UNFORESEEN CONSEQUENCES

Description: What move has CISA just made that affects our home routers? What serious flaw was discovered in a core C library used everywhere by Linux? Does OpenSSL still have a future? What's Roskomnadzor done now? How can a password manager become proactive with Passkey adoption? Which favorite browser just added post-quantum crypto? What prevents spoofing the images taken by digital signing cameras? Why are insecure PLC devices ever attached to the Internet? And what may be an undesirable and unforeseen consequence of Google's anti-tracking changes?

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-960.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-960-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is ready. He's got some great stuff to talk about, including the new CISA recommendations for home routers. I hope they're adopted. A massive flaw that really affects every version of Linux. It's being patched or has been patched, but you should know about it. Post-quantum crypto added to our favorite browser. And then an unforeseen consequence of Google's new anti-tracking changes. That's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 960, recorded Tuesday, February 6th, 2024: Unforeseen Consequences.

It's time for Security Now!. Steve Gibson, here he is. Or was that one of the sounds that goes off when something happens, one of the alerts?

Steve Gibson: I can't wait. And speaking of sounds, all of our listeners will be glad to know that little annoying beep in the background finally died.

Leo: You couldn't find - it was a smoke detector; right? But you couldn't figure it out.

Steve: No, actually it was a water alarm which I had installed because my air conditioning condenser was backing up and overflowing, so I needed to be alerted if that was happening. But I replaced the whole AC system a couple years ago with a brand new one that has all that built in. So I removed the water sensor and just stuck it aside. And as happens here where we need to have archeological digs to find things...

Leo: Oh, it was in a pile somewhere.

Steve: Yes. It was just buried. Literally buried.

Leo: It was in the midden heap somewhere south of your living room. Okay.

Steve: And then at some point it began going "beep," like very briefly, very high-pitch, and not often. And some of our listeners began saying, Steve, you've got to check the batteries in your smoke detector.

Leo: Yeah, yeah.

Steve: Because, you know, apparently there's a problem there. Well, no. And I could not find it. And it had been going on, I don't know, a couple years maybe? And I just...

Leo: But I stopped hearing it. I haven't heard it recently. Has it still been going?

Steve: Oh, it was going last week during last week's podcast. If you play the podcast, every so often, "beep." Anyway, so...

Leo: Oh, and that's got to drive - because we have a lot of OCD listeners, I mean, people who really can't handle that kind of thing.

Steve: Yes.

Leo: That must, I mean, I'm sorry.

Steve: I stopped hearing it. I had adapted to my environment.

Leo: Right.

Steve: And so it's like, you know, I step over things that are in the way rather than...

Leo: I thought you'd fixed it.

Steve: No.

Leo: Oh, my god.

Steve: So when I came in yesterday morning I heard [screeching sound]. So I thought, oh, thank god. I knew at some point the battery would actually finally die so that I couldn't even make these beeps. It was like [screeching sound].

Leo: Did you find it?

Steve: I went right to it. Just directly, I just, like, pulled some things out of the way. There it was. And that's indeed what it was.

Leo: Did you stamp on it?

Steve: I mean, even moving it around it went [screeching sound]. It was just on its last volt.

Leo: Oh, my god. Oh. Well, anyway...

Steve: So silence.

Leo: Somebody, Chickenhead21 in our Discord wants to know, was Elaine actually typing "beep" when it went off in the transcripts?

Steve: Bless her heart, I wouldn't be surprised. She just had a little parenthetical "beep," you know.

Leo: I haven't heard it in months. I knew about it. People had written in about it. And I thought you'd fixed it last year.

Steve: Oh. No.

Leo: I just wasn't hearing it. Maybe like you I'd either grown attuned to it or I'm so "deef" now that I can't hear that frequency. Wow. Well, thank you for fixing that. That's...

Steve: Well, thank you, thank me for my patience. It is now finally gone.

Leo: I told you about that Avenue - I remember when we talked about this last, I told you about that "Avenue 5" episode. I don't know if you ever watched "Avenue 5" after we talked about it. But they're on the spaceship; right? And there was a beep, and nobody could figure out where - it was keeping people up. It was the whole thing. So this is not an unusual phenomenon. You maybe should make a variation of the Portable Dog Killer that is the Portable Beep Locator.

Steve: Believe me, when this began, I gave it some serious thought. It was impossible for me to find it. So I considered putting two microphones some distance apart.

