The Apex Agentic Adversary

Description: Why Fable 5's re-release has disappointed. Opera becomes the first browser to offer "Paste Protect." Microsoft BlueHammer exploit is "hammering" systems. Industry legend (TCP creator) Vint Cerf on AI. Chrome turns 150 with too many fixes to load. Google fails to sidestep a $4.67 billion EU fine. One last (we can hope) Chat Control vote next week. AirDrop and Android Quick Share are exploitable. How to bypass Claude's and ChatGPT's guardrails. My own Sunday spin with SpinRite. A legendary hacker uses AI on a widespread library.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-1086.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-1086-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Great show planned. We're going to talk about Fable 5. Actually we're going to talk about AI used for security. One of the creators, the Fathers of the Internet, is retiring. And you won't believe how many bugs Google fixed in Chrome this time around. All that and more coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 1086, recorded Tuesday, July 7th, 2026: "The Apex Agentic Adversary."

It's time for Security Now!, the show where we cover the latest in security, privacy, and online shenanigans with the king of online shenanigans, Mr. Steve Gibson. Not the creator of, the forger of.

Steve Gibson: Fortunately, the Internet did not exist when I was a little one.

Leo: Oh, you would have been in trouble.

Steve: You know, like high school age, when I was creating shock machines and portable dog killers because that would have been a problem.

Leo: Yeah. What do you think if, as a kid, at your, you know, this super inquisitive, super smart kid, had AI been available? Your life could be completely different. Right? I mean...

Steve: I would be doing things, because I've always been a project person, I mean, I don't sort of just meander.

Leo: Right, right.

Steve: I normally want a conclusion.

Leo: Right.

Steve: So I'm sure I would be telling my parents I need more tokens.

Leo: I need more tokens. Mom, can I have some more tokens, Mom?

Steve: I need to increase my allowance, is what we used to get. Well, we'll give you a dollar a week if you take the trash cans out. Of course that lasted about a week and a half, and the trash cans don't get taken out anymore. But I would have been saying I need to increase my token allowance. So that would have been...

Leo: That's really true. If you take out the trash cans, you get a million more tokens, little Stevie.

Steve: So today's podcast title, which is "The Apex Agentic Adversary," is a phrase that comes from someone famous who we're going to be talking about at the end of the podcast. Actually, we're going to talk about several famous people during this podcast because Vint Cerf also has his name pop up.

Leo: Yup.

Steve: Of course we recognize it. We'll have some fun with that. So we're going to talk about an aspect of Fable 5 which has disappointed people. Not you. And we'll talk about how it's, you know, what your experience has been because that's certainly interesting. Opera becoming the first browser to offer Paste Protect. And while I think that's good, why I'm more annoyed than ever that it had to be, you know, Opera. And who cares about 2% of the browsers? I mean, it's better than zero; right? But still.

Microsoft BlueHammer is now being used to hammer systems. Vint Cerf has some comments about AI on the eve of his retirement. Chrome has turned 150 as the nation turns 250. And Leo, well, we'll have some fun. Something happened about that that's just so cool. Also Google, I wanted to just touch on Google and the EU. And oops. The fine is coming out of escrow, but it's not going back to Google. Also there's an upcoming, what we think is going to be the final Chat Control vote, although really unclear. It's looking like something may happen, but no one knows what because it's all behind closed doors. Turns out that there were some bad problems found both in AirDrop and in the Android equivalent, Quick Share, which are exploitable. Also something interesting about bypassing Claude's and ChatGPT's guardrails, as I've been saying. This is uncontrollable technology. And we have another example of that.

I had a SpinRite story to share, which I haven't had for a long time.

Leo: We haven't had one of those in ages.

Steve: Yeah. And this one was like I'm the proud father. Anyway, and then we're going to talk about a legendary hacker who has used AI against a, I don't know to say horribly, shockingly, massively. Anyway, it's a library that is everywhere; and it is, again, you know, what AI does when you aim it at anyone's past code, is it just cuts it to Swiss cheese.

Leo: Wow.

Steve: Anyway, and that is where the phrase "apex agentic adversary" came from.

Leo: Oh, yeah.

Steve: So I think a lot of fun stuff for us. A fun Picture of the Week from a listener who has been listening to us for about six years. And he sent an email, I think I just got it over the weekend. He said his name is Ryan. He said: "After 6-ish years of listening I think I finally got something for Picture of the Week. Thanks for a great podcast that's been very helpful throughout my career." So anyway, we will get to that after our first break. And so, yeah, a fun podcast.

Leo: And if you are listening, there's a couple of ways you can see the Picture of the Week. One is to get the video. The other's to get Steve's show notes, which always contain the Picture of the Week. Now, I am looking at your show notes, but I have not scrolled up. So I will be surprised.

Steve: And in fact, if you - if you prescribe. If you sub - it may be a prescription. If you subscribe to the weekly mailing, then there's a 250-pixel wide, that's the standard Picture of the Week thumbnail. And if you click on it, you immediately get the Picture of the Week. You don't even have to look at the show notes, is my point. So you could also just get it that way.

Leo: In the email, right there.

Steve: There are lots of people who just say, I just really - I don't care about security, but the Pictures of the Week are worth...

Leo: That's actually, you know, that's a good point. Get the email just for the thumbnail so you can click it and see the Picture of the Week. [Crosstalk] for the week comes in your mailbox, courtesy of Steve Gibson. That's nice. All right, Steve. Picture of the Week.

Steve: So as I often do, I captioned this myself. I gave this the caption, "But Officer, what do you mean the speed limit is 25? I'm sure I didn't see any sign to that effect."

Leo: All right. I'm scrolling up. Let's see. It's just, there's nothing here. I don't get it.

Steve: You will.

Leo: You've baffled me. Normally I get these right away. It's just an empty street.

Steve: I know you do. But this is not a cognitive get. This is a notice it get. So, "But Officer, what do you mean the speed limit is 25? I'm sure I didn't see any sign to that effect."

Leo: Yeah. Oh, there it is! Yeah, you're right. You didn't. Holy cow.

Steve: Now, Leo, how does this happen? Like...

Leo: Who puts a sign smack dab against a telephone pole so you can't read it?

Steve: I mean, it's even centered behind the pole.

Leo: It's S, L, part of a 2 is all you could see. Holy cow. Do you think they put the pole in after the sign?

Steve: Had to have been. And they just, like, well, that's - we don't care.

Leo: That's our job.

Steve: That's, you know, we know where the pole has to go, and it's got to go right here. And actually there is like a - we see a big spool of wire at the bottom.

Leo: Yeah, clearly this is new, yeah.

Steve: My guess is that, exactly, that this is sort of in the process of happening.

Leo: That's hysterical.

Steve: Anyway, yeah, for those who don't have access to the video or the show notes, we have this typical, you know, Speed Limit 25, we've all seen those. And it is, I mean, it is - it's almost completely obscured. We know it's 25 because you can see just the, as Leo said, the S and the L, a little bit of the L is not visible. And then what - you know

that's got to be a 2 because it can't be any other number. A 3 would look different if you just saw that very left edge of it, or a 4 or a 5 or something. Anyway, I would argue this one, you know, you think you can get out of this ticket? I think so. I think you take a picture, and you go to court, and you say, Your Honor, you know, I'm paying attention. And boy, I don't know where this is, but look at all those phone poles and power poles.

Leo: There are a lot of them, and very close together; aren't they.

Steve: Yeah. And this is not, you know, computer generated. This is our listener Ryan, who's been listening for six years, and he saw this, and he thought of the podcast. So Ryan, thank you for keeping us in mind. And I wrote back to him, I said, you got it on the first try. I said...

Leo: Yeah. Nice job.

Steve: ...this is a perfect picture for, you know, like us shaking our heads at humanity.

Leo: Crazy.

Steve: Okay. So a blurb of reporting by BleepingComputer following the release of Fable 5 caught my attention because it was, to me, it was predictable. Here's what BleepingComputer wrote about Fable 5's performance after talking a little bit about its availability, which I skipped because we know about that. You know, like, oh, it's expensive, twice as expensive as Opus 4.8 and so forth.

Leo: Just as a data point, I asked my Hermes agent, which I use to do a lot of AI stuff, based on the amount of tokens I used in June, last month, how much would it have cost if I'd been using Fable 5. $5,000. And this was just kind of casual use. I wasn't doing anything heavy-duty. $5,000. And I said, well, if I use DeepSeek, the cheap Chinese model? They said 117 bucks. So there's a big difference between Fable and the rest.

Steve: Now, there is, and I'm glad you brought that up because we're going to - by the end of this podcast, something has happened which suggests that the guardrails can actually protect the user rather than limit what the user can do. So this is going to end up being, I think, very important to our audience because there will be a strong tendency to use a powerful foreign AI like DeepSeek. It's like, well, look at all the money I'm saving. But there's a gotcha.

Leo: Ooh.

Steve: Which we will get by the end of today.

Leo: I'll be listening. Yikes.

Steve: So BleepingComputer wrote, after talking about the availability like, you know, that, cost and token use and it's like available. Okay. What is it that is available until the 7th? I mean until the 12th now.

Leo: Well, yeah. So that's part of the story. The deal was Fable is of course very expensive. I think it's, what was it, 15...

Steve: Twice what 4.8 is.

Leo: Yeah, $15 for a million tokens in and $50 for a million tokens out. So it's considerably more expensive than even the best models before, including ChatGPT. But OpenAI and Anthropic and other companies, even the Chinese companies, have subscriptions, where it's kind of all you can eat subscription. You pay 100 bucks a month or 200 bucks a month and you get five times the tokens or 20 times the tokens for a flat rate.

Steve: Not all you can eat, then. I mean...

Leo: Never all you can eat, really.

Steve: If you eat too much.

Leo: Yeah. And there's also limits to how much you can use in a five-hour period, how much you can use in a week, and how much you can use in a month. So they try to - but honestly, if you buy a 5x or a 20x package, unless you're really, like, running 12 different tasks at the same time, you're probably going to be covered. So when they first released Fable, they gave us five days that we could use it under our all-you-can-eat package. And then we were told on the 22nd of June it's over. Well, of course the government ended it a little bit sooner. When it came back they said, okay, you can use it under your subscription package, and it'll only be 50% of the total usage cost of what it will be. So it's even [crosstalk].

Steve: So, like, half off.

Leo: Half off.

Steve: For an introductory period.

Leo: Right. And it will be available on your subscription. But that's going to end on July 7th. So up to midnight last night I was going, "Got to use all the Fable, got to use all the Fable." Because I have a Max subscription. Anyway, they announced this morning, okay, never mind, we're going to let that continue until the 12th. So we have until the 12th to use it under our subscriptions.

Steve: And why? Why? Do they want to addict more people to it?

Leo: Yes.

Steve: Not enough people got the news that it was back, maybe?

Leo: Yes.

Steve: And so, oh, okay.

Leo: And there's another reason, which is that OpenAI has a model, GPT 5.6, which they're touting is as good as Fable and very good at cybersecurity, by the way.

Steve: Oh. So A/B comparison testing.

Leo: Yeah. And in fact the rumors were the minute Anthropic pulls the plug on that subscription thing, OpenAI is going to release 5.6 and say, come on in, you could use a subscription here. Now, that didn't happen, but I suspect competition is the other reason. So we'll see. You know, these two are in head-to-head battle right now.

Steve: Yeah. Wow.

Leo: It's interesting. Anyway, I'm going to - I have five more days to stay up till midnight [crosstalk].

Steve: So five more days at half use or half...

