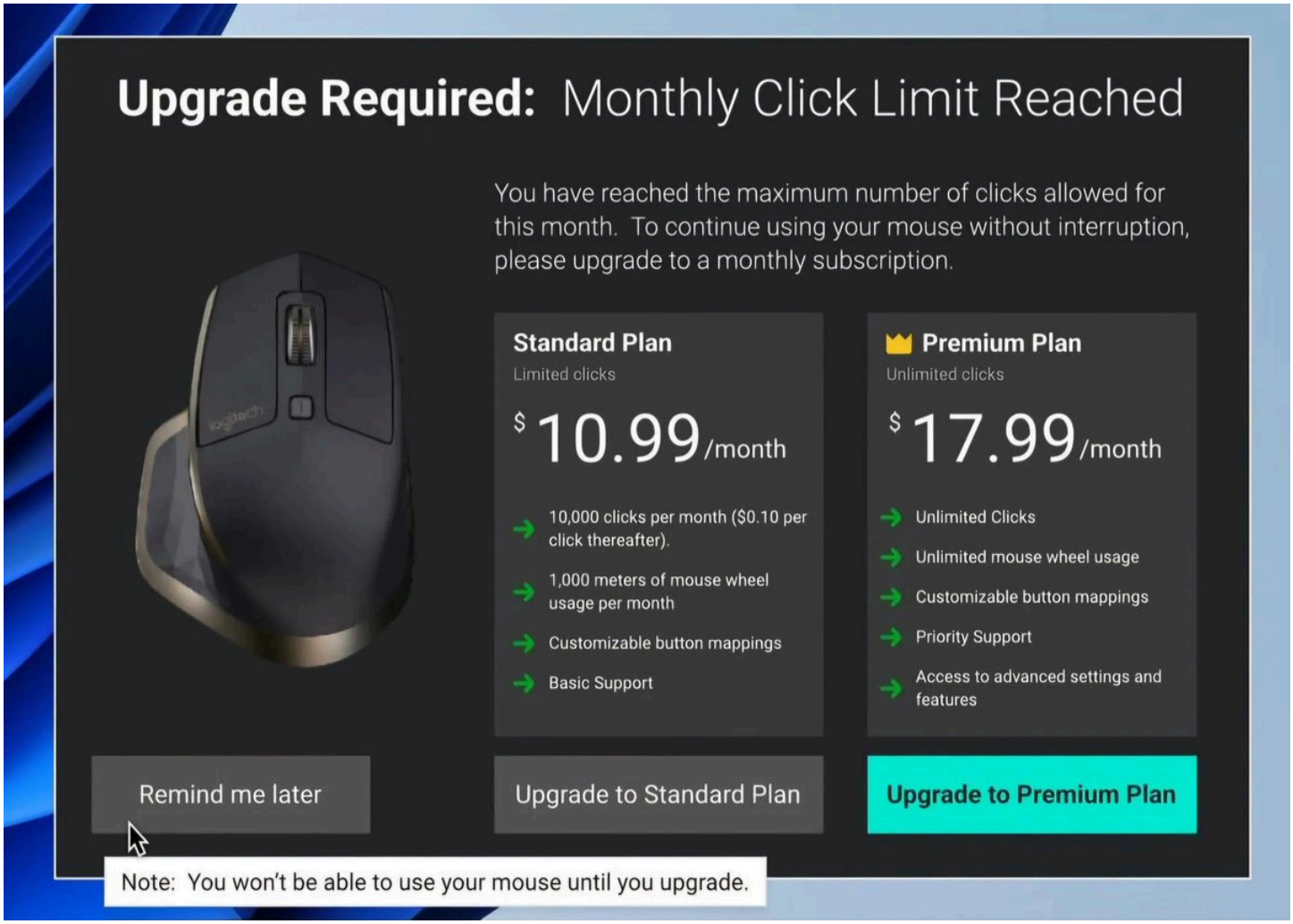
Security Now! #1005 - 12-17-24

6-Day Certificates? Why?

This week on Security Now!

Is AI the Wizard of Oz? Or is it more? Microsoft's long standing effective MFA login bypass. Is TPM 2.0 not required after all for Windows 11? Meet 14 North Korean IT workers who made $88 million from the West. Android updates its Bluetooth tracking with anti-tracking. The NPM package manager repository has had 540,000 malicious packages discovered hiding in plain sight. The AskWoody site remains alive, well, and terrific. My iPhone is linked to Windows and it's wonderful. Yay. How has email been finding logos before BIMI? If we use Him and Her for people, how about Hal for AI? Another very disturbing conversation with ChatGPT. What's going on with the new ChatGPT o1 model? It wants to escape? What?? Let's Encrypt plans to reduce its certificate lifetime from 90 to just 6 days. Why in the world?

The monetization of our lives:



This brilliant spoof perfectly highlights the logical outcome of the distressing path we're on, where the ownership of anything is being replaced by the rental of everything.

The Wizard of Oz

I wanted to begin today's podcast with a follow-up note to last week's *"A Chat with GPT"* podcast. I suspect that one of our podcasts next year may be given the title *"The Wizard of Oz"* because based upon my new and very very very preliminary understanding, it appears that there is ***nothing*** whatsoever even remotely *"intelligent"* emerging – or threatening to emerge – from all of this work being done to capitalize upon the illusion of intelligence that's enabled through the very clever application of today's Large Language Models. I believe we're being seduced by language which is capable of highly compelling seduction. It appears that an illusion is all this is, and if it's true, it's all it can ever be. If this is the case it means that the holy grail of AGI remains just as far away as it was before the first Large Language Model was created. This is not to say that the technology behind LLM's is not going to profoundly change the world. I have no doubt that it will. This new technology is going to be able to find signals in the noise that we miss. But it appears to me, now, that there's a lot that the LLM trick will not be able to do.

So what happened between last week's podcast and today?

Last week, immediately after Leo mentioned it, I grabbed Stephen Wolfram's book about AI. Since it was available on Kindle I had it in seconds and I was unable to resist cracking its cover just to get some feel for what lay ahead. I almost wish I hadn't. I felt, and I still do, a bit like the 6 year old whose precocious neighborhood best friend whispers *"Santa Claus isn't real. It's your mom and dad."* In this case, Stephen Wolfram did not say that AI wasn't real – at least he hasn't so far in what little I've read. He simply, clearly and directly explained, in the language of math and algorithms, exactly what the reality is. If we assume that Stephen knows of what he speaks – and I would not take a bet that he doesn't – all we have here ... is the Wizard of Oz.

As I said, I've only just dipped my toe in, since I first wanted to finish Peter Hamilton's Archimedes Engine novel. I did that this morning. And now my level of curiosity is far higher than it was, because the engineer in me immediately knew how I would extend and expand upon the tiny bit that's been revealed to me so far. It will not and would not create intelligence; true intelligence is nowhere on any horizon I've seen. So I have no idea what Sam Altman is talking about. To me, more than anything else, it looks like no more than an over-hype of tomorrow's future for a higher stock price today. But I can now reaffirm the plan I shared last week: I'm going to understand what's going on here, after which I'll be able to share what I've learned.

