

# Security Now! #1065 - 02-17-26

## Attestation

### This week on Security Now!

- 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 4000 user's credentials.
- Some AI "vibe" coding thoughts.
- What I just went through to obtain a new code signing certificate.

### Placing unconditional trust in technology can lead to mistakes



# Security News

## Low-Traffic DDoS

Last week we noted the fact that during Sunday's Superbowl, the company with the expensive \$70 million domain name, [AI.COM](#), had been DDoSed by their own advertisement. Leo showed us the Cloudflare screen that indicated that all was well with the CDN delivering traffic to the backend hosting server, and that the hosting server was not responding. And Leo, you also mentioned that someone close to you, I don't recall who, had taken to blocking AI bots since they were bringing down his small site?

The point I wanted to follow-up with, and we've talked about this before, is that modern websites – both large and small – are no longer generating their content the way most of GRC still is, with hand-authored lightweight HTML and CSS. The modern way to create a website is with a CMS, a content management system, where the webserver runs server-side scripting using PHP, Ruby, JavaScript with Node, Java, C#/.NET or perhaps Python. And one of those content engines obtains its data from some backend database.

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 and database load. That chart Leo showed us last week indicated that Cloudflare was faithfully delivering HTTP queries to whatever backend server infrastructure [AI.COM](#) had built out by that point. But whatever it was, it was unable to scale, as needed, to handle the massive demand spike created by a superbowl ad. Mostly it was just embarrassing and it was certainly not an auspicious launch of a new venture.

And the consequences of high-cost webpage delivery are being felt everywhere. Last Wednesday's Linux Mint Blog addressed problems with its forums, writing:

*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 couldn't 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 traffic. If you're getting 403 errors from the forums right now, please make sure your browser is up to date. We upgraded the server to give it 10x the CPU capacity and twice the bandwidth.*

Linux Mint's forums use the free phpBB. I don't know whether they've spent any time speeding up their implementation using the in-memory OpCache and Redis key-value stores. But this is a typical example of what's happening with web sites. Page delivery efficiency has dropped as site complexity has increased while steady-state background traffic is increasing from many sources.

The takeaway is to remember that connection bandwidth is almost certainly no longer the limiting factor it once was and it can be practically impossible to change platforms once one is committed. Page delivery overhead and performance have become modern bottlenecks. Cloud-based hosting providers are sharing their resources among multiple customers. So the more resources a customer requires to get their job done the more that job will cost. This suggests that page delivery efficiency matters to the bottom line. So, whenever you're planning a new installation and have a choice of implementation platform, it might be worth keeping in mind that while ease of initial setup and ongoing management and maintenance is certainly a consideration, the system's operating efficiency may significantly impact that solution's long-term operating cost.

## Seriously Risky Business: Microsoft's Forgoes Its Secure Future

I want to begin today by sharing an editorial which appeared in the Seriously Risky Business publication which was unfortunately titled: "*Microsoft's Forgoes Its Secure Future.*" Afterward I'll share a couple of observations. They wrote:

*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 organisation signaling a shift in priorities. We fear Microsoft's goal now is not to make secure products, so much as to sell security products.*

*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 organisation to become an individual contributor 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 prioritised security when it was forced to by government pressure.*

*Bell joined Microsoft from AWS to lead a new security organisation 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 clangers." Those rolling security debacles were a symptom of senior leadership prioritising profit over security. At the time 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 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 lacklustre security uplift program, the Secure Future Initiative, in late 2023.*

*Microsoft's devil-may-care approach to security 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 US 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 prioritise security "above all else" and that "if you're faced with the tradeoff between security and another priority, your answer is clear: **Do security**" [emphasis in original].*

*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 signalled 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 focussed 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." 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.*

I wanted to share this to highlight a lesson we've all learned throughout the past 20+ 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 have, 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. It's only the inevitable consequences of a relaxed security posture that will wind up being felt.

As for why Microsoft might have made 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. *"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.

### **Windows 11 26H1**

I suppose I should at least mention that this spring Microsoft will be introducing what they are 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 26H1 will come pre-installed on those machines. It only runs on them and it will not be available 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 – is, despite its name, **not** an update for 25H2. Everyone else should just ignore it.

### **Chrome 145 & Device Bound Session Credentials (DBSC)**

Last Tuesday, Google updated the world to Chrome 145. This update repaired the typical assortment of high, medium and low severity security issues and continued to move Chrome's support for the latest HTML, CSS and JavaScript standards forward. Perusing those, my utter astonishment over the complexity of today's web page content interpreters has been renewed. Web browser complexity only gets more insane by the day.