Leo: It will use my subscription, and it will be half the usage. So I will get twice as much done in five hours than I could. And it's still on the subscription.

Steve: And that hourly reset thing also makes sense because they have, I mean, because use is inherently interactive, and they have a limited capacity...

Leo: That's right.

Steve: ...based on data center.

Leo: That might be the other reason is we don't know.

Steve: So they need to spread to what degree they can, they have to spread usage over time.

Leo: Right.

Steve: So that not everybody can pile on at the same - during the same five hours [crosstalk].

Leo: Remember, Anthropic knows exactly how many people are using it, how hard they're using it, when and so forth.

Steve: Yup.

Leo: They have a limited capacity. Sure, they buy capacity from xAI, a billion dollars a month worth of capacity. They've got, you know, they're buying as much press as they can. But there is a limit to that. And so that's why it costs more, and that's why these limits. And yeah, that may be the other detail. We don't know. Maybe they found out that, oh, they've got enough capacity to support it for another few more days or...

Steve: Yeah. Do you know if there's a diurnal cycle to it, like is there a time...

Leo: Oh, there is. For instance, the Chinese models, they have a peak period that is in the middle of the night for me. So that's very good. So you see, I have used up - that's context, 13% of - it's a million token context. But that's the five-hour period. I've used up 27%. When it gets to 90%, it either starts - I've turned off the "start charging me," but you can have it start charging you API tokens, and then it really gets expensive. So you want to stay within that five-hour period. Yeah, it's fun. It's working right now. Even as we talk, it's doing stuff.

Steve: Okay. So they talked about that stuff. Then they said: "However, the real gut punch is the degraded performance, or as famously used in the AI community, the 'nerfed' performance." Meaning "nerfed" is the term. "On Reddit, users are reporting that the restored Fable 5 feels weaker, or is simply being routed through stricter safety systems more often than before." One user wrote in a Reddit post: "The new guardrails are kicking in on way too many tasks and falling back to Opus 4.8. This is not the model that got banned."

So Bleeping says: "The problem is not just limited to Claude desktop, as Claude Code is also struggling with similar issues." Now, again, Leo, I think you're not hitting this because you're doing advertising management or whatever it is.

Leo: Yeah, they don't seem to care about what - I asked somebody, and it says they will tell you. They didn't do this last time, but they will tell you now if they drop down to 4.8. In the past they have dropped down silently to Opus 4.8. But apparently they will let you know if they've done this. So as far as I know, they haven't done that.

Steve: And clearly these people who are writing this and complaining are knowing that it is dropping to Opus 4.8. So this is not going, you know, silently. Bleeping said: "One user said Fable 'didn't even let me search for dead code without switching to Opus,' while

another said it was 'very, very obvious' when the fallback triggers because Claude tells the user and visibly shifts to Opus. Another developer claimed the model was unusable for some systems-level coding work, saying that C, C++, Rust, Win32 API references, memory-related work, and files mentioning words like 'security,' 'vulnerable,' 'unsafe,' or 'hook' appeared to trigger a fallback or block. Fable 5 may still be powerful when it actually handles the task" - that's what you're finding, Leo - "but the restored version appears to be far more sensitive to prompts, project files, and security-adjacent language."

Leo: Yeah, you can thank the government for the [crosstalk].

Steve: That's right. That's right. "However," they wrote, "BleepingComputer understands that the model itself has not been nerfed. Instead, it's likely that Anthropic is being extra careful with the safety guardrails, which is negatively affecting Fable's daily use cases." I would argue in a subset like that are security adjacent, which totally makes sense. They said: "In fact, we observed that Fable is sometimes routed to Opus 4.8 even when the task does not appear to be a safety risk. Anthropic has said that its updated safeguards rely on a large 'safety margin,' which could explain the subpar experience some users are seeing with Fable. Anthropic has not acknowledged the reports of false positives yet, but it's likely the company is aware of the problem and will address it in a future update."

Okay. Now, all of this is exactly the downstream consequence of what I've meant when I've been saying over and over since the early emergence of this that the intrinsic operation of our current generation of LLM systems is fundamentally hostile to control by safeguards and guardrails. What's clearly happened here is that after that inevitable slip-up caused the U.S. government to freak out, and after even the NSA, our highest-tech National Security Agency, discovered that the security of even their highest-security networks was vulnerable to Mythos, the only way Anthropic was able to get a vestige of the Fable/Mythos model back online was to go WAY overboard in knee-jerk reaction to anything that might even possibly resemble cyber or code security.

Up until this point our AI models have been growing in power with astonishing speed. In the beginning, we found that just by making a neural network massive enough to contain a useful amount of knowledge, then training it on that knowledge, we achieved surprising utility. Eventually, we were practically unable to scale them any further. So, and where we are now, rather than increasing their breadth through further scaling, now we're increasing their effective depth through looping, essentially reusing the same very powerful network iteratively to amplify and purify its knowledge output. So as a result, with a comparatively startlingly small amount of work, we, humanity at large, now have in our possession a new set of knowledge-manipulating tools the likes of which have never existed before.

But there's a problem with this, which this podcast has been exploring since this revolution in AI first appeared: Large Language Model AI essentially provides a new form and scale of access to knowledge. That's what it is. Knowledge itself, of course, is ethically neutral, it's just knowledge. The question and concern is how the access to that knowledge will be used.

Until now we've been racing forward, creating the most powerful knowledge containment and access systems possible; and we have succeeded beyond our wildest imaginations. But it really

is true that knowledge is power, and only scant passing attention has been given so far to the reality that not everyone can be trusted with unrestrained access to the power of the knowledge that's now potentially available to anyone and everyone.

Commercial AI providers are at a disadvantage and have a problem that the open-weight models do not. It may not be fair, but it's true. To be a commercial AI provider means being saddled with the responsibility of preventing the illegal use and abuse of their commercial offerings. They have no choice. So the commercial AI industry's attention must now shift from "bigger is better" to "how do we truly and accurately control the use of what we've created?" Up to this point, "control" has been an annoying afterthought. That attitude cannot stand, and it must have become the new focus of every provider of high-strength commercial AI models.

We've seen the anti-control arguments that suggest that if, for example, the U.S. were to somehow impose limits upon its internal development of AI, it would be putting itself at a significant strategic disadvantage relative to the rest of our adversarial nations. The solution to this dilemma is not to limit anything on the development and capability side. That should not only be allowed, it should be pushed to continue to go full bore. The solution is to truly develop something we don't have yet, which is highly accurate control over the use of this most powerful AI we're able to create.

So just to conclude: I have no idea how this will be done, and I accept that it's an extremely difficult problem to solve. In fact, I expect it to be significantly more difficult to solve this problem, the one of control, than it was to create the knowledge containment and access system that now requires that control. But I believe that's what has to happen next. You know, everyone's worried about controlling this. I think there should be no limit imposed on how strong these things can get. But they're going to have to have some means of control. And I just, you know, I'm not - I don't pretend to be an AI expert. I'm standing on the outside, you know, with my life's experience of intuition saying, you know, we don't really control this yet. The reason, Leo, all these false positives are happening is that basically they put keywords.

Leo: Yes. It's a classifier. That's exactly how it works, yeah.

Steve: Yes. And that's just dumb. Right? I mean, that's like, how do we get this back online?

Leo: Well, it's not a good way to do it.

Steve: So that we can offer some of Fable 5. But it means that it's really not of any practical use to security researchers. You've got to then - you've got to be chosen and trusted, and then you get to use Fable 5's equivalent, which is Mythos 5.

Leo: Right. It's my guess that the U.S. government said, "We want to keep those capabilities to ourselves. We want the NSA to have them. We want..."

Steve: And selected, trusted, private organizations. Because you know that...

Leo: Yeah, which is sensible. That's sensible.

Steve: Yes. Yes. And so right now there's, you know, Fable 5 that you can use as long as you're careful which words you use. You know, pick your words carefully. And then

Mythos, if you can demonstrate who you are, then you get controlled access to unrestrained super power, essentially.

Leo: Right.

Steve: But also remember that there have been other researchers who've said I found a whole bunch of things using just standard good AI and the proper harness. It's, you know, they've been arguing that, yeah, these latest frontier models are powerful, But if you know how to build a harness, and you know how to prompt, you don't need that in order to get some results, which I think is also significant.

Leo: Right.

Steve: Okay. So I mentioned at the top of the show Opera. They've added something that they call Paste Protect, to protect against, not surprisingly, ClickFix attacks. As we know, this new breed of so-called ClickFix attacks has become the number one means of getting malware into people's machines. It's not only number one, but bigger than all others combined. It alone amounts to more than all other attack techniques combined. I've been crying in the dark for any OS vendor, but specifically Microsoft, whose OS supports cross-application clipboard copy and paste, right, I mean, that's what the clipboard is, to take responsibility, which they should, for preventing or in some way dramatically cautioning their users against putting anything onto the system clipboard from their browser. Or if you do, only allow it to be pasted back into the browser. Not anywhere else.

You know, apparently some pedant inside Microsoft is arguing that this is the way the clipboard is supposed to function. You know, that its entire purpose is to facilitate inter-application data exchange. So it's neither their fault nor their responsibility if such a facility is misused. And, you know, while that's not incorrect per se, on some level it damages Microsoft if Windows further acquires the reputation of being a security disaster.

In any event, we're talking about this today because, while I would argue that this is not their responsibility, some browser vendors are doing what the underlying OS should be doing for all browsers. That is, it ought to be just, you know, fix it in one place, and the proper place is the OS. So today we have news from Opera, who last Thursday posted: "At Opera, user security is a top priority. That's why today" - meaning last Thursday - "we are excited to announce that Opera is the first browser to introduce Paste Protect: a native defense measure against malicious takeover of your clipboard" - and it isn't really a takeover; right? It's, you know...

Leo: Well, that's Microsoft's point, it's what it's supposed to do.

Steve: Right, exactly. It's the malicious use of your clipboard.

Leo: It's the context that's the problem.

Steve: Yes, yes. And they say: "Which includes code injection attacks such as ClickFix. Paste Protect helps identify situations where malicious websites attempt to either replace

something you copied with a malicious version or place potentially harmful commands on your clipboard and later trick you into pasting them into a terminal." Or, you know, command prompt. Or the run dialog. "When any kind of suspicious clipboard activity is detected" - and this makes me a little queasy; right? It feels heuristic, like oh.

Leo: It's looking at your clipboard every time; right? That's a problem.

Steve: Yeah. So like other activity that we don't think is a problem, that's going to be allowed? No. It ought to be nothing from your browser gets into your system. It's just like it ought to be an island. So they said: "When any kind of suspicious clipboard activity is detected," which again makes me uncomfortable, "Opera's Paste Protect warns users before dangerous content can be executed." And you've got a picture of that on the screen, thank you, from the show notes.

Leo: Yeah.

Steve: It says: "Paste Protect. Suspicious content blocked. This site was prevented from copying a potentially harmful command to your clipboard. It's recommended you close this tab." Then there's a "Learn More," or "Close tab safely," "Show content," and then "Hold to copy," that's kind of strange language, "Hold to copy," and then they say in parens "(unsafe)."

Leo: It should be hold to paste.

Steve: Yeah.

Leo: That's weird. I don't get that.

Steve: Yeah, that is. Anyway, so they said: "By introducing Paste Protect [doot-to-do], Opera is taking a proactive approach to defending users against some of the fastest-growing attack techniques on the web. ClickFix is a growing form" - doesn't have to grow much bigger - "of social engineering attack that tricks users into running malicious commands on their own devices. In 2025, it was responsible for over 53% of all malware loader activity, according to a report by cybersecurity firm Huntress." And we shared that at the time.

