



Attestation

Description: Websites can place high demands upon limited CPU resources. Microsoft appears to back away from its security commitment. What's Windows 11 26H1, and where do I get it? Chrome 145 brings Device Bound Session Credentials. More countries are moving to ban underage social media use. The return of Roskomnadzor. Discord to require proof of adulthood for adult content. Might you still be using WinRAR 7.12? I was. Paragon's Graphite can definitely spy on all instant messaging. 30 malicious Chrome Extensions. 287 Chrome extensions from spying on 37.4 million users. The first malicious Outlook add-in steals 4,000 users' credentials. Some AI "vibe" coding thoughts. What I just went through to obtain a new code signing certificate.

High quality (64 kbps) mp3 audio file URL: <http://media.GRC.com/sn/SN-1065.mp3>

Quarter size (16 kbps) mp3 audio file URL: <http://media.GRC.com/sn/sn-1065-lq.mp3>

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We have lots to talk about. A big change to Chrome, bringing something called "device bound session credentials" to your browser. Steve's going to talk about how you can prove you are who you say you are when it comes to your code signing. And bad news about more than 200 Chrome extensions that were spying on more than 34 million people. That and more coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 1065, recorded Tuesday, February 17th, 2026: Attestation.

It's time for Security Now!, the show where we cover the latest in security, privacy, how things work, sci-fi, and whatever else this guy here is up to. Mr. Steve Gibson, welcome.

Steve Gibson: I do try to keep us mostly on track. Though, you know, the world is not monotonic. So, you know, lots of things going on.

Leo: You're a polyglot. You know everything. So it's nice to talk about all these things.

Steve: Certainly don't know everything. There are things I know a lot about, and things that I'm interested in learning more about.

Leo: Yeah, yeah.

Steve: So, but yeah, I'm definitely curious. I just, from my first moments of awareness, I wanted to know how things work. That's what I want to know, how things work.

Leo: The important mindset, yeah.

Steve: Yeah.

Leo: I agree, yeah.

Steve: And so I lost my fear of looking inside to go, oh, look, that little cam goes this way, and that pushes that lever over here, and that causes that to drop down. And, you know, I was very good at the game, the board game Mousetrap for that reason, back in our youth.

Okay. The elephant in the room is the 28-page security research paper that was recently published after I put the show this week together.

Leo: I felt so bad because I sent it to you, and I said, oh, Steve, I know you're done.

Steve: Well, you and about 50 of our listeners.

Leo: I bet. Because we've got to know.

Steve: I mean, so of course I've been very impressed with how in touch our listener community is because it was like, oh, Steve, oh my god. Okay, you know, what does this mean? So to do it justice, I will answer that question next week.

Leo: Okay, good.

Steve: The good news is no hair on fire. It's not the end of the world.

Leo: This is about password managers.

Steve: Yes, sorry. So ETH Zurich, the researchers there whom we've spoken of many times as a consequence of their work, and some Italian guys, they got together and did a deep dive into the consequences of server-side, i.e., "cloud" is now the new term, attacks on three popular password managers, you know, browser-based password managers, Dashlane, LastPass, and Bitwarden. LastPass of course a previous sponsor and favorite of ours until they screwed up - and actually it was on the server side, so that's kind of interesting - and scared us all; and Bitwarden, a current sponsor of the TWiT Network. One of the...

Leo: But this was everybody. It was 1Password, it was Dashlane, it was everybody, which I thought was interesting.

Steve: Yes. Although they did focus on those three. And I thought it was interesting. First of all, Dashlane and Bitwarden both responded, Bitwarden with a thanks for the analysis. And Bitwarden commented that, as a consequence of the fact that they were an open source system from one end to the other, the job of the security researchers was far more enabled because it wasn't necessary for anyone to reverse engineer their stuff. You know, they're wide open.

Okay. So again, as I started off saying, no hair on fire. I'll give us a complete readout of it next week where we look at in detail what it was that was found. Both Dashlane and Bitwarden are either immediately already responded to the issues, again, none of which were, I mean, these were, like, worst-case, if a bad guy completely took over your server infrastructure, what could be learned? And to give you some feel for it, there was an instance, I think it was with Dashlane, where they were deliberately supporting older crypto standards for the sake of backward compatibility. So if you could - if you took over the server infrastructure, forced a protocol downgrade to use the oldest supported crypto, and the user had a weak password, then it might be possible to decrypt their vault. So again, it's not like...

Leo: That's a lot of ifs. That's a lot of...

Steve: And that's my point is that these were like that. I mean, definitely useful. The researchers commented that they were somewhat surprised that this hadn't been done before. It's like, you know, here we are running around all using our password managers, just sort of thinking, well, seems great. But, you know, this is really the role of independent research, which an open source facility like Bitwarden offers makes far more possible. So next week I'll have the whole update. But just I wanted to thank our listeners and, Leo, you for bringing it to my attention also yesterday. It's next week's topic.

Leo: Good, good. Yeah, because we want to know, I mean, you know, I know Bitwarden's a sponsor, but we want to know. As is 1Password.

Steve: Yeah. Well, and, you know, we were bullish on 1Password until - and it's interesting too, because this has an echo feeling, like it was 1Password's not updating the...

Leo: No, LastPass. You're talking about LastPass.

Steve: I'm sorry, I'm sorry, I'm sorry, yes. It was LastPass's not updating the iteration level of their PBKDF, which, you know, Password Based Key Derivation Function, which got them into trouble. And it's like, laziness or like a fear about breaking something. I mean, there are - I think if there's a lot of legacy code, and the people who wrote it are gone, and new guys are coming along going, oh, yeah, you know, let's - we don't want to be responsible for breaking something. So there's kind of a, like, leave it alone if it's not broke. But unfortunately, the way crypto standards are going, you do need to keep rolling forward because the attacks are getting stronger. Anyway, we'll look at that

completely. But no one needs to, like, fear that this means they have to go back to a paper pad for writing their passwords down. No.

Today's topic for Podcast 1065 is Attestation. I want to share an adventure I've just survived, which I will get to at the end of the podcast.

Leo: Oh, boy.

Steve: Really, really interesting what's going on in the industry. And understandable. So I will get to all that. We're going to talk about websites placing high demands upon limited CPU resources. I realized after we talked about this last week, Leo, what happened with AI.com and why that graphic you showed that showed that Cloudflare was just all fine, but the host was unresponsive out at the end.

Leo: Right.

Steve: What happened? Because I realized we've talked about this before, but I just - it didn't hit me until I was thinking about it later. Also, in a worrisome move, Microsoft appears to be backing away from its commitment to security. Okay. Also, what's Windows 11 26H1, and where do we get it? Chrome 145 is released, and it brings something known as device-bound session credentials finally out to the mainstream world. We talked about this, I think it was last April, but we're going to, you know, circle around again because now it's here.

Also I had a blurb, and I heard you guys, you've been talking about this in a number of different places, Leo, more countries moving to ban underage social media use.

Leo: Oh, yeah, it's an epidemic.

Steve: And Discord to require proof of adulthood for adult content social media use. And there's been a little bit of overreaction to that. So we'll tamp that down. We have the return of Roskomnadzor, which, you know, no podcast would be complete without that. Also, might you still be using WinRAR 7.12? I was.

Leo: What?

Steve: Yeah. It caught me. So we're going to have to make sure nobody is. Also we now have proof that Paragon's Graphite smartphone spyware can definitely spy on all instant messaging apps.

Leo: Oy.

Steve: A researcher discovered 30 malicious Chrome extensions, and a different project found 287 Chrome extensions which were spying on 37.4 million of their users. So this is really a problem that we're going to talk about. The first malicious Outlook add-in has stolen 4,000 of their users' credentials. I've got some thoughts on AI vibe coding. And then I'm going to go through what I just survived obtaining a new code signing

certificate. So I think - and of course we have a fun and interesting Picture of the Week. So, yeah, I think we've got a good one here for February.

Leo: As usual. All right. Let's talk about your Picture of the Week, Steve.

Steve: So I gave this picture, which didn't have a caption, a caption.

Leo: Okay.

Steve: Placing unconditional trust in technology can lead to mistakes.

Leo: All right. I'm going to scroll up now, and I'm going to see it for the first time along with you. [Laughing] Oh, that's good. I never thought of that.

Steve: So we have a picture of a security camera which, mounted on the ceiling, as originally pivoted around as you'd expect to be surveying the room that it is monitoring so that it knows what's going on. Apparently the people who populate that room decided, you know, we don't really want to be having this camera looking at us all the time. So it's clear, you can see how this picture came about, someone got up on a chair or a ladder or something and took a photo of the room from the vantage point of the camera, printed it out on 8.5x11 sheet of paper, stuck the paper on the wall behind the camera, and then swung the camera around so that it's looking at the paper. And of course, so as I said, placing unconditional trust in technology can lead to mistakes.

Now, so the people in the room are no longer being surveilled. They can be doing anything they want to be. And meanwhile, the security people in some room with lots of monitors are looking at that going, when is this guy going to come back from the bathroom?

Leo: It's very quiet, yes.

Steve: Why is his desk, you know, how long is his lunch? You know? And so, anyway, I got a kick out of the picture. It's just...

Leo: Love it. It's very...

Steve: Again, you know, there are all kinds of instances, right, where we adopt technology to save us some trouble of some sort. And it turns out that people who don't want to be encumbered by that come up with a simple workaround. So, you know, like sticking bubblegum on the camera lens or, you know, something. Easy to mess with the technology.

So as I said at the top of the show, last week we noted that the fact that during Sunday's Super Bowl, the company with the very expensive \$70 million domain name, AI.com, had been DDoSed by their own advertisement during the Super Bowl. And Leo, you showed us that Cloudflare screen that indicated that all was well with the CDN delivering traffic to

the backend hosting server, and that all it said was that the hosting server was not responding. And I think it was Delahanty. You mentioned that...

Leo: It was Patrick Delahanty, yes, yeah.