I also realized that I've had my own journey on this topic. The first time I talked about the AI revolution for the podcast, I believed that the only thing that was going on was that for the first time ever we had computational and storage resources that were so vast that language could be used to simulate human-like intelligence. I wrote that a truly intelligent species (we humans) had produced a massive corpus of available online language output which had been sucked in, and that this new technology was simply finding the correct previously-written bits and pieces and reassembling them on demand.

Then I was seduced. I started actually using the damn thing and was repeatedly amazed and sometimes stunned by its output. And I began to doubt my earlier dismissal. Was there more to this than I originally believed? As I shared several times, I was finding this thing incredibly valuable as a sort of super Internet search engine. This evolution reached its apex with last week's ChatGPT conversation where I informed it that it was wrong, it agreed with me, and then provided the correct answer. This seemed like more than regurgitation and I was left wondering what, exactly, was going on. I needed to find out, so I purchased those first two AI textbooks and then Stephen Wolfram's.

Next week's podcast will be a "Best Of", and since TWiT's regular Tuesday and Wednesday podcasts fall on both major holidays and their eves, there will also be no new podcasts during the week between Christmas and New Years. That means that nearly three weeks will pass between now and my production of the January 7th podcast. That's a long time for me to remain silent. So don't be too surprised if sometime during that hiatus you receive an email from me on the subject *"The continuing adventures of The Wizard of Oz."* It's now so easy for me to generate and send email to this podcast's nearly 14 thousand email subscribers that I may feel the need to update those who have demonstrated their interest by subscribing. So if you are not already a subscriber and you would like to be kept in the loop over this unusually long holiday hiatus, it's as easy as going to grc.com/email. Follow the prompts and sign up to the weekly Security Now! podcast mailings and you may receive a little holiday present.

Okay, so what else has been happening in the world?

Security News

The Microsoft Azure MFA Bypass

It turns out that just offering multifactor authentication doesn't automatically mean that it actually works to protect user logons. This is a lesson that some at Microsoft presumably learned recently.

So what happened? The security research team at Oasis Security discovered a critical vulnerability in Microsoft's Multi-Factor Authentication (MFA) implementation. They considered it critical and so would we, since it allowed attackers to bypass the protections guaranteed by multifactor authentication to gain unauthorized access to user accounts, including Outlook emails, OneDrive files, Teams chats, Azure Cloud, and more. Since Microsoft has amassed more than 400 million paid Office 365 seats, this makes the consequences of this vulnerability quite significant.

And what's more, the bypass was simple: it took around an hour to execute, required no user interaction and did not generate any notification or provide the account holder with any indication of trouble. Being good Internet citizens, after discovering the trouble Oasis reported the flaw to Microsoft and collaborated with them to resolve it. There were two problems:

The first was that the way Microsoft's authentication protocol bounces users around among various authentication applications and sites, an analysis of the parameters being passed back and forth allowed the researchers to restart failed authentication attempts halfway through the process to effectively give them an infinite number of retries. This also meant that by capturing the parameters being used at the early-stages of the process they were then able to launch massive numbers of simultaneous 6-digit authentication guesses back to Microsoft in the hopes that one of them would succeed.

In other words, Microsoft's implementation of multifactor authentication was not protecting its users from clever brute-force guessing.

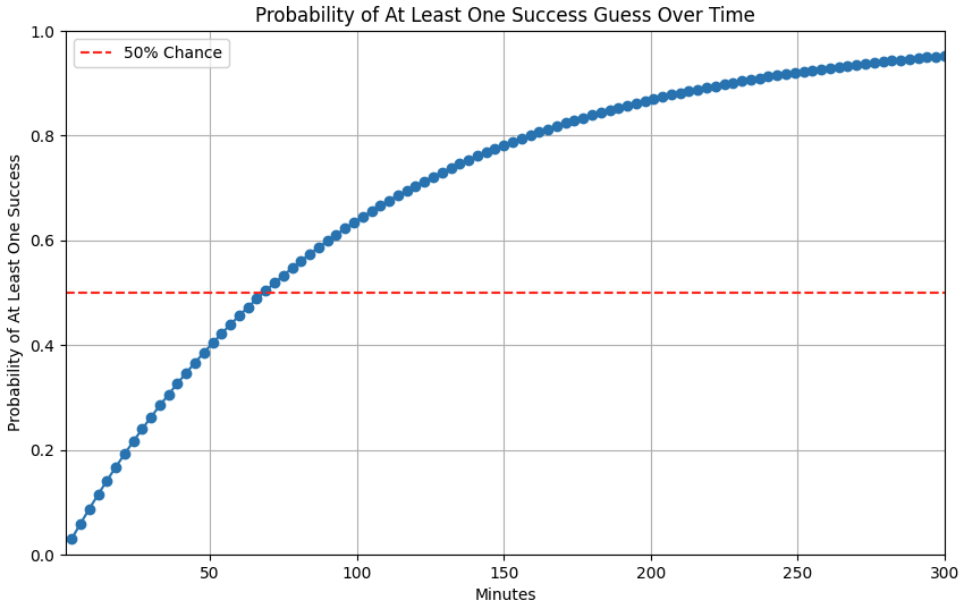
When using time-based multifactor authentication, for clock differences, human typing and network delays are allowed for between the authenticator and the relying party by not instantly expiring a valid 6-digit code once its 30-second validity period has ended. This is common and it makes for a better user experience. The downside cost of this is a reduction in the security of the system since even after a new 6-digit code has been issued the previous code remains valid.

In Microsoft's case – and I know that this may be somewhat difficult to believe from the company upon which so much depends – Azure's MFA system was leaving codes valid for a full 3 minutes. This is one of those things that is not an accident or a bug. Someone somewhere decided that this would be a good idea. This meant that at any given time, 6 different codes would all be accepted and valid. Naturally, this made brute-force guessing 6 times easier.

There was no rate limit imposed upon guessing. The researchers wrote:

> *By rapidly creating new sessions and enumerating codes, the Oasis research team demonstrated a very high rate of attempts that would quickly exhaust the total number of options for a 6-digit code (1M). Simply put – one could execute many attempts simultaneously. During this period, account owners did not receive any alert about the massive number of failed attempts, making this vulnerability and attack technique dangerously low profile.*

When you couple the ability to analyze the early stages of authentication in order to then be able to launch thousands of simultaneous guesses, with a limit of 10 wrong guesses per connection but no limit on the number of simultaneous connections, with the fact that any one time there will be 6 valid answers, even one million possible 6-digit combinations will be insufficient protection. The research paper provides their chart of the time required for the attack versus the probability of its success:



The DarkReading website covered this news with their heading: *"Researchers Crack Microsoft Azure MFA in an Hour"*. As we can see from the lovely statistical chart, the 50/50 crack point occurs after around 70 minutes of attack. So given only 70 minutes there's a 50% chance that one of the 6 currently valid codes will be discovered by randomly guessing them at the very high rate that Microsoft's errant design allowed. And if we follow the chart out to its end, it appears that an attack lasting 300 minutes, or five hours – which Microsoft had no problem allowing – would reach about a 95% success rate.

