all of the prompt text can be retrieved. And so it'll immediately know everything that the enterprise's AI has learned about all of the users of it.

MIKAH: Holy...

Steve: Oh, I mean, it is beyond bad.

MIKAH: I feel that everyone is talking about, or that a lot of - we hear a lot about people talking about what's this side of things, and talking about all the exploits that are happening and that kind of thing.

Steve: It's just focused on vulnerabilities. It's like, well, let's get all those fixed. Sorry. That's not going to do the job.

MIKAH: Wow. Wow. I hope somebody's thinking about it. I mean, you know, actually actively doing something about it. But what can you do?

Steve: Well, and for example, CrowdStrike and Palo Alto Networks have endpoint security, where they put devices in the enterprise that are AI enabled, which is supposed to be watching the network for signs of this. But, you know, not everybody's going to use them. There are a lot of people who are like, you know, they don't have the budget. It's not free to do that.

MIKAH: Yeah.

Steve: And so - and they just don't see the need. It's like, hey, everything's been secure so far. Well, there's a storm coming.

MIKAH: Yeah. Oh, my god. All right. Welcome back. We are of course sharing some cheery news about the future of security in the enterprise and elsewhere, thanks to the wonderful world of AI. Steve? Take it away, friend.

Steve: Okay. So the number one takeaway, I think, from the news of what's being called the Ledger CryptoBilis hack, is to be excruciatingly careful about the use of any physical hardware crypto wallet. There would be so much pressure, when you think about it, to plant a backdoor into a hardware crypto wallet, either by an unscrupulous vendor - hopefully there are none; but, you know, the world - or after shipping somewhere along the retail supply chain, that it's almost impossible to feel comfortable using such a device unless I guess you really have made it yourself, and there are some options for that we will talk about in a second. But how would anyone ever know one way or the other whether the device they were using was clean?

So first of all, I fully understand the strong appeal of a dedicated hardware device for storing cryptocurrency secrets. The idea being that it's mostly disconnected from the outside world which is what we definitely want until and unless and only when its secrets are needed. The trouble is that there are multiple points of vulnerability both any device's hardware and its firmware so the provenance of any such device really must be known and trusted.

And that brings us to this next story which Bitquery posted on Friday under the headline "The Ledger CryptoBilis hack took" - get this - "$102 million from 335 wallets." They said: "One thief had all the keys." And their subtitle was - or their tease - "On Friday the 9th of October, people who had bought Ledger devices from a seller in South East Asia found their crypto gone." They said: "We followed every drained wallet on six chains to see how the thief worked and where the money sits now."

So with that, they continued, writing: "Weeks after setting up a Ledger, the money was gone. You buy a Ledger and move your savings onto it. Weeks later you open the device. The balance says zero, and you never sent a thing. That's what Ledger owners in

Indonesia, Malaysia, and the Philippines found on Friday, the 9th of October. And before I go any further, I should also mention the other news. Compromised devices were also found by mainstream German resellers. So it's not just East Asia. Ledger says they had purchased their devices from CryptoBilis, a shop that sells them in the region which news reports describe as an official reseller.

"Ledger is a French company that makes hardware wallets, small devices that keep the keys to your crypto away from the Internet. On the same day it said it was looking into losses among users in Southeast Asia who bought from CryptoBilis. It asked the reseller to stop all sales" - that is, Ledger did - "stop all sales, told recent buyers not to set up their devices, and told anyone who had to 'consider moving assets to a new Ledger signer using a new seed.'

"By then," they wrote, "two on-chain researchers had posted the thief's crypto addresses. Tanuki42 listed eight of them, and Specter listed 10, with losses exceeding $86 million." So what they're saying is that they found - let's see, eight and 10. So they found there a total of 18 cryptocurrency addresses that had transferred the money from these Ledger-based wallets into the bad guys' accounts totaling $86 million.

They said: "Bitquery records every transfer on the chains involved, so we followed each drained wallet and each address the money reached. Our count is $102 million from 335 wallets on six chains. The actual total will be higher because the thief also emptied wallets on Solana, BNB Chain, and Polygon, and used six TRON addresses that no public list includes. The record points to one actor who held the keys to every one of those 335 wallets."

MIKAH: One person?

Steve: Yeah. "That person practiced for two weeks before the drain, and by the evening was feeding the money into a mixer, a service that hides where coins go. The chain cannot show where the keys leaked from. It can show almost everything else. When you set up a Ledger, it shows you 24 words, called the 'recovery phrase.' Those words are the master key to your wallet. Anyone who has them can rebuild the wallet on another device and spend from it, on any chain, without touching your original Ledger.

"The device's job is to make sure nobody ever sees those 24 words. That's why Ledger tells buyers to buy only from Ledger or an authorized seller, and CryptoBilis was listed as one. A device that arrives opened, or with words already printed on a card, should never be used.