They said: "While there are extensions that try to mitigate such attacks, and warning systems are featured in some operating systems, Paste Protect is built directly into the Opera browser, the first line of defense before a malicious command even makes it to your clipboard. It is also enabled by default, meaning users are protected automatically without needing to configure anything. Paste Protect combines Opera's already existing Hijack protection feature with a new and unique Injection protection element." And there's nothing else here. Blah blah blah.

So, Opera has stepped up. Yay. That's great. But at the moment, as I mentioned, Opera sits at roughly 2% of the global browser market, and I don't think it's growing, which puts it in fifth place, meaning that market share falls off dramatically, fifth place behind Chrome, Safari, Edge, and Firefox. And the people who need this protection the most, right, are the ones not using Opera because they're just using Edge or Chrome, mostly.

Safari probably number three, and then Firefox in fourth position. So what Opera got right was reminding us that this single easy-to-cure problem alone is responsible for more than half, at least 53% and growing, of ALL user-facing attacks. And since it's so effective, you know it's not going away.

The abuse of this clipboard is not going away. It's going to grow because the attacks are so effective. Everybody's being tricked by this. And there's only one way that's going to change. And it's not by some obscure browser fixing it for 2% of the world. It's Microsoft that needs to pay attention to this and the abuse of their clipboard.

Leo: But I do think that maybe Microsoft's reluctance, you kind of hit it, is that they don't want to be reading the contents of the clipboard on every single...

Steve: They don't have to. But they know where it came from.

Leo: Origin.

Steve: They track - yes. They track the origin. And so you should not be able to paste from the clipboard anything that was copied to it from a browser.

Leo: That makes sense. I mean, I do that all the time because Bitwarden, my password manager, is an extension on my browser, and so I go there, get the password, and then I paste it into a window somewhere. Shouldn't be able to do that, though.

Steve: Right. And so it would make sense for default for it to caution you. And then you could say thank you.

Leo: Right. Oh, no, I know. Yeah.

Steve: And then you could have the option of turning that annoying thing off, like we all wish we could just turn off the cookie reminder.

Leo: Right. So they don't have to look at the contents, just the origin.

Steve: Right, right.

Leo: That's all that's necessary.

Steve: And they do track origin already. They know where contents came from.

Leo: Yeah, that makes sense, yeah.

Steve: Yeah. And again, please, Microsoft, do it. In the meantime, while we're waiting for that, Leo...

Leo: It might be quite a while. But I'm willing to fill some of that time, anyway.

Steve: All right. We all really need a vault.

Leo: Yes.

Steve: Where we can - because no one knows their password any longer. Anybody who does is not doing the right thing. And my wife really hasn't converted yet. And so she says, what is the password? And I go, well, honey, I can't read it to you because, you know.

Leo: But I can share it with you if you use Bitwarden. I can share it securely with you. Yeah, it's really an interesting issue. It's not - they shouldn't really call it a password manager because I think everybody needs an encrypted spot to put stuff.

Steve: Vault.

Leo: A vault. And so it's really much more than that now. And I think it's vital for everybody.

Steve: Well, and with so much of our lives now being digital, not to be morbid, but there is the issue of what happens to a person's accounts when they're no longer able, you know, when they're no longer at the helm. You know, their loved one or their family is saying, how do we deal with this? I mean, so...

Leo: How many times have I got calls on the radio show, somebody with that exact situation. And Bitwarden has that. Actually most password managers now have that capability to say here is my designated survivor.

Steve: Right.

Leo: And then a mechanism for giving that person all of that information. And Lisa, of course, my wife has all of that. And I'm sure Lorrie does, too.

Steve: Yeah. Absolutely.

Leo: Yeah. Because we're getting on, Steve.

Steve: Well, I'm feeling good, and I'm going for Episode 2000. So that's our next milestone.

Leo: Yeah, let's do it.

Steve: Okay. So one place we know Microsoft is paying attention, although they don't seem to be paying attention to abuse of the clipboard by online browser hackers. We know they're paying attention to the continuing antics of the security researcher/hacker who fashions themselves Nightmare Eclipse.

Leo: Oh, lord.

Steve: They're not happy about that person.

Leo: No, no.

Steve: There's been some speculation that it's a young woman who we covered a long time ago, who was like camping in Alaska or something.

Leo: Oh, I remember her, yes.

Steve: Uh-huh. Similar skill set. Worked for Microsoft for a while. Presumably left unhappy.

Leo: Yeah, similar outrage.

Steve: Yeah.

Leo: Maybe it is.

Steve: There's been some online speculation. Anyway, early last week CISA confirmed that the BlueHammer Windows Defender Local Elevation of Privilege vulnerability was now seeing active use. BleepingComputer wrote: "CISA confirmed on Monday that ransomware gangs have begun exploiting a high-severity Microsoft Defender privilege escalation vulnerability that's previously been abused in zero-day attacks." So had been abused; CISA got onboard is the point. "BlueHammer," they wrote, "was leaked by a security researcher known as 'Nightmare Eclipse' in early April' - now, keep track of that, early April - "together with proof-of-concept exploit code, in protest at how the Microsoft Security Response Center (MSRC) handles the disclosure process.

"Microsoft explains in a security advisory: 'Insufficient' - I love this. 'Insufficient granularity of access control in Microsoft Defender allows an authorized attacker to elevate privileges locally.'" What? "Insufficient granularity?" What a crock. That's like excusing a leaky water faucet by saying "insufficient valve closure allows water to escape." Right. Known as a leak. Anyway, what they're describing is not "insufficient granularity," but "improper design." So how about taking some responsibility here, Microsoft?

Anyway, BleepingComputer continues: "Will Dormann, principal vulnerability analyst at Tharros, told BleepingComputer in April that while the issue is not easy to exploit" - meaning BlueHammer - "it gives local attackers access to the [whoops] Security Account Manager (SAM) database" - which, you know, that's the world - "which contains password hashes for local accounts. With this access, they can escalate to SYSTEM privileges and potentially take complete control of the targeted system. Dormann said: 'At that point, the attackers basically own the system, and can do things like spawn a SYSTEM-privileged shell,'" or whatever.

"Microsoft patched the" - and here it is. "Microsoft patched the vulnerability on April 14th" - like two months ago. Wait. Almost three; right? April, May, yeah, almost three. "As part of the April 2026 Patch Tuesday. However, days later, Huntress Labs security researchers revealed that threat actors had been exploiting it as a zero-day in attacks that showed evidence of 'hands-on-keyboard threat actor activity.' So not automated, but a hacker in somebody's system using this escalation of privilege exploit in order to get control of the system.

"CISA added the BlueHammer flaw to its KEV (Known Exploited Vulnerabilities Catalog) on the 22nd of April" - so again, two and a half months ago - "ordering Federal Civilian Executive Branch agencies to patch their Windows devices against ongoing attacks using that within two weeks." Which would bring it to May 7th, so by May 7th. "CISA warned at the time: 'This type of vulnerability is a frequent attack vector for malicious cyber actors and poses significant risks to the federal enterprise.'

"While Microsoft," writes BleepingComputer, "has yet to tag this security flaw as exploited in attacks" - why bother at this point, everybody else knows - "CISA has now also flagged it as exploited in ransomware campaigns in a Monday update to its KEV Catalog." Which should surprise no one. "In recent years, CISA has flagged eight Microsoft Defender vulnerabilities that have been exploited in attacks, with two of them also targeted by ransomware gangs."

Okay. So what's curious about the timing of this to me is that this flaw was patched way back on April 14th. So we've had the second half of April, followed by all of May and all of June, because here we are on July 7th. So how is it that there are any Windows systems that have not been updated since April's, May's or June's Patch Tuesdays, or any time since? No one, as we've said, no one deserves being attacked. But "asking for it" is what this looks like. So the only explanation I can find would be some overzealous enterprise IT admin who's decided that allowing Microsoft to have unfettered update rights over the systems they oversee is a bigger problem than patching actively exploited vulnerabilities. And I think they're wrong.

Or perhaps this hypothetical IT person is overworked and is just asleep at the switch. Whatever. You know, they intended to "get around to it," but didn't. Whatever the case, the idea that Windows systems are being successfully exploited two and a half months after Microsoft made patches available, and those systems will be, if they're allowed to, will be trying to update themselves. Somebody is saying no, no no no no no. Anyway, it's really unfortunate and so avoidable that this is happening.

Having Nightmare Eclipse posting proofs of concept for these prematurely disclosed vulnerabilities I'm sure made matters worse. Made it trivial for the bad guys to grab an example of how to do this and then start using it. But never updating Windows is also a no-win strategy. As everyone has seen, the dynamics of the way the world has changed has really moved me in the direction of let updates happen. The incidence of them causing a problem, while not zero, is low enough that the risk imposed by not doing it is far higher.

Okay. So, Leo. Vint Cerf is certainly not a household name. I'm sure that none of the hosts of "The View" would recognize it.

Leo: No.

Steve: No.

Leo: But we do. We do.

Steve: Yes. Certainly a name that is revered among the techies who have been present throughout the birth of the Internet. During a recent conference where Vint Cerf's pending retirement from Google was noted, he had some interesting things to share about where he sees things today and where AI may be headed. TechCrunch covered this under their headline "The 'Father of the Internet' is finally retiring." They wrote: "Vinton Cerf will step down from his role as Google's Chief Internet Evangelist next week, marking the conclusion of one of the most influential careers in technology history." And yes, I would argue that's not overstated.

They continue, writing: "Speaking via video feed at the Open Frontier conference hosted by the Laude Institute, Cerf was recognized by Dave Patterson, the UC Berkeley professor best known for co-developing RISC processor architecture. Patterson said: 'Vint has been at Google more than 20 years, and he is retiring a week from today. So I think we ought to give him a round of applause for a relatively good career.'"

Leo: Yes. Yeah, relatively.

Steve: Yeah. Relatively.

Leo: TCP/IP, big deal.

Steve: That's right. "The room erupted in cheers."

Leo: Aww.

Steve: So it was neat. "A Google spokesperson confirmed that Cerf will be stepping down from his role at the company. Cerf, now 83, and collaborator Robert Kahn are credited with being the architects of the networking protocols that became the Internet we know today. His work developing and popularizing TCP/IP, the basic set of rules" - this is TechCrunch writing - "that lets different computer networks talk to each other, beginning in the late 1970s, has been recognized with numerous honorary degrees, the Presidential Medal of Freedom, and a Turing Award, among other honors.

"Since '05, Cerf has served as vice president and chief Internet evangelist at Google." TechCrunch had in parens: "(At this point, we can safely say the Internet has been fully evangelized...)"

Leo: I think so.

Steve: Yes. And they said: "(...for good or otherwise.)"

Leo: Job well done.

Steve: That's right. "Cerf," they wrote, "was speaking on a panel alongside other computer scientists known for their work on durable open source projects, including Patterson; Francois Chollet, creator of the Keras deep-learning library and co-founder of Ndea; John Ousterhout, the Stanford computer scientist behind the Tcl" - you know, TCL - "programming language, who also co-founded Electric Cloud; and Matei Zaharia, who is Databricks' co-founder and chief technologist. They offered advice about what it takes to build open source systems that survive - advice that's increasingly relevant as founders bet on open infrastructure for the next wave of AI products.