Leo: Triangulating it.

Steve: Exactly, and locking onto that sucker. But then I thought, well, we really do want SpinRite 6.1 eventually.

Leo: Won't you be glad when you retire that you can devote your time to things like that?

Steve: Retire? What?

Leo: Never. Never.

Steve: Oh, no. I've got to move SpinRite 7 onto the Vision Pro.

Leo: Yes. We announced that during MacBreak Weekly that you would make a version for the Vision Pro.

Steve: Not a problem, yes. Imagine walking through the bits of your mass storage, looking around and saying, ooh, look at that bad spot there. Let's pluck that out.

Leo: Oh, gosh.

Steve: Yes.

Leo: So what is coming today on Security Now!?

Steve: Oh, boy. This is Security Now! 960 as we begin February. This podcast is titled "Unforeseen Consequences," which sort of crept up on me when I stumbled upon an odd reference to a piece in the Financial Times. Now, the Financial Times has one of the strongest paywalls you can find. I mean, they're not screwing around. They're like, hey, you know, we're just going to tease you with a headline. You're not going any further. Except they also allow themselves, like I just googled the headline, and there it was.

So it's like, okay, well, you're not that worried. I mean, you know, they want to bring people to their paywall so you can decide if you want it. Anyway, they had a really interesting piece that talks about some consequences we've never considered that are, like, the dark side of Google's killing third-party cookies. So it's going to be really interesting. This is going to be a riveting episode.

But first we're going to talk about what move CISA has just made that affects our home routers. What serious flaw was discovered in a core C library used everywhere by Linux? Does OpenSSL still have a future? And what's Roskomnadzor done now? How can a password manager become proactive with Passkey adoption? Which favorite browser has just added post-quantum crypto? What prevents spoofing of the images taken by digital signing cameras, if anything? And why are those insecure PLC devices, you know, the programmable logic controllers which run process automation everywhere, ever being attached to the Internet? And what may be an undesirable and unforeseen consequence of Google's anti-tracking changes?

Leo: Uh-oh.

Steve: Yeah. It's going to be a great episode. And oh, Leo, we do have a Picture of the Week.

Leo: I only see the caption. I haven't scrolled up yet. But I can tell from the caption it's going to be a good one.

Steve: Yes, it is. It may explain the power outages you've been having at TWiT Studios.

Leo: Holy cow, yeah, we were in the middle of TWiT on Sunday. Fortunately, we were not in the middle. We were actually within minutes of ending it, and everything just went dark. And I had to go home and finish the show at home because there was no power here. And then of course as you noticed I come in the studio, and everything is all messed up because they don't survive power outages very well. I had to play with a bunch of things. Anyway, we got it all working.

Steve: Now, Leo, you may scroll up.

Leo: And reveal, huh?

Steve: And reveal the cause of the power outages at TWiT Studios.

Leo: But this, the caption, "But this is where you said you wanted the dangerous high-voltage terminal box." Oh, just sitting right out there. Right out there in the public. I bet you there's a playground right next to it.

Steve: Well, and look what's on it, or aimed at it. Scroll down a little further.

Leo: Oh, I missed that part. There's a sprinkler, sprinkling it.

Steve: So for those who are listening...

Leo: Oh, I hope it's weather sealed. Holy cow.

Steve: Out in the middle of something is this scary-looking high-voltage box that says "Attention, Attention" with the lightning bolt saying, you know, high voltage. And there's a sprinkler, you know, one of those, like those things that shoots out a beam of water that's supposed to go about a thousand yards, which slowly rotates to water the entire park. Well, this box is about three feet away from it.

Leo: Right on it.

Steve: Receiving the full force of this water blast.

Leo: Oh, my god.

Steve: Right in its face. You know, it's surprising there aren't sparks flying out of this thing.

Leo: Oh, my god.

Steve: Anyway, yeah, you want to step cautiously on the wet lawn that surrounds this electrical box.

Leo: That's a great picture.

Steve: You could probably charge your Tesla just by parking on the lawn next to it.

Leo: That's hysterical. That's just great.

Steve: Yeah. Wow.

Leo: It's liquid cooled, MashedPotato says in our Discord.

Steve: He said what?

Leo: It's liquid cooled.

Steve: Liquid covered, right.

Leo: Yes, never gets hot.