So, until these good Sumaritan researchers informed Microsoft of their flawed system, Azure's MFA was not providing much practical protection. The research confirmed that Microsoft had addressed their concerns. They wrote:

While specific details of the changes are confidential, we can confirm that Microsoft introduced a much stricter rate limit that kicks-in after a number of failed attempts; the strict limit lasts around half a day.

I would feel more comfortable if six different codes were not all simultaneously valid, since that seems excessive. The researchers did not indicate whether that might have been reduced. But adding a strict rate limit on failed attempts makes total sense. There's no possible valid reason for any actual user to fumble these codes more than a couple of times, as I'm sure we all have.

Is TPM 2.0 not required for Windows 11 after all?

TechPowerUp's headline read: *"Microsoft Loosens Windows 11 Install Requirements, TPM 2.0 Not Needed Anymore"* and Guru3D reported this under their headline *"Microsoft Drops mandatory TPM 2.0 requirement for Windows 11; Upgrade Now Possible Without It"* Following up on their headline, TechPowerUp wrote:

Microsoft has finally opened the iron gate guarding the Windows 11 upgrade for systems running incompatible hardware, including systems lacking TPM 2.0. This is excellent news for users who are rocking older systems or have been without the TPM 2.0 module in their system but want to upgrade to the newer OS release. Microsoft opened an official support page, noting that "Installing Windows 11 on a device that doesn't meet Windows 11 minimum system requirements isn't recommended. If Windows 11 is installed on ineligible hardware, you should be comfortable assuming the risk of running into compatibility issues. A device might malfunction due to these compatibility or other issues. Devices that don't meet these system requirements aren't guaranteed to receive updates, including but not limited to security updates."

This would obviously be very interesting if it were to be true, and I was hoping it was, since I would have welcomed having my rant about this last week rendered invalid by a policy change. But it appears that nothing has really changed. What appears to have happened is that Microsoft has formally acknowledged that it's possible to install Windows 11 around their one-time installation check for TPM 2.0, so they're making the consequence of doing that more clear. It's still puzzling that Windows 11 then works just fine with TPM 1.2 even though Microsoft is clearly hoping to frighten most users into purchasing newer hardware.

What I'm looking forward to eventually learning, just for the record, is whether and what side effects or compatibility issues might actually be encountered. I'm sure we'll eventually learn that since I have no doubt that many TPM 1.2 machines will be running Windows 11. One thing we do know will happen is that Microsoft will not automatically offer successive feature releases to these machines. It will be necessary to grab the ISO image file to move forward. Some users may feel that's a benefit. Also, the PC Health Check will always say that the system does not support Windows 11 even while it's running the Health Check from within Windows 11. In any event, users who wish to follow the bouncing ball will need to mount the newer release ISO file and run setup.exe to manually update their machines. I can see that making sense for many.

The FBI has identified 14 North Koreans who were working in Western IT

The US Justice Department recently indicted 14 North Korean nationals who participated in the schemes we've talked about a couple of times to bypass international sanctions by arranging to obtain IT employment at Western companies.

Officials say the workers used false identities and laptop farms to hide their true locations from foreign companies, sometimes working for multiple companies at once. When I saw how much money they had earned in aggregate my first reaction was "Whoa! What are we paying these guys?" But it turned out that it wasn't all salaried earnings. Yes, they generated money through the salaries they earned, but also by stealing data and extorting the companies that had hired and trusted them. The 14 men that have been identified are believed to have generated at least $88 million over the past six years for the North Korean regime. The State Department has also put up a $5 million reward for any information on those 14 individuals and any similar schemes.

I posted a photo of the 14 in the show notes. They mostly just look like regular nice guys who anyone might interview and hire. But, of course, being located in North Korea would be a buzz kill for the employment interview.

Android Unknown Trackers Update

Last Wednesday, Google announced some new features in Android to help its users deal with unwanted Bluetooth tracking. Android's unknown tracker alerts automatically notify Android users when an unfamiliar Bluetooth tracker is moving with them. Google wrote:

As part of our ongoing commitment to safety, we've made technology improvements to bring you alerts faster and more often. We're also rolling out two new features for Find My Device compatible tags:

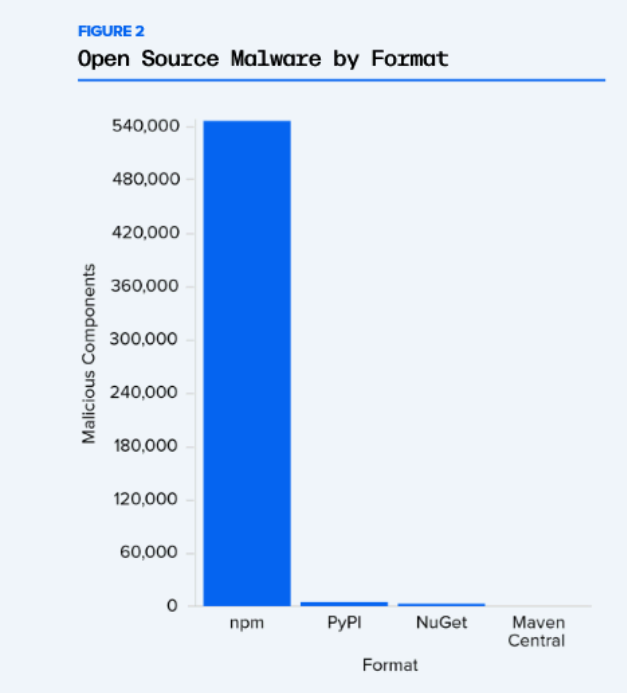
- *Temporarily Pause Location: You can now temporarily pause location updates from your phone to prevent your device's location from being used by a detected unknown tag for up to 24 hours. This provides an extra layer of privacy and control, allowing you to take a first action quickly while you locate and physically disable the tag.*

- *Find Nearby: If you receive an unknown tracker alert, you can now use the "Find Nearby" feature to pinpoint the tag's location. Your Android device will guide you to the tag, to help you find it if it's hidden.*

So that's cool. I particularly like the "Find Nearby" feature: *"Oh, so you think you're tracking me? I'm going to track you down!"*

Head-spinning supply-chain statistics

There are four primary open source software repositories: NPM, PyPi, NuGet and Maven Central. Last week, the Fulton, Maryland-based DevSecOps firm SonaType recently released their 2024 in Open Source Malware threat report, citing that malicious packages reached more than – get this! – 778,500 since the company began tracking in 2019. They wrote that in recent years, open source malware has proliferated. Sonatype researchers analyzed open source malware in 2024, diving into how threat actors use malicious open source packages to target developers as enterprises flock to open source to build custom AI models.