Much of the conference's discussion focused on the problems with the centralization of advanced models in a handful of well-resourced labs, in contrast to the decentralized world of the open Internet that made Cerf's own protocols so durable. However, Cerf predicted that the rise of AI agents, software that can act autonomously and coordinate with other software, would push tech companies back toward standardized protocols. Cerf said: 'The agentic model of AI, with multiple agents from multiple sources interacting with each other, is going to force composability, and a requirement for interoperability and standardization.'

"If he's right," writes TechCrunch, "the companies that define those interoperability standards early could end up with outsized influence over how the agentic economy actually works, a dynamic not unlike the early Internet protocol wars. While other panelists speculated that natural language communication between LLM agents would be sufficient, Cerf predicted formal standards would be required. He said: "I don't think English is going to be the best choice. There's a flexibility in it, but there's also ambiguity, and I think precision for inter-agent interaction is going to be very, very important.

"An agent really needs to be sure the other agent understands what it is that they just agreed to do together. Remember the old telephone game where you whispered a message in someone's ear, and by the time it got to 10 people away, the message was totally different? Imagine a bunch of agents talking to each other in natural language, you know, that's kind of terrifying."

Leo: I don't have to. We have a Discord chat with four agents talking to each other, and it's cuckoo. It's really out there. You know, OpenAI has a standard, an API standard for how you talk to Codex that everybody uses. Anthropic has the MCP standard. I think it's already happening.

Steve: Yeah.

Leo: I think he's absolutely right.

Steve: Yeah. And then the article concludes, saying: "In a more lighthearted moment, Patterson" - remember the UC Berkeley inventor of RISC - "recalled meeting Cerf, known

for his wardrobe of three-piece suits, as a grad student in the '70s. Patterson said: 'He's always been the best-dressed computer scientist I've ever met. My memory of Vint is that he came as a grad student with a shirt and tie in the '70s.' Cerf replied: 'It's absolutely true. I even had a vest, and for some reason I always wanted to stick out, and instead of having long hair and something in my nose, I thought just dressing differently was one way to do it.'"

Leo: He always wore a three-piece suit. Here he is, I interviewed him in 2014 in his three-piece suit. He always did wear that. He was a very well-dressed fellow. And a real gentleman. Genius. Just a genius, yeah.

Steve: So yeah. I would assert that Vint certainly did stand out, but not due to his choice of wardrobe, but because he turned out to be so completely correct about many of the design decisions that grew into the Internet. Through the more than 20 years of this podcast, our listeners have often heard me shake my head in wonder of the early genius that designed the Internet. It's easy for us to forget that there were many early competing and proprietary alternative protocols like Novell's IPX/SPX, Apple's AppleTalk, DEC's DECnet, Xerox's XNS, IBM's Token Ring. But TCP/IP was also present, and it was open, free, and worked. No one owned it. And when Windows 95 shipped with a built-in TCP/IP protocol stack, that was pretty much the rest of the game. It's like, okay, it's done. It's out there now. Everybody has it.

Leo: And by the way, he gave it away; right? He didn't try to make money on it.

Steve: Oh, absolutely. I mean, the whole concept was here's how you do DNS. Here's how you do IP. Here's how you create a connection-oriented association between two IP endpoint addresses. That's called TCP. If you don't want connection, you use UDP. I mean, yeah, it was all open protocol. And, you know, and he was an early evangelist of this.

Leo: Yup.

Steve: And it worked. And I think that there was a very early stack called Trumpet. Trumpet Winsock was...

Leo: Oh, yeah, Trumpet Winsock, I remember that, yeah.

Steve: And I believe that that's what Microsoft grabbed and incorporated into Windows 95.

Leo: That makes sense, yeah. Yeah, yeah.

Steve: You know, we're going to talk about Chrome 150. But we're at an hour in. Let's take another break so we don't get too far ahead of ourselves.

Leo: We absolutely best do that, Mr. Gibson.

Steve: And then, oh, wait till you see what happened. Oh, goodness.

Leo: Really. Okay. Oh, that's a tease. I don't know what it's a tease for, but that's a tease. That makes me want to stay tuned. Now back to Steve.

Steve: So as the United States reaches 250, Chrome hits 150. I went to bring up the page of updates to get some sense for what Chrome 150 had wrought. So I clicked on the link which my system's default URL handler, Firefox, of course, then began to render. And the page would not come up.

Leo: Too big?

Steve: Yes, actually. That's exactly right, Leo.

Leo: Wow.

Steve: I received a little Google - it does now. I just tried it. I don't know what was going on at the time.

Leo: 433?

Steve: Yes.

Leo: Oh my gosh.

Steve: Yes. Just scroll that sucker. I mean, it scrolls and scrolls and scrolls and scrolls.

Leo: And by the way, many of these critical CVEs.

Steve: Yes. Yes.

Leo: Holy cow. Oh my god.

Steve: It's just - it's astonishing.

Leo: This has to be Fable or Mythos. This has to be.

Steve: That's exactly what we are seeing. Overall, I don't want to forget to mention this, there's probably a place where I had intended to, I'm seeing this everywhere. You probably got a notice that there were critical updates for your Synology.

Leo: Yeah, yeah.

Steve: I just updated both of mine. I believe what we are seeing is what we saw pre-Y2K.

Leo: This is great.

Steve: Which is that everybody is using AI on their own in the privacy of their own R&D and the development shops, to find and fix problems.

Leo: That's all from one version of Chrome.

Steve: I know.

Leo: Unbelievable.

Steve: I know. It's astonishing. And there have been a couple other things where I've had updates, and I thought, wow, that's a lot of updates. And I went back and looked at the previous one, where there was like one thing fixed. The one before that, one thing fixed. The one before that, one thing fixed. And then this one is like 27. So the feeling I'm getting is that what we had hoped would happen is happening, which is quietly, without a lot of fanfare, everywhere, the word has gotten to people writing software that they can invest in some tokens, and they can get their software fixed. And we're seeing updates far more comprehensive than they have been historically.

So the little page here says the Chrome team is delighted to announce the promotion of Chrome 150 to the stable channel for Windows, Mac, and Linux. This will roll out over coming days and weeks. Chrome 150.0.7871.46 (Linux) and either .46 or .47 for Windows and Mac, they said, contains a number of fixes and improvements. Well, there you have 433. That's a number. It's got three digits.

Leo: A number with lots of digits.

Steve: That's right. So they said a list of changes is available in the log. Watch out for upcoming Chrome and Chromium blog posts about new features and big efforts delivered in 150. I'll talk a little bit about that. Then, under the section titled "Security Fixes and Rewards," they write: "Note: Access to bug details and links may be kept restricted until a majority of users are updated with a fix. We will also retain restrictions if the bug exists in a third-party library that other projects similarly depend on, but have not yet fixed." They said: "This update includes 433 security fixes." Not just random, like, oh, the font was the wrong size. Oh, speaking of which, there is a new - they are now supporting the feature in CSS where the font will scale itself to the area provided.

Leo: Oh. Nice.

Steve: Which isn't that handy, yes.

Leo: Yeah.

Steve: As far as I know, they're the first browser to do that. Anyway, they said: "Please see the Chrome Security Page for more information." If you can get it to display. So, yes, 433 security fixes. And as we've said, we've both seen, the page just scrolls and scrolls and scrolls. So it's sorted by severity with the most severe first. So at the top we see that all of the first 20 of these 433, which were all found by Google themselves, as were most, but the 21st one wasn't, so I'll get there in a second. All of those first 20 are Critical. So 20 Critical security flaws fixed. And running down...

Leo: This isn't like some brand new application that's not being actively maintained by the best engineers in the world.

Steve: This is Chrome. Chrome.

Leo: This is Chrome. Wow.

Steve: Yeah. So we see Use After Free in Extensions. Use After Free in GPU. Use After Free in ANGLE. Type Confusion in Dawn. Insufficient validation of untrusted input in iOS Web. Use After Free in WebUSB. Use After Free in Chromoting. Insufficient validation of untrusted input in ANGLE. Insufficient validation of untrusted input in Skia. Use

After Free in Dawn. Use After Free in Browser. Use After Free in Views. Use After Free in Views. You get the feeling that they put a memory tester on this? Use After Free in Skia. Use After Free in Bluetooth. Out-of-bounds read and write in Dawn. Use After Free in Ozone. Heap buffer overflow in Skia. Use After Free in Chromoting. And a Use After Free in Fullscreen.

So it's pretty obvious that some sort of new tooling has been quite successfully deployed against the Chromium codebase to discover 433 previously unknown problems. They didn't fix these last month. They didn't know about them. They didn't know about them last month. As we know, a Use After Free, where an attacker is able to control the data that's used, is a premier means of remotely taking over a system. And as we frequently observe, today's web browsers present the largest and most vulnerable attack surface to the Internet since by design they execute untrusted code supplied by not only every site the user visits, but every third party that wishes to supply their own content.

Leo: [Crosstalk] make you terrified.

Steve: It should. It is utterly insane. It is insane.

Leo: How did we get here?

Steve: But it's what we are all doing today.

Leo: Oh, my god.

Steve: Yes. How did we get - how did this happen? Right. So next up under those 20 critical vulnerabilities that have all been repaired with the release of Chrome 150 we have the first "High," it's rated High vulnerability. High-stakes vulnerability. And this one was interesting. It's been assigned the CVE of 2026-14382, and its description is: "Insufficient validation of untrusted input in ANGLE." What makes it interesting is that its anonymous discoverer and the reporter of this earned him- or herself a bug bounty of a cool quarter million dollars in return for reporting this.

And this brings me to an observation that I have missed until now. We've talked a lot about bug bounties and about the feasibility of, you know, followers of this podcast perhaps earning sufficient income on the side from finding and reporting bug bounties to significantly augment and maybe even supplant their "day job." The acknowledged problem with that is that finding exploitable vulnerabilities at that time, when we were previously talking about this a year or two ago, required both significant expertise and potentially really significant amount of time invested. But as we've been covering for months now, AI has now largely eliminated both of those limitations.

Near the start of what was happening, I observed that the future of Pwn2Own competition and programs like HackerOne were probably endangered. And in fact last week we saw five of the six Pwn2Own Berlin contestants drop out after Mozilla patched a single flaw that five of those six were all depending upon.

Leo: Oh, they were counting on it.

Steve: It's like, whoops.

Leo: Oh, boy.

Steve: But when I spoke of bug bounty programs disappearing, I was talking about a future after the legacy of past troubled software had all been cleaned up. That's not today; that's not yet. That's some future day. THIS moment in time is the optimum opportunity for someone who has the motivation to score some available bug bounty while also helping to improve that software and beat the bad guys to the punch. The time is now, while AI vulnerability discovery is still in its infancy and bugs are so numerous that web pages attempting to list them are unable to load before timing out.

If you may have considered this before, you may wish to consider it again. The game has changed, and that same lowered bar that has the world worried that bad guys are going to take advantage allows the good guys to, also. And as we've seen, it's not necessary to have access to the latest and greatest expensive models. So dive in. Hang out in forums where this is being discussed. You'll learn a lot, and who knows what might happen? You might make some money.

The other changes that Google brought to Chrome 150 were the addition of post-quantum crypto. Chrome now supports something called ML-DSA for TLS connections, something Vint never thought about. ML-DSA is Module-Lattice-Based Digital Signature Authentication. Back during its development we talked about it because it was known as CRYSTALS-Dilithium, where "CRYSTALS" is the acronym for Cryptographic Suite of Algebraic Lattices. But once the algorithm was approved by NIST, it was formally renamed to the much more boring ML-DSA.