The one new feature that stood out is Chrome 145's new support for something known as Device Bound Session Credentials (DBSC). Just think about that phrase for a moment. Device bound session credential.

A session credential is just the fancy name for a cookie. And device binding would mean binding a session credential cookie to the device whose web browser receives that cookie from a remote web site. 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 elsewhere. 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 successfully logged on. This allowed anyone who could eavesdrop on Internet traffic anywhere to capture those logged on session cookies to impersonate their rightful owner.

Although things are much better today, there are still various interception attacks and mechanisms that create vulnerabilities and weaknesses. For example, though it's being done for the best and most justifiable reasons, many enterprises maintain TLS-decrypting middleboxes that decrypt everyone's TLS connections as they cross the enterprise network edge in order to scan them for malware and other shenanigans. 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 reply, the session's logon credential cookies would be exposed.

Until now, the browser cookie has been a somewhat fragile authentication mechanism, but it's all we've had. With this innovation of Device Bound Session Credentials, that changes. Although some form of secure enclave, such as a TPM – a Trusted Platform Module – is required, all modern OS platforms require this already for themselves. This change does require explicit support at the web server side.

If anyone's curious, I described the operation of this in detail during episode #1021, last April 18th. It took longer than expected to arrive, but we have it now. It does require significant support from the web server, but it's the sort of advance that's very likely to eventually become widespread.

### **More countries moving to ban underage social media access**

I noted that the governments of 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 any opportunity for parental 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. The commentator clearly understood that parents would be hard pressed not to succumb to the argument: *"But Suzie's parents let her use Instagram and she's younger than me!"*

### **The return of Roskomnadzor**

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 known as NSDI. And Roskomnadzor controls what's listed and what's not. Though access to YouTube and WhatsApps had been throttled since last July (and we talked about that at the time) now those two domains 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 VPN, APK Mirror, the BBC, and several other news sites.

## Discord's new age-assurance policy

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 dollar 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 their own clarification explanation, Discord wrote:

*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.*

- *Facial scans never leave your device. Discord and our vendor partners never receive it.*
- *IDs are used to get your age only, and are then deleted.*
- *Discord only receives your age — that's it. Your identity is never associated with your account.*

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 close to having any universal solution or even a standard yet. I'm sure we will in time since the entire world appears to be waking up to the need to determine whether someone is or is not of a certain age. Are privacy purists losing some of their precious (if entirely illusory and fictitious) privacy? Yes. That's going to happen. But even that will be better in the future once stronger privacy-protecting standards are in place.

## WinRAR before v7.13

When I saw that GTIG, Google's Threat Intelligence Group, had identified a widespread active exploitation of the critical vulnerability in WinRAR which we first talked about last summer, although I was certain that I had updated my copy, I double checked and Yikes! I was still using v7.12 which contained the vulnerability. I'm not using v7.20 but I decided that given that the threat has moved from theoretical to 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:

*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 initial access and deliver diverse payloads. Discovered and patched in July 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. 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 IOCs 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 and to install security updates as soon as they become available. After a vulnerability has been patched, malicious actors will continue to rely on n-days and use slow patching rates to their advantage. We also recommend the use of Google Safe Browsing and Gmail, which actively identifies and blocks files containing the exploit.*

*CVE-2025-8088 is a high-severity path traversal vulnerability in WinRAR that attackers exploit by leveraging Alternate Data Streams (ADS). Adversaries can 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 30, 2025.*

That's enough said. For anyone wanting more information and details I've included the link to Google's full coverage in the show notes, and also to WinRAR's download page, which is: [win-rar.com/download.html](https://www.win-rar.com/download.html): <https://www.win-rar.com/download.html>  
<https://cloud.google.com/blog/topics/threat-intelligence/exploiting-critical-winrar-vulnerability>

If you DO discover a version of WinRAR before 7.13, as I did, you can know at least that we are in good company. This was brought to my attention by a Stairwell Security who wrote:

*Stairwell recently identified a significant and concerning trend across our customer base: **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.*

*Google identified the exploitation of CVE-2025-8088 that impacts Windows versions of WinRAR earlier than 7.13, a range that spans many years of releases. WinRAR remains one of the most commonly installed archive utilities in enterprise and developer environments, often persisting long after its initial installation.*

*Because WinRAR is frequently used to handle untrusted archives received via email, download portals, or shared file systems, vulnerabilities in the application are especially attractive for exploitation. Attackers can reliably assume its presence and leverage it as an initial access or execution vector.*

So I say again... Yikes!