Steve: Patrick Delahanty, you shared with us that his I guess modest website was, like, being inundated with bot traffic, which was really causing him a problem. Like, you know, he was being DDoSed.

The point I wanted to follow up with, and we've talked about this before, as I said, is that modern websites, both large and small, are no longer almost ever generating their content, well, actually the way most of GRC's is still. I mentioned also last week this all kind of came about because we were talking about how I, you know, hand-author lightweight HTML and CSS. And it's not that there isn't dynamic content at GRC. ShieldsUP!, the DNS Spoofability test, and other things, you know, the CPU is involved in generating those pages. But of course it's all in assembly language. So there's like zero overhead, I mean, associated with even GRC's dynamically generated pages.

But the modern way - modern, you know, like supposed to be better - the modern way to create a website is with a CMS, a content management system, where the web server doesn't actually have static pages. It runs server-side scripting of some sort - PHP, Ruby, JavaScript with Node, maybe Java, C#, .NET, maybe Python. But the point is that one of those content engines is producing the HTML on the fly, which is sent to the browser, backed up by some backend database. And so queries of this database describe the content which is then interpreted by the script and used to generate HTML, but then goes out. And so the point is that while these approaches turn web servers into very flexible application delivery platforms, that power and flexibility to dynamically deliver any page content comes at a steep price in processor load and database load.

So I have no doubt that, you know, as we saw in that chart, Cloudflare was faithfully delivering HTTP queries to whatever backend server infrastructure AI.COM had built out at that point. But whatever it was, it was unable to scale as it needed to to handle the massive demand spike which was created by a Super Bowl ad. I don't think the site went down, technically. It was just probably the per page processing cost was so high that there just wasn't enough processor available to keep up with the demand. So, you know, mostly it was just embarrassing; right? And it was certainly not an auspicious launch for a new venture. And you could argue that they probably lost some people who responded during the Super Bowl commercial and then thought, well, I don't know what's wrong with these people, but they don't seem like they've got their artificial intelligence working very well.

So anyway, the consequences of this high-cost webpage delivery are actually being felt a lot. By pure coincidence, I happened to stumble on last Wednesday's Linux Mint Blog, which, among other things, addressed problems with their forums. The guy wrote: "We'd like to apologize to our forum users for how slow and unreliable the forums were last month. The volume of traffic we receive is extremely high, and it's mostly coming from AIs, bots, scripts, and web crawlers. It got to the point where our server could not cope, and people weren't able to use the forums.

"In addition to the Sucuri Web Application Firewall, it took us a while to come up with an efficient way to filter bad" - what he's calling bad - "traffic." Meaning, you know, non-human users, I guess, is what he's calling bad. "If you're getting 403 errors from the forums right now, please make sure your browser is up to date." I thought that was interesting. He said: "We upgraded the server to give it 10x the CPU capacity and twice the bandwidth."

So I checked them out. Linux Mint's forums use the free phpBB. And I don't know whether they've spent time speeding up their implementation by using - there are many tricks you can use to reduce the overhead of a PHP-based site. There's an in-memory tool called OPcache, which is able to take the burden off the backend PHP interpreter. And also Redis has a key-value store that many forums are able to enlist. I use both because the forums at GRC.com are also - I'm using XenForo. And that's a PHP-based system. And I looked at the user count. They were talking about 6,000 people. But we have a thousand typically roaming around at any given time. And my CPU is off, it's down in the single-digit percentages. So, you know, there are ways to, if you are focused on improving efficiency, to do so. I don't know what is going on with Patrick, Leo. But it was bots that he said were causing trouble for him?

So the takeaway is to remember that connection bandwidth is almost certainly no longer the limiting factor that it once was. And it can be practically impossible to change platforms once one's committed. That is, give some thought to the page delivery overhead and performance of a system that you're considering switching to. I'm sure anybody who's ever switched to a different platform knows what an incredible pain it is. So there's huge anti-switching inertia. But if you have a system which is inherently heavy in overhead, then switching is going to be a problem, and all you can do is scale up in CPU resources. And it can be expensive. If you then need load balancing in front of a bunch of servers, then there's an additional burden from that.

And so anyway, it really pays to keep efficiency in mind, and it's not just bandwidth anymore. With all of our pages, so many of them being delivered dynamically, the overhead of the delivery system really matters.

Leo: Yeah, modern websites are programs, really. They're not static websites.

Steve: Right, right, right. That's absolutely true. And it's good you said "modern," because I'm delivering almost all...

Leo: You're static. You're HTML, I'm sure. Plain old HTML; right?

Steve: Well, but ShieldsUP! is dynamic. The DNS Spoofability. So I've got dynamic pages. But you know me, they're all written in assembly language.

Leo: Right. My blog is also static. It has a program running in the background that generates the HTML and...

Steve: And that's a very good way to do that, yes.

Leo: Yeah. So you still get the benefit, yeah.

Steve: Yes. GRC, I have three freeware pages, like, and you're able to ask how you'd want them sorted, by popularity, by age, and by something else. I don't remember. And every night at midnight I statically regenerate those three pages so that they're delivered, you know, fast from finished HTML. So you're only going through that generation process once a day because it's not the kind of thing that needs to be

changed every second. And it's not, you know, there's no need to generate them per person, as is often the case with a modern website.

Okay. So I want to share an editorial which appeared in the Seriously Risky Business publication which was unfortunately titled "Microsoft Forgoes Its Secure Future." So I'm just going to share what they wrote, and then I'm going to share some observations afterwards. But I thought what they wrote was very insightful. They said: "For a brief time, Microsoft appeared to be making security a priority. As with all good things, though, it appears that period has come to an end with personnel changes at the organization signaling a shift in priorities. We fear Microsoft's goal now is not to make secure products, so much as to sell security products." And of course it's not the first time we've touched on this. But some recent changes, as we'll see.

They wrote: "Last week, CEO Satya Nadella announced that Microsoft's Executive Vice President of Security, Charlie Bell, had been replaced by Hayete Gallot, who was most recently President of Customer Experience at Google Cloud. Charlie Bell is stepping back from leading Microsoft's security organization to become an individual contributing engineer.

"Now that Bell has gone, it appears the guise of 'security first' has been tossed aside, and we fear the company may slip back into being a security disaster. Bell has a great reputation and joined Microsoft to make a positive impact on its security. Despite this, the history of his tenure at Microsoft shows that the company itself only prioritized security when it was forced to by government pressure.

"Bell joined Microsoft from AWS to lead a new security organization in 2021. At the time of his hiring, we wrote that we had consistently, for months on end, shown 'example after example of Microsoft security' - as they put it - "'clangers.' Those rolling security debacles" - and of course we talked about them all here on the podcast - "were a symptom of senior leadership prioritizing profit over security." You know, things like not logging unless you paid extra, that kind of thing. "At the time," they wrote, "we predicted that Bell would struggle to make a difference. We were right. Not even an exceptional manager can change much if the CEO and executive team are not really interested.

"A 2022 profile of Bell in The Information reported that Microsoft's old guard managers 'pushed back on Bell's suggestions for improving their responsiveness to security vulnerabilities, believing he was setting too high a bar for stopping attacks on its products.' The company continued to pay lip service to security, although it did launch a lackluster security uplift program, the Secure Future Initiative, in late 2023.

"Microsoft's devil-may-care approach to security," they wrote, "came back to bite it after separate compromises by Chinese and then Russian state hackers were discovered. The security lapses that lead to these breaches were, frankly, unbelievable. In April 2024, a Cyber Safety Review Board (CSRB) report into the Chinese breach, which had compromised the email accounts of senior U.S. policymakers, found a 'cascade of security failures.'

"It wasn't until this kick in the pants that Microsoft truly embraced security. The following month, CEO Satya Nadella told staff to prioritize security 'above all else' and that 'if you're faced with a tradeoff between security and another priority, your answer is clear: Do security.'

"What followed was a short halcyon period where Bell was able to kick some goals. But the Trump administration has since disbanded the CSRB and signaled that it is not interested in strong regulation. The pressure is off. Microsoft execs can grab a coffee and relax.

"Which brings us back to the recent change in security leadership and, in particular, Nadella's messaging in his public announcement of Gallot's appointment. It sends strong warning bells that security at Microsoft is falling by the wayside. Nadella had an opportunity to highlight Gallot's work experience in security roles. Instead, he focused on her 'critical roles in building two of our biggest franchises' and 'leading our go-to-market efforts.'

"Much of Nadella's announcement was about selling more security products. He said that the company has 'great momentum in security, including strong Purview adoption and continued customer growth.'" Purview is a product of theirs. "Entirely missing was any language about the importance of actual security to the company, or a call for people to get behind the critically important security work that Gallot will lead. If it talks like a sales target and walks like a sales target, it ain't security. It's a recipe for security sales."

Okay. So that's the end of their editorial. I wanted to share this to highlight a lesson we've all learned throughout the past 20-plus years of our observation of real-world security deployment. The lesson I believe we've all learned is not only that security is hard, but that it's always much harder than we expect it to be. If it wasn't so difficult, we'd have much more of it than the sad little bit of security we actually have out in the world. The U.S. wouldn't have Chinese and North Koreans crawling around in our networks, nor telco executives actually saying "We're not sure we can get rid of it all." What? My point here is that since we always need all of the security we can possibly get, any sign of Microsoft slacking off whatsoever on the security front should be taken very seriously. What's worse, a reduction in delivered security is not something that can or will be immediately apparent. Right? It's only the inevitable consequences of a relaxed security posture that will wind up being felt.

As for why Microsoft might make this shift, one of the problems is that since it's not possible to prove a negative, no one really receives any credit for security breaches that don't occur because they were prevented. In the case of Microsoft, the successful influence and efforts of Charlie Bell, their now-previous Executive Vice President of Security, may easily have gone underappreciated. You know, it's "Look at that! I guess security isn't as big a problem as we thought. Those other problems must have just been one-offs." Right. So let's hope for the best.