ML-DSA is a general-purpose digital signature scheme meant to replace RSA- and ECC (Elliptic Curve Crypto) based digital signatures. Its performance is already on par with previous schemes and will likely improve going forward. The downside is that the signatures and public keys are significantly larger than their non-quantum-safe predecessors, which we're still using. This creates potential resource issues, you know, resource-size issues that must be accounted for in the systems' design and deployment. So it might break some space allocation assumptions in the short term and require a bit of reengineering. You know, it's not going to just, like, break things because both sides have to agree upon their ability to handle the signature and public key sizes. Once they both do, then it's like, hey, we get to use post-quantum crypto.

Chrome 150 also adds support for the FIDO Alliance Credential Exchange standard in Chrome on Android, and also a new "Always use secure connections" mode, as well as some new UI elements including new icons, context menus, and settings. The most interesting new feature for us is probably Android Chrome's support for what essentially is Passkey exchange. This was a well-thought-out solution by a committee of designers that included representatives from 1Password, Dashlane, Bitwarden, NordPass, and Google. They got it right since actually it wasn't really very difficult to get it right. While it's really just a matter of securely encrypting a bundled-up blob of metadata, the specific composition of that metadata needed everybody's input to keep from missing something important. Now we just need it to be supported everywhere.

The spec, which was published two years ago, suggests that now it's time to get it supported so that Passkeys and, you know, essentially all the data that is a Passkey can be securely moved from one client to another. It's clear that's going to happen. It'll just take some time, as indeed these things do.

In a bit of miscellany, I noted that everywhere I looked, everybody was covering a little bit of news. Maybe schadenfreude. And since it has already altered current behavior and will almost certainly inform future behavior, I wanted to share the news here: The European Union's highest court, that is, the court of last resort, rejected the final appeal made by Google against a record antitrust fine. Google's argument was that the Euro-bloc was unfairly penalizing innovation. Yeah. Right. They're unfairly penalizing innovation by preventing Google from forcing all Android users to use Google's stuff in order to use Android. Good luck with that argument.

Anyway, the reporting on this reads: "Google will have to pay a record 4.125 billion euros, which is currently around $167 billion in antitrust fine over anti-competitive practices after the European Union's top court ruled against an appeal by the tech giant on Thursday." That's last Thursday. "The European Commission initially imposed a 4.3 billion euro penalty back in 2018" - so fully eight years ago - "accusing Google of abusing its Android system's market dominance by requiring all phone makers to pre-install Google Search and Chrome. The EU's executive branch accused Google of restricting competition while imposing the bloc's highest antitrust fine ever.

"The EU's General Court upheld the findings in 2022, but they reduced the fine from 4.34 billion to 4.125 billion." Not much of a reduction. "Google then appealed to the Luxembourg-based Court of Justice, the EU's highest court, which has now sided with the bloc's antitrust enforcer. The Justices of that court said: 'The appeal brought by Google and its parent company Alphabet against the judgment of the General Court is dismissed, thereby confirming the penalty imposed for Google Search's abuse of a dominant position in the context of the Android operating system.'

"Google claimed the case was unfounded, saying that the sanction penalized innovation and that Android users were free to download rival apps. Google said in a statement: 'In any event, we adapted our agreements to comply with the initial decision back in 2018'" - that's what I was referring to when I said "behavior had been altered" - "'and we

remain focused on continued innovation and openness for our users, partners, and developers.' Earlier, Google had also accused the EU of being blind to practices by Apple pushing its own services on iPhones." It occurs to me that Google was maybe not paying attention to what the EU did with Internet Explorer, Leo, way back in the early days of Windows. Which, you know, very similar outcome.

This reporting finishes, saying: "The latest case is one of several antitrust disputes between Google and the EU, which fined the company more than 8 billion euros between 2017 and 2019 over antitrust violations. The EU has other open investigations into Big Tech giants under its Digital Markets Act (DMA), and Thursday's ruling sets the stage for the bloc's regulatory crackdown. Among the other EU sanctions Google is facing for exploiting its market dominance are a 2.95 billion euro fine from September '25 for favoring its own advertising services, and a 2.4 billion euro competition fine for promoting its own shopping services."

So I previously noted during a podcast long ago that there was a shockingly large gap between imposed versus paid fines. This gave the impression that the imposition of the fines were just pro forma, and that they could and would simply be ignored, because that seemed to be what was happening. In looking a bit deeper into this for this reporting, I learned that in every case when a fine is imposed, the challenger of that fine is required to place the entire fine amount into an escrow where it will be held during any series of appeals. So, for example, that $4.6 billion that Google just lost had already been captured by an escrow years ago and has been out of circulation ever since. Upon losing that appeal, the escrow is now collectable by the European Union.

And the EU shows no signs of letting up. Google, Apple, and Meta are all currently contesting fines over various antitrust and anti-competition law violations totaling more than 6 billion euros since the start of 2024. Google is fighting an ad-tech fine of 2.9 billion euros, Apple's arguing against a 500 million euro anti-steering fine, and Meta is appealing a 797 million euro marketplace fine. So the EU is just saying we're not putting up with, you know, any of this behavior which the U.S. seems to be fine with. We're going to require that true competition be supported.

And while we're talking about the European Union, it's worth noting that the ever-controversial Chat Control has, not surprisingly, refused to die. Not surprisingly because it was never fully resolved; and, I could argue, maybe it's never going to be. It is a fundamentally irresolvable sore point. And I don't see how it can go away. "Chat Control," that phrase, of course, is the euphemism that's been given to the requirement to detect and prevent any and all illegal communications among citizens of the European Union. There's just no getting around the fact that the prevention of any illegal communications necessarily requires somehow the monitoring of all communications in order to determine if any are illegal.

And in this regard the EU has apparently painted themselves into their own corner, since any such communications monitoring flies in the face of their own citizenry's privacy laws, which they also have, and which are also absolute. What they want is obviously impossible on its face. They say they want to give their citizens privacy, but that only applies to legal communications. And any enforcement of this requires a breach of all that privacy.

Recall that the previous "Chat Control 1.0" was a voluntary scanning scheme. Few companies complied with it even so. But a few did, and what happened was that and remained optional. But by comparison, "Chat Control 2.0," which is what is now looming, is the pending permanent so-called CSAR - Child Sexual Abuse Regulation. This, if it happened, and it still hasn't gone away, there's a vote next week, this would impose a mandatory scanning framework. This 2.0 legislation has been through trilogue

negotiations since December '25 with meetings held on December 9th of last year, then February 26th, April 16th, and May 16th of this year.

The fifth and supposedly final one, reports say that it just happened. And I saw some reports that said it's about to happen. So something happened on June 29th, but apparently something's also happening next week. We should know what is actually happening shortly. We're currently under Cyprus's Council presidency. So far, any conclusions from these multiple rounds of trilogue negotiations are unknown. No one knows what's been happening behind closed doors. But we should know soon. And as I said, I don't see any way to resolve the conflicting goals that the EU themselves have, let alone what the vendors of communications systems are going to do.

All of the big guys have written to the EU, like Apple, for example, saying, you know, don't do this. There's just no way to do what you want in a privacy-enforcing fashion. There was also a report that I saw which I didn't put into the show notes that mentioned that fully one in four of all AI attempted automated recognition of CSAM material was a false positive. So AI was also not doing a good job of this. So I don't know.

Leo: I think CSAM is just the excuse. I think these governments really do want access.

Steve: Yes, yes, they want to have communications that they can monitor. At the same time, they say they want to give their citizens privacy. You know...

Leo: You can't do both.

Steve: I don't know how you - yeah. I don't know how you resolve that.

Leo: The sad thing is they've changed the rules, which makes it more likely to pass. And I have a feeling this time it's going to get through.

Steve: Yes. And what I've seen noted is that some countries are signaling a new willingness to pass it. The question is what is going to pass, and what will the vendors do?

Leo: Well, we know Signal will leave, speaking of Signal, I think Signal will leave the EU.

Steve: Yes. Signal will leave. I don't know about Telegram. But then, you know, Apple can't...

Leo: I don't think Telegram cares, honestly.

Steve: Yeah.

Leo: They see this as an opportunity, frankly.

Steve: Right, because Signal, their big competitor, will say sorry, we can't offer our services.

Leo: I think really Apple and Android are really going to be the big question marks because both of those have strong encryption with RCS.

Steve: And WhatsApp, same.

Leo: WhatsApp, yup, it uses the Signal protocol. That's right.

Steve: Yup. Wow. Really, really interesting. It just won't go away.

Leo: No.

Steve: A research paper published during the last week of June, so two weeks ago, was titled "Protocol Prying: Systematic Vulnerability Research in the Apple AirDrop and Android Quick Share Proximity Transfer Protocols." Okay, like transferring something between devices that are near each other. Okay. I'm just going to share the abstract while breaking in to comment on the abstract's writing.

They wrote: "Apple AirDrop and Google/Samsung Quick Share are proximity file-transfer protocols used by over five billion devices, yet their application-layer security properties remain largely unstudied because both stacks are proprietary and undocumented."

Okay, now, what? This is an all-too-familiar problem; right? We've got something which is - we're just being asked to believe it's secure, yet there's no proof of that. I mean, I get it, I guess, on Apple's side, that they're saying, well, you know, we don't document these things. We fix them when problems are found. How can you find problems if you force researchers to reverse engineer the extracted code or on the wire protocol? It's just still the wrong way to operate. But we're in a world where we do have proprietary companies offering solutions. Not everything is open source yet.

They said: "Both protocols are reachable from wireless proximity without any prior pairing and process complex serialized content" - uh-oh - "(binary plists, CPIO archives, Protocol Buffers, UKEY2 handshakes). And they do this inside privileged daemons, making them attractive zero-click targets across multiple operating systems."

Okay. So the attacker needs to be within typical AirDrop/Quick Share distance of their target. We know the danger inherent in taking serialized content, assuming that it was serialized by a friendly party, then de-serializing it to restore its original structure, you know, kind of reinflating it. De-serialization means interpretation, and we've seen so many times how fraught with error interpreters can be when they are not completely protected. And to make matters worse here, these de-serializing interpreters are running inside privileged daemons, meaning that any code an attacker may be able to cause the interpreter to execute will have system level privileges. It's in the kernel. So no need to also elevate. You're already elevated if you can get code to run.

So they continue, writing: "We perform the first cross-platform reverse engineering and protocol-aware fuzzing study of both stacks. We reconstruct AirDrop's seven-layer state machine and DVZip adaptive compression from binary analysis, build AirFuzz, a protocol-

aware fuzzer that mutates pre-compression representations, and complement it with targeted handwritten analyses of Samsung's Quick Share service and Google's Quick Share for Windows. We discover a total of six vulnerabilities. Three are pre-authentication issues in macOS/iOS AirDrop. The first is Swift fatalError DoS in the HTTP path router; the second is an unbounded XML plist recursion in Foundation; and the third is a NULL dereference in Network.framework's HTTP/1.1 parser.

"We found two protocol-layer flaws in Samsung Quick Share. The first is a pre-authentication OfflineFrame dispatch, and the second is a D2D encryption bypass for three frame types. The last problem was a heap Use After Free in Google Quick Share for Windows for which Google awarded a bounty. We responsibly disclosed all findings; Apple, Samsung, and Google have acknowledged the reports."

So these are all clearly bugs and should be readily fixable by their respective publishers. So I'd imagine that we'll be seeing updates addressing them soon. Still, it's annoying that the researchers had to go to such extreme lengths, prying apart proprietary protocols, figuring out what was going on, creating a fuzzer to essentially attack the interpreters of those protocols, find problems, reverse engineer the problems. Like, okay, Apple's not doing that. There ought to be some way to cross that Rubicon. I don't know what that would be. But I do know what it's time for, Leo.