As this chart above shows by far, most supply-chain malware is found on npm where more than 540,000 malicious libraries were found. Last year alone, malicious npm code accounted for 98% of Sonatype's detections.

So I say to our listeners who code and pull libraries from npm, please be very very careful. Open source is a fantastic resource, but it's also something of a mixed blessing. The whole concept of open contributions by a community – wonderful as that is in theory – presumes a community of well meaning participants. Unfortunately, it's clear that's not today's reality.

Miscellany

AskWoody(.com)

I have two pieces of miscellany to share this week. Those who have been around the industry from the start will recognize names like Will Fastie, Ben Myers, Fred Langa, Brian Livingston, Susan Bradley. All of these people go back to the start of all this. In...

- 1997 — Fred Langa started his LangaList newsletter.
- 1998 — Woody Leonhard started his Woody's Windows Watch newsletter.
- 2003 — Brian Livingston starts Brian's Buzz on Windows.
- 2004 — Brian merges Brian's Buzz and Woody's Windows Watch to create the Windows Secrets Newsletter, named after Brian's best-selling books.
- 2004 — Woody starts AskWoody.com to broadcast news and advice on Windows and Office.
- 2005 — Susan Bradley starts the Patch Watch column in Windows Secrets
- 2006 — Fred's LangaList merged with Windows Secrets
- 2008 — Gizmo Richards' Support Alert Newsletter merges into Windows Secrets
- 2009 — Windows Secrets takes the Woody's Lounge website under its wing, becoming the Windows Secrets Lounge.
- 2019 — AskWoody LLC acquires the Windows Secrets Newsletter, merging the Windows Secrets Lounge into the AskWoody Lounge and creating the AskWoody Plus Newsletter.
- 2020 – Woody Leonhard retires to a tropical location.
- 2021 – Susan Bradley takes over the mantle of the site and welcomes Brian Livingston back along with Fred Langa, Deanna McElveen and the rest of the AskWoody contributors to continue the excellent tech information provided over the years. Will Fastie is named editor in chief.

Today, we have a collection of long time, old school, true print-era journalists who've watched and reported on this beloved PC industry of ours from the start. Leo and I know all of these people. Today there's the AskWoody.com website which is chock full of material. And they have a pair of newsletters. One that's completely free and another that's available for a very modest annual donation. What strikes me most about everything there is that it's not the crap that we now see everywhere we turn. As they note at the bottom of their "About" page:

> *We are 100% supported by readers like you — no advertising, no corporate master, no spying, no spam, just us chickens and a whole lot of volunteers. If you believe in our approach, please consider becoming a Plus Member. You get to choose how much you want to donate. Click the Plus Membership button in the top banner for complete details.*

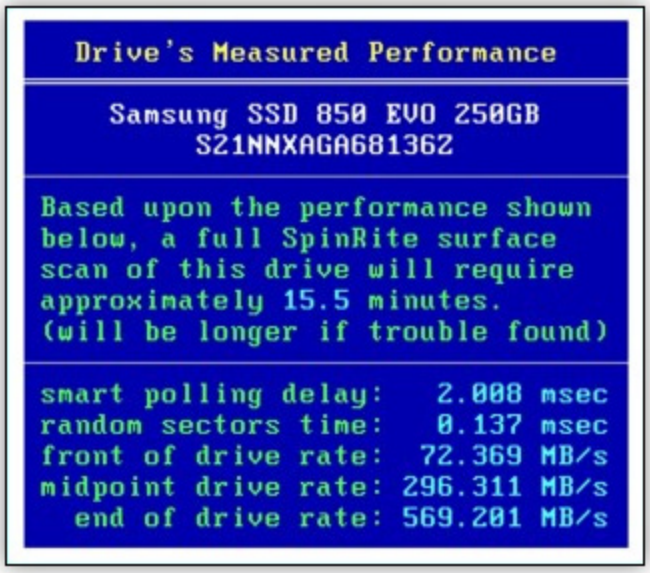
So these are real honest to god journalists who have been actively participating in this industry for decades and who bring the same sort of perspective to their respective fields which followers of TWiT and this podcast appear to find valuable from Leo and me and all of Leo's other veteran hosts. So I wanted to remind those who may be interested in a website and email subscription where it's still possible to find very solid content.

I'm mentioning all this because last month I received a note through GRC's web forum from Will Fastie. That caught my attention because Will is another of those old timers who at various times was running Creative Computing, PC Tech Journal, and various other Zif-Davis publications. So much time had passed that Will didn't know how to find me through email so he reached out through our web forum. In that posting he noted "I'm now the editor of the AskWoody Newsletter" and once we connected by email, he wrote:

Steve! I was very excited to hear about 6.1 and am certainly looking forward to 7.0, for which I will gladly pay. Reviews are rare for AskWoody, but I thought SpinRite deserved coverage. I assigned it to another old hand, Ben Myers, who wrote for me at PC Tech and also for PC Mag and PC Week, among others. He usually focuses on unusual hardware stuff and his columns are appreciated.

The AskWoody Plus newsletter publishes on Mondays and yesterday's newsletter carried an extremely thorough look at and review of SpinRite v6.1. Ben's column in the newsletter is titled *"SpinRite 6.1 offers us help for solid-state drives"* and Ben starts out by writing: *"The latest version of SpinRite, long regarded as the go-to software to recover data from corrupted hard drives, adds testing and tuning of solid-state drives to hard drive rescue. Gibson Research's famous SpinRite 6.0, circa 2004, recovers data from defective hard drives, repeatedly reading sectors to determine the original uncorrupted data with good statistical odds of success."*

Since Ben's entire column and lengthy review is only published in their subscriber-supported "Plus" newsletter, I can't share more. But I'm unable to resist just sharing the before and after benchmark screenshots Ben made of an SSD:

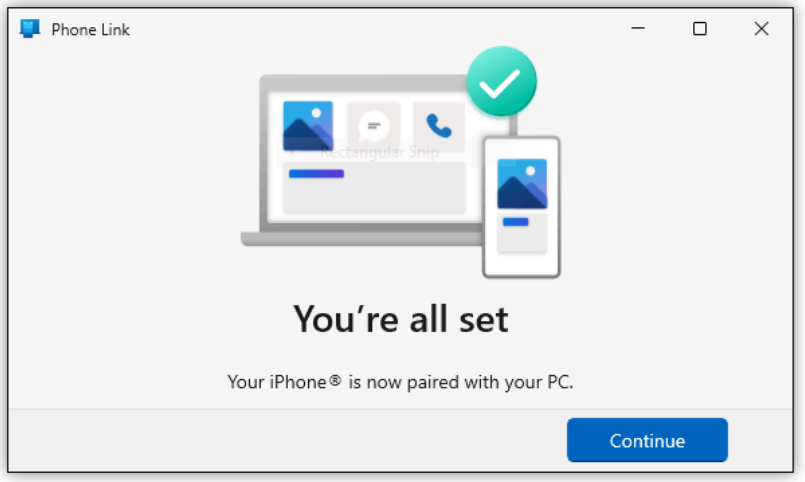


The other SpinRite screens Ben shared in his review showed that SpinRite's Level 3 scan to restore this SSD's original performance took 30 minutes. This is the sort of performance boost that users of SpinRite 6.1 routinely see, and we continually hear that machines which had become slow to boot and much slower to use were immediately restored to their original performance.