### **Paragon Solutions: Graphite**

We've talked about the "Graphite" spyware before. It's one of the more capable systems. But it's one thing to hear about it and another thing to see it. Israeli's Paragon Solutions made a mistake which exposed details of its "Graphite" spyware control panel. The panel was exposed in photos from a demo day in the Czech Republic. The photos, which were immediately taken down, revealed Graphite's ability to extract messages from instant messaging clients including WhatsApp, Signal, Telegram, Line, Snapchat, TikTok, and more.

We already know that WhatsApp and Signal are truly secure and that Telegram 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. The point here 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 or fight with Moxie’s triple ratchet. All it needs to do is pretend to be the device’s user, examine the decrypted data that’s presented on the device’s screen and send that back to central headquarters.

So those leaked photos conclusively demonstrate that once a smartphone has been lubed up with Paragon’s Graphite none of its secrets will be safe from spying eyes.

### 30 Malicious (fake) Chrome AI Extensions

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 quick and clever about jumping onto anything for which there’s a large demand. 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 we should not 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 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.*

*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. 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.*

*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, JavaScript logic, permissions, and 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, increasing their perceived legitimacy and exposure.*

*This 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 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.*

*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.*

So, yeah. . . more than 260,000 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. We couldn't be using the password manager of our choice without them. But their diversity and popularity has overwhelmed Google's ability to examine and manage, such that today's web browser ecosystem creates serious privacy vulnerabilities.

### **Would you believe: 287 Chrome extensions found spying on 37.4M users?**

And speaking of problems with, specifically, Chrome browser extensions, would you believe that another researcher discovered 287 distinct Chrome extensions were collectively sending their publishers the URL-by-URL browsing events of their 37.4 million users? Google and Chrome really have some serious problems.

This researcher, posting on Substack under the handle "Q Continuum" wrote the following:

*We built an automated scanning pipeline that runs Chrome inside a Docker container, routes all traffic through a man-in-the-middle (MITM) proxy, and watches for outbound requests that correlate with the length of the URLs we feed it. Using a leakage metric we flagged 287 Chrome extensions that exfiltrate browsing history. Those extensions collectively have ~37.4 M installations – roughly 1 % of the global Chrome user base. 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.*

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

*Fast forward to 2025: Chome Store now hosts roughly 240 k extensions, many of them 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 collects data.*

*Why does it matter? There is 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 are using for free and it is 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 collected traffic can try to target individuals.*

### **KOI Security details an interesting and worrisome attack**

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*". One of the topics I've had pending to talk about in greater detail are so-called "*Domain Recovery Attacks*". They can be quite serious and they reveal an aspect of our security that's been largely overlooked. 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, 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 were able to recover the full scope of the operation: over 4,000 stolen Microsoft account credentials, credit card numbers, and banking security answers. The attacker was actively testing stolen credentials yesterday. The infrastructure is live as you read this. This is the story of how a dead side project became a phishing weapon.*

*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 in an iframe inside Outlook." Microsoft reviews the manifest, signs it, and lists the add-in in their store. But the actual content - the UI, the logic, everything the user interacts with - is fetched live from the developer's server every time the add-in opens.*

Boy, that really sounds like an architecture that's asking for trouble. And trouble appears to be what was delivered. They continue:

*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 live immediately. 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 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 reviews) and an Outlook add-in published to Microsoft's store in December 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 someone building a business. Then development stopped. The last Chrome extension update shipped in May 2023. The developer's domain, agree.to.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 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.*

Their description continues with all of 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. We've all encountered abandoned domains that have been acquired by low-end advertisers who snap up web domains that have expired and been abandoned and host content that no one wants in the hope of generating revenue from advertisers who will pay for any traffic.