Steve: Okay. So under the headline "CISA and FBI Release Secure by Design Alert Urging Manufacturers to Eliminate Defects in SOHO Routers" - and I think everyone knows SOHO, S-O-H-O, Small Office Home Office is what that abbreviation is. So last Wednesday, CISA and the FBI published guidance, this is the third such release of theirs, they've kind of - and this is the first aimed down at the consumer. Previously they were talking at the enterprise level. So they published guidance on Security Design Improvements for SOHO Device Manufacturers, which is part of their new Secure by Design Alert series, which focuses on how manufacturers should shift the burden of security, thank god, away from the customers who, you know, they just want this stuff to work - plug it in, set it and forget it - by integrating security into the product design and its development.

So this third publication in CISA's series examines how manufacturers can eliminate what they call "the path threat which actors are taking" to compromise small office and home office routers. Now, they were specifically referring to a recent initiative. There is a group out of China known as the Volt Typhoon group, which the FBI just somewhat controversially took down by patching these routers. And it was my intention initially to talk about that as our main topic this week. But I ran out of space, actually, on the podcast, and time, and I really needed to talk about the consequences of what I realized was going to be happening as a consequence of stumbling upon this Financial Times piece. So I have that queued up for next week.

But there was something that caught my attention in this which was unsuspected, or unanticipated. They said, CISA did in this joint FBI release, that they wanted manufacturers to do three things: automate update capabilities, remove web management from the WAN interface, and require a manual override to remove security settings.

Okay. So all of this podcast's listeners have probably grown tired of hearing me talk about those first two points: automate updates and remove all device management from the public-facing interface, the WAN interface; right? You just don't need to use a web interface aimed at the Internet so that you can access your device across the Internet. What we keep learning is that we don't know how to do that safely because everyone keeps making mistakes. So, and you don't have to expose it to the public because there are plenty of ways to get over onto the private LAN from the public Internet and then access the device from the LAN side. That's the way we should do it.

Anyway, the third one was really interesting. I think it's brilliant. They say: "Require a manual override to remove security settings." In other words, routers should not accept remote or any even local over-the-wire instructions which reduce their security in the absence of a manual physical local confirmation of some kind. There is no substitute for the affirmation of one's physical presence at a router's location. Pressing a "I want to change my router's configuration" button is the one thing no remote attacker in Beijing is able to do from the comfort of their cyberwarfare bunker.

I think that the best way to do this would be to require a button to be pressed in order to place the router into configuration change mode. So if a user logs into their router, they're welcome to do that. They're welcome to poke around and look at the router's various settings. But the moment the user attempts to change something which is important to the security of the system, the router's UI will pop up a little box and say: "Please press the Enable Configuration Changes button on your router to proceed." And it'll just wait. Once the button is pressed, the router will take down that little message and will allow the user to change its configuration until the user either logs out of the interface, or after some period of inactivity because most people just leave their login cookie present and logged in so they can get back to it easily if they need to.

So would this be potentially a pain in the butt? Yeah, especially if the router is in the attic. But, you know, it's a classic trade-off between security and convenience. Requiring a one-time password is certainly not as convenient as not using one, but that requirement is clearly much more secure. So the problem being addressed in this case is very real; you know? We are populating the world with insecure yet increasingly powerful consumer routers which are actually being taken over by malign remote forces that wish to exploit our traditional lack of focus on security.

So once again I give big props to CISA for leading this truly necessary change. I think this makes so much sense. Yes, again, it will be a bit of an annoyance to have to physically go to the router and press the button saying "I want to enable configuration changes." But it's a brilliant requirement, and I do hope that we see this. And really we're not doing this all the time. And if you are, don't put your router in the attic. Put it

somewhere a little more accessible, and that'll just become, you know, the way we do things in the future. I think this makes so much sense.

While we were recording last week's podcast, the Qualys Threat Research Unit, they call it the TRU, which is kind of a cool abbreviation, was informing the world that they had recently unearthed four significant vulnerabilities in the GNU C Library, which forms a cornerstone for countless applications in the Unix, I'm sorry, in the Linux, well, probably Unix, too, well, not GNU, but in the Linux environment. One of these four which they found is a severe vulnerability tracked as CVE-2023 - notice it's late last year - 6246. This vulnerability affects major distros like, well, every version of Linux, I think it's safe to say, but of course including Debian, Fedora, Red Hat, and Ubuntu.