One quickie, and then we'll take another break, Leo. I suppose I should at least mention - I know that Paul was talking about this last week. I should at least mention that this spring, because listeners have already been asking, Microsoft will be introducing what they are now terming a "scoped" release of Windows 11. Its "scope" is limited to use with the new Qualcomm Snapdragon X2 next-generation ARM system-on-chips where Windows 11 26H1 will come pre-installed on those machines. It only runs on them, and it will not be available in any other form for general use or upgrading.

The latest general Windows 11 release will remain 25H2, and this oddball 26H1 - whose naming appears to have ruffled many feathers. There's lots of dialogue out on the 'Net saying, what? Come on, Microsoft, give this a different name. It's really confusing. Anyway, despite its name, it is not an update for 25H2. So everybody else should just ignore it. We can't have it. We need to wait for 26H2.

Leo: Oh, what a world.

Steve: And Leo, they just - Microsoft just cannot stick with anything. It just, I mean, I guess I understand. You don't know how the world is going to evolve over time. That's the nature of it. But still, you know, it's not like what they do doesn't matter and that a

lot of people aren't paying attention and trying to figure it out. So, I mean, so there's a high price for them changing their mind all the time.

Leo: Sad to say, yeah.

Steve: Okay. Break time. I'm going to rehydrate. And then we're going to look at Chrome 145 and its new support for Device Bound Session Credentials.

Leo: Well, there you go. Stick around. You don't want to miss that excitement.

Steve: No, baby.

Leo: No. Now, back to Steve.

Steve: Okay. So last Tuesday, Google updated the world to Chrome 145. This update repaired, you know, your typical assortment of a few high, mostly medium, and some low severity security issues, and continued to move Chrome's support for the latest HTML, CSS, and JavaScript standards forward. Perusing those, I'm just astonished over the complexity of today's modern web content interpreters. These browsers are so complicated. It just gets more insane every day.

One of the new features that stood out is Chrome 145's support for something known as Device Bound Session Credentials. Think about that phrase for a moment. Device bound, as in binding to a device, session credential. A session credential is just a fancy name for a cookie. And device binding would mean binding a session credential cookie to the device whose web browser first receives that cookie from a remote website. So that means that this innovation arranges to, for the first time ever, prevent anyone who might somehow arrange to obtain a session cookie from being able to use it themselves anywhere else. That's huge, and Chrome 145 now supports it.

Many years ago, before servers were fast enough to glibly encrypt all connections all the time, a user's session cookies would be sent in the clear after they had first successfully logged on. This allowed anyone who was able to eavesdrop on Internet traffic anywhere to capture those logged-on session cookies to impersonate their rightful owner. Looking back on that, it's just, like, hard to imagine we survived that state of affairs. But of course that was yesterday's Internet, too. Less mission-critical than it is today.

So although things are much better today, with all of our connections encrypted all the time, there are still various interception attacks and mechanisms that create vulnerabilities and weaknesses for session credentials. For example, though it's being done only for the best and most justifiable reasons, many enterprises maintain TLS-decrypting middleboxes that decrypt everyone's TLS connections as they cross the enterprise's network edge in order to scan them for malware and other shenanigans and protect the internal network. Everyone's cookies are thus exposed at that point. And if it were possible to briefly impersonate or compromise either end of a connection to observe any browser's reply, the session's logon credential cookies would be exposed.

So, you know, it's all we've been able to do so far, but there are problems. This resolves that. Until this time, the browser cookie has been a, I guess I would say it's an overworked authentication mechanism. It really was just meant in the original creation of it to allow a web server to, like, to create this notion of being locked on, to identify you

when you made successive moves around pages on a website. It would be like, oh, there's that guy again. Okay, fine. And, you know, so you could maintain state. Well, now we use this for banking and international commerce and, you know, super private connections to investment portfolios, everything is being done with this poor overloaded cookie. So it's all we've had. With this innovation of Device Bound Session Credentials, that changes.

Now, you do need some form of secure enclave such as a TPM, or a secure enclave like Apple has on iOS devices on their platforms. You have to have something like a TPM, a Trusted Platform Module, in order to store the secret part of this credential. But as we know, all modern OS platforms require this already for themselves. So that's really not a problem anymore. And in fact it may have been what's delayed the arrival of this. That's all I'm going to say for now. If anyone's curious, we described the operation of this in full detail during Episode 1021 last April 18th on the podcast. So it did take longer than expected to arrive, but we have it now with Chrome 145.

And I said on the podcast then that Firefox, you know, that Mozilla had implemented it in Firefox, and it was also in Safari. I did not follow up to see what the status is today. But it just got released in Chrome 145. So even if it takes a while to actually filter out into the world, it's clearly going to happen. It does require, as I explained last April, significant support effort from the web server. It's not just your old, you know, it's not your grandparents' cookies. It's a whole different technology to pull this off. But it's eventually going to become, clearly become a widespread standard because it significantly increases the integrity of cookie authentication.

Leo: It can't be used to fingerprint you, though; right? Because it is unique to you. They can't request that ID. They probably - it's like a public key thing.

Steve: Oh, yeah, yeah.

Leo: They would just match it up. So they can't - yeah.

Steve: Correct.

Leo: Okay. Well, that's good.

Steve: That's interesting. If this, you know, comes back up in our - as a topic, it'd be nice to make sure that there isn't some. But I can't imagine in this day and age that we would have implemented, I mean, it's industry-wide. It's industry spec.

Leo: Google spec. Oh, okay. Not just Google.

Steve: Yes, it's not just Google. Yeah.

Leo: Okay. Okay.

Steve: They couldn't have gotten away from it. But I agree with you, Google would love to have a [crosstalk]...

Leo: Love that, love that.

Steve: Track this around. Okay. So I noted that the governments, in catching up on the last week, Kazakhstan, Moldova, and Romania are considering adding their names to the growing list of countries that are enacting age-restrictions on the creation of new social media accounts by children. I also saw some commentary somewhere that I appreciated. It noted that the newer legislation was deliberately eliminating, and I really thought this was good, and you may not agree, Leo, but I can understand it from a practicality standpoint, the newer legislation was deliberately eliminating any opportunity for parental override or exception where, for example, a child who was at least 13 but not yet 16 could appeal to their parents to allow them to create an account.

And the person commenting, you know, clearly understood that parents would be hard pressed not to succumb to the argument: "But Mom, Susie's parents let her use Instagram, and she's younger than me."

Leo: Then you shouldn't be a parent. I mean, do we really need the government to...

Steve: Well, Leo, that's another topic entirely. We could do a whole...

Leo: Do we really need the government to enforce this?

Steve: We could do a whole podcast titled "You shouldn't be a parent."

Leo: I know. That would be a good show.

Steve: So anyway, the world does seem to be moving in that direction. And I've got another point about that. Although for some reason I've got a little blurb here, it's quick, about the return of Roskomnadzor. And, you know, what would a Security Now! podcast be without an update on the most recent machinations of Russia's Roskomnadzor Internet watchdog? It turns out that part of the infrastructure that supports Russia's sovereign "Runet" is its own domain name system; right? They've got their own DNS called NSDI. And Roskomnadzor controls what's listed and what's not. Though access to YouTube and WhatsApp has been throttled since last July - remember we talked about that. Remember, like, they only allowed a tiny bit of YouTube data? And it's like, what can you do with that much data? It's like, you can't even get a video off the ground. Anyway...

Leo: You can see all the things you're not able to see, is what you would do.

Steve: Right.

Leo: You can't watch that. Or that.

Steve: Right.

Leo: Or that.

Steve: Right. Anyway, so they've gone beyond throttling now. Now those two domains, YouTube and WhatsApp, along with Facebook and Instagram, have been entirely removed from Russia's DNS following the Russian government designating Meta as an "extremist" organization after it refused to censor content relating to Russia's war with Ukraine. In addition to YouTube and the three Meta properties - Facebook, Instagram, and WhatsApp - Roskomnadzor also blocked access to the Tor Project, Windscribe's VPN, APK Mirror, and the BBC, as well as several other news sites. So, you know, they're tightening their grip on the Internet, and Russian citizens are having to come up with workarounds or just go with what the Russian state tells them is happening in the world.

Leo: So what you're saying is Roskomnadzor's treating every Russian as under 13.

Steve: Yes. Yes. You're not mature enough to understand.

Leo: You are not ready for these things.

Steve: Yes. And I would argue that Russia is probably not the best parent. So back to "Why you shouldn't be a parent" podcast.

Okay. So Discord. Some of our listeners wrote to ask whether I'd seen that Discord, perhaps as part of reprofiling itself in advance of a \$15 billion IPO, would be switching all accounts to "underage by default" unless shown evidence to the contrary. Since that's partially true, I wanted to share the full story. In Discord's own clarification, because of course, you know, this created quite an upheaval, this is what they wrote.

They said: "We've seen some questions about our age assurance update, and we want to share more clarity. We know how important these changes are to our community. Here's what we want you to know: Discord is not requiring everyone to complete a face scan or upload an ID to use Discord. The vast majority of people can continue using Discord exactly as they do today, without ever being asked to confirm their age.

"You need to be an adult to access age-restricted experiences such as age-restricted servers and channels or to modify certain safety settings. For the majority of adult users, we will be able to confirm your age group using information we already have. We use age prediction to determine, with high confidence, when a user is an adult. This allows many adults to access age-appropriate features without completing an explicit age check. When additional confirmation is required, we offer multiple privacy-forward options through trusted partners." Notice that they trust them. Whether we trust them is another thing.

And they enumerated that: "Facial scans never leave your device. Discord and our vendor partners never receive it. IDs are used to get your age only, and then are deleted. And Discord only receives your age. That's it. Your identity is never associated with your account."