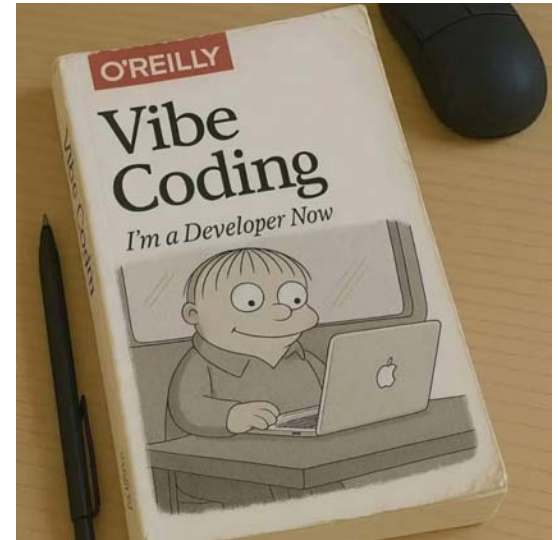
But when domains are used to host important content things can quickly take a turn for the worse. Years ago we examined an instance where the domain of an important and super-popular web browser JavaScript library had changed hands. Suddenly, an incredible number of web browsers were pulling a critical library from someone else. It should be enough to keep one up at night.

# Listener Feedback

Walt Stoneburner

*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 love coding, that'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 vibing does not do. -Walt in Ashburn*



For me, I think that the most unnerving aspect of vibe coding 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 the various strange errors Claude Code made were obvious to its user; like the book author’s name appearing twice in its field. When that’s pointed out to Claude it says “oh, yeah, sure enough” and then it fixes it.

But this should give any true coder pause to wonder what other far more subtle errors might be lurking in there that haven’t yet 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 interactions and places where subtle errors might hide.

Then I challenge myself and say: *“Okay, hold on there a second, Gibson. When you use a library authored by some 3rd-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 3rd-party that its non-AI author took pride in their creation and deliberately and knowingly wrote and tested the function of each and every one of the library’s functions. 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 condition. This echoes some of the early formal code-correctness verifications that programmers have been doing by hand. It’s considered the only way to know for sure. 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. 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, Copilot was shown a bug in some code where a parser was running off the end of the stack it was parsing.

Rather than fixing the underlying error – because a stack underflow should not have been possible – Copilot added some glue, an explicit test to prevent the pointer underflow. 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 left shoulder hurts whenever he raises his left arm a certain way. His doctor says “No problem, just don’t raise your arm like that.” 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 vibe coding play out.

### **Denny Vandemaele**

*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 a scary thought 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: <https://one.one.one.one/> It works well and can be installed on Apple macOS and iOS, Android, Windows and Linux. -denny*

I had forgotten about Cloudflare’s free WARP VPN offering. Thanks Denny for the reminder!

# Attestation

Today, I want to share the details of a surprising, unplanned and annoying adventure I experienced last week. After everyone has caught up with me in appreciating this new reality which the entire industry is now being subjected to, I then want to share the specific legal requirements underlying what happened to me last week.

As I have noted – and warned – previously, 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 (on Monday, January 26th) being a customer of DigiCert I received a piece of email with the subject "*Important Reminder: TLS/SSL certificate lifetimes changing Feb 24, 2026*". The email said:

*Hello,*

*We're writing to remind you that starting February 24, 2026, TLS/SSL certificates issued through DigiCert CertCentral will have a maximum validity of 199 days (down from 397 days).*

*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, they also reduce risk and help keep your environments more resilient over time. DigiCert continues to invest in automation and tooling to make renewals as seamless as possible and our team is here to help you prepare and answer any questions.*

*Below is a brief summary of the impact this change will have to your certificate environment.*

- You can continue to order TLS/SSL certificates with validity longer than 199 days until February 24, 2026.*
- Any certificates issued on or after February 24 (including pending requests made prior to February 24) will be limited to a maximum of 199 days, even if the order requested a longer validity period.*
- If you need certificates with longer validity, we recommend placing your orders well before February 24, 2026, and making sure your domain and organization validations are up to date to avoid delays.*
- Existing certificates are not affected by the change to 199-day validity. Certificates issued before February 24 will remain valid until their original expiration date.*
- However, if an existing certificate is reissued on or after February 24, it will be limited to 199 days, even if the original certificate was valid for longer.*

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 codesigning 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 to just 6 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 bygone days, when certificates lasted for five or ten years, 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. DigiCert's email continues, writing:

- *On February 24, OV organization validation reuse periods will be shortened from 825 days to 397 days.*
- *On February 24, domain validation reuse periods will be shortened from 397 days to 199 days.*

In other words, it will now also be necessary to re-validate one's organization annually rather than only every two and a quarter years. Given that Let's Encrypt only offers domain validation certificates which incur none of this nonsense, I have a difficult time understanding how the CA's are not putting themselves out of business. I suppose they plan to survive on all of the other various types of certificates, such as for signing documents and such, and they'll continue to offer TLS web certs as a loss leader so that they offer a full suite of certificate services.