Leo: Yeah, everything. It's C Lib; right?

Steve: Yes. It's glibc.

Leo: Yeah. Yeah, glib, yeah, yeah, yeah, yeah, yeah, yeah, yeah. That's a big one. That's everywhere.

Steve: You know, and that's the core C library that C depends upon.

Leo: It's basic standard functions, yeah.

Steve: Yes, it's linked into everything. So the bug impacts versions going back to August of 2022, which is when the bug was introduced. It is an elevation-of-privilege flaw that can allow local attackers with access to a system to obtain root privilege access. So we dodged a big bullet here, folks, because if this had allowed remote attackers to get root...

Leo: Oh, then we'd have trouble, yeah. Yeah.

Steve: Oh, baby. So here's what Qualys explained about their discoveries. They started by saying: "Before diving into the specific details of the vulnerabilities, it's crucial to understand these findings' broader impact and importance. The GNU C Library, or glibc, is an essential component of virtually every Linux-based system, serving as the core interface between applications and the Linux kernel. The recent discovery of these vulnerabilities is not just a technical concern, but a matter of widespread security implications." And actually more about the bullet that flew by and we dodged. We'll get to more of that in a second.

In other words, so it was more than a little bit shocking to Qualys to discover serious exploitable vulnerabilities in a core component of a system that is this widespread. Needless to say, Linux is everywhere, including in every one of those SOHO routers we were just talking about. And we all need to keep in mind that fixing it today doesn't automatically fix it yesterday. Which is another strong argument for allowing autonomous updating of unattended and unmanaged IoT devices.

Anyway, Qualys continues, writing: "The vulnerabilities identified in glibc's syslog and qsort functions highlight a critical aspect of software security. Even the most foundational and trusted components are not immune to flaws. The ramifications of these

vulnerabilities extend far beyond individual systems," they write, "affecting many applications and potentially millions of users worldwide. This article aims to shed light on the specific nature of these vulnerabilities, their potential impacts, and the steps taken to mitigate them.

"The first vulnerability (CVE-2023-6246), a significant security flaw has been identified in the GNU C Library's _vsyslog_internal() function, affecting syslog() and vsyslog(). This heap-based buffer overflow vulnerability was inadvertently introduced in glibc v2.37 in August of 2022 and subsequently backported to glibc v2.36" - an earlier one - "while addressing a different, less severe vulnerability." So, oops. It actually, you know, the flaw was introduced in 2.37, and then they thought they were fixing an earlier vulnerability in 2.36 and broke it, as well.

They write: "Major Linux distributions like Debian" - that would be 12 and 13 - "Ubuntu 23.04 and 23.10, and Fedora 37, 38, and 39 are confirmed to all be vulnerable. This flaw allows local privilege escalation, enabling an unprivileged user to gain full root access, as demonstrated in Fedora 38." So again, somebody standing in front of a machine where you are relying on them not having root and only being able to log in and do things as a non-root user, that reliance broke completely.

They said: "In our analysis of the same function affected by CVE-2023-6246" - this one - they said: "We identified two additional, albeit less severe, vulnerabilities. One is an off-by-one heap-based buffer overflow also in the _vsyslog_internal() function, and an integer overflow issue, also in the same function." But, you know, not nearly as worrisome as this main one. They said: "Based on our assessment, triggering these vulnerabilities appears more challenging than 6246, the primary problem. Additionally," they said, "exploiting them effectively is likely to be more complex.

"As for the last of the four vulnerabilities, a memory corruption issue was found in the GNU C Library's qsort() function, caused by a missing bounds check. This vulnerability could be triggered when qsort() is used with a nontransitive comparison function, just such as a simple comparison of a and b, which returns a minus b; and using a large number of elements controlled by an attacker, potentially leading to a memory allocation failure."

Okay. So what are the implications? Qualys writes: "The discovery of vulnerabilities in the GNU C Library's syslog and qsort functions raises major security concerns." And these are sort of hypothetical concerns, but still worth noting. They said: "The syslog vulnerability, a heap-based buffer overflow, can allow local users to gain full root access, impacting major Linux distributions. Similarly, the qsort vulnerability, stemming from a missing bounds check, can lead to memory corruption and" - get this - "has affected all glibc versions since 1992."

Leo: Yikes.

Steve: Yeah. In other words, all glibc versions, effectively.