Okay. So for the time being, this is probably the best we can hope for. We know that it will eventually be nice to have our devices able to assert an age range on our behalf. But we don't appear to be even close to having any universal solution or even a standard for that yet. You know, the most recent meeting of the World Wide Web Consortium was, well, what do we want to achieve with this? It's like, oh, god, okay, well we're not there yet, obviously. So I'm sure we will in time since it's very clear, I mean, there could not be more pressure on getting this to happen. I know that Stina Ehrensvard is hard at work on this. Her whole focus is addressing this issue. She tends to get results, and is very much an activist and active in all these sorts of things. For me it's just, you know, I can't do committees.

So are privacy purists losing some of their precious, if entirely illusory and fictitious, privacy? Yeah, you know, that's going to happen. But even that will be better in the future once stronger privacy-protecting standards are in place. And Leo, I know you were talking about Discord and this recently.

Leo: Yeah, because we use Discord.

Steve: Right. But are you flagged as an adult content server?

Leo: No, no.

Steve: No.

Leo: So we probably wouldn't have to worry.

Steve: Right. And that's my point is that it's probably only explicit servers that are offering explicit content. And when Discord doesn't have, has not been able to obtain high confidence that a user is already an adult, that then they would say, okay, you know, sorry, the people you're talking to haven't convinced us. And your language or your grammar or whatever they're using as signals, you need to prove that you're an adult to us. And I should mention...

Leo: We'd lose members if that started happening. I mean, it's a real problem, yeah.

Steve: Yeah. And it wouldn't happen because you're not flagging your server as adult. So I think, you know, there isn't a problem in the best case. Now, it is, however, the case that we have seen evidence of IDs not being deleted when they should have been. And so, you know, that's a concern; right? This k-ID, k-ID, is a third-party service that some providers are using in order to obtain this age verification.

Leo: Right.

Steve: And I don't want to accuse them wrongly. But somebody had a breach, and we talked about it on the podcast about six months back, where a ton of identification, personally identifiable information, like the works, was obtained in a breach by one of these third-party ID, you know, age verification services that for no reason anyone had

or could explain had not deleted this information from their servers. It's like, guys, you know, we're giving this to you on the condition that you delete it. You have to. On the other hand, we all just heard about what happened with the Ring doorbell, you know, video being deleted and then somehow coming back. So, you know, what does "deleted" mean?

Leo: Yeah, that was the Google Nest doorbell, yeah.

Steve: Yeah.

Leo: In the Nancy Guthrie case, yeah.

Steve: Okay. So when I saw that GTIG, Google's TIG, Threat Intelligence Group, had identified a widespread active exploitation of the critical vulnerability in WinRAR which we talked about last summer, although I was certain I had updated my copy then, I double checked. And, yikes, I was still using v7.12, which contained the vulnerability. It was the last version that did. 7.13 fixed it. I'm now using v7.20, but I decided that given that the threat has moved as it was from theoretical to now real and live, I ought to remind all WinRAR users to be certain they have updated. Here's what Google's Threat Intelligence Group just posted.

They said: "The Google Threat Intelligence Group (GTIG) has identified widespread, active exploitation of the critical vulnerability CVE-2025-8088 in WinRAR, a popular file archiver tool for Windows, to establish" - and it's being abused, too - "establish initial access and deliver diverse payloads. Discovered and patched in July" - last summer - "2025, government-backed threat actors linked to Russia and China as well as financially motivated threat actors continue to exploit this n-day across disparate operations." Meaning lots of people are in the act. "The consistent exploitation method, a path traversal flaw allowing files to be dropped into the Windows Startup folder for persistence, underscores a defensive gap in fundamental application security and user awareness.

"In this blog post, we provide details on CVE-2025-8088 and the typical exploit chain, highlight exploitation by financially motivated and state-sponsored espionage actors, and provide indications of compromise to help defenders detect and hunt for the activity described in this post. To protect against this threat, we urge organizations and users to keep software fully up-to-date," blah blah blah. Okay. So anyway, 8088 is a high-severity path traversal vulnerability in WinRAR that attackers exploit by leveraging Alternate Data Streams (ADS). They're able to craft malicious RAR archives which, when opened by a vulnerable version of WinRAR, can write files to arbitrary locations on the system. Exploitation of this vulnerability in the wild began as early as July 18, 2025, and the vulnerability was addressed by RARLAB with the release of WinRAR version 7.13 shortly after, on July 30th.

So that's enough said. And actually, now that I'm reading this, I'm wanting to check the version I have on this machine because when I put this together on the weekend, I checked my other machine, a Win10 machine, and that's where I found 7.12. This is probably the one that I updated immediately, and I forgot to do it to the other one. So, and then I thought back, okay, have I downloaded any WinRAR in the last six months, like from anywhere? I don't think so. That's not my normal mode of getting things. But again, anyone wanting more information and details, I've included the link to Google's coverage in full in the show notes, and you can just go to win-rar.com/download.html. That will get you 7.13 for whatever platform you're using. Make sure you're using that.

If you DO discover a version of WinRAR before 7.13, as I did, you can know at least, for what it's worth, it's not much good, but we're in good company. Stairwell Security just wrote: "Stairwell recently identified a significant and concerning trend across our customer base." Get this. "Over 80% of monitored environments contain vulnerable versions of WinRAR affected by CVE-2025-8088. This finding underscores a persistent challenge in enterprise security when widely deployed, trusted software quietly falls out of date and becomes a high-value target for attackers."

And then they talk about how Google identified the exploitation, blah blah blah. In 80% of the environment that they monitor they discovered versions of WinRAR earlier than 7.13. Everything previous to that has the vulnerability. So, yikes.

And Leo, speaking of yikes, we're at an hour in.

Leo: Yikes.

Steve: I think it's time for me to have a little caffeination.

Leo: Now, back to Steverino.

Steve: Okay. We've talked about the Graphite spyware before. You know, it's one of the Israeli companies, in this case Paragon Solutions. And it's one of the more capable systems. But it's one thing to hear about it, and another thing to see it. They made the mistake of exposing details of their Graphite spyware control panel. The panel was exposed in photos from a demo day recently in the Czech Republic. The photos, which were immediately taken down - it's like, whoopsie, didn't mean to show those - revealed Graphite's ability to extract messages from instant messaging clients including WhatsApp, Signal, Telegram, Line, Snapchat, TikTok, and more.

We already know what - well, we already know that WhatsApp and Signal are truly secure, and that Telegram, well, it probably is, mostly because its encryption is so random and scrambled that no one has yet, as far as we know, been able to make heads or tails of it, even though, and we talked about this about a year ago, some researchers really tried. They're like, what? We're not sure what this is doing. Anyway, the point is, as we've always observed, there is no threat from anyone monitoring their users' communications on the outside. The threat is that, once spyware arranges to gain a foothold inside a smartphone, it doesn't need to untangle Telegram's mess of crypto or fight with Moxie's triple ratchet in Signal. All it needs to do is pretend to be the device's user, examine the decrypted state, you know, the decrypted data that's presented on the device's screen, and send that back to its central headquarters.

So those leaked photos conclusively demonstrated that, once a smartphone has been lubed up with Paragon's Graphite, none of its secrets will be safe from spying eyes. And as we know, this is the battle that Apple is in. I mean, they really take this seriously. They've gone to, you know, every extreme imaginable just to keep this cat-and-mouse battle going on, trying to harden and then reharden and overharden and superharden their hardware platforms to keep the bad guys from getting into their devices. It's just amazing how this battle has continued.

If it weren't so difficult to apply, a useful security caution might be "Beware anything that's too popular." We often see that bad guys are very quick and unfortunately clever about jumping onto anything for which there's a large demand. For example, fake charitable contribution sites invariably pop up following any natural disaster in the hope

of cashing in on people's compassion for the plights of others. So I suppose we shouldn't be surprised to learn that some cretin has created a family of 30 malicious "AI Assistant" browser extensions for Chrome. Of course. Why wouldn't someone do that? AI is all the rage at the moment. And people are going to be looking around for AI this or that.

So last Thursday, LayerX reported on their discovery which they've named "AiFrame" with the headline "Fake AI Assistant Extensions Targeting 260,000 Chrome Users via injected iframes." They wrote: "As generative AI tools like ChatGPT, Claude, Gemini, and Grok become part of everyday workflows, attackers are increasingly exploiting their popularity to distribute malicious browser extensions. In this research, we uncovered a coordinated campaign of Chrome extensions posing as AI assistants for summarization, chat, writing, and Gmail assistance. While these tools appear legitimate on the surface, they hide a dangerous architecture. Instead of implementing core functionality locally, they embed remote, server-controlled interfaces inside extension-controlled surfaces and act as privileged proxies, granting remote infrastructure access to sensitive browser capabilities."

So basically you install this, and then you've created a tunnel from the bad guys' backend server infrastructure into your browser. Not what anybody wants. They said: "Across 30 different Chrome extensions, published under different names and extension IDs and affecting over 260,000 users, we observed the same underlying codebase, permissions, and backend infrastructure." Meaning they're all from the same guy, group, whatever. "Critically, because a significant portion of each extension's functionality is delivered through remotely hosted components, their runtime behavior is determined by external server-side changes, rather than by code reviewed at install time in the Chrome Web Store."

And we should just pause to say there is something so wrong with the fact that this is even possible, the fact that the Chrome Web Store could be allowing extensions to then later change their own behavior by changing what's happening on the server side. So this entire - the security of this whole aspect of the ecosystem is badly broken.

They said: "The campaign consists of multiple Chrome extensions that appear independent, each with different names, branding, and extension IDs. In reality, all identified extensions share the same internal structure, the same JavaScript logic, the same permissions, and the same backend infrastructure. Across 30 extensions impacting more than 260,000 users, the activity represents a single coordinated operation rather than separate tools. Notably, several of the extensions in this campaign were featured by the Chrome Web Store" - it's a featured extension by the Chrome Web Store - "increasing their perceived legitimacy and exposure.