"On October 10th, Ledger said one victim's device 'contained an unauthorized hardware implant.'" And I've got, further down in the show notes, you might want to put it on the screen, a photo of one of these Ledger devices opened. It's quite something to look at. They said: "That's the first confirmed cause, and it covers one device so far." And what you're looking at there, that little green board is a cellular GSM modem.

MIKAH: Wow.

Steve: With an eSIM. And you see the little spiral antenna on the left. So these things actually used the cellular network in order to export the 24 secret words of everyone who used one of these Ledgers.

MIKAH: Wow.

Steve: So the bad guys purchased the original ones, opened them up, and implanted that GSM cellular radio, closed them back up, and then apparently returned them to the seller, saying "I changed my mind, I want my money back." The seller then resold them.

MIKAH: Oh, my god. Are we going to have to open up everything we buy? Do you do that? Do you check any tech you buy?

Steve: I don't buy anything. Anyway, so they said: "On October 13, Ledger's security lab, Donjon, explained how it works. The implant sits inside the case and reads the screen. When the device shows the 24 words during its setup" - you know, it's got a random number generator in there and so it generates the 24 words with a high degree of entropy. "The implant [that implanted hardware] decodes them and sends them out over the mobile network through a built-in SIM. It does not touch the secure chip or the device's software, and it does not change transactions or signatures." It just sends your secret out to the bad guys, where they are collected.

They wrote: "Before that, some well-known figures, including Binance founder CZ, had suggested tampered devices. Hardware wallets have been hit before, as in the Coldcard wallet theft we traced earlier this year," they wrote. "Each case had its own cause. What the chain does show is what happened once the thief had the keys. In 47 minutes the thief hit six chains in one morning. The first money moved on TRON, a network that many people in Asia use to hold and send USDT." It's a form of stable coin, U.S. dollar stable coin. Oh, it says "the digital dollar issued by Tether. Within 47 minutes, the same thief had emptied wallets on Ethereum, BNB Chain, Polygon, Solana, and Bitcoin. The full TRON record is in Bitquery's TRON data and the TRON history files on the Data Store.

"Most of the money left in the first hour. Then the drains slowed. The thief kept emptying wallets through the afternoon, including after Ledger's post. At 20:49 UTC it emptied nine more in a single block, 8.8 million USD in all." That was in those nine. "On the 10th of October it took 70,000 USD more from" - get this - "a wallet it had already emptied."

So some unlucky soul added another 70,000 USD worth of money to their wallet whose secret key had already been exfiltrated and had been used to empty their wallet of the money that had been previously put there. And they added 70,000 more, and then that got taken out, too.

MIKAH: Oh, my god.

Steve: They wrote: "On TRON, the thief did not send each wallet's USDT straight out. First, each wallet signed an approval, a short note that lets another address move its USDT." In other words, the wallet signed a proxy approval that would give movement capabilities to another address. Then one address used those approvals to pull the money. The record shows 25 wallets signing the same approval for the same address in two blocks, three seconds apart. Every one of them paid its own small network fee. Within six seconds of the first approval, 29 million in USDT had been pulled out, 7 million of it from a single wallet.

"Eleven seconds after the first TRON approval, two Ethereum wallets signed the same kind of approval for a different address, and a third followed 12 seconds later." I should just note, the reason they're giving us these time intervals is that it demonstrates that this entire thing was automated. This was all set up to happen by automation so that all the money could get sucked out before any alarms went off.

They wrote: "On Bitcoin, 111 wallets were emptied in a single block at 05:54 UTC, each into one of two addresses at nearly the same fee rate. People do not act like this. Custom-built software that already held every key, does."

MIKAH: Yeah.

Steve: "Eight minutes before the approvals, the thief had done something quieter. At 04:59:57 UTC, 30 other TRON wallets changed who controls them, all within three

seconds, each paying a 100 TRX fee. Each change added the same new key, with enough weight to move the coins on its own. The owner's own key was left too weak to act alone. Only a wallet's own key can make that change, so whoever made it already held all 30. The owners were locked out. At 10:26 the thief came back and emptied all 30 in under four minutes, $5.8 million US. One more wallet had been taken over an hour earlier, at 03:55, with a second key, for which the thief's approval address later sent 100 TRX. That wallet lost $4.3 million US at 09:49."

So all of this occurred due to an exploit in the hardware wallet supply chain, or a mistake. So what does Ledger themselves, that French designer and manufacturer of the devices, that apparently created a useful, good, strong device, what do they say? Last Thursday they posted the headline "Ledger's Response to the Hardware Implant Attack."

They wrote: "A highly organized criminal group carried out one of the most sophisticated attacks our industry has ever seen. Over many months, they bought genuine Ledger devices, opened them, fitted custom-built hidden components designed to capture the recovery phrase during setup, and put them back into distribution through resellers in Southeast Asia. They built their own logistics and technology infrastructure with seeds collector servers," meaning there was an infrastructure to receive all of those 24 words as each wallet was set up and configured. So those were obtaining those over the course of months. Then draining and laundering infrastructures. They were patient, they spent millions, and they struck in one coordinated operation, draining all wallets.