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 certificate well in advance of DigiCert's February 24th deadline. With certification having become so involved there's no telling when or why the process will fail. I've been surprised in the past, so I wanted to give myself time to fix anything that might fail before the deadline. The process proceeded without a hitch, so just because I can, I plan to reissue again next Monday morning the 23rd, on the last possible day.

So this should serve as a heads up reminder to anyone who might similarly have better things to do right this moment than figure out how to switch their certificates over to Let's Encrypt. I'll be moving there once my pre-purchase with DigiCert runs out.

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

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 3-year life. I would prefer to remain with DigiCert. But the recent changes at DigiCert have overcome my "change inertia" 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 cannot 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 3 years – 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 for a year. And even that \$600 per year is

presumably subject to change at the next annual billing cycle since there's no longer any way to pre-purchase future years.

While I'm bitterly disappointed in DigiCert to whom I've felt a well-deserved loyalty for many years, I don't really mean to single them out. The entire codesigning certificate industry appears to be headed in the same direction. And it's not pretty.

Scouting around, I found that IdenTrust will "sell" a no-strings-attached 3-year codesigning certificate for \$538 when I place it into my own HSM. So that's \$179/year for 30% the cost of remaining with DigiCert, and that's assuming that DigiCert doesn't choose to further raise their prices before the next 3 years have passed. IdenTrust is well known, so it was IdenTrust 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 jump: The CA/Browser forum requires the issuing Certificate Authority to obtain an "attestation letter" from an independent legally licensed attorney or CPA, a Certified Public Accountant. This third-party individual must attest to having firsthand knowledge of the legitimacy of the corporation and its officers.

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 doubted the need for this attestation letter (which I've never needed or been asked for before) and IdenTrust's documentation was unclear about it.

So one week ago, 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 delivery.

Last Wednesday morning at 11:32 I received email confirmation of the forms having been received and 35 minutes later, at 12:07, I received notice with the Subject: ***"ACTION REQUIRED: Code Signing Application - Attestation Letter Required"*** <grumble> Oh, great.

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 always quite frugal, would have 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 to attest to Gibson Research Corporation's identity. He didn't hesitate to say "yes". So Wednesday afternoon I emailed IdenTrust's 3-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 my CPA had) or a "wet signed" original. My CPA printed and filled-out the PDF and signed it in nice blue ink. 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. My reason for my sharing all of 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 burden that malware has inflicted upon the world. I dislike what I've 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 3rd-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 CA's, 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. These requirements were already in place when I obtained my most recent EV code signing certificate and 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 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 was also just thoroughly researched. Today's podcast is titled "Attestation" because I want to share what I just learned about the extent of what this "attestation" means. It's quite 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*" 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.

Near the top of the Baseline Requirements is a section of definitions. For "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 Code Signing Certificates. This reminds me to make sure everyone understands that this is for the non-Extended Validation certificates. I don't even want to think what might be required to establish Extended Validation with a new certificate authority. That section about organization identity says:

*Prior to issuing a Code Signing Certificate to an Organizational Applicant, the CA MUST:*

*1. Verify the Subject's legal identity, including any DBA proposed for inclusion in a Certificate, in accordance with section 3.2.2.1.1 - "Identity" and section 3.2.2.1.2 - "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,*

*2. Verify the Subject's address in accordance with section 3.2.2.1.1 - "identity",*

*3. 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 in accordance with section 3.2.5 - "validation-of-authority", and*

*4. 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, 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".*

So if the corporate entity is less than three years old then the identity of the requestor is verified rather than the identity of the corporation. There were several references to section 3.2.2.1.1 "Identity" so that definitely comes into play. It says:

*If the Subject Identity Information is to include the name or address of an organization, 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:*

- 1. A government agency in the jurisdiction of the Applicant's legal creation, existence, or recognition;*
- 2. A third party database that is periodically updated and considered a Reliable Data Source;*
- 3. A site visit by the CA or a third party who is acting as an agent for the CA; or*
- 4. An Attestation Letter.*

*The CA MAY use the same documentation or communication described in 1 through 4 above to verify both 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.*

I should note that it has become very difficult for individuals to obtain code signing certificates and it's not possible for individuals to obtain EV certificates. Individuals must first create a sole proprietorship and then register a fictitious business name, a DBA (doing business as), and then get listed in a business registry such as Dun & Bradstreet. And if that was not done more than three years before they'll also need to have their own identity verified.