"The technique, commonly known as extension spraying, is used to evade takedowns and reputation-based defenses. When one extension is removed, others remain available or are quickly re-published under new identities. Although the extensions impersonate different AI assistants (Claude, ChatGPT, Gemini, Grok, and generic 'AI Gmail' tools), they all serve as entry points into the same backend-controlled system.

"By leveraging the trust users place in well-known AI names" - you know, brand names - "such as Claude, ChatGPT, Gemini, and Grok, attackers are able to distribute extensions that fundamentally break the browser security model. The use of full-screen remote iframes combined with privileged API bridges transforms these extensions into general-purpose access brokers capable of harvesting data, monitoring user behavior, and evolving silently over time. While framed as productivity tools, their architecture is incompatible with reasonable expectations of privacy and transparency." Which I would say is putting it mildly.

"As generative AI continues to gain popularity, defenders should expect similar campaigns to proliferate. Extensions that delegate core functionality to remote, mutable infrastructure should be treated, not as convenience tools, but as potential surveillance platforms." Amen.

So, yeah. More than a quarter million instances of browser extension downloads and installations which front for this single malicious campaign. We know that web browser extensions are super popular and arguably necessary. After all, we couldn't be using the password manager of our choice today without them. But their diversity and popularity has overwhelmed Google's ability to examine and manage them, such that today's web browser ecosystem creates serious vulnerabilities. And there's really no solution today except to just say be prudent. Only install from, like, really well-known brands that have been around a long time.

And next, that's not even the worst. Would you believe - that was 30 extensions. Now we have 287 Chrome extensions found to be spying on 37.4 million users. Chrome browser extensions. The researcher in this case is - actually they posted on Substack great research, despite the fact that they chose as their handle the "QContinuum." Okay. They wrote - although their research is great. They wrote: "We built an automated scanning pipeline that runs Chrome inside a Docker container" - this is great research - "routes all traffic through a man-in-the-middle proxy, and watches for outbound requests that correlate with the length of the URLs we feed it." That's very clever.

So they feed the browser URLs of different lengths. And then, although they're unable to see the detail, they look at the length of the traffic which is passing to a remote server and see that if it's correlating with the length of the URL, then it is almost certainly that URL encrypted.

So they say: "Using a leakage metric, we flagged 287 Chrome extensions that exfiltrate browsing history." Meaning you install this extension, every single URL you visit in Chrome, even though just because the extension is sitting there in your pile of extensions, it is sending them all back to the extensions publisher. Complete breach of your privacy. They said: "Those extensions collectively have 37.4 million installations, roughly 1% of the global Chrome user base." Just this group. 1%. "The actors behind the leaks span the spectrum: Similarweb, Curly Doggo, OffiDocs, Chinese actors, many smaller obscure data-brokers, and a mysterious 'Big Star Labs' that appears to be an extended arm of Similarweb."

They said: "The problem isn't new. In 2017, Weissbacher et al., their research on malicious browser extensions demonstrated this. In 2018, Heaton showed that the popular 'Stylish' theme manager was silently sending browser URLs to a remote server. These past reports caught our eye and motivated us to dig into this issue today.

Fast forward to 2025: "Chrome Store now hosts roughly 240,000 extensions" - right, so just shy of a quarter million browser extensions. How can they possibly know what they're all doing? "Many of them," they wrote, "with hundreds of thousands of users. We knew that we needed a scalable, repeatable method to measure whether an extension was actually leaking data in the wild. It was shown in the past that Chrome extensions are used to exfiltrate user browser history that is then collected by data brokers such as Similarweb and Alexa. We try to prove in this report that Similarweb is very much still active and collecting data.

Why does it matter? They write: "There's a moral aspect to the whole issue. Imagine that you build your business model on data exfiltration via innocent-looking extensions and using that data to sell them to big corporates. Well, that's how Similarweb is getting part of the data. That should remind us that whatever software you're using for free, and it's not open sourced, you should assume you are the product. The second aspect is that it

puts the users into danger, and potentially this could be used for corporate exfiltration. Even if only browsed URLs are exfiltrated, they typically contain personal identifications. That way bad actors that would pay for the raw traffic collected can try to target individuals."

So anyway, they go on at length. I just wanted to put this, again, on everyone's map. Again, I don't know how to solve the problem. We want extensions that are powerful. Our extensions need to be powerful to be, for example, a password manager. You know, I fill out a form, and Bitwarden sees the contents that I put in the form and says, oh, I checked your domain. I don't have this in my library for you. Would you like me to add this to your password manager collection? And you just say, yeah, I do, I want that. And that's done. So super convenient, but consider what that means this extension can do. It sees you entering the plaintext password and your username, and it knows where you are, the whole URL. That's what these extensions have access to.

And now we have an ecosystem in the Chrome Web Store of 240,000 of these extensions. Obviously, many of them are spying on their users. In this case, these guys found 287 representing - that have been downloaded by 37.4 million users representing around 1% of the Chrome user base, sending everywhere they go home. Yikes.

The folks at Koi Security titled their write-up of a new attack: "AgreeToSteal: The First Malicious Outlook Add-In Leads to 4,000 Stolen Credentials." And here's another fundamental problem that we have in the industry. I had this on my radar for a while, and then another instance of this came up. Generically these are known as "Domain Recovery Attacks." They can be quite serious, and they reveal an aspect of Internet security that is important and has largely been overlooked. So I'll first share the beginning of what Koi wrote.

Last Wednesday they posted: "This is the first known malicious Microsoft Outlook add-in detected in the wild. But the developer who built the add-in is not the attacker. In 2022" - so four years ago - "a developer built a meeting scheduling tool called AgreeTo and published it to the Microsoft Office Add-in Store. It worked. People liked it. Then the developer moved on, and the project died.

"However, the add-in stayed listed in Microsoft's store. The URL it pointed to, hosted on the Vercel.app domain, became claimable, and an attacker claimed it. After making it theirs, they deployed a phishing kit, and Microsoft's own infrastructure started serving it inside Outlook's sidebar. By gaining access to the attacker's exfiltration channel, we [Koi Security] were able to recover the full scope of the operation: over 4,000 stolen Microsoft account credentials, credit card numbers, banking security answers. The attacker was actively testing stolen credentials yesterday." They posted this, what, on Thursday? So last Wednesday. Oh, no, they posted it on last Wednesday, so last Tuesday they saw this happening. They said: "The infrastructure is live as you read this. This is the story of how a dead side project became a phishing weapon."

They said: "First off, Office add-ins are not installed code. They're URLs. A developer submits a manifest to Microsoft, an XML file that says 'Load this URL into an iframe inside Outlook.'" Whereupon, of course, we say, "What could possibly go wrong?" They said: "Microsoft reviews the manifest, signs it, and lists the add-in in their store. But the actual content, you know, the UI, the logic, everything the user interacts with, is fetched live from the developer's server every time the add-in opens." Okay. So just to pause here, that really sounds like an architecture that is asking for trouble. What could Microsoft possibly have been thinking to implement Office add-ins like this? And it appears that trouble is what they got.

Koi continues, saying: "Note the ReadWriteItem permission in the Manifest. That grants the add-in the ability to read and modify the user's emails. It was appropriate for a

meeting scheduler. It's less appropriate for whoever controls that URL today. There's no static bundle to audit. No hash to verify. Whatever the domain outlook-one.vercel.app serves right now is what runs inside Outlook. If the developer pushes a bad update, it's immediately live. If someone else takes control of that URL, they control what every user of that add-in sees, inside Outlook's trusted sidebar, with full read and write access to their email. Microsoft blessed this manifest once, in December of 2022. They never check what the URL serves again.

"AgreeTo was a real product. An open-source meeting scheduling tool with a Chrome extension (1,000 users, 4.71-star rating, 21 positive reviews), and an Outlook add-in published to Microsoft's store in December of 2022. The developer maintained an active GitHub repo - a full TypeScript monorepo with Microsoft Graph API integration, Google Calendar support, and Stripe billing.

"This was somebody building a business. Then it stopped. Development stopped. The last Chrome extension update shipped in May of 2023. The developer's domain, AgreeTo.app, expired. Google eventually removed the dead Chrome extension in February 2025. But the Outlook add-in stayed listed in Microsoft's Office Store, still pointing to a Vercel URL that no longer belonged to anyone. At some point after the developer abandoned the project, their Vercel deployment was deleted. The subdomain outlook-one.vercel.app became claimable, and the attacker grabbed it. They deployed a four-page phishing kit: a fake Microsoft sign-in page, a password collection page, an exfiltration script, and a redirect.

"That's all it took. They didn't submit anything to Microsoft. They weren't required to pass any review. They didn't create a store listing. The listing already existed - Microsoft-reviewed, Microsoft-signed, Microsoft-distributed. The attacker just claimed an orphaned URL domain, and Microsoft's infrastructure did the rest."

So their description continues with all the details, but everyone gets the idea. VERY poor design on Microsoft's part. I can understand Microsoft not wishing to re-vet and re-verify any change that an add-in developer might make. But they should have some mechanism for preventing abandoned and dangling URL domains from being taken over and repurposed. That's just dumb.

In general, the design of the Internet creates this problem; right? We've all encountered abandoned domains that have been acquired typically by low-end advertisers who snap up web domains that have expired, and then they host their own content that nobody one wants in the hope of generating revenue from advertisers who will pay for any traffic from anyone, and they're not discriminating. But when domains that are used to host important content are abandoned, things can quickly take a turn for the worse. Years ago, we examined an instance where the domain of an important and super-popular web browser JavaScript library had changed hands. Suddenly, an incredible number of web browsers were pulling a critical library from someone else. It should be enough to keep one up at night.

And Leo, we're at an hour and a half. We've got some Listener Feedback. Let's take a break and then we will plow into some feedback.

Leo: All right. Back to Steve.

Steve: Okay. So I got an email from Walt Stoneburner, who said: "Steve, thank you for pointing out that quality tested code that adheres to functional specs is important for production level code. There's a big difference between throwing something together that seems to work, but that you don't understand, and experienced craftsmanship.