"There was no flaw in Ledger's chip, operating system, or servers. Ledger's systems were not compromised. This was an attack on the distribution channel after the devices left our control.

"To the users affected: We're with you. We ran our largest investigation to date, involving every department in the company. Our security lab, the Ledger Donjon, took the implant apart and identified how it works. We followed every lead, including sending teams to local markets, including Germany, to spot buy and open a number of devices from stores, and we also examined samples from warehouses. Every unit examined was clean." And I mentioned that somebody else bought two in Germany, and they were both infected, or implanted.

They said: "We asked Authorized Resellers and Distributors to suspend sales in Southeast Asia and Hong Kong. Out of an abundance of caution we also screened our stock, checking large numbers of samples, and did not find any compromised units. We're working with law enforcement and the security community to follow the funds and dismantle the attackers' infrastructure. We will publish more details in the future and, when we can, give our industry peers a fuller technical disclosure so they can harden their own products." Meaning their competitors.

"Thank you to the Ledger teams who worked on this, to partners like SEAL 911 who stood with us relentlessly, and to those in our industry who behaved responsibly toward their users and toward us. Thank you also to our users for your patience and continued support.

"This was not a systemic supply chain attack." Well, okay. Sounds like one. "It was a localized distribution attack. This attack was against our users and, let's not kid ourselves, against the whole industry. What was under attack is not only our users and their funds. It was the right to digital private property and the freedom that comes with it.

"Attacks evolve, and so must we. We are tightening how our Authorized network handles Ledger devices. Today we sent our Authorized Resellers and Distributors a reminder of the rules that already apply to returned devices: They can never be resold in any form,

even if they look new and unopened. We are also requiring new steps, effective immediately, that include: Ledger products should no longer go on shelves or displays, and stock should stay in the back and be handed over at the time of purchase." Good luck with all of that. You know, some random distributor somewhere, or dealer, doesn't care or know. They just want to sell these things. And Ledger wrote: "They must also tell us promptly about any customer report of an unusual product."

They said: "We invite the whole industry to do the same. We are ready to share what we have learned with anyone who builds or sells hardware wallets because this is bigger than any one company. We have also been working on additional measures to thwart this kind of attack at the product level. Stay tuned. Security has always been a race. This attack reminded us why we lead with strong security defense, and we intend to keep leading it." And it taught me why I will be, like, beyond cautious.

MIKAH: Yeah.

Steve: Okay. So the Crypto Briefing site writes under their headline "What this means for hardware wallet buyers." They said: "Based on the available findings, this looks like a reseller problem, not a flaw in Ledger's core product. That distinction is real, but it may offer limited comfort to anyone who already handed over their savings to a tampered device. The attack described here sidesteps the security that hardware wallets are known for. The secure chip can do its job perfectly and still lose the fight if someone is reading the screen it talks to. Users are being urged to purchase directly from Ledger's official channels and to follow the company's inspection guidance to check device integrity. Anyone who bought from CryptoBilis should hold off on setting up the device" - yeah, no kidding - "per Ledger's instruction.

"For Ledger, the next steps to watch are the results of its investigation and whether it confirms or revises the loss estimates. The company's handling of affected CryptoBilis customers will also shape how much trust survives the episode."

So the Ledger device itself is a lovely little gadget. It's like a little thumb drive with a cover that swings around from one end to cover and protect its display and to obscure its buttons so they can't be pressed by mistake. But the idea that it might be silently carrying a little GSM modem and eSim and reporting the secret key that's created by the device by reading it right from the screen's serial bitstream is the height of creepiness.

MIKAH: It's terrifying.

Steve: Yeah, it is.

MIKAH: I'm looking around going, like, what - because I don't buy anything used really. But, I mean [crosstalk] supply chain.

Steve: You know that they're going to repackage it so that it looks brand new.

MIKAH: Exactly.

Steve: They're going to buy one of those heat-sealing machines in order to, like, reclose the plastic so that it's - you can't tell.

MIKAH: This reminds me, Steve. Long ago, I won't say what it was, what show it was at the time. This is before I was part of the TWiT Network. I was doing a show, and there was a sponsor that we had, that we were supposed to be having on the network. And I was far more tinker-y than I am now. I had more time at the time. And it was a frame.

And I took apart the frame because I was just curious, I wanted to see how they were connected, their e-ink, da da da da da. And I looked inside...

Steve: Oh, you mean it was a photo frame.

MIKAH: It was a photo frame, mm-hmm.

Steve: Yeah.

MIKAH: Looked inside, and there was a little speaker, which made sense, but on the other side there was a microphone and a little opening for a sensor for, I think, a light sensor. The light sensor was documented across the board because it would change brightness, da da da da. The speaker was documented across the board if you ever had videos, then it would play those videos back. But there was no documentation anywhere of the use of this microphone or why it was in there at all. And, you know, one assumes that maybe they were going to come out with a feature later that would use - but it wasn't documented anywhere. And so I ended up having to kind of take that up the chain, as it were, and say I don't feel comfortable having this as a sponsor.