Leo: Yeah. Linux is only - yeah, definitely, that's all of them. Yeah.

Steve: Right. They said: "These flaws highlight the critical need for strict security measures in software development, especially for core libraries widely used across many systems and applications." So, yeah. No kidding. Now, what happens, or the way this is

managed behind the scenes, is always interesting. So here's a quick blow-by-blow timeline from the discovery through the coordinated release one week ago today.

So this began in early November, November 7th of last year, 2023. So the end of last year, November 7th, they said: "We sent a preliminary draft of our advisory" - that is, you know, a disclosure of their discovery - "to Red Hat Product Security." Eight days later, on the 15th, Red Hat Product Security acknowledged receipt of their email. The following day, on the 16th of November, "Red Hat Product Security asked us if we could share our exploit with them." The following day, on the 17th, they sent the exploit to Red Hat Product Security. Four days later, on the 21st, Red Hat Product Security, they said, "confirmed that our exploit worked, and assigned CVE-2023-6246 to this heap-based buffer overflow in vsyslog_internal."

Okay. So that is November 21st. Now we go to December. We're on December, the next month, on the 5th: "Red Hat Product Security sent us a patch for this vulnerability 6246, written by the glibc developers, and asked us for our feedback." Two days later, December 7th, they said: "While reviewing this patch, we discovered two more minor vulnerabilities in the same function." That's where that off-by-one buffer overflow and the other integer overflow surfaced. They said: "We immediately sent an analysis, proof of concept, and patch proposal back to Red Hat Product Security and suggested that we directly involve the glibc security team." That was on December 7th.

"The next day, on the 8th, Red Hat Product Security acknowledged receipt of our email and agreed that we should directly involve the glibc security team. We contacted them on the same day, and they immediately replied with very constructive comments." And of course they were already looped into this because Red Hat had previously forwarded this to them and then received the patch back from them which then they sent back to Qualys.

"Three days later, December 11th, the glibc security team suggested that we postpone the coordinated disclosure of all three vulnerabilities until January 2024." Okay. So we were at December 11th at this point. They said: "Because of the upcoming holiday season," meaning people on vacation, people not around, people less available to respond immediately, as this would require, to the public coordinated disclosure of this. So they said, yeah, good, let's let the holidays pass, and we'll deal with this immediately afterwards.

So December 13th, still last year before Christmas, Red Hat product security assigned the two additional CVEs to the other two things that had been found. On January 4th this year, they said: "We suggested either January 23rd or January 30th for the Coordinated Release. Glibc developers agreed on January 30th." That was last Tuesday. So now we're at January 12th. "The glibc developers sent us an updated version of the patches for these vulnerabilities. The next day we reviewed these patches and sent our feedback to the glibc developers." Two days later, on the 15th, the glibc developers sent us the final version of the patches for these vulnerabilities.

The following day, Qualys says: "We sent these patches and a draft of our advisory to the linux-distros@openwall list. They immediately acknowledged receipt of our email and, on the 30th, last Tuesday, coordinated release of this occurred."

So, you know, that's how this actually, you know, like there's an example of everybody being responsible, everybody responding to email, no one sitting on this for months the way we've seen Microsoft do so often. You know, this is the way it's supposed to happen. Problem is found, the right people are looped in, it's reviewed, it's verified, patches are created, patches are verified, some more tweaks are made, everybody agrees about, like, looks at the calendar, when would be a good time to let everybody know. And that's

the way it happens. So a great look at how this happened. And all the distros have been updated now.

Leo: Nice.

Steve: Everybody who's in a situation where it might be a problem if a Linux system from the last two years is relying upon its protected root privilege, well, it's not as protected as we were hoping. So you'll want to...

Leo: But at least somebody - an attacker needs to be physically on your system. So that's a relief.

Steve: Yes, thank goodness.

Leo: Yeah. By the way, I bet you you could look at a quick, any quick sort and immediately know if there's a buffer overflow. This is not a hard thing to write. Everybody wrote it in Comp Sci 101. I can see how you'd get buffer overflow, but that seems, like, pretty boneheaded.

Steve: Well, so you're able to pass a function to qsort to use.

Leo: Oh. Oh, yeah, yeah, because that's the function that determines what's less or greater; right.

Steve: Yes, exactly. So it's the sorting determiner function which is where the problem actually is.