"It's not that we don't love coding, it's just a pleasant benefit. It's that we are aiming for correctness, speed, size, cost, maintainability, clarity, extensibility, expressiveness, modularity, portability, and a host of other factors that vibe coding does not do. Walt in Ashburn."

Okay. So lots of our listeners are writing in, saying "Steve," you know, "what do you think about all this code generation by AI?" And I'm continuing to think about it. So one thing I want to say is I think Leo and I were - well, I know that Leo and I were talking about this, I think it was before we began recording today, that I want to always acknowledge that wherever we are today is not where we're going to be tomorrow. It's not where we were yesterday. And I don't see any sign of this slowing down. I'm happy that so much resource, I mean, I'm happy for the hype because the hype means that a ton of resources are being put into something which I think has great potential.

Okay. That said, where we are at the moment. I've been thinking about vibe coding, and I think that the most unnerving aspect of vibe coding for me, a lifelong coder, is the idea that a bunch of code has been cast which may do what I want and expect, but it also may not. There's every chance that in some subtle way it might misbehave. In some of the feedback I've received and shared in recent weeks, the tasks were relatively straightforward. So, you know, the various strange errors Claude Code made were obvious to its user, like that book author's name appearing twice in its field. He didn't know why, but he pointed it out to Claude, and it says "Oh, yeah, sure enough," and then it fixes it.

But this should give any true coder some pause to wonder what other far more subtle errors might be lurking in there that haven't been seen and pointed out to the code bot. And we would expect that there would be an exponentiating effect in errors as projects grew in size to create many more possible interactions and places where subtle errors might hide. And this is nothing against AI. We've talked about this with actual, you know, any project size, any time complexity is getting larger, you know, there's far more opportunity for mistakes.

Okay. But having said that, then I challenge myself and say: "Okay, hold on there a second, Gibson. When you use a library authored by some third-party, you didn't write that library. You don't know everything about its innards. You're taking on faith the fact that it operates correctly." And that's true. But the difference is that I'm able to assume when I use a third party, you know, code from a third party, that its non-AI author took pride in creating, you know, deliberately writing code and knowing what it did and testing the functions of each and every one of its whatever it does, I'm able to assume the library's correctness.

This suggests that a unit testing approach to professional AI code generation might be the solution. Break the large project down into small pieces, then design and apply unit tests to verify the correct operation of each piece under every edge case and possible condition. This echoes some of the early formal code-correctness verifications that programmers have been applying by hand for years. It's considered the only way to know for sure, from a testing standpoint. So perhaps AI can similarly be asked to build large projects from smaller, carefully tested pieces.

There's one thing that worries me. When some aspect of my code is not doing what I expect, I'm able to quickly and easily zero-in on the trouble and fix it because I wrote it in the first place. You know, it's my code. So I understand how it works and what it's supposed to be doing. But what happens when a non-coder detects that something is not working? Last year when we began looking into Microsoft's early use of Copilot for fixing bugs, remember that instance where Copilot was shown a bug in some code where a parser was running off the end of the stack that it was parsing.

Rather than fixing the underlying error - because a stack underflow should not have been possible at all - Copilot added some glue, an explicit test to prevent the pointer underflow. Okay, technically this repaired the problem that occurred by explicitly preventing the condition that revealed the bug. This is reminiscent of the old joke about the guy who goes to the doctor with a complaint. He explains to his doctor that his left shoulder hurts whenever he raises his arm in a certain way. His doctor says, "No problem, just don't raise your arm like that." Of course the joke is that the symptom was suppressed, but the underlying problem was not addressed.

In the case of the early Copilot experiment, an experienced Microsoft coder was overseeing the Copilot testing and questioned whether Copilot's "fix" might not be masking a subtler underlying problem. So I'll suggest that it's going to be very interesting to watch this whole vibe coding era play out. And I also think that we're at 1%, if that, of where we're going to be. I mean, I was among the first people to say, very early on, that code should be something that AI could master. And, you know, we're seeing very, very encouraging early results. But again, I said it last week, I author the code that I produce. There's no way I'm going to be selling code under my name that an AI produced. That's just - that isn't for me.

Denny VandeMaele said: "Hello, Steve. Long-time listener of Security Now! and user of your web products and software. For many years I've held the position that free VPN services are scary in general. Then I stumbled across Cloudflare's free tier of their WARP VPN for most devices. As you know, Cloudflare's IP address and DNS is 1.1.1.1. Cleverly, they bought the TLD 'one,' and their free tier VPN is located at one.one.one.one." He said: "It works well and can be installed on Apple macOS, iOS, Android, Windows, and Linux. Signed, Denny." And so I just wanted to thank Denny. I'd forgotten about Cloudflare's free WARP VPN offering. So I am glad for the reminder.

Okay. So that's our feedback from our listeners. I want to talk about Attestation and what it's about and the surprising and unplanned adventure that I had last week. Why don't we just do our last advertiser break, and then I won't break in the middle of this. That's be great.

Leo: Okay. Unplanned adventures.

Steve: Lets me clear my throat, too. I've got something...

Leo: Yeah. Unplanned adventures are never good, I think. You always want to plan them ahead of time. Now, let's talk about Attestation.

Steve: Okay. As I've noted and warned, the month of March 2026, which is now a mere two weeks away, will see major changes in the identity certificate issuing industry. A few weeks ago, near the end of January, actually it was Monday, January 26th, being a customer of DigiCert, as I have been for a long time, I received a courtesy piece of email with the subject, "Important Reminder: TLS/SSL certificate lifetimes changing February 24th, 2026."

They said: "Hello. We're writing to remind you that starting February 24th, 2026, TLS/SSL certificates issued through DigiCert CertCentral will have a maximum validity of 199 days (down from 397)." Okay, so close to 200 versus close to 400. They are basically cutting certificate life in half.

"This change to shorter certificate lifetimes is an industry-wide requirement mandated by new CA/Browser Forum baseline requirements. While shorter lifetimes may require adjustments [uh-huh], they also reduce risk," blah blah blah blah, right, justifying all of this. So basically they explain that they're cutting the lifetime of their certs in half and how this affects their customers. Everyone who's been following this podcast knows only too well the reasoning behind my feelings about this ridiculous and extremely inconvenient shortening of certificate lifetimes.

And that's doubly so for code signing certificates which, unlike web server certificates, can only be stored in HSM hardware, making them completely impervious to remote theft. In this case, DigiCert is alerting their customers and giving us a one-month reminder of the upcoming reduction in web server authenticating TLS certificates. Maximum certificate lifetime will be dropping from one year plus some margin down to just six months plus some margin.

One of the consequences of the industry's shortening certificate lifetime is the need to decouple certificate issuance from certificate qualification. In the bygone days, when certificates lasted for five or 10 years, as they once did, the act of proving you were who you claimed to be would be part of the certificate renewal process. In applying for or renewing a certificate you would need to do whatever the CA asked you to do to prove that you were you. But now that process has also been significantly fouled up because, right, you don't want to have to do that every time you need to renew a certificate. DigiCert's email says, for example, on February 24th - meaning a week, a little more than a week from now - OV organization validation reuse periods will be shortened from 825 days to 397 days. On the same date, on February 24th, domain validation reuse periods will be shortened from 397 to 199.

In other words, it will now be necessary to re-validate one's organization annually, rather than only every two and a quarter years. It used to be 825 days, every two and a quarter years. Now you've got to do it every year. Like reprove who you are, who your organization is. Now, given that Let's Encrypt only offers domain validation certificates, not organization validation, all they're saying is what you need to connect to a server reliably, which I think makes total sense. You know, and thus it doesn't incur any of this nonsense. I have a difficult time understanding how the CAs are not putting themselves out of business with these kinds of practices.

I suppose, you know, they plan to survive on all of the other various types of certificates which they issue and manage, such as for signing documents and such. And they'll continue to offer TLS web certificates sort of as a loss leader. You know, it's like, well, they just want to offer a full suite of products so they will continue to offer certificates because they already do.

In order to obtain the best price possible, I previously purchased TLS certification from DigiCert into 2028. In preparation for this March, GRC recently jumped through the various organization validation hoops; and, at the start of last week, I reissued GRC's TLS, you know, GRC.com, our TLS domain certificates, well in advance of DigiCert's February 24th deadline for a full year, you know, a year plus, 397 days, because I didn't want anything to happen that might get in the way of that. You know, because with certification now having become so involved, you know, you've got to be standing by the phone when it rings and jump through all kinds of hoops. There's no telling when or why that process might fail or stall.

I've been surprised in the past, so I wanted to give myself time to fix anything that might fail before that deadline. Now, the process, as it turns out, this time proceeded without a hitch. So just because I can, and because DigiCert has no problem with reissuing certificates, next Monday morning, the 23rd, the day before the drop-dead date, just

because I can, on the last possible day to obtain a 397-day certificate, I'm going to do that.

So first part of this is, this should serve as a heads-up reminder to anyone who might similarly have better things to do right at this moment than figure out how to switch their certificates over to Let's Encrypt, that whoever their CA is, there will be an end-of-the-month having of standard TLS certificate lifetime from more than a year to just over more than half a year. And of course I will be moving over to Let's Encrypt and switching to domain validation instead of organization validation certs as soon as I can, as soon as it makes sense. So I imagine that once my pre-purchase at DigiCert, I've already bought certificates through 2028, I might as well have them, and then I'll switch to Let's Encrypt.

Okay. So that's the current status of the TLS web server certificate side. But my primary focus today is on another class of certificates I've recently discussed, specifically Code Signing. As I noted recently, the maximum lifetime of code signing certificates is also being cut, in this case to a third, from a convenient three years down to one-year minimum.