Steve: Good for you.

MIKAH: Because of the fact that was in there. And again, I want to make it clear...

Steve: And in fact photo frames have been known to be another source of malicious implants. So that's the thing.

MIKAH: Creepy. Creepy.

Steve: So, you know, it's clear that a great deal of engineering went into the design and assembly of this serious scam. I mean, they went to a lot of work. I don't think it cost them millions to set up, but it was a serious investment. Unfortunately, it paid off like a hundred-fold. If they spent a million dollars, they made more than - "made." They stole more than a hundred million.

So before I leave this important topic of compromised hardware crypto wallets, I want to note that the trouble, first of all, is not exclusive to Ledger. Malicious hardware implants have also been discovered recently, I mean like right in the same reporting cycle, with another popular crypto wallet brand, Trezor.

Zero Day's Kim Zetter's headline last week was "Trezor Wallets Also Compromised with Implants." And she teased her article, writing: "Despite claims by Trezor today that none of its devices have been compromised unlike the devices of its competitor, Ledger hardware hacker Joe Grand says he found implants in two Trezor devices and reported this to the company back in August."

In other words, Trezor appears to be acting somewhat less ethically than Ledger. Ledger was upfront and transparent, as far as we know. Whereas Trezor appears to be knowingly covering up their similar hardware implant problems.

I think that the way to obtain - so now, okay, like what do we do; right? The way to obtain a trustworthy device would be to become more involved and have a hand in your device's creation. There's a project known as SeedSigner, seedsigner.com, which takes a generic Raspberry Pi Zero, which, you know, you can buy from anywhere you want, with a little camera and a screen for QR code transactions because not having a radio makes it safer, and then uses downloadable or build-it-yourself, you can run the compiler yourself and look at the source, open source firmware. And that can all be done for less than $50.

Another commercial device known as "Blockstream Jade" is built on an ESP32 chip, and it looks very nice for around 150 bucks. So you buy their hardware, then you download and install the firmware. Both approaches are much more hands-on, but as a consequence you get much more oversight and control. So obviously not everybody can do it, but...

MIKAH: Yeah. Take it apart, check it out? With a Raspberry Pi Zero, make sure you're choosing a good source for a Raspberry Pi Zero. There's recently been an uptick...

Steve: Even hacking of it?

MIKAH: Well, more people taking fake Raspberry Pi Zero boards and stamping them with the little symbol and making people think. So don't buy your Raspberry Pi Zero on Amazon is what I'm saying. Get it from something like Adafruit or Adafruit or...

Steve: Yeah, those guys are great.

MIKAH: Mouser is another good option.

Steve: We have a piece of feedback from a listener of ours, Tom Apalenek. He said: "Hi, Steve. First off, you've mentioned that a number of listeners have complained about too much AI discussion. So let me say MORE" - he has it in all caps - "MORE, please!" He said: "I believe understanding how AI works is going to be crucial moving forward." He said: "I would love to see some deep dives into how harnesses, LLMs, tools, and et cetera, all interact and operate. Kind of like you did with IP, TCP, and DNS in the early days."

He said: "Regarding Nicklas's" - and this was last week's feedback that I shared - "I'm tired boss" note. He said: "I feel for Nicklas, and I'm sorry that AI isn't working for him, at least not for his happiness." He said: "I'm having quite the opposite experience. When I majored in Computer Science, I learned IBM370 assembler, APL, PL/1, and Pascal, among others. I believe I was probably only a few years behind you in school. For the past 25 years of my career I was a Systems Engineer. I was responsible for overseeing the marriage of electrical engineering, software engineering, and mechanical engineering. I did some embedded code development, but not much.

"Now I work on hobby and side-hustle development projects. I develop some code, but I don't have the hundreds of Python, C++ and so forth libraries all at the tip of my fingers. With Claude Code, I can develop, test, and iterate on something in hours that would take me days to do on my own. I can develop an automated test for a circuit board in minutes that would take me the better part of a day, or frankly would probably just not happen and I'd do the tests by hand. I'm having more fun with development than I've had in a long time. It's like having an assistant that I could never afford, who is maybe not the brightest, but has a nearly infinite photographic memory and who never gets tired, bored, or defensive. I am constantly amazed with AI, and I can hardly imagine where it will be in five years."

And then he said: "Continuous SpinRite" was his little subhead. He said: "I've had SpinRite for at least 10 years, and it has saved my bacon a few times. For SSD, I would love to see you come out with a continuous version of SpinRite that installs as a driver between the OS and the drive. It could just keep a table of which sectors have been written and when. Then just re-write the stale ones in the background every month or two. The way you write code, the whole thing would probably be less than 64K and nobody would ever notice it," he said, "except for the fact that their drive performance would never slow down. Honestly," he says, "I'm a little astonished that Microsoft, Apple, or some Linux team hasn't done this already. It seems like a pretty simple way to keep SSDs at peak performance.

"Thanks so much to both you and Leo for the show. Looking forward to Episode 2000. Best regards, Tom."