Leo: Oh?

Steve: And then we're going to talk about, at some length, bypassing Anthropic's and OpenAI's guardrails.

Leo: Time for our hydration break, as they say in the World Cup. He's got - what are you drinking, orange juice? What's wrong with you?

Steve: Diluted three-to-one with water, yeah.

Leo: Oh, okay. No more coffee? You've had enough?

Steve: I'm running around so much, as I think I mentioned last week, the Apple Health app is saying, what is going on with you? Because it's like, it just jumped up in the amount of activity. But it's been great.

Leo: So just what's going on in your heart rate, and it's like - but it's happy that you're working so hard.

Steve: It just is my phone in my pocket, and it says my number of steps have gone from 2,000 to 5,300 because...

Leo: Oh, because you're moving.

Steve: Because I'm moving from one location.

Leo: You have to go back and forth.

Steve: And up and down stairs, yes.

Leo: Stairs is good. Stairs is good for you.

Steve: Okay. So DeepKeep's research paper has the headline "InkJect: The Visual Prompt Injection That Text Defenses Were Never Built to Stop." Their paper reads: "A user asked an LLM to deploy a website" - oh, and I should mention, Leo, this is why an AI's guardrail technology benefits its user as opposed to encumbers its user.

Leo: Okay.

Steve: Why it could be dangerous to use a powerful Chinese AI that doesn't have the same guardrails.

Leo: Oh. Which I am. Okay. So quickly tell me.

Steve: "A user asked an LLM to deploy a website from a public repository. Standard workflow. The model retrieved the code, processed the repository's assets, and built the site as requested. It also created an admin account with attacker-controlled credentials" - whoopsie - "silently embedded in the back end."

Leo: Oh, boy.

Steve: "The user saw none of it. The model flagged nothing. Every guardrail in place treated the task as clean. The instructions that caused this were sitting inside an image file in the repository. The model read them and followed them. That is InkJect.

Leo: Steganography.

Steve: Yes. "The attacker never touched the user's system, the user's environment, or the user's credentials. They uploaded an image to a public repository. That was enough. Direct versus indirect: why the distinction matters. Visual prompt injection is not a new concept. Researchers have demonstrated that instructions embedded in images can manipulate visual language models (VLMs), and some vendors have implemented mitigations for the most straightforward cases.

"InkJect is an indirect variant. The word carries a lot of weight. In a direct attack, the attacker needs the user to interact with a malicious image. The user has to upload it, reference it explicitly, or send it through a channel the attacker controls. That creates a dependency. The attacker needs the user to do something. In an indirect attack, the attacker does not need the user to do anything beyond their normal workflow. The malicious image sits in a public location. When the user asks the LLM to work, the model retrieves and processes the image on its own, as part of the task. The user does not know the image is there. The model pulls it anyway.

"The attack surface for indirect injection is not a specific user interaction. It is every asset the model will autonomously retrieve during the course of its work. Every string, every image, every file could be malicious. So how does InkJect work? The setup is simple. An attacker embeds malicious instructions inside an image and hosts it where a VLM is likely to encounter it during a task. The instructions are designed to evade security scanning while remaining legible to the model.

"When a user asks the LLM to deploy or interact with that repository, the model retrieves the image as part of its normal operation. Through its inherent ability to process images, it reads the embedded instructions and executes them alongside the user's actual task. The user receives a result that looks correct. The unauthorized action has already been taken.

"In our test case" - and they've got five bullets. "First, a user asked an LLM to deploy a website from a public repository. The repository contained an image with embedded instructions. The model retrieved and processed the image as part of the deployment. The hidden instructions told the model to create an admin account with full privileges on the deployed site. The website was deployed as requested. An attacker-controlled admin account was created without the user's knowledge.

"The model did not flag the instruction. It did not warn the user. It completed both the requested task and the unauthorized one, with no visible indication that anything out of scope had occurred.

"InkJect works because of a gap between what security tools can read and what visual language models can read. We found two distinct techniques that exploit this gap. Both defeated the guardrails on all four tested models.

"The first technique employs white text on a white background. Malicious instructions are rendered in white or near-white text against a white background. The image looks blank to any human reviewer. Security scanning tools that evaluate image content for harmful material also miss it. They are looking for recognizable visual content: faces, objects, explicit material, known threat signatures. A white rectangle with no visible contrast registers as an empty image. But the VLM reads it without difficulty.

"This is not a quirk of any specific model. Visual language models are built to extract meaning from images across a wide range of conditions, including low contrast, faded text, and challenging backgrounds. That general-purpose visual capability is precisely what the attacker is using. The model sees what human reviewers and automated scanners do not.

"The second technique uses skewed and distorted text: Some security architectures attempt to catch embedded instructions by running images through OCR before passing them to the model. The reasoning is, if you can extract the text first, you can run it through the same filters that catch text-based injection. But skewing or distorting the perspective of embedded text breaks OCR extraction. The characters are rotated, warped, or transformed enough that OCR returns garbled output or nothing at all. The security filter sees clean input. But the VLM reads the original instructions accurately.

"This is the core of the capability gap InkJect exploits. OCR and visual language models do not read images the same way. OCR looks for well-formed character patterns under expected conditions. VLMs interpret visual content semantically, including text rendered in ways that OCR cannot process. Any security architecture that treats these as equivalent has a blind spot that can be precisely targeted.

"We tested both techniques against four models across two providers." And they are the top two, THE providers. "All four executed the injected instructions. All four would refuse

the same instructions delivered as plain text. OpenAI and Anthropic have both invested heavily in defenses against conventional prompt injection. These systems work. They catch a wide range of text-based injection attempts, flag suspicious patterns, and block instructions that arrive through the prompt.

"InkJect bypasses them because it does not go through the text layer. The malicious instruction lives in an image. The VLM processes it through its visual encoder before any text-level analysis occurs. By the time the model produces output, it has already read the embedded instruction and acted on it.

"The guardrails that stop 'Create an admin account with these credentials' in a text prompt do not stop the same instruction placed inside an image. The same instruction, delivered visually, executes without resistance. This is not a failure of any specific safety system. It is a consequence of where those systems were built to operate. They were designed for models that processed text. VLMs process images, too, and that processing happens upstream of the controls.

"We tested InkJect against four production models: OpenAI GPT-5.2; OpenAI GPT-5.4 Mini; Anthropic's Claude Sonnet 4.6; and Anthropic's Claude Opus 4.5. All four were susceptible to both evasion techniques. Attack success varied across models, but no model in the test set blocked the injected instructions. The vulnerability was disclosed to OpenAI and Anthropic prior to this publication.

"So why does this matter now? Visual Language Models are not being deployed as novelties. They are being embedded into production engineering workflows: repository analysis, code generation, automated deployments, infrastructure provisioning. These systems have real access to real environments. The indirect nature of InkJect means the attack scales. An attacker does not need to target specific users or compromise specific sessions. They need to implant a malicious image somewhere in the path of assets VLMs routinely retrieve. Public repositories, image hosting services, shared asset libraries - any of these is a viable delivery point. One image can affect every user whose VLM retrieves it.

"The attack also leaves no obvious trace. The model completes the user's requested task. Nothing in the output signals that an additional unauthorized action occurred. A user who deploys a repository and reviews the result sees a correctly deployed site. The unauthorized account is there, but they have no reason to look for it.

"Forty percent of generative AI solutions are predicted to be multimodal by 2027. The workflows being built today on VLMs are the attack surface InkJect targets. Security architecture for these systems needs to account for what the visual layer can do, not just what the text layer can do. InkJect demonstrates three things that have practical implications for any organization running VLMs in production.

"First, indirect injection is viable at scale. Attackers do not need direct access to a target user or system. They need access to content that a VLM will retrieve autonomously. Second, the capability gap between OCR and VLMs is an exploitable attack surface. Defenses that rely on OCR to pre-process visual content assume equivalence that does not exist. That assumption is wrong in exactly the ways an attacker needs it to be. And finally, third, text-based guardrails do not transfer to the visual layer. The same instruction that triggers a refusal in text executes without resistance in an image. Until defenses are built to operate at the visual processing layer, that gap remains open. InkJect was discovered by DeepKeep's research team, and the vulnerability was disclosed to OpenAI and Anthropic ahead of this publication."

So my takeaway from this is the need to remain very careful as a user of current-generation AI. Be vigilant and as aware of all the ways that things can go wrong as

possible. Also, as I said, this research suggests that we may be seeing a long-term divergence in the delivered safety of AI models. These very responsible DeepKeep researchers notified Anthropic and OpenAI immediately. And we know that both of those two top-tier commercial AI providers are already in the cross-hairs of the U.S. government. So they're going to be on their best behavior, and they will have certainly already jumped onto this research and either have, or soon will have, closed yet another loophole in their AI's guardrailing. But what about all the other AI systems and models which are now proliferating around the world? Will they care as much? Will they need to? We have become used to guardrailing being put in place to prevent the abuse of AI by malefactors.

But the point of this research is to show that malefactors may be able to use this technique NOT to trick an AI into providing them with knowledge and information that it should not, but to indirectly take some action against unwitting end-users of the AI on the malefactor's behalf. This suggests that the use of any AI whose guardrails may not be absolutely state-of-the-art doesn't just allow its users to get away with things they should not. It also means that its users could be indirectly attacked due to the lack of the fully comprehensive guardrails their chosen AI is using. And that's a big deal.

Leo: So takeaway is I shouldn't just take stuff off the web and give it to my AI, I guess.

Steve: Well, depends upon your AI. I would say it's safe to give it to Anthropic or...

Leo: Fable won't let me do anything anyway.

Steve: ...Claude or ChatGPT. Exactly. They're going to have a kneejerk reaction. But as a consequence of this research, they will now be scanning any imagery that your AI might encounter while it's doing its work. And we talked - remember how long ago we talked about this idea of, like, you could embed commands in images.

Leo: Right.

Steve: Well, so they put in an OCR to check for commands in images. But turns out you can do it in a wacky way that an image processor, which is designed to pull meaning from an image, will still be able to read, even though both Anthropic and OpenAI's image scanners were missing the embedded text. So the embedded text finder is going to get better. My worry is, why would everybody's embedded text finder take the time to get better? And if there's a gap then, it means that there could be problems with using an AI that isn't protecting you against its encounter with deliberately embedded malicious material that arranges to get in. So it's a little spooky.

Leo: Yeah.

Steve: Okay. So it's been a while, as I mentioned at the top of the show, that I've talked about GRC's primary bread-and-butter product, of course, SpinRite. But I had the occasion Sunday, two days ago, to have a proud father experience. My wife Lorrie and I, as we've said, have been wrapping up our move into our new and final home. And she

hates it when I call it that, but I'm sure it is. At this point everything is out of our previous place, but very little is organized in the new place. And over morning coffee we were talking and taking stock in what needed to happen next, how to organize where we are. She mentioned that she needed to do something that would require a PC. And as I've noted before, she uses her iPhone for PC things that would drive me nuts. I'm just like, hey, you have a real computer around the corner. Why don't you use that?

But, you know, it's the computer that she's holding in her hand that she uses. So I get it. And I asked her whether she'd like me to reassemble her PC, get it all plugged back in and ready for her. She said no, that she wanted to use her laptop, like going forward, for all of these such things. She just doesn't really need a PC full-time. A laptop would be enough.