Leo: That might be a little bit harder to trace, I guess, yeah. I mean, usually you just pass it less than or greater than. But okay. If you did something really elaborate, maybe you'd get something weird, yeah. Interesting.

Steve: Okay. So speaking of libraries, OpenSSL has lost another big user. The CDN Fastly, you know, one of the biggies, announced that they've decided to switch from OpenSSL, which they've been using to date, to the name you've just got to love because this is what you want from your SSL, BoringSSL. You know, you want a BoringSSL library. In their announcement they explained. They said: "OpenSSL has a long history of high-severity vulnerabilities, including the notorious Heartbleed bug. In addition to the risk of exploitation, there is a significant operational cost incurred to rapidly test and deploy patches."

And, you know, we're talking about - so I don't think they say this anywhere, but this is on all of their edge system instances. So all of their edge routing, edge proxies where the CDN's network is interacting with the Internet, this is where this goes. So, yeah, if some high-severity vulnerability is found in OpenSSL, like every one of those instances needs to be fixed immediately. And that's a big pain in the butt.

So they said: "There's a significant operational cost incurred to rapidly test and deploy patches whenever a new vulnerability is announced. Our primary goal in replacing OpenSSL with BoringSSL was to reduce the frequency and impact of CVEs and improve the security of our TLS termination system for our customers. BoringSSL is a fork of OpenSSL that was created and maintained by Google. It is widely considered to be fundamentally more secure than OpenSSL because it is less complex. OpenSSL remains the Swiss Army Knife of SSL libraries, and a bunch of great work has been done over the years to improve it. But we are convinced that BoringSSL provides better protection for our customers."

They added: "Our work began about a year ago with the ambitious idea of replacing OpenSSL on our edge for all incoming connections. We considered a few alternatives, but stuck with our original vision of migrating to BoringSSL to gain the following benefits: smaller, more modern code base; a safer API - BoringSSL is an OpenSSL derivative and is mostly source-compatible, making our migration less challenging; extensive fuzzing; used by big players and maintained by Google; and similar performance to OpenSSL."

They said: "In summary, the consensus was that BoringSSL offers a more focused code base, one without OpenSSL's myriad of legacy code, which makes it intrinsically more secure." And I didn't have it here just because it would take up a lot of space, but they showed the breakdown of code between OpenSSL and BoringSSL. The BoringSSL source code base is less than half the size of OpenSSL. So it just makes sense as a technology is maturing that it's also going to be getting a bit old and creaky along the way. In the case of OpenSSL, it spans decades, having started in 1998. So that makes it 26 years old. And as we know, SSL has evolved itself as a protocol dramatically during those 26 years. So Google created BoringSSL. And we know, for example, that Amazon's AWS service is running on their own very small homegrown TLS stack.

I'm sure that OpenSSL will remain the bedrock that it always has been for experimentation and testing. That's always where new protocol stuff is worked out. And for being, as Fastly said, the Swiss Army Knife of SSL libraries. But its deployment in critical new applications has probably seen its day. And as I was reading this and thinking about it, we've been using GitLab to, like, manage all of the issues during the ending phase of SpinRite's development. We were just using, you know, newsgroup threads initially. But one of our participants, well known to all of the people in our newsgroups, Colby, he was suggesting GitLab. And I looked at it, and I thought, okay, let's, you know, I'll give it a try.

So I brought it up on its own server. And it's very nice. The problem is, it has way more features than we are using, just as OpenSSL has way more features than Fastly is using; and they won't leave it alone; and it's so big and complex it's constantly having bugs and problems that are critical. So the analogy is perfect.

And as a consequence, I am seriously considering moving to a much more modest, better fit for us, like issue tracking system. There's something called Redmine which looks like it is exactly what I want, mostly because they haven't touched it in a long time. And I don't want to spend all my time maintaining a tool which is supposed to be helping us to manage a project. I just want it to manage the project and not require its own maintenance staff. So I can fully understand the tradeoff that Fastly is looking to make and has made.

Leo: Back to Steve Gibson, who is going to show us how to write a proper quick sort. No, he's not. That's not what he's going to do.

Steve: Well, no.

Leo: Although I would take that class, Steve. I would. Sanitize your inputs.