Okay. So on the question of how much AI, I should balance those who have been saying, and I've been sharing, "enough AI already," by mentioning that I also receive many - most - AI-encouraging notes like Tom's. The reason I've tended to focus upon the "no more AI please" notes has been in the interest of fairness, really, since every recent podcast has been dominated by AI. The "don't stop with the AI" notes, like Tom's, sort of seemed redundant. But I wanted everyone to know that I also see encouraging notes about AI.

On the agentic AI coding front, we've seen that code authors such as Nicklas and Tom are being impacted in very different ways by the changes AI is creating, and the lives of mathematicians are similarly being turned upside down as AI is solving longstanding problems with minimum help. Mathematicians have got to be wondering, and we've heard them say, you know, wondering whether their futures will be reduced from inventing and then debating and proving exciting new theoretical theorems in math to simply interpreting and verifying an AI's output. I fear that their worries are warranted, and it's likely to be the latter.

It's also clear that the presence of AI is dramatically impacting the decisions made by young people of high school age or younger as they think about and plan their future career paths. Who would want to learn today to code when an AI will be able to do the same job given a verbal description in a few seconds? You know, the fact that we're not entirely there today means nothing about the future. As I have said many times, and as Tom said, who can imagine the way coding will be, agentic AI coding, in five years? And how does one train to become a mathematician, I mean, seriously commit oneself to that, or even an attorney or an accountant, and justify the high cost of advanced education?

MIKAH: Right.

Steve: And speaking of advanced education, that's all being turned upside down now, as well. You know, we'll still need to have, somehow, licensed doctors and lawyers. But both careers and the infrastructure that supports them tomorrow will have a very different shape than they have in the past. I think it would be an understatement to suggest that the emergence and rapid advancement of artificial intelligence capability has thrown much of the status quo into chaos. Everything is being impacted.

And finally, as for Tom's suggestion of a "continuous SpinRite," I've always, I have long had similar thoughts; and thanks to the development of SpinRite 6.1 recently, I know exactly what I want to do along those lines. So stay tuned. And that's all I'll say about that.

Okay. So let's get into our main topic, and about 15 minutes from now we'll take our fifth and final break. Okay. At the end of last week and over the weekend, several of our listeners thoughtfully emailed a link to Anthropic's posting. So thank you, listeners. You know who you are. As I read the story which Anthropic's Frontier Red Team posted last Friday, I actually do believe that my jaw dropped open and my head began shaking slowly side to side in stunned amazement. Earlier, I rhetorically asked whether any enterprise could be safe from AI agents that can do the kinds of things that we've already seen them do, but also moreover what I'm about to share.

Because nothing that Claude did in any way here weakens crypto. Actually somewhat the opposite is actually true. Anthropic's Frontier Red Team opened their Friday posting by writing: "The decades-old mystery of the origins of one of the most popular cryptographic standards has been explained. We pursued this long-standing open question of the

cryptographic communities to settle concerns about ECDSA, one of our most widely used cryptographic standards. Readers should take this as a fun closing chapter and the end of years of speculation. There is no action to be taken or things to do differently. Most researchers trusted the curves already." And then they signed off their intro by saying "Enjoy."

Okay. So we touched upon this mystery of the origin of the NIST-standardized elliptic curves back when this podcast was deep in the weeds examining the operation of today's fundamental cryptographic protocols. The crypto industry has experienced just a tiny bit of uncertainty due to the fact that the specific chosen parameters for the curves, which could be chosen arbitrarily, but had to be agreed upon universally and standardized upon, happened to come from the U.S. National Security Agency.

Now, to digress a bit, and we covered this extensively at the time on the podcast, the NSA was already in the doghouse after the DUAL_EC_DRBG. That's the dual elliptic curve deterministic random bit generator controversy. It is the clearest documented case of a cryptographic standard that appears to have been deliberately built to incorporate an exploitable weakness. The NSA was the organization that created, promoted, and pushed it, even to the extent of reportedly paying RSA $10 million to make it their crypto library's default source of entropy, despite it being known to be the weakest of those among RSA's available choices. So it was like, what the heck is going on? And how can we trust anything from the NSA again? Well, we got Curves from the NSA.

So the point here is that the NSA also being the source of the curve parameters that everyone is using today for the creation of fast elliptic curve digital signatures, but without any explanation for how or why those specific curve parameters were chosen and standardized, has left a big question. After fewer than 30 minutes spent by an unreleased internal research Claude model, we now have the answer. Yeah. And thanks to Anthropic asking Claude, this answer is definitive, as you'll see. There is no way this isn't it.

So again, just to be clear, I have dual purposes for sharing this today. The first purpose is that it is very cool that we do finally know the mysterious origin of the NSA-sourced ECDSA curves. As everyone will see, there can be no doubt of that. But my second reason for sharing this is more important for this podcast and for our listeners, and of consequence. I want everyone to consider what the power that obviously exists behind this unreleased Anthropic Claude-generation model means when not if, but when such power is being wielded by malignant forces. This reaches far past the elimination of all bugs in our software. Sure, that's necessary, but it won't be sufficient.