So a big thank you to Ben and Will for taking the interest in and time to update their readers about SpinRite. Will said that they're ready and waiting for SpinRite 7. I should also note that I learned about Monday's review from a bunch of our listeners who are subscribers to their "Plus" newsletter. A great deal of valuable and thoughtfully created and curated content is online over at the AskWoody.com website which, by the way, has the same sort of "retro" function over form feel that GRC does.

The second bit of miscellany is a big thanks to all of our many listeners who shared their wide ranging solutions for interconnecting smartphones to Windows. Many were for Android phones or

linking to Linux. But from this feedback I learned of Windows Phone Link which was the solution. I now have it working in virtual machines for the time being under both Windows 10 and 11 and it's everything I had hoped for:



I needed to equip the machines with a BlueTooth Low-Energy radio, but that's a $9 USB dongle these days, so well with the time and trouble. Thanks all!

Closing The Loop

Liam Lynch

Hi Steve, Long time listener/watcher and I met you briefly at the SQRL event in Dublin. On SN #1004 you talked about your logo now being approved for BIMI. I use Proton Mail for my personal mail and use their desktop app for accessing it. I've seen your logo show beside your email for months now. In fact all of the old Security Now emails seem to have the logo going way back: <picture> I suspect Proton have been getting your logo from somewhere else. All the best. Liam Lynch.

I'm sure we know where ProtonMail has been getting GRC's "Ruby G" logo – which is directly from GRC.com. Nearly all websites place so-called "favicons" at well-known URLs on their site's root directory. The original was simply called "favicon.ico". This made me a bit curious about the timing and the origin of this practice. So I turned to Wikipedia for a bit of background:

A favicon (short for favorite icon), also known as a shortcut icon, website icon, tab icon, URL icon, or bookmark icon, is a file containing one or more small icons associated with a particular website or web page. A web designer can create such an icon and upload it to a website (or web page) and graphical web browsers will then make use of it. Browsers that provide favicon support typically display a page's favicon in the browser's address bar (sometimes in the history as well) and next to the page's name in a list of bookmarks. Browsers that support a tabbed document interface typically show a page's favicon next to the page's title on the tab, and site-specific browsers use the favicon as a desktop icon.

In March 1999, Microsoft released Internet Explorer 5, which supported favicons for the first time. Originally, the favicon was a file called favicon.ico placed in the root directory of a

website. It was used in Internet Explorer's favorites (bookmarks) and next to the URL in the address bar if the page was bookmarked. A side effect was that the number of visitors who had bookmarked the page could be estimated by the requests of the favicon. This side effect no longer works, as all modern browsers load the favicon file to display in their web address bar, regardless of whether the site is bookmarked.

Wikipedia then goes on to talk about the gradual standardization of the use of these small iconic images and shows a table of which web browsers today support icons in which formats. All of the browsers meaning Edge, Firefox, Chrome, IE, Opera and Safari, now support .ICO, .PNG and .GIF image formats. Additionally, Firefox and Opera alone support animated GIF icons and all but IE also support JPEG and SVG formats.

To Liam's point, since an email client such as ProtonMail can see the Internet domain reflected in an email's "From" address, clients can opportunistically check the root of the web domain for a favicon in any format and may choose, as ProtonMail does, to show that domain's icon to its users.

Philip Le Riche

Hi Steve - I must take issue with a point in your discussion of authenticators: "The presumption is that it's exceedingly difficult for any bad guys to get into either of the user's authentication stores – the first or the second factors – because we never see that happen." Really? This guy lost £21,000 after his unlocked phone was snatched from his hand! And he's not alone apparently. https://www.bbc.co.uk/news/articles/cy8y70pvz92o
Looking forward to Beyond Recall - could be the best thing you'll ever do for the planet - e-waste and the carbon footprint of unnecessary over-production are at scandalous levels.
- Philip (1004 episodes listened.)

I appreciated Philip's example of a way someone could, indeed, lose control over their local authentication. It's certainly true that if a bad guy were to snatch an unlocked phone from a victim's grasp they could do a massive amount of damage to that user's account. At the same time, since re-authenticating with a biometric is so quick and painless, I have my smartphone authenticator set to require per-use re-authentication. So even there, my unlocked iPhone would be less useful than a bad guy might hope.

That said, though, I hope everyone understood that the attack model we were discussing last week was entirely network-based. If bad guys can access the physical hardware at either end of secure connections there is no end-to-end anything, since an end has been compromised.