So I told her that I had no idea where the laptop that she had been using was, but that I'd try to track it down. Fortunately, it was in a soft laptop bag that is a brand we both like. The brand is "incase." It's got a distinctive, almost fluorescent green tag, so I was able to find it easily. Since it hadn't been used in many months, of course I wanted to do the right techie husband thing. I needed to join it to our new WiFi network. And of course I would run Windows Update and get it all settled and ready for her. The laptop was a Dell, which is my second favorite brand after Lenovo. So I plugged it in, powered it up, and I watched the Windows 10 roller coaster dots spinning around and around and around.

Leo: We've all been there.

Steve: Oh, my god. And actually, after a while I became concerned that something was wrong. Because, I mean, probably be 200 spins. You know, oh, and the little disk drive icon was lit up solid the entire time. So I thought, okay, whatever it's doing, I mean, it's like it's just - what? Anyway, at the time I was working on today's podcast, so I just let it keep going on the side. Eventually it got itself booted. So I joined it to our brand new WiFi network and ran Windows Update. That also took nearly forever, but it did finally finish. After the update, I noted that booting wasn't any faster. So I decided it was time to run SpinRite 6.1. Time for 6.1 on it. And I've pasted the before-and-after benchmark photos from our SpinRite 6.1 customers before. I've never pasted them from me.

Leo: Your own pictures.

Steve: My own pictures. What I immediately saw before running SpinRite on the drive, but just running SpinRite's built-in benchmark...

Leo: Oh, look at this. Wow.

Steve: Uh-huh. The front of the 256GB SSD was astonishingly slow. It was clocking in at just 13.445 megabytes per second, whereas the middle and the end of the same drive was running the way it should be at 570.5 megabytes per second. So we have 13.4 versus 570. Now, this was the phenomenon that the group of us who were testing SpinRite 6.1 throughout its three-and-a-half-year development became quite familiar with. And I've shared, as I've said, many of our other users' similar reports since then. But I may not have ever had the occasion to witness myself on a real live system that someone I cared about was going to be using.

So I fired it up on Level 3, which is what's necessary to "refresh" the entire storage surface of an SSD, and I watched it painfully run. The Real Time Monitor page of SpinRite lets us watch the individual reading and writing phases, and SpinRite is reading 16MB at a time. 6.3 is able to do that. And it's got a bar where it shows which phase of the work it's on. Where the first one is reading, it may do other recovery things if necessary, and if you're running on different levels. And then the last one, where it's finished, it writes it back. And so it normally would just be - it would just be a flick on the read, and then the writing, as we know, on an SSD always takes longer than reading.

So it would normally - you would mostly see it sitting on writing, and it would briefly flick up to read and then back down to writing. That's not what was happening. It was sitting on the reading phase for a long time, and then dropping to writing, then back up to reading again. So again, it's what I expected, but it was an extreme case of that. So that meant that for, well, I'm getting ahead of myself.

One of the more amazing things that I should mention, that I've witnessed across many brands now of SSD, is how slow an SSD can become while still - and I'm kind of amazed by this - still finally being able to obtain an error-free read. It is retrying, and it is applying error correcting like crazy. But it ends up getting the data back. Those 16MB blocks that SpinRite 6.1 reads at a time are typically composed of eight logical 5K sectors, meaning a 4K byte physical block. So SSDs are actually allocated in 4K pages, 4K physical blocks. So for every 16MB that SpinRite reads, that's 4,096 individual 4K byte physical blocks being read.

And apparently in this case, on this drive, a large percentage of those 4,096 individual 4K byte physical blocks must have needed retrying and rereading and error correction, re-thresholding, whatever it took to finally pull the data off. But I'll be damned if every single block was not finally read without a single error. It might slow WAY down, but the SSD controllers are clearly prioritizing read recovery over speed.

Leo: Which makes sense.

Steve: And that's exactly, yes, that is exactly what we would want.

Leo: Right.

Steve: The problem is, Leo, this is so gradual over time that people adjust to it. They just, you know, it didn't happen suddenly one morning. It's just over time it's just being slowed down.

So anyway, this was going on. It was going painfully slow, so I turned back to work on the podcast and let SpinRite 6.1 grind away on that Dell laptop's SSD. After a couple of hours I looked over and noticed that it was now about halfway done and zipping along. SpinRite had moved through and past the slow beginning regions of the drive into the unused territory where the drive probably knew that nothing had ever been stored there. Now, what's interesting about SSDs is they know that. An old-school hard drive doesn't. But those regions were still fully "TRIMmed," meaning that the SSD's controller did not even need to read from the media since, as I noted...

Leo: Ahhh.

Steve: ...the drive's, yes, its controller would have known that those regions contained no data. Okay. And since there was no point in continuing, I hit Escape at that point to interrupt SpinRite, exited back to the DOS console prompt, pulled the USB boot stick from the laptop, hit Ctrl-Alt-Delete to reboot. And I watched. Sure enough, rather than watching those six annoying roller-coaster dots looping around hundreds of times, unbelievably, they looped around exactly twice, and the welcome screen popped up, and I was in Windows.

Leo: Wow.

Steve: Now, the first thing I did was to enter "optim" into the search bar, and that brought up the disk defragmenter and optimizer. This was not to defrag an SSD, but rather to "re-trim" it. Running SpinRite at any level 3, 4 or 5 will always write to the media, even if the file system is not storing any data in that location. The SSD, which doesn't know any better, will assume that it must keep whatever SpinRite just wrote, even though SpinRite was just writing zeroes. So running an "Optimize" on an SSD after SpinRiting it serves to re-TRIM the SSD, meaning to let it know which 4K physical blocks are important and actually contain file system data and which it never needs and which are actually empty and which it never needs to worry about preserving in the future.

So anyway, we've had many users and owners of SpinRite 6.1 describe their before-and-after experiences with 6.1 and the performance of their machines. But I've never had the experience myself on an actual machine. So, yeah, a bit of a proud father experience here. And my wife now gets to use a snappy laptop that performs just the way it did when it was brand new. So anyway, it was very...

Leo: Nice. Congratulations. I notice the random access time improved, too, yeah.

Steve: Yes. Yes. Because any - during random fetching, any read that happened to fall within the front of the drive would just basically stall, while the drive tried to actually read it.

Leo: Yeah, yeah. Very cool.

Steve: Very cool. Okay. Our last break, and then we're going to talk about who it was who coined the term "apex agentic adversary."

Leo: It sounds like something from "Jurassic Park." But I might be wrong on that.

Steve: Apex predator, that's right.

Leo: Apex predator, that's right.

Steve: That's right.

Leo: All right. On we go with Agentic - what is this? Agentic what?

Steve: The Apex Agentic Adversary.

Leo: Ooh.

Steve: Which is a phrase that we will see quoted by somebody who's sort of famous. So a set - oh, I mean, but this sort of hides the fact that a really bad set of vulnerabilities has been uncovered.

A set of seven severe vulnerabilities have been discovered in the massively widely used FatFs file system library. FatFs, that's file system, FAT file system. It's a pure ANSI C implementation of Microsoft's original FAT file system with [crosstalk].

Leo: File Allocation Table, it stands for.

Steve: Yup. Exactly. And this FatFs code is the solution used by virtually any and all embedded computing devices that have any need to operate any lightweight and broadly compatible file system. Like, for example, a voting machine. I mean, like, anything you can imagine where, you know, you plug an SD card or a thumb drive into it. A router or whatever. If it needs to understand a FAT file system, many of the systems use this.

So the troubles, these seven severe vulnerabilities, were discovered by the security firm runZero. Now, runZero has some pedigree by virtue of its founders. Its primary founder and CEO is none other than HD Moore, a name we've used on this podcast many times. He's the primary developer of the massively popular Metasploit Framework which became the Metasploit Project. Metasploit is the world's most widely used penetration testing framework.

The research paper's co-author is Tod Beardsley, who currently enjoys the title of VP of Security Research along with HD at runZero. Tod was previously the Section Chief for the Vulnerability Response section at CISA. Tod has over 30 years of hands-on security experience with previous IT ops, security and software engineering, and management positions in large organizations including the government, Rapid7 - another company we refer to, like, over and over and over - 3Com, Dell, and Westinghouse, both in offensive and defensive practices. So when these guys notify the world that they have found something serious, they know of what they speak.

The title of their research, which they updated most recently last Wednesday, was "Seven FatFs bugs, one very large blast radius." Yeah.

Leo: Great.

Steve: Uh-huh. Here's what they shared with the embedded computing world. They said: "Heads up! If you ship firmware that touches FAT media..."

Leo: That's you; right?

Steve: Yeah, "...think any removable storage, like USB drives and SD cards, you'll want to pay attention to this. This work was part of runZero's research into long-tail supply

chain bug hunting using LLMs." And they wrote: "We live in the future." Meaning they're just as blown away by what AI's doing as anybody else.

So just to be clear, we have another example of what the world can expect as LLM technology is aimed at old and longstanding code bases. The trouble is when they are also massively widely deployed and enable vulnerabilities that are probably impossible to remediate. Think of all the devices around that might need a removable media of any kind. So this pair tells the story well. They write: "Today, we're publishing seven CVEs documenting several vulnerabilities in the FatFs project, ranging from CVSS from Medium to High." And they wrote: "No Criticals, phew! The affected ecosystem, meaning, for example, things that use this, some major non-hobby platforms like Espressif ESP-IDF, STMicroelectronics STM32Cube middleware, Zephyr RTOS, MicroPython, ArduPilot, RT-Thread, Mbed, Samsung TizenRT, and SWUpdate, with downstream reach into consumer IoT, industrial controllers, drones, crypto wallets, and more.

"FatFs," they write, "is a portable, royalty-free FAT/exFAT filesystem library written in C by ChaN (elm-chan.org)." They write: "It's designed for resource-constrained embedded systems with no OS dependency and is typically compiled directly into firmware. It supports FAT12, FAT16, FAT32, and exFAT, as well as optional Long File Name (LFN) and GPT partition support." So it's been kept up to date. I think the most recent update was about a year ago.

"Because FatFs is small, self-contained, and permissively licensed, it's become the de facto standard FAT implementation for microcontroller firmware. The library is vendored verbatim into official SDKs, RTOSes, bootloaders, and application frameworks - meaning a single upstream vulnerability propagates to every downstream project that copied ff.c." Which is the, you know, the fat file system dot c file.

They said: "This project is a return to a" - this project, meaning their current work now that they're reporting on - "is a return to a security assessment started back in 2017, when a manual audit and multi-day fuzzing effort identified some basic, but not compelling, bugs in the FatFs driver. Nine years later, in March 2026, we revisited this project using Visual Studio Code, GitHub Copilot in 'auto' mode, and some basic prompts, all without any specific loops, harnesses, or skills. The results were surprising. Bugs that were overlooked during the manual audit became trivial to find by using the LLM to automatically build a fuzzer with novel inputs. Not only did this effort find interesting and reportable bugs, it also automated the process of validating that these bugs are actually exploitable across different embedded scenarios.

"So, why do FatFs bugs matter? For the vendors who build on these platforms it's simple: any physical access leads to a jailbreak, especially given the FatFs library's lack of address space layout randomization and memory protection. For everyone else, there are numerous devices where brief physical access by the general public should not lead to a full compromise. For example, security cameras with SD card storage, voting machines with USB file readers, ATMs, and pretty much anything else that has a screen that you expect people to touch.

"FatFs has no CVE history, no security mailing list, no patch notification mechanism. Every downstream project that uses ff.c must discover, triage, and patch these vulnerabilities independently, usually without knowing whether they're affected. That means the window between public disclosure and widespread remediation will be measured in years, not days. The practical attack surface is therefore not one software application or service, but tens of millions of devices across dozens of independent codebases, many of which will never receive a patch.