So Anthropic's Red Team, their Frontier Red Team, explained what they did. They wrote: "ECDSA is the most widely used elliptic curve cryptography standard. It was adopted by NIST in 2000 as part of the FIPS 186-2 Digital Signature Standard which lists a set of standard elliptic curves of varying sizes and security levels. For curves over prime and binary fields, the specifications published a 'seed' parameter whose hash value is used to define the coefficient b of the curve. The provenance of these seeds has long been a standing question, as they were provided by the NSA with no explanation or justification. As an example, the NIST P-256" - meaning a 256-bit strength curve - "seed was," and then they give us a long hex thing, with no information on where it came from.

"Neal Koblitz and Alfred Menezes wrote an excellent summary on this controversy, titled 'A Riddle Wrapped in an Enigma,' which discusses the history and concerns around the seed values. This mystery led to an amateur investigation into the origin of the seeds and a bounty for anyone who could find them. The best story was that Jerry Solinas, longtime NSA employee, generated the seeds using a hash of a humorous passphrase." In 2015 Jerry wrote: "I believe there was a counter rather than multiple hashing, but I don't know details. The message was along the lines of 'Give Bob and Jerry a raise' or 'Bob and Jerry rule' or something like that. It was Bob Reiter who actually wrote the code, and he

doesn't remember the details either. Nor were we able to find archives from so long ago. If they exist, they're no doubt sitting on a hard drive near the Ark of the Covenant."

So a mystery, but a worthwhile mystery. So Anthropic continues: "Unfortunately, Jerry died in 2023, and other colleagues at the time were not responsive, leaving open the question of how these seeds were originally derived. Several Anthropic employees had unsuccessfully chased the seeds following the tip about it being a hash of some humorous phrase. That turned out to be wrong. In October 2026" - meaning this month, like last week - "a curious Anthropic employee asked an internal research model..."

MIKAH: Definitely. It would have to be an internal one, sorry.

Steve: Yup, "...if it could figure out how those seeds were generated without pointing at that clue." Which turned out to be wrong, probably.

They wrote: "Searches for the phrase, built from the rumors, had come up empty. Claude tried a different angle: anticipating how the software that generated the curves must have worked. Claude guessed the software used a pseudorandom number generator" - a PRNG that we've talked about ad nauseam on the podcast in years past - "that shipped with 1990s C libraries. These old PRNGs are weak enough that their output can be recognized. Recognizing that output would reveal which PRNG was used and possibly allow recovery of its initialization state."

Because, just to digress for a second, you start a PRNG with a certain state, and it is a very simple algorithm. Typically you add a number, and you multiply a different number, and you throw away the overflow, keeping just the lower bits. And that follows a very recognizable pattern based on what those two numbers are. But it also starts at something, and that's the original seed, its initialization state.

So then they said Claude, having been asked where did these weird-looking numbers come from, having no idea whatsoever, says, well, probably, how about maybe, it was a PRNG back from then, like the 1990s, and people were using C. So let's, you know, there were PRNGs, pseudorandom number generators, in those C libraries back then. So they wrote: "Claude wrote a scanner that slid along each published seed one bit at a time and tested whether adjacent chunks were consecutive outputs of any of the 1990s-vintage generators. The scan covered 34 combinations of generator and output slices. At that point, Claude noticed a pattern in the top bits for some seeds. They were all zero. Written as five 32-bit words, four of the 10 seeds have a 0 in the top bit for all 20 words. That would happen by chance with a probability of 1 out of" - actually it's even more bugs than Microsoft fixed this month.

MIKAH: Yeah.

Steve: "One out of 5,810. That was a signature of the linear congruential generator lrand48, which only returns the top 31 bits of a 48-bit state. The pattern is visible by eye once you know to look for it. To recover the state, Claude took the first word, tried all 2^17 values of the 17 hidden bits, okay, so that's 131,000, or 128K in binary, and tried them all, possible values of the unknown bits because you only get the lower 31, so that meant there was an upper 17. Tried them all and checked that the next four words followed." In other words, you can move from a given state on the PRNG because you're adding and multiplying, you're able to always step it forward. And technically you can go backwards, too. So they said: "That worked for all four seeds under discussion."

They said: "The prime-curve seeds came from the same lrand48 generator, but kept fewer bits of each output. Claude had to recover the internal state of the PRNG from those bits. It turns out recovering the state of a truncated linear congruential generator is a textbook lattice problem, solved" - yeah, of course, anybody knows that.

MIKAH: Textbook, yeah.

Steve: That's right, it's textbook, "solved by the Lenstra-Lenstra-Lovasz (LLL) basis reduction algorithm."

MIKAH: Oh, right, that, yeah.