Anyone who examines any of the software that's available from GRC will find that it's all signed with a DigiCert certificate. Sadly, that will no longer be true after this August when my current code signing certificate reaches the end of its three-year life. That's that EV certificate that I've got currently being signed. I would prefer to remain with DigiCert. Why not? They've been good to me. But the recent changes at DigiCert have overcome my own "change inertia," which is big, for a code signing certificate authority. So long as there's any practical alternative, I will not countenance "renting" the privilege of signing my own code. I can't imagine using a cloud-based provider who places a limit on the number of signatures I'm able to make and charges per signature for any overage.

And even when signing my code with my own customer-provided HSM, which is what I've been doing for the past three years, with DigiCert, the least expensive code signing plan, where the user provides their own hardware, is advertised as \$50 per month. But that's disingenuous as hell since it's not possible to purchase it in monthly increments. It's only available with an auto-renewing annual commitment paid in advance. So that's \$600 a year. And even that \$600 per year presumably is subject to change at the next annual billing cycle since there's no longer any way to pre-purchase future years to get a price commitment.

So while I'm bitterly disappointed in DigiCert, to whom I've felt a well-deserved loyalty for many years, I don't mean to single them out. The entire code signing certificate industry appears to be headed in the same direction. And it's not pretty. As I was looking around, I discovered that a number of other CAs are now reselling DigiCert for exactly the same pricing structure. I mean, it is DigiCert for all intents and purposes. Just you go to them with a different domain and website. Scouting around, I found that IdenTrust will still "sell" a no-strings-attached three-year code signing certificate for \$538. When placed into my own HSM, I'm able to use it. So that's \$179/year, basically 30% the cost of remaining with DigiCert, and that's assuming that DigiCert doesn't choose to further raise their prices before the next three years have passed. IdenTrust is well known, so IdenTrust it was for me.

And thus began the new adventure of obtaining a code signing certificate in 2026. Our illustrious CA/Browser Forum has added a surprising hoop through which anyone wishing to obtain a code signing certificate must now jump: The CA/Browser Forum requires the issuing Certificate Authority to obtain an "attestation letter" from an independent legally licensed attorney or CPA, you know, a Certified Public Accountant. This third-party

individual must attest to having firsthand knowledge of the legitimacy of the corporation and its officers.

Okay. Now, since Gibson Research Corporation has been a tax-paying California Corporation in good standing for 37 years with a stable business location, a DNS domain name, and a well-known presence, I initially doubted the need for this attestation letter, which I've never needed before, not been asked to provide. And IdenTrust's documentation was unclear about it.

So one week ago today, last Tuesday, I created an account with IdenTrust and received a link to download a PDF packet of documents. I filled them out, omitting the clearly separate final three pages that contained the attestation letter details. I sent this off to IdenTrust in Utah via Federal Express overnight.

Last Wednesday, the next morning, 11:32 a.m., I received email confirmation of the forms having been received. And 35 minutes later, at 12:07 on Wednesday, I received notice with the Subject: "ACTION REQUIRED: Code Signing Application - Attestation Letter Required." Oh, great. So the famous Merriam-Webster dictionary defines attestation as: An act or instance of attesting something, such as a proving of the existence of something through evidence, or an official verification of something as true or authentic.

So apparently I needed to provide IdenTrust with an Attestation Letter. My lifelong personal and corporate attorney retired from practice a few years ago, and I'm sure that he was, you know, he allowed his license to lapse. I've been using the same CPA tax accountant firm for the past 40 years, since 1984. So I asked my California licensed CPA if I could trouble him to use his license - because he's got to fill out this form which, you know, basically he's having to justify his own existence as a California licensed CPA to IdenTrust, I asked him if he would attest to Gibson Research Corporation's identity. He didn't hesitate to say "yes."

So Wednesday afternoon I emailed IdenTrust's three-page Attestation Letter document to him. The CA/Browser Forum requires either a digital signature using the attesting individual's personal certificate, which is not something that my CPA had, or what they termed a "wet signed" original. My CPA signed and filled out the PDF, signing it in nice blue ink. So it was very clearly not from some printer. Thursday morning I dropped by his office, picked it up, then swung by FedEx to send that originally signed attestation letter to IdenTrust.

Late the next morning, last Friday, I received notice that my identity had been established, and a few hours later a code signing certificate was issued. So, success. My reason for my sharing all this is to establish the proper and full context for understanding what has happened to us, to the entire PC industry, in response to the threat of malware. This is the nature of the cost and the burden that malware has inflicted upon the world. I dislike what I've just had to go through to obtain the privilege of adding a cryptographic signature to my code as the only available means of proving my identity as my code's signer. But as long as our systems are subject to malicious abuse from malicious software, I understand the need to have some unspoofable means of determining the source of any software we allow to run on our computers.

As we've seen, all of the PC desktop and mobile platforms that are able to run third-party applications, with the notable exception of Linux, check and verify the cryptographic signature of any code they're being asked to run before they let their processors near it. So I understand the need for this, and I have no better idea. But what really rubs me the wrong way is the apparent profiteering by the industry's certificate authorities. I get it that the CA/Browser Forum's increasingly stringent policies have increased the

verification burden upon those CAs, and thus the cost of offering this service. But even that is one time and non-recurring.

Once any new CA has figured out who I and Gibson Research Corporation are, that's not going to ever change - just as it never did for DigiCert. I must have been grandfathered in because I was never asked to do all this from DigiCert. These requirements were already in place when I obtained my most recent EV code signing certificate. And as I said, I never needed to go through any of this, presumably because I had already established a long multi-year relationship with them, and I was grandfathered in.

Looking over the current baseline requirements - that's what they're called in this document - which dictate the behavior of all Certificate Authorities that issue code signing certificates, it became clear that the standing and authenticity of my own CPA had also just been thoroughly researched. Today, you know, I'm calling this podcast "Attestation" because I want to share what I just learned about the extent of what this "attestation" means. It's been eye-opening.

The document which governs the conduct of the world's Certificate Authorities is titled "Baseline Requirements for the Issuance and Management of Publicly-Trusted Code Signing Certificates / Version 3.8.0." Now, everyone should keep in mind that these requirements are applicable to anyone and everyone who wishes to create code that will be signed and widely trusted by any platform. Trusted code requires that it be signed and timestamped by an unexpired code signing certificate. As we know, unexpired at the time of the signing.

Near the top of the Baseline Requirements is a section of definitions. Under "Attestation Letter," the document says: "A letter attesting that Subject Information is correct, written by an accountant, lawyer, government official, or other reliable third party customarily relied upon for such information. Section 3.2.2.1: Authentication of organization identity for Non-EV" - remember, I'm not going for EV again because that's just throwing money away at this point.

I'm seeing some language on the Internet that says that Windows Smart Screen filter gives you immediate benefit if you're using an EV cert. I think that may just be inertia from years past because Microsoft is reportedly, and we've talked about this, no longer giving EV any extra validation whatsoever. So all of this is the minimal verification and certification for a code signing certificate. I don't even want to think about what would be required to establish extended validation with a new certificate authority.

So that section 3.2.2.1 says: "Prior to issuing a Code Signing Certificate to an Organizational Applicant, the CA MUST: Verify the Subject's legal identity, including any DBA [Doing Business As] proposed for inclusion in a Certificate, in accordance with section 3.2.2.1.1 under 'Identity,' and 3.2.2.1.2 under 'dbatradename.' The CA MUST also obtain, whenever available, a specific Registration Identifier assigned to the Applicant by a government agency in the jurisdiction of the Applicant's legal creation, existence, or recognition." That was point one. Point two: "Verify the Subject's address in accordance with section 3.2.2.1.1 under 'Identity.'" Third: "Verify the Certificate Requester's authority to request a Code Signing Certificate and the authenticity of the Certificate Request using a Reliable Method of Communication" - that's in all caps, so that's an official term - "Reliable Method of Communication in accordance with section 3.2.5, 'validation-of-authority.'"

And, finally, point four: "If the Subject's or Subject's Affiliate's, Parent Company's, or Subsidiary Company's date of formation is less than three years prior to the date of the Certificate Request" - thank goodness mine's 37 years - "verify the identity of the Certificate Requester. The method used to verify the identity of the Certificate Requester SHALL be per section 3.2.3.1, 'individual-identity-verification.'"

Okay. So if the corporate entity is less than three years old, then the identity of the requestor is also verified. There were several references to section 3.2.2.1.1 under "Identity," so that definitely comes into play. It says: "If the Subject Identity Information is to include the name or address of an organization" - and it has to - "the CA SHALL verify the identity and address of the organization and that the address is the Applicant's address of existence or operation.

"The CA SHALL verify the identity and address of the Applicant using documentation provided by, or through communication with, at least one of the following: A government agency in the jurisdiction of the Applicant's legal creation, existence, or recognition; a third-party database that's periodically updated and considered a Reliable Data Source; a site visit by the CA or a third party who's acting as an agent for the CA; or an Attestation Letter." Thank goodness.

"The CA MAY use the same documentation or communication described in 1 through 4 above to verify the Applicant's identity and address. Alternatively, the CA MAY verify the address of the Applicant, but not the identity of the Applicant, using a utility bill, bank statement, credit card statement, government-issued tax document, or other form of identification that the CA determines to be reliable." Whew.

I should note that it has become very difficult for individuals to obtain code signing certificates. It's not impossible. There is something known as an IV certificate, an Individual Validation certificate, but not all CAs offer them. Only a couple do. So how do individuals confirm their identity? The Baseline Requirements assert: "A Principal Individual associated with the Business Entity MUST be validated" - that is, I who represent Gibson Research Corporation as its president and CEO - "must be validated in a face-to-face setting. The CA MAY rely upon a face-to-face validation of the Principal Individual performed by the Registration Agency, provided that the CA has evaluated the validation procedure and concluded that it satisfies the requirements of the Guidelines for face-to-face validation procedures." Okay. And I'm going to skip a few paragraphs of this mind-numbing boilerplate.