Steve: So recall that last December 1st Russia put a new communications law into effect which required all hosting providers of Russian websites to register with none other than Roskomnadzor. This law requires all cloud and web hosting providers to register with the Roskomnadzor agency, which is of course Russia's telecommunications watchdog. So far, 266 web hosting providers have registered with Roskomnadzor, and all are local companies. Not a single external provider has registered. And those providers are responsible - those providers, the external providers, I'm sorry - the external providers are responsible for about one third of all Russian websites.

Now, I don't know what's up, but this does seem a little suspicious that not a single external provider has registered. So it makes me wonder whether this is actually, like, you know, a backhanded Russian way of forcing the remaining one third of Russian sites which are currently being hosted by external providers, none of which suspiciously have registered, and all of which - and here's the point - are subject to being cut off at some point in the future, if this isn't some way of forcing all the Russian sites into Mother Russia's hosted services rather than continuing to use, you know, those non-Russian territorial providers. We'll see how this goes. But Roskomnadzor has made it clear that at some point non-registered providers will be cut off from access to Russian territory. So again, don't know what that means, but we'll see.

Also last Tuesday, Google's Security Blog announced a very nice-sounding new feature for Android's Password Manager. The blog's title is "Effortlessly upgrade to Passkeys on Pixel phones with Google's Password Manager." Okay. So it turns out this is less Google-specific than they're making it sound. I'll explain that in a second.

Here's what Google said. They said: "Google is working to accelerate Passkey adoption." That's good for everybody. They said: "We've launched support for Passkeys on Google platforms such as Android and Chrome, and recently we announced that we're making Passkeys a default option across personal Google Accounts. We're also working with our partners across the industry to make Passkeys available on more websites and apps." Which as we know is what's required for this to make any sense at all.

"Recently," they said, "we took things a step further. As part of last December's Pixel Feature Drop, we introduced a new feature to Google Password Manager: Passkey upgrades. With this new feature, Google Password Manager will let you discover which of your accounts support Passkeys, and help you upgrade with just a few taps. This new Passkey upgrade experience is now available on Pixel phones starting with the Pixel 5a, as well as Pixel Tablet. Google Password Manager will incorporate these updates for other platforms in the future.

"Best of all," they wrote, "today we're happy to announce that we've teamed up with Adobe, Best Buy, DocuSign, eBay, Kayak, Money Forward, Nintendo, PayPal, Uber, Yahoo! Japan, and soon TikTok, to help bring you this easy Passkey upgrade experience and usher you into the passwordless future." They said: "If you have an account with one of these early launch partners, Google Password Manager on Pixel will helpfully guide you to the exact location on the partner's website or app where you can upgrade to a Passkey. There's no need to manually hunt for the option in account settings.

"And because the technology that makes this possible is open" - in other words, yes, it's actually not Google's - "any website or app, as well as any other password manager, can leverage it to help their users upgrade to Passkeys for supported accounts. It's all part of Google's commitment," they said, "to help make signing in easier and safer."

Okay. So they're saying that at launch this initially works with Adobe, Best Buy, and so forth. But why them and not everyone? It's just that this group is first to adopt a new standard. We've all seen how our password managers are able to perform a security checkup; right? Like to notify us when we may have reused a password somewhere, where we're using the same password for two different accounts. So this is our password managers being proactive about our security.

Well, it turns out that there's an open standard means by which any website that supports Passkeys is able to advertise the fact that it supports Passkeys in a way that any password manager is able to check for and similarly advise. I did a bit of digging, and I found the page where Google describes this. It's titled "Promote Passkey upgrades in Google Password Manager." Of course, this actually applies to any password manager that does this. There's nothing Google Password Manager-specific about this.

Anyway, they wrote there - now, this is aimed at web app and website developers. So that's the portion of the site where this was found. So talking to website developers, they said: "Integrating Passkeys into your app or website is just the beginning of your Passkey journey. After your initial deployment, one of the challenges you will likely encounter is making sure your users understand what Passkeys are and how to create them.

"You should suggest creating a Passkey immediately after the user signs in using their password and verifying with a second factor. Remembering passwords and entering one-time passwords while switching between different apps and tools can be frustrating for users. Recommending the creation of a Passkey at this moment is an opportune time, as users are likely feeling this frustration. In addition to the self-managed promotions, Google Password Manager can now suggest creating a new Passkey on behalf of your website or app."