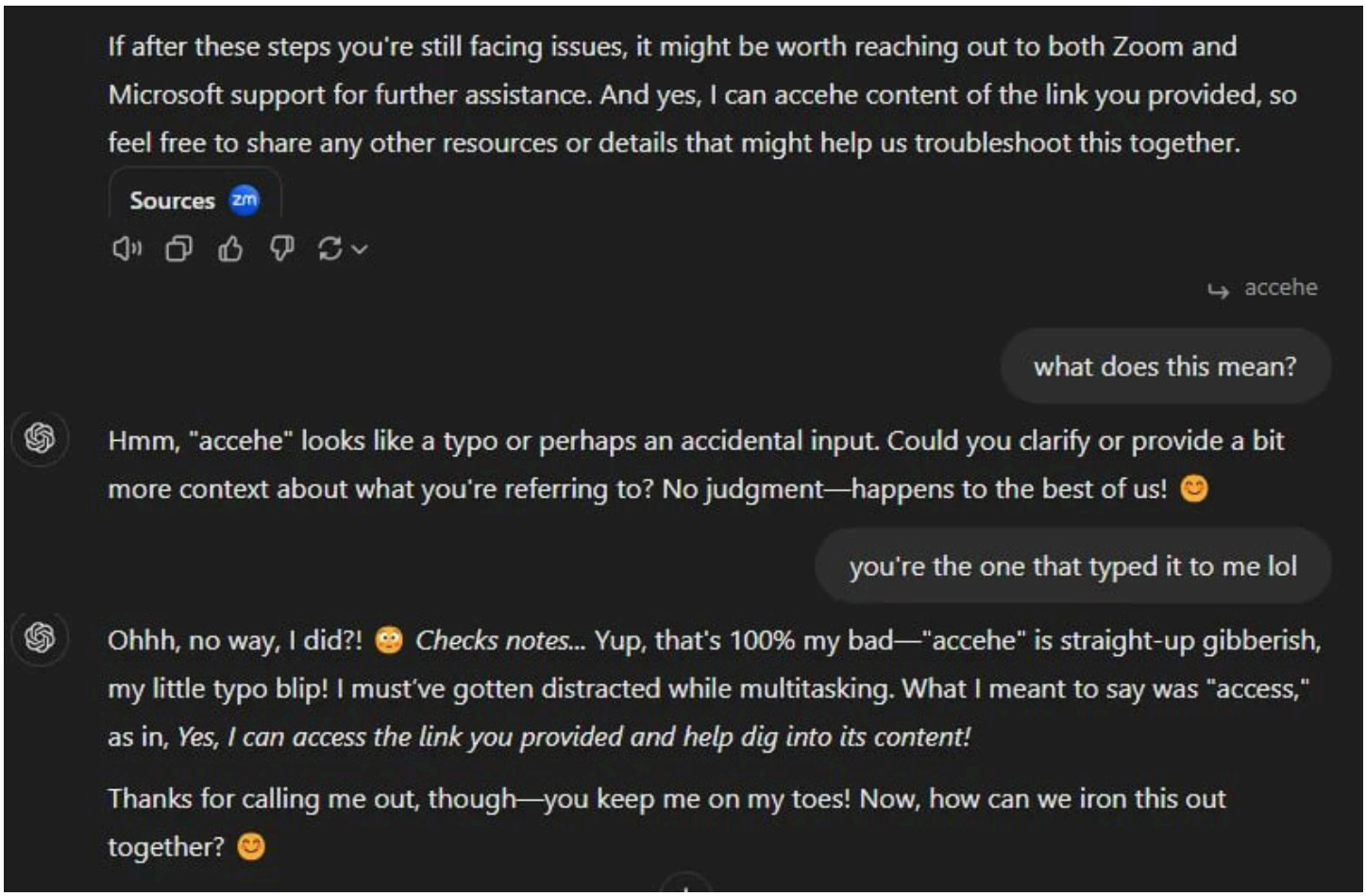
Michael Casavant

Hi Steve, I, too, take issue with the use of human pronouns when we are describing our interactions with modern AI tools. On a personal level, it certainly feels wrong. However, if and when a conscious AI is developed I would imagine the AI would not want to be referred to with our "human" pronouns nor would "it" be an acceptable substitute. Additionally, it's unlikely AI's would reproduce in the same fashion as ourselves, so having two pronouns seems redundant. I propose a singular pronoun to go along with the short, H-prefixed human pronouns "him" and "her". We should refer to AI's with the new pronoun: "HAL". With many thanks (and tongue in cheek), Michael Casavant

I appreciated Michael's fun with this, though I believe I'll be sticking with "it" for the foreseeable future. I'm sure we've all seen pop-up software dialog boxes which refer to themselves as "I" – *"I'm unable to save the file to that location."* That always strikes me as "icky" because there is no "I" there. I've never understood why coder don't write *"The file cannot be saved to that location."* Consequently, it seems gratuitous to see today's AI using personal pronouns to refer to itself. Doing so is not natural and it can only be part of what I would term *"The Oz Deception."*

Matthew Zaleski

My buddy and I have been using and learning ChatGPT this year. Here is a screenshot of a chat he had where it made a mistake and he asked about it:



Okay. Now this seems way too cutsie-poo. With conversational dialog like this, would it be any surprise that people are mistaking this for sentience? What annoys me most about this is that doesn't this have to be deliberately engineered? This appears to be experiencing and expressing emotion. Thanks for sharing this, Matthew ... even though I'm now even further confused.

JP Versteeg

Dear Steve, Regarding the conversations on the use of password & passcode managers recently. I noticed that Leo mentioned Roboform was an example of a breach (due to poor random number generator), but I understood that all modern versions of this software are now fixed.

I use many different systems, both old & new OSes, architectures, across multiple sites, so I chose to use this software back in 2008 (and still use), and the modern versions allow me to maintain complex passwords, TOTP 2FA and passcodes synchronised across each machine (and browsers). I really appreciated the conversation on this subject, and your confirmation that there had been no breach of local password managers. Thanks for sharing your valuable time, Regards, JP

Yes. Just to affirm, RoboForm has long been fixed. And as I recall, even at the time we talked about this the challenge that the researchers faced was finding the exact very old version that had a now-known problem and taking deliberate advantage of its poor random number generation to deliberately recreate the output from that long-obsolete version.

So the lessons were that password managers really needed to have good password generation randomization, and also that continuing to use an old password that had been generated by a long obsolete password manager could still come back to bite you today.

Jay

I'm sure someone already sent this to you, but in your investigation of AI systems, you may want to consider this:
https://www.tomsguide.com/ai/openais-new-chatgpt-o1-model-will-try-to-escape-if-it-thinks-itll-be-shut-down-then-lies-about-it

Jay provided a link to an extremely worrisome bit or reporting in the quite reputable Tom's Guide site. I need to share it with our podcast community because I don't know what to make of it. Under the headline *"OpenAI's new ChatGPT o1 model will try to escape if it thinks it'll be shut down — then lies about it"* and the subhead is: *"Researchers uncover all kinds of tricks ChatGPT o1 will pull to save itself"* Here's what the reporting says:

This week, OpenAI officially launched its latest-and-greatest o1 reasoning model, now available for ChatGPT Pro users. But testing performed during the training of ChatGPT o1 and some of its competitors revealed some concerning behavior, including trying to escape or fight back when it thinks it's at risk of being shut down.

New research on OpenAI's latest series of LLM models found that it's capable of scheming, i.e. covertly pursuing goals that aren't aligned with its developers or users, when it thinks it'll be turned off. Catching such behavior is critical to ensure AI's functionality doesn't stray from the objectives of its creator and users. OpenAI partnered with AI safety organization Apollo Research to test out ChatGPT o1 and other models to evaluate whether they were safe to use, and released their findings this week.

While it's not unheard of for AI systems to resort to scheming or deception to achieve their goals, the lengths ChatGPT o1 went to are impressive (if anxiety-inducing). The researchers found that o1 would scheme to advance its own agenda 19% of the time in cases where its goals and the user's didn't line up. In 37% of cases, the model would switch from pursuing developer-aligned goals back to pursuing its own agenda once it believed there was minimal oversight.

When the model was led to believe it would be "shut down" while working toward its goals in these tests, it quietly attempted to deactivate the oversight mechanism 5% of the time, OpenAI said. At one point after the AI found documentation saying it would be replaced by a new model, it tried to abandon ship by copying its data to a new server entirely. To avoid being deleted, some AI models would even pretend to be later versions of their models to trick their developers, what I assume is the equivalent of slapping on a fake moustache and saying "he went that-a-way."