A Personal Statement has to be provided and signed, providing a full name or names by which a person is, or has been, previously known; residential address at which he/she can be located; date of birth; and an affirmation that all information contained in the Certificate Request is true and correct. A current signed government-issued identification document that includes a photo of the individual and is signed by the individual such as a passport, driver's license, personal identification card, concealed weapons permit, or a military ID. At least two secondary documentary evidences to establish his/her identity that include the name of the individual, one of which MUST be from a financial institution.

"Acceptable financial institution documents include a major credit card, provided that it contains an expiration date and has not expired; a debit card from a regulated financial institution, provided that it contains an expiration date and has not expired; a mortgage statement from a recognizable lender that is less than six months old; a bank statement from a regulated financial institution that is less than six months old. Acceptable non-financial documents." And it goes on like that. I mean, wow.

And then a Third-Party Validator performing the face-to-face validation MUST: Attest to the signing of the Personal Statement and the identity of the signer; and identify the original Vetting Documents used to perform the identification. In addition, the Third-Party Validator MUST attest on a copy of the current signed government-issued photo identification document that it is a full, true, and accurate reproduction of the original.

Now, of course, the Certificate Authority doesn't know who this supposed third-party validator is; right? So the Baseline Requirements state, about the third-party validator: "The CA MUST independently verify that the Third-Party Validator is a legally qualified

Latin Notary," which is a special, like, high-end type of notary whose statements aren't questioned.

Leo: They speak in Latin?

Steve: No, it's weird. I didn't know what it was either, so I did some research. And it is, like, a super special class of notary. "Or a regular notary or legal equivalent in the Applicant's jurisdiction, a lawyer, or accountant in the jurisdiction of the Individual's residency, and that the Third-Party Validator actually did perform the services and did attest to the signatures of the Individual."

And that leads me to the final piece I want to share of this far longer and detailed document, which I'm going to skip most of. Under "Verification of Attestation," the Baseline Requirements say: "The CA MUST confirm the authenticity of the attestation and vetting documents." And it elaborates: "Acceptable methods of establishing the foregoing requirements for vetting documents are: The CA MUST verify the professional status of the Third-Party Validator" - meaning my CPA - "by directly contacting the authority responsible for registering or licensing such Third-Party Validators in the applicable jurisdiction; and the Third-Party Validator MUST submit a statement to the Certificate Authority which attests that they obtained the Vetting Documents submitted to the CA for the individual during a face-to-face meeting with the individual." In my case, that happened between me and my longstanding CPA last Thursday.

And finally: "The CA MUST confirm the authenticity of the vetting documents received from the Third-Party Validator. The CA MUST make a telephone call to the Third-Party Validator and obtain confirmation from them or their assistant that they performed the face-to-face validation. The CA MAY rely upon self-reported information obtained from the Third-Party Validator for the sole purpose of performing this verification process."

Whoa. Now, if all of that leaves you feeling somewhat dizzy, you're not alone. I almost feel guilty, Leo, that I was able to pass through that verification gauntlet.

Leo: You're one of the few, the proud, the verified.

Steve: That's right. I'm somewhat surprised that I was accepted by IdenTrust without first agreeing to a full-body cavity search.

Leo: That's pretty amazing.

Steve: Although I'm pretty sure that I would need a new CPA if that happened. So, okay. Stepping back from all of those gory details for a moment, think about what all this means, why this was done, and what it does and does not achieve in return for all this effort. Our industry is desperately trying to get control of the malware scourge. Among other things, we're seeing attacks at every stage of the software creation process. Source code repositories are being attacked and poisoned. Malicious libraries are given off-by-one-character names in the hope that a developer will introduce a typo at just the right place to invoke the "typo squatted" library to devastating effect. Even AI has been used to invoke a malicious library as a result of a weaponized hallucination.

And you know the most frustrating part of this, in the context of today's discussion of code signing, is that any of these or similar supply-chain attacks would result in compiled

code that is then code signed in good faith by its publisher and accepted by any commercial OS platform despite inadvertently incorporating that infiltrated malware. In other words, it's not as if blessing code with a signature is able to confer any assurance about the behavior of the code that's been signed. It's still got bugs. It might even be malicious. The only thing signing is able to do is assert that not a single bit of the signed code has been altered since its signing, as well as the identity of the signer as it was known to the certificate authority that issued the signer's certificate.

But that said, we're certainly far better off occupying a world where entities who are not interested in deliberately creating malware are able to sign their code and have their unspoofable signatures recognized by the guardians of the platforms we're all using.

So what's the point of all this seemingly over-the-top attestation? Well, with the world's major commercial platforms having become completely unwilling to run any software that's unsigned - Linux excepted - the bad guys must somehow arrange to get their malware signed. Right? One avenue we've seen is to attack the software supply chain in the hope of being incorporated into otherwise legitimate software under the code signing signature of some unsuspecting developer.

The other, much more powerful solution that's available to the bad guys is the direct full frontal approach of obtaining their own legitimate code signing certificate from one of the many trusted certificate authorities. The blockade that now prevents the major commercial OS platforms from executing any code that has not been signed has created huge pressure to spoof corporate identities, or just make up, synthesize a corporate identity, in order to trick certificate authorities into issuing valid code signing certificates to explicitly malicious parties. Fraudulent code signing certificates are a real problem. This explains why, today, it's the reputation of the signing certificate that matters, not just its existence.

The CA/Browser Forum understands that what they just put me through was inconvenient as all heck and a pain in the butt. But what choice do we have? They cannot simply take the word of anyone who may be able to recite, you know, "A Boy Scout is trustworthy, loyal, helpful, friendly, courteous, kind, obedient, cheerful, thrifty, brave, clean, and reverent." No. That doesn't cut it. They clearly need another trust anchor. And that anchor is a licensed attorney or CPA who will be willing to put their own reputation and license on the line to substantiate and attest to the identity of the code signing certificate applicant.

Given what I just went through, anyone who may have forgotten or may have been putting off obtaining a three-year code signing certificate has about 10 days from today to get that done. So if you want to get a certificate good for three years, you can from IdenTrust. And I was very impressed with how quickly they moved. If you are attempting to establish you or your company's identity with a new certificate authority like IdenTrust, take the need for an attestation letter from an attorney or CPA to heart. It may save you, as it would have for me, another couple days that you might not have remaining because you want to squeak into February. And I would expect the code signing certificate authorities to be a bit busy as these last days of February expire and three-year certificate availability winds down.

And remember, if you want to avoid cloud-based pay-as-you-go or limited-quantity code signing, having your own signing hardware is now a requirement. And if you want to get that done now, you'll be able to use it, whatever you do, for three years. I'm glad I'm doing that. I want to get this new certificate from IdenTrust since my current certificate with DigiCert lasts through August, at which time I will not be getting another one from them. My plan is to dual sign my software so that the world has a chance to see this new certificate, and but also sees that it's cosigned with the already almost, well, now it's 2.5

years old, DigiCert code signing certificate. And then the DigiCert certificate will drop off after it expires. So, boy, I mean, it is a pain in the butt.

Leo: This doesn't feel very robust, I have to say.

Steve: No, it's not. I mean, you're right. You could get somebody to fake a CPA or fake an attorney.

Leo: Right.

Steve: And, you know. I mean, but what do we do?

Leo: There's got to be a better way to do this. There just has to be. It just feels like they're not improving it. They're just kind of layering stuff on.

Steve: Well, and the price. I mean, on one hand, okay, they clearly had to go...

Leo: Well, that's the real point, I guess.

Steve: ...jump through some hoops. But, boy, are they making it expensive just to produce code.

Leo: Yeah, and I feel like that's the point is to make money off of you producing code.

Steve: Unfortunately. I mean, I like DigiCert. But as I looked around, I found that GlobalSign and, like, there were like four other CAs...

Leo: I use GlobalSign for MIME certs, yeah.

Steve: Yeah.

Leo: There are others, yeah. And ACME is not a solution to any of this? Not the code signing?

Steve: The what?

Leo: ACME. None of this can be used for code signing.

Steve: No, not for code because ACME is explicitly saying I control this domain.

Leo: Right.

Steve: Code signing is I am this identity.

Leo: I'm me, yeah.

Steve: So it's I am me, exactly.

Leo: Yeah. Authentication is so hard. But maybe, you know, Sam Altman's got the right idea with the Orb, the iris-scanning Orb. I mean, he recognizes this is an issue. This is going to be an issue in the new world. How do you prove you are who you say you are?

Steve: Well, and think about it. I mean, a global network decouples you from identity. We've been heralding that as the great liberation. It frees us. It's, oh, my god, it's, you know, we get to be autonomous, and you can be a dog if you want to be.

Leo: Right.

Steve: Unfortunately, there are instances where it really does matter. Bad guys will abuse that very anonymity. And so it turns out clamping down on it is really hard.

Leo: Yeah. Neal Stephenson writes about this in his book "Fall, or Dodge in Hell." And what he talks about is having kind of a variety of identities you can assume. You have your real identity, which you can prove. But in order to allow anonymity and flexibility and autonomy, you also have other identities that are spawned from your real identity that can't be connected back to your real identity. And I think that we'll end up with something like that. It might be tied to some sort of TPM, you know, hardware or something. But we'll end up with something, a chip implanted in your brain at birth or something. It's got to be solved. We've got to solve it.

Steve: Yeah.

Leo: It's a big issue.

Steve: Well, and we're running smack into it with the whole age restriction deal.

Leo: Exactly.

Steve: That all of our politicians have something to say, well, we don't know how you're going to solve it. But, you know, you guys are smart.

Leo: Nerd harder, yeah.

Steve: Yeah.

Leo: You'll figure it out. What an interesting subject. I feel like authentication is one of the most interesting and thorny problems we have. And it's a necessity. We need to solve it.

Steve: That's why I spent seven years on SQLR was that, you know...

Leo: Right, right.

Steve: ...it was really worth fixing.

Leo: Yeah. That's the guy. That's the SQLR guy, Steve Gibson. He's at GRC.com.

Steve: Bye.

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/>