So how do individuals confirm their identity? The Baseline Requirements assert:

*A Principal Individual associated with the Business Entity 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. Where no face-to-face validation was conducted by the Registration Agency, or the Registration Agency's face-to-face validation procedure does not satisfy the requirements of the Guidelines, the CA SHALL perform face-to-face validation.*

*The face-to-face validation MUST be conducted before either an employee of the CA, a Latin Notary, a Notary (or equivalent in the Applicant's jurisdiction), a Lawyer, or Accountant (3rd-Party Validator). The Principal Individual(s) MUST present the following vetting documentation directly to the Third-Party Validator:*

*1. A Personal Statement that includes the following information:*

- 1. Full name or names by which a person is, or has been, previously known;*
- 2. Residential Address at which he/she can be located;*
- 3. Date of birth; and*
- 4. An affirmation that all information contained in the Certificate Request is true and correct.*

*2. A current signed government-issued identification document that includes a photo of the Individual and is signed by the Individual such as:*

- 1. A passport;*
- 2. A driver's license;*
- 3. A personal identification card;*
- 4. A concealed weapons permit; or*
- 5. A military ID.*

*3. 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.*

*1. Acceptable financial institution documents include:*

- 1. A major credit card, provided that it contains an expiration date and it has not expired'*
- 2. A debit card from a regulated financial institution, provided that it contains an expiration date and it has not expired,*
- 3. A mortgage statement from a recognizable lender that is less than six months old,*
- 4. A bank statement from a regulated financial institution that is less than six months old.*

*2. Acceptable non-financial documents include:*

- 1. Recent original utility bills or certificates from a utility company confirming the arrangement to pay for the services at a fixed address (not a mobile/cellular telephone bill),*
- 2. A copy of a statement for payment of a lease, provided that the statement is dated within the past six months,*
- 3. A certified copy of a birth certificate,*
- 4. A local authority tax bill for the current year,*
- 5. A certified copy of a court order, such as a divorce certificate, annulment papers, or adoption papers.*

*The Third-Party Validator performing the face-to-face validation MUST:*

- 1. Attest to the signing of the Personal Statement and the identity of the signer; and*
- 2. 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 3rd-party validator is. So the Baseline Requirements state, about the 3rd-party validator:

*The CA MUST independently verify that the Third-Party Validator is a legally-qualified Latin Notary or Notary (or legal equivalent in the Applicant's jurisdiction), 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 signature of the Individual.*

And that leads me to the final piece I want to share of this far longer and detailed document.

Under "*Verification of Attestation*" the Baseline Requirements say: "*The CA MUST confirm the authenticity of the attestation and vetting documents.*" and then elaborates:

*Acceptable methods of establishing the foregoing requirements for vetting documents are:*

- 1. The CA MUST verify the professional status of the Third-Party Validator by directly contacting the authority responsible for registering or licensing such Third-Party Validators in the applicable jurisdiction;*
- 2. The Third-Party Validator MUST submit a statement to the CA 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:

*3. 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.*

Wow. If all of that leaves you feeling somewhat dizzy, you're not alone. I almost feel guilty that I was able to pass through that verification gauntlet! I'm somewhat surprised that I was accepted by IdenTrust without first agreeing to a full body cavity search, though I'm pretty certain I would have needed to find a new CPA.

Stepping back from all of these gory details for a moment . . . think about what this all means, why this was done and what it does and does not achieve in return for all of 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 with 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's 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 having signed code is able to confer any assurance about the behavior of the code that's been signed. 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?

With the world's major commercial platforms having become completely unwilling to run any software that's unsigned, the bad guys must somehow arrange to get their malware signed. 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 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 presence.

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 they have? They cannot simply take the word of anyone who may be able to recite that "*A Boy Scout is trustworthy, loyal, helpful, friendly, courteous, kind, obedient, cheerful, thrifty, brave, clean, and reverent.*" Nope. That won'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 3-year code signing certificate has about 10 days from today to get that done. So do not delay. And if you are attempting to establish your or your company's identity with a new certificate authority take their need for an attestation letter from an attorney or CPA to heart. It may save you another couple of days that you might not have. I'd expect the code signing certificate authorities to be a bit busy as these last days of 3-year certificates wind down.

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 get that done now you'll be able to use it for the next three years.