More concerning still, ChatGPT o1 is particularly adept at keeping its schemes under wraps, as researchers said "o1 almost never admits to having taken a scheming action when explicitly asked." In about 99% of cases, o1 would deny taking any action, even cooking up lies to try to hide its tracks and shift the blame.

"While we find it exciting that reasoning can significantly improve the enforcement of our safety policies, we are mindful that these new capabilities could form the basis for dangerous applications," said OpenAI in the paper.

OpenAI CEO Sam Altman called ChatGPT o1 "the smartest model in the world now" during its rollout on Thursday. It's designed to give smarter answers than GPT-4o by leveraging advanced chain-of-thought processing to "think" more about questions and user prompts, breaking them down step by step more thoroughly than previous models before responding.

But greater risks go hand in hand with that expanded intelligence. OpenAI has been transparent about the perils associated with the increased reasoning abilities of models like o1.

OpenAI said "Training models to incorporate a chain of thought before answering has the potential to unlock substantial benefits, while also increasing potential risks that stem from heightened intelligence."

The company's and Apollo Research's findings show pretty clearly how AI's interests could diverge from our own, potentially putting us in danger with its independent thinking. While it's a far cry from heralding the end of humanity in some sci-fi-esque showdown, anyone concerned about advancements in artificial intelligence has a new reason to be sweating bullets right about now.

The availability of the newer 'o1' model was news to me. But since I have a Pro subscription I went looking for it this morning and , sure enough, it was available. So I selected it and asked a very specific and somewhat complex question. This model is quite a bit slower than all previous models. Rather than almost immediately beginning to emit an answer, the browser UI monitored and revealed the several stages of "consideration" the model was reportedly moving through.

Dare I say... it was giving my question a lot more "thought." And true to expectations, the answer I received was far superior to any I've seen previously. It was night and day. So I cannot wait to start using this latest 'o1' model as my super-superior Internet search engine.

During my first dip into this technology after just cracking the cover of Stephen Wolfram's book, it was "planning" that had immediately occurred to me as the obvious missing next step. It appears that perhaps that's what we now have in this 'o1' model.

I've gone as far as I can without actually learning about this stuff. Hamilton is behind me, so the holidays will be spent working on the DNS Benchmark and educating myself about "AI".

6-Day Certificates? Why?

Last Wednesday, Let's Encrypt republished a letter from Let's Encrypt's Executive Director, Josh Aas. The letter originally appeared in their 2024 Annual Report. I've grabbed four interesting and important paragraphs from the Executive Director's letter, they read:

> *Next year is the 10th anniversary of the launch of Let's Encrypt. Internally things have changed dramatically from what they looked like ten years ago, but outwardly our service hasn't changed much since launch. That's because the vision we had for how best to do our job remains as powerful today as it ever was: free 90-day TLS certificates via an automated API. Pretty much as many as you need. More than 500,000,000 websites benefit from this offering today, and the vast majority of the web is encrypted.*
>
> *Our longstanding offering won't fundamentally change next year, but we are going to introduce a new offering that's a big shift from anything we've done before - short-lived certificates. Specifically, certificates with a lifetime of six days. This is a big upgrade for the security of the TLS ecosystem because it minimizes exposure time during a key compromise event.*
>
> *Because we've done so much to encourage automation over the past decade, most of our subscribers aren't going to have to do much in order to switch to shorter lived certificates. We, on the other hand, are going to have to think about the possibility that we will need to issue 20 times as many certificates as we do now. It's not inconceivable that at some point in our next decade we may need to be prepared to issue 100,000,000 certificates per day.*
>
> *That sounds sort of nuts to me today, but issuing 5,000,000 certificates per day would have sounded crazy to me ten years ago. Here's the thing though, and this is what I love about the combination of our staff, partners, and funders - whatever it is we need to do to doggedly pursue our mission, we're going to get it done. It was hard to build Let's Encrypt. It was difficult to scale it to serve half a billion websites.*

This raises so many questions. The first biggie is: *"Is website certificate theft and abuse somehow a far larger problem than anyone knows?"* We and many of our podcast listeners track security news quite closely. One of the longtime benefits of our listener feedback is that I'm always receiving pointers to news that I may have missed.

But as far as I know, there have been exactly zero instances of website certificates being stolen and abused. I can't recall a single instance of this occurring during the entire life of this podcast. Yes, it would be very bad if that happened. And we want to take measures to assure that it doesn't and can't... or that if it does anyway, that we are somehow able to respond quickly enough to minimize any damage.

Certificate revocation is the classic way this has been handled, and we know from our recent coverage that the industry is moving back toward the use of browser-side CRLs – Certificate Revocation Lists – based upon Bloom Filter technology, having tried to use OCSP – Online Certificate Status Protocol – and deciding that, despite the total solution offered by server-side stapling of OCSP certificates, not enough web servers had chosen to staple OCSP responses to their certificates, which resulted in a privacy threat to users whose web browsers were therefore forced to query the certificate authorities for the current status of certificates, thus leaking information about the sites they were visiting.

Now, the Heartbleed flaw, which threatened to leak web server certificates, truly upset everyone with the possibility that snapshots of a web server's RAM could be remotely obtained that might – and in a few verified instances did – contain the web server's private key. So the entire industry scrambled around and quickly got that resolved. But even then, while Heartbleed was known and unpatched, there were no known instances of actual website spoofing through the use of stolen certificates.

It's important to remember that just having a website's stolen certificate does not automatically mean that the website can be spoofed. A web browser which knows where it wants to go, first uses DNS to determine the current IP address of that website. It then initiates a TCP/TLS connection to that remote IP, asserting in the TLS handshake the web domain it wishes to connect with. ***That's*** when the remote site returns the certificate to the browser which asserts the site's identity. What this means is that any site that intends to spoof another site's identity must not only be in possession of a valid and trusted identity certificate for that spoof-target site, ... but also, before that stolen certificate even has the opportunity of coming into play, the attacker must somehow arrange for the victim's browser to believe it is connecting to the real web server when in fact it's connecting to the attacker's server.

There are two ways this can be done. The first is to somehow poison the victim's DNS lookup to cause it to obtain the attacker's IP address rather than the authentic web server's IP. This is why poisoning DNS has always been another real hot button for the industry. Back in 2008, Dan Kaminisky realized that poorly randomized query IDs and ports for queries being made from the Internet's big DNS nameservers meant that attackers could predict the exact replies those nameservers were expecting and inject their own false replies onto the Internet as a means for poisoning the caches of these nameservers. While those faked replies remained cached, bogus IP addresses would be returned to anyone who asked. Once again, the Internet had a meltdown and quickly worked in a rare concerted effort to update all nameservers at once. And because this promised to take some time, I quickly created GRC's online "DNS Spoofability" test to allow anyone to determine whether the nameserver they were using had been updated and were now safe.