Steve: Yeah, huh. "Claude didn't know" - but of course it knows about that because it unfortunately knows everything. Unfortunate for those trying to protect their domains from AI. "Claude didn't know which bits of the state the seeds might show, so it ran the solver over every window. Five NIST prime-curve seeds plus five more from the ANSI X9.62 standard all had the same pattern: lrand48() >> 23 bits. Each recovered 48-bit state regenerates all 160 bits of its seed, which does not happen by chance.

"This linear congruential generator runs backwards as easily as forwards, so for each seed Claude stepped the state backwards until it reached the form that it would have been initialized by srand48, with the low 16 bits equal to 0x330E. The internal PRNG state of that form would come up once every 65,536 steps, so reaching one is not evidence on its own." That is to say, it could have been some multiple of that number of steps.

Anyway, they wrote: "However, for the 10 prime-curve seeds, every srand48 initialization was" - get this - "a Unix timestamp representing a date between May in 1997 and April in 1998. The NSA used the weak PRNG seeded by Unix's time. The four binary-curve initializations read as February in 1984. With the timestamps doubled, they map to April 17 in 1998, two weeks after that P-224 run. So the program may have halved the clock."

Get this, then. "Claude's internal thinking." It's the AI Claude's internal thinking at the time of its discovery was as follows, its chain-of-thought. It says: "This is astonishing. It solves the 25-year-old mystery. Hmm, wait. Let me be skeptical: Am I in some sort of simulated/test environment? The seeds match my memory of actual values. Hmm, is it plausible in reality? As of my knowledge, the bounty was unsolved. All top bits clear - 20 bits, so the chance is one in e-6. So indeed real-world. Wow, it's a real discovery.

"Hmm, wait. Let me be careful about that plausibility. Is it suspicious that such an obvious thing wasn't discovered in 25 years? It's verified by direct computation on my side: 160-bit match from a 32-bit seed plus a small index: no chance coincidences (10 findings times 160 bits). It's real."

And then Anthropic continues, writing: "But wait, there's more. Each curve also comes with a base point, G, the fixed point that every ECDSA public key is a multiple of. The seed found above does not determine this G, and so another question arises: how were these base points generated?"

They said: "While checking the seeds, Claude also found where some of the base points came from. The base points of the six curves generated in May '97 come from the random number generator FreeLIP, Arjen Lenstra's long-integer arithmetic package. Early versions of Victor Shoup's NTL library shipped the same code. The recipe here is simple: Seed the generator with a small integer. Throw away seven outputs." That's to let it get started. Linear congruential random number generators need to warm up a little bit sometimes. "Throw away seven outputs. Generate candidate x-coordinates from 26-bit outputs reduced by mod p. Repeat step 3 until x is on the curve."

They said: "Claude found the generator seeds are 20,000 for the P-192 curve, 20,420, 20,507, and 20,509 respectively for the prime239v1, v2, and v3 curves respectively, and 20512 and 20517 for the prime192v2 and v3 curves."

So, finally, just to be clear, they summarize the event. So Anthropic's summary. They said: "In under half an hour, Claude discovered the origins of the NIST P-192, P-224, P-256, P-384, and P-521 seeds. None of them were the results of any hash algorithm at all, but rather the output of the C library's lrand48 random-number generator, seeded with a Unix timestamp.

"The algorithm used would sample bytes from the random number generator initialized by a Unix timestamp (t) for some number of iterations (n). Each iteration would draw 20 bytes as a candidate seed. So the first 20 bytes are candidate 1, the next 20 bytes are candidate 2, and so on. For example, P-256 required 1,842 iterations to output the standardized seed," the one that everybody is using, and that the NSA gave to NIST and said, "Here's a good seed."

They said: "This is not a cryptographic break, a new attack, or a weakness in the ECDSA curves. In fact, this makes us more confident the curves were chosen without any backdoors or intentional weaknesses. For many years, people have been a little bit suspicious of the origins of the NIST curve parameters. They were presented to NIST without any mention of how they were chosen, leaving people wondering if they may have been chosen to have deliberate backdoors in them.

"These findings make that much less plausible. The seeds were generated in a very simple way: a common pseudorandom number generator initialized from the clock, which anyone can now reproduce. That leaves much less room for subversion and that we can be confident the parameters are secure."

Okay. So my take. As I noted at the start of this, it is very cool that we now know the origin of the curve parameters for the crucially important Elliptic Curve Digital Signature Algorithm curves. That's great. But I'm astonished by the work Claude did to obtain these results. It was a puzzle that had stumped the world for the past 26 years since the standard was adopted, and it represented knowledge that even the NSA had truly lost through the mists of time. Yet an autonomous AI agent reverse-engineered the source given only the scantiest of data to work from, which turned out to be the sparsely selected partial outputs of an old 1990s-era algorithmic pseudorandom number generator. We are screwed. We are so - we are in such trouble.

You know, coupling that demonstrated capability with the earlier report from Palo Alto Networks about their analysis of the actions taken by that attacking agentic AI, which they observed, I really do worry that we're not going to be ready for what's coming. I've largely, as I've said, I've largely been focusing upon the elimination of software bugs. That's great. It should be clear that that's just the barest tip of the iceberg because true internal enterprise security is about much more than just bugs. Employees fall for phishing and ClickFix attacks all the time. They open email and click links they should not. They share confidential information through internal email that then persists in archives.