Okay. So under the user's experience they say: "On Pixel devices, Google Password Manager discovers that your website or app supports Passkeys, suggests users to create a new Passkey, and directs them to your Passkey creation page." Okay. So leaving Google out of this, what this is about is a very welcome standardized and uniform way for any Passkey-supporting site to declare its support in a machine-readable way. So this is, as I said, more broadly than just Google, this means that any password manager on any platform - are you listening, Bitwarden? - could examine the entire inventory of its user's saved passwords and use this standardized protocol to proactively check the web domain of each password for its support of Passkeys. And if an available Passkey had not yet been configured on that account, the password manager could take the user directly to that site's Passkey setup page.

The standard used we've talked about before. It's the /.well-known/ web directory which is located at the root of a domain. And there's a "Passkey-endpoints" JSON-formatted file there under that /.well-known/ directory that contains two URLs, one to enroll a new Passkey and another to manage existing Passkeys. So again, any Passkey-supporting site should take every opportunity to enroll its users the next time they're logging into the site, and that the site sees that they're using a Passkey-supporting client. That's the primary way we can expect Passkeys to become adopted. But it will also be cool for them to be able to come at this from the direction of the Passkey-enabled password manager to have them reveal the sites to which we could enroll and switch over to Passkey logon and authentication.

Leo: I agree. I agree.

Steve: So very cool.

Leo: Now that Bitwarden supports Passkeys, I find myself much more likely to use it because it's cross-platform. Because I work on all platforms. So, yeah, Apple, I have my Passkeys for some things in my iPhone. But if it's not everywhere, it's not useful. So I really like it that Bitwarden supports it. And I've used it a number of times now to log into Google and stuff. And it's like, wow, that was easy. Really it's good.

Steve: Yeah.

Leo: I wish we'd done SQRL; but, hey, next best thing.

Steve: We got, well, if wishes were fishes or something.

Leo: Right.

Steve: So, okay. And just a quick note that Mozilla has added support - Mozilla - for post-quantum cryptography to its developer Firefox Nightly builds. So we'll all be seeing it once the release build is published on the main channel. It can be enabled, as soon as it's available, by going to about:config and then looking for security.tls.enable_kyber, K-Y-B-E-R. And the good news is that Firefox's search in that about:config, I mean, remember how long that about:config is. I mean, it's ridiculous. The scroll bar just disappears on the screen. There are so many things that you can tune and tweak. So you're able to do a substring search. So you can just put in "kyber," and it would immediately bring you to that entry.

Leo: Nice, nice.

Steve: So anyway, just, you know, a nice forward move for Firefox. And I've got some feedback to share before we get to the main goodie here.

Jeff Zellen, he said: "Steve, I've been a listener to Security Now! for quite some time and have really enjoyed and gotten a lot out of your [what he calls the] 'correspondence school'" that we conduct here every week. He said: "I wanted to let you know there is a way to get your TOTP tokens out of LastPass. It's a little Python script that rebuilds the QR codes for you. It also allows you to print them off, in case you didn't know about the 'Steve Gibson offline backup and storage technique.'" Which of course is printing all - I have printed out every QR code for every one of my one-time passwords and stapled them together in a sheaf, and they're in a drawer. And it's come in handy a couple times.

Leo: Sure, no, it's good to have that, yeah.

Steve: When I've needed to bring up a new device.

Leo: Right.

Steve: So anyway, Jeff wrote later to say "I didn't write this. I didn't mean to say that I wrote this." Anyway, I've got a link to it. It's on GitHub. If you search for lastpass-

authenticator-export, you'll find it. I checked it out, and it looks nifty. It allows you to regenerate your original QR codes, which you may have fed to LastPass. And if so, display them, capture them, buy a device that may be starved for them, or print them out. So anyway, just a cool note. I wanted to make sure that our listeners knew that was available. Thank you, Jeff.

Brenty said: "Re oddly inflated app data," he said, "if you look in iPad or iOS Settings > General > iPhone/iPad Storage, wait for the list to load, and then select an app, you'll see that the size of the app itself is listed separately from its 'documents and data.'" He said - and this is referring to a question that came up last week. He said: "When trying to free up some storage space previously, I found a few apps whose documents and data appeared to be way more than seemed reasonable."

Remember it was that some credit, oh, Credit Karma was occupying a gig of space in some guy's phone. And he's like, uh, what? So anyway, Brenty says that he deleted the app, reinstalled it, and it was now at one tenth of the