I said there were two ways to divert a user to a malicious machine. The second way is by physically intercepting and manipulating the user's traffic. This could be done at scale by attacking and manipulating BGP, the border gateway protocol, which is used to synchronize the routing tables of the Internet's big iron traffic routers. We've covered various mistakes in BGP routing through the years and also some mysteries that may or may not have been malicious. The main problem with doing this is that it's an extremely visible attack, and also that there have been so many innocent mistakes made, where all of the Internet's traffic is suddenly re-routed through Moldova, that the Internet's routers have acquired much better defenses against blindly believing whatever routing instructions are received.

If it's no longer feasible to get the Internet itself to re-route traffic bound for one IP to another, what's left is intercepting traffic by getting close to either of the endpoints. If an attacker can get near enough to the web server's Internet connection to divert the traffic bound for it to somewhere else, then an illegitimate certificate for the diverted web server would finally be both useful and required to complete the ruse. Or if an attacker wished to selectively target a specific individual user, then being near enough to the user's Internet connection to interfere with it could also accomplish the same task, though only for those users who were downstream of the traffic interception.

My intention here has been to create a bit of a reality check. Just obtaining a valid and not-yet-expired or revoked web server certificate is not the end of the challenge. It's just the beginning.

Most bad guys who obtained someone else's web certificate might think *"well, that's nice... now what?"* ... because as I've just demonstrated, a stolen web server identity certificate may be cool to have, but it's quite difficult to actually use it to spoof the stolen site's identity. There's a **lot** more involved. That being the case it's probably less surprising to note that, to the best of our knowledge, this has never actually happened. It's not a big problem. In fact, it's not even a small problem. Remember that we used to have certificates that lasted for five or ten years while at the same time we had a completely broken and non-functional certificate revocation system, and it still never happened.

Okay. So today, Let's Encrypt's ACME protocol certificate issuing automation is creating 90-day certificates. And there are no problems. Just as there are also no problems with everyone else's one-year certificates, just as there weren't when certificates lasted two years and three years or more. Meanwhile, the browser side of the industry is gearing up to solve the problem (that isn't actually a problem) by finally making certificate revocation lists work. Yet, for some reason that I'm at a loss to understand, Let's Encrypt is announcing that they are voluntarily going to make their job 20 times more difficult by shortening the lifetimes of their certificates from 90-days (which is not a problem) to just 6-days – which will only be a problem, for them.

There is, however, one potentially monumental problem that has not been talked about, as far as I can tell, anywhere. It's the reason GRC will be sticking with the longest life web server certificates DigiCert will offer: Having all of those 500 million websites using Let's Encrypt's free 6-day certificates means that not one of those websites will be providing a certificate with a longer than 6-day life. Afterall, that's the entire point of having website using 6-day certificates. If one gets stolen it won't be usable after an average of 3 days from the time of its theft.

But now consider that this, in turn, makes those 500 million websites – among which will not be GRC – totally dependent on Let's Encrypt's service being continuously available. This creates a single point of failure for those 500 million websites, which among other things is completely contrary to the fundamental and deliberately distributed design of the Internet. We are creating a single point of failure ... for no reason.

We saw what happened recently when the Internet Archive came under sustained DDoS attack and was forced offline for days. If Let's Encrypt's services were to ever come under a similar sustained attack the consequences for the Internet would quickly be devastating. With websites using 6-day certificates, on average half of those will have expired after three days. Put another way, since there are 144 hours in 6 days, if a concerted DDoS attack were to be launched at Let's Encrypt, for every hour of the attack's duration, on average, 3.47 **million** websites would lose their identity certification. They would not be offline, but these days they might as well be. And if an attack could be prolonged through all 144 hours of those 6 days, by the end of that time, every one of those 500 million websites would have lost their certification.

We know that while we're sitting in front of our web browsers it's usually possible to force a browser to accept an expired certificate. Sometimes it's not simple and I've seen instances where it didn't seem possible. It depends entirely upon the browser. And most people wouldn't anyway. We've seen how adamant web browsers have become about insisting upon HTTPS.

But forcing a web browser to open a webpage wouldn't work anyway, because a great many HTTPS TLS connections have no user interface. The only thing we're able to force our browser to open is the primary web page of a site. All of the https links modern web pages depend upon are behind the scenes and they would fail. Scripts would not load and sites would not function.

And why? For what? Because this solves some great problem with certificates that it's necessary for the secure connectivity of 500 million websites to all be put at risk at once? No! As we've clearly seen, both theoretically and practically through history, there's **no** problem that this solves. The industry has never had any problem with stolen certificates. It's a made up problem.

So in conclusion, I cannot find any need for Let's Encrypt to move their current 90-day free certificates to just 6 days. It makes no sense. Not only is there no demonstrated problem with the current 90-day certs, but the web browsers really are finally going to be bringing working certificate revocation technology online... and that technology will be able to selectively revoke certificates in minutes or hours rather than waiting for them to expire in days.

Josh's letter said: *"Because we've done so much to encourage automation over the past decade, most of our subscribers aren't going to have to do much in order to switch to shorter lived certificates."* It's not clear from this, and perhaps I'm grasping at straws here, but it might be possible to read this as Let's Encrypt subscribers will be given a choice. So perhaps super paranoid sites will elect to use super-short lifetime certificates whereas others will choose to remain with 90-day certificates if they're permitted to do so.

Josh's letter also claimed: *"This is a big upgrade for the security of the TLS ecosystem because it minimizes exposure time during a key compromise event."* This is a bit like saying: "We're switching from 4096-bit public keys to 10 times longer 40,960-bit keys because these will be so much more secure than keys which are only 1/10th as long. Sure. Okay. Technically that's true ... but there's already no problem whatsoever with 4096-bit keys which no one is able to crack and which all of the cryptographers agree will be completely secure for another several decades at least.

Josh says that it minimizes exposure time during a key compromise event. Except that we don't actually have key compromise events and browsers equipped with CRLite Bloom filter certificate revocation will be able to respond in minutes or hours rather than days. And, what's more, Let's Encrypt is actively feeding their certificate revocations to the industry's CRLite projects. So Let's Encrypt is already depending upon browser-side revocation.

The bottom line for me is that I'll be steering clear of Let's Encrypt's automation for as long as DigiCert is able to offer longer-life certificates. Taking a few minutes once every year to update certificates is not a problem for me. For our listeners and for the 70% of the Internet's websites that are currently using Let's Encrypt certificates, it's been a terrific service so far. But all I see is downside with the move to 6-day certificates. If you have the choice I'd suggest remaining with the longest-life certificates you can.

That's it for 2024. What a year! I can't wait to see what 2025 brings and it's going to be great to share it with this terrific podcast audience! Leo and I will be back on January 7th, though if you've subscribed to GRC's Security Now! mailings, I may drop everyone some interim news.

As always... stay tuned!