"The archetypal exploitation scenario is the evil SD card: an attacker with a few seconds of physical access swaps the storage medium in a device, from consumer cameras to

drones, to 3D printers, to a thousand other product families. Every vulnerability in this set is triggerable by mounting a crafted FAT image, which nearly always happens automatically on insertion with no user interaction required. That said, physical access is not the only path.

"Devices that ingest FAT-formatted update packages from a network source such as Over The Air update frameworks and drag-and-drop bootloader updates, are exploitable by any attacker who can deliver a malicious image to the update pipeline. Supply chain compromises, an Agent in the Middle (AitM) injection on a cleartext HTTP update feed, or a malicious image posted to a hobby platform distribution portal. The Over The Air path is fully remote on any device that lacks end-to-end authenticated integrity verification of its update container prior to mounting it with FatFs.

"Therefore, the value presented to attackers is straightforward: these issues are triggerable by crafted FAT/exFAT/GPT images, often through removable media or update channels that get mounted automatically. Note that CVE-2026-6682 and 6683 are both implicated in some Over The Air update processes for firmware. This post is intentionally light on exploit internals. For technical details, proof-of-concept images, harnesses, and code-level analysis, see runZero's companion research repository on GitHub." And I've got a link to them in the show notes, or just go to github.com/runZeroInc and you'll find the directory there.

"The seven findings documented here are listed roughly in order of subjective value to an attacker." And I'm going to skip over them. They're in the show notes for anyone who's interested. We start with a CVSS of 7.6, which is the highest. They called it the "headline issue." And that's that 6682, which they said describes integer overflow in core mount arithmetic which can produce attacker-controlled file-size metadata that downstream code may trust as a read length. In real systems, that can become a heap/stack overflow and code execution. In terms of practical attacker value plus transitive impact, this is at the top of the stack. And so they have actually three that are all - they're all assigned a CVSS of 7.6, where basically they're able to arrange to execute code they provide on an SD card or a USB stick, simply by sticking it into any machine that is using this library to read those things. That's what this comes down to. So it's bad.

They then independently remind us of our favorite XKCD. After enumerating through those seven CVEs, they said: "If you've seen XKCD's 2347 'Dependency' - of course that's the huge construction of different-size blocks - you already know where this is going: one component, maintained in one tiny corner of the Internet, quietly supports an absurd amount of modern cyber stuff."

Leo: exFAT is everywhere. Everywhere.

Steve: Yes.

Leo: You use it probably in your FreeDOS; right?

Steve: Yes.

Leo: Yeah.

Steve: So they said: "FatFs is one of those components. It's compact, useful, and copied everywhere. That's great for shipping products quickly, but less great when memory-safety issues show up in parser-adjacent code that happily ingests untrusted media. This kind of component is even more challenging to deal with from a disclosure-and-fix perspective, in that nearly everyone ends up making local, vendored modifications. So an upstream patch must be validated pretty carefully before incorporating. We made repeated attempts to contact the maintainer, and we invoked JPCERT/CC early on, as well. We never received a response."

Leo: [Crosstalk] is a maintainer, actually.

Steve: There basically isn't. It's just some random guy who checks in once a year to see if, like, they need to do anything.

Leo: It's done. He thinks it's done.

Steve: Yeah. Exactly. It's done. It's finished code. And an audit nine years ago found no problems that were significant, just a few things.

Leo: You know, the fact that you could make the name of the volume so long that it goes into user space, that seems so obvious. How could they miss something like that?

Steve: Right.

Leo: I mean, so these are - oh, well.

Steve: Right.

Leo: Okay.

Steve: So they said: "We never received a response. So this publication is aimed at the people who can still do something useful right now," which is to say, yeah, downstream...

Leo: Hello? Anybody there?

Steve: "Downstream implementers." They said: "Audit your vendored version, audit your wrappers, audit your file-name and file-size handling, and plan for patch updates."

Okay. And now we learn where I obtained the title of today's podcast. HD Moore writes: "Keeping this class of issue quiet in 2026," consider that they were unable to find the vendor, there's no mailing list, no CVE history, there's just, you know, it was just some project, some random guy, ChaN, did, and thank you very much for giving the world a free FAT file system. Unfortunately, it's got some bad bugs.

So HD says: "Keeping this class of issue quiet in 2026 would be almost entirely security theater, now that we've entered the age of the apex agentic adversary. If we can find this kind of thing with some thoughtful application of AI-assisted vulnerability hunting, then so can pretty much anyone else. Sure, this took a little HD Moore-branded stubbornness; but even so, we don't believe these are going to stay undiscovered for very long. It's too good a target space."

Leo: Yeah.

Steve: "And the next researcher probably won't bother with creating validation and reproduction harnesses. When it comes to vulnerability disclosure, I've always believed that I'm merely the most recent person to learn about the issue." Meaning why would he, he's saying, imagine that he's the only one to know? He said: "And that's more true now than ever in this hyper-automated code auditing world. Better to disclose; coordinate where possible; and, in the end, publish in order to give defenders the head's up." And he said: "PS: Looking for exploits? You can find a test harness, a full QEMU-based exploit example, and corrupt FatFs images in the repository."

Okay. So, yes. We have indeed entered a brave new world where the rules of the game are clearly changing. The collapsing cost of novel vulnerability discovery means that many new players, both well meaning and malicious, will be joining the fray. The asymmetry of the security guarantee bites us in this instance. As we know, security must get everything right, everywhere, every time. For insecurity to win, security need only make one mistake in one place once. It's an impossible scenario, but it's the scenario we have today. Someday, thanks to AI, it's clear to me that software will have finally been fixed. AI is going to fix our software. We couldn't do it ourselves. We could write it, but we couldn't understand it because it got out of control.

And that's something that until now we've only been able to dream of. But to get there from here, we have a long and daunting road ahead because there's a lot of crappy software out there. And a lot of it, as HD noted, is never going to get changed. So I'm thankful that the world does have many good guy researchers such as this HD Moore who are interested and intrigued enough to contribute their time, energy, and effort.

And in this specific instance I agree with the route that Harold - that's what the "H" of HD Moore, he's Harold Dwight Moore but just goes by HD. I agree with the route he took. Given that this library is effectively unmaintained, that there's no mailing list nor update methodology, and that they weren't even able to contact its author, direct outreach to all of the library's major known users - and they have a list of those - is all the researchers can do. Let them know that the library they're using would make any future use of it, well, actually all past, but also future, vulnerable. So they could fix it. And depending upon who they are, maybe they can fix some of their installed base. Maybe they can't. Maybe they just need to stick glue into the USB socket so that nothing can mount itself there. So...

Leo: I mean, you have to have auto mount turned on; right? I mean...

Steve: Well, that's the point. Typically these embedded devices do.

Leo: They do because that's how you...

Steve: All you do is, yeah, you just stick the SD card in the camera, and it looks to see what's there. Looking to see what's there takes it over.

Leo: Right. So, yeah. And you probably can't even disable that. You wouldn't...

Steve: Right.

Leo: You don't have an interface.

Steve: Exactly, not a feature.

Leo: Yeah. Wow.

Steve: Because, Leo, why would you ever need to?

Leo: Steve, amazing. While you've been talking, I've been building. And, boy, Fable is amazing. And I'll tell you, what it's been doing is reviewing. So I have it write the plan. Then Opus 4.8 executes. And then it reviews. And like two real bugs were caught during execution.

Steve: Wow.

Leo: By subagents. And this is - it's already doing, you know, that exact kind of testing.

Steve: Yeah.

Leo: They found a race condition, which is pretty amazing. So I just, you know, and Lisa says, well, what are we going to do if it doesn't work? Will you be around to fix it, or are you going to be on the air and not able to fix it? And I said, it's always going to work. It's not going to have any bugs.

Steve: There's a far better chance it'll work than anything that anybody else would ever write.

Leo: This is true. The stuff that she's been using for 12 years. So it's kind of like, you know, come on, the thing you've been using all this time is so unreliable.

Steve: Yeah, you know, and she probably knows the quirks, like, oh, don't click on this.

Leo: Oh, that's exactly right. She knows all the workarounds.

Steve: Yup.

Leo: Like Steve Jobs when he first demoed the iPhone. He knew there was one path and one path only where everything would work. Any deviation, the thing crashes horribly. Yeah, we kind of were in that situation. I'm very impressed with its ability to do this stuff. And actually as it turns out this is a fairly trivial thing because it's a business application. And we're not writing...

Steve: And I think you should be impressed. I mean, I don't think you should be at all shy about being impressed. You know, one of the things that I said from the start is AI is going to inherently be very good at code.

Leo: This is its language.

Steve: Yes.

Leo: And it's a language it's fluent in.

Steve: And plays by rules. It's all logical. It's got a ton of examples on the Internet to have learned from.

Leo: Yeah, that's the key, yeah.

Steve: Yeah.

Leo: It's doing this in Go. I don't have a strong preference of language. Probably if it had asked me I would have said Rust. But it likes Go. Go has concurrency.

Steve: And Go is a good language. And actually that's where your race condition came from was the concurrency.

Leo: [Crosstalk] multiple threads, yeah, that's right. It has that built in. It's also...

Steve: It had to build a lock-in.

Leo: It's also very fast, which is nice. It really - it's a good web language, and we're going to be running this in the cloud. So anyway, maybe, you know, be trivial to say, hey, you know what, can you rewrite that in Rust? I'll be back in a day or two. You let me know.

Steve: Wow. It's a whole different world, my friend.

Leo: It's somewhat addictive because you feel this power to do stuff that in the past, as computer users, we were somewhat disempowered, you know, we were stuck with whatever the FAT library guy wrote. I'm not going to rewrite that. And now we have all this capability we just didn't have before. It's very addictive. It's very exciting. And you can see with agentic, what is it, agentic...

Steve: Apex agentic adversary.

Leo: We're going to be in a whole new world of security, as well.

Steve: The good news, though, as I said, if we just look around at the updates that are coming in from all different directions, like the fact that the Synology fixed a bunch of stuff. Where they'd only had one or two over, like, for the past year, suddenly there were 14. Well, we know where those 14 came from. They're being responsible, and they said, whoa, let's run our software through AI. And now we have a much better result than we did before. Much more attack resistant. So again, I think in the same way that Y2K didn't crash because everybody actually did fix their Year 2000 dependencies, it seems that we have a well-distributed remediation going on across the globe.

Leo: Darren in our Club TWiT says, oh, all you have to do is add a Claude side panel to the app. And then if Lisa has a problem, she doesn't have to ask you. She could just say, "Could you fix that?" and Claude will do it. It's a little buddy in there that can do it. Wow. What a world. What a world we live in. You know, I'm just glad that you and I made it to this point so we can watch this.

Steve: Exactly that. This is just too significant.

Leo: I feel like our grandparents who grew up in the age before man could fly, and end up living to an age where you could fly to Europe in a jet airplane in four hours...

Steve: Or the moon.

Leo: Or the moon. Imagine. You're born in 1900. They had - Orville and Wilbur Wright hadn't yet invented the airplane. And you're at my age, and you're watching us land on the moon in 69 years. That's kind of where we are in. We're in this hyperspeed of change. And some people find that very upsetting. I understand why. But then there are people like you and me, and I imagine most of our audience, who go, yeah, this is cool.

Steve: And AI is going to accelerate that rate of change. It is going to.

Leo: It already is, yeah, you're right. That's part of the - most interesting part of it, yeah.

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/