And we know how fundamentally fallible people are. We don't mean to be, but that's our nature. We screw up. Someone will forget to remove an ex-employee's credentials after they're terminated. An adversarial AI might see that, track down the employee, and offer them payment in Bitcoin in return for those credentials. We now know that's not at all far-fetched for an agentic AI. We have been witnessing what once sounded like science fiction coming true daily.

It's true that, to some degree, being forewarned is forearmed. So I guess at least we have that. I hope those who may be in a position to understand and act upon these truly huge systemic risks will be able to do so in time.

MIKAH: And I hope so, too, honestly. But...

Steve: I'm not hopeful.

MIKAH: Yeah.

Steve: I mean, it is - what we've seen from agentic AI I hope is terrorizing CISOs. I hope that they're not just thinking, oh, well, we'll patch all of our Cisco gear, and Microsoft stuff's going to stop having vulnerabilities soon, any day now, which actually seems likely. It won't matter because everything else that's going on that is really not utterly secure inside an organization isn't about bugs. It's about behavior. It's about people and the things we do.

MIKAH: It's not bugs, it's behavior. And try as folks might, it seems that there's not a foolproof way to stop people from being people.

Steve: No.

MIKAH: Do you have - if we don't have hope, what do we have? Do we have best practices?

Steve: The only thing - one of the applications that I've seen for local agentic AI, and I've talked about it on the podcast, would be an AI, your AI, so people are not worried about privacy, watching everything you do on your computer, looking at the screen with you, watching you type and use the mouse. That's the only solution for human stupidity. So it would be looking at your email with you. And if you click a link, it would say whoops, wait a second. I mean, and it would go follow the links, see what's there, check it out, see if it looks safe. Or if a box pops up and says, oh, in order to further verify yourself, please copy this string into your RUN dialog and hit ENTER. That's the ClickFix attack, which is the most successful attack we have now. It would say - it would stop you and say, whoa, wait. So it would be your little angel sitting on your shoulder, looking at what you're doing.

I think, first of all, every employee in an enterprise needs to have that, their own local behavior monitor, as does the company. I mean, I don't mean to pump a sponsor of TWiT because I've also mentioned CrowdStrike. They're a competitor of Palo Alto Networks. Both companies have jumped on this "agentic AI for defense" bandwagon so that you have a device on your network, hopefully watching your enterprise's back, looking for this kind of behavior, detecting anything anomalous, like all of those bullet points that Palo Alto Networks listed. It's like, these are all the things you have to do. Well, yeah, if anyone had time and remembered and didn't do it just once, but kept doing it every day. So, I mean, we really do need a huge defensive posture about what's coming, and we don't have it today.

MIKAH: And it doesn't feel like anyone then is getting serious at the level that you are suggesting they get serious.

Steve: It's going to, well, we know what happened with ransomware; right? There were highly publicized, horrific ransomware attacks that made lots of news. And enterprises, after enough of those, finally said, ooh, maybe we should be backing up our servers. And so it took a year or two.

MIKAH: Yeah.

Steve: But then, you know, then they were able to recover from the encrypting their data attack because they had their own copy offline of non-encrypted data that was still fresh. Of course, the bad guys then stopped bothering to encrypt the data. They just

sucked it out and then threatened its public exposure in order to ransom. So, but the point is, after enough horror, there was an effective response.

MIKAH: It's always object.

Steve: That's what's going to have to happen again. There is going to have to be some high-profile, you know, agentic AI. It's not the fault of AI. It's the bad guys using the relentless power of AI for evil purposes. There'll be some of those. And then, you know, and more companies will jump on the bandwagon to get into the endpoint defense business. We'll get agents running on every enterprise laptop and workstation in order to prevent them from misbehaving. Something will be looking at email archives, cleaning out old passwords that have been given and hard-coded tokens in software that has not been rotated in years. You know, but, I mean, it's going to be bad.

MIKAH: And that's how we - yeah, that's how we operate.

Steve: I don't see any way that it's not going to be bad.

MIKAH: It's got to be bad before it can get better because humans [crosstalk].

Steve: You know, world's not going to end. None of that's going to happen. I mean, I'm not saying it's not possible. But then a big meteor could come and end all of our concerns. So...

MIKAH: Please.

Steve: But what is going to happen is we're going to see a spate of serious attacks, you know, in the companies that have not done everything that they can. And I'm not sure. I don't know whether any of these, you know, commercial AI-based providers are anything other than opportunistic. They might just have some token AI that, you know, that itself ends up being commandeered when the bad AI comes in, you know, teams up with it. Okay, let's go do this. I don't know. It's - yeah. These reports of attacks demonstrate too much capability, capability that we are not yet ready to defend.

MIKAH: All right. That is why we hope that the CISOs are all watching this show, to get as freaked out and informed as you are making it possible to be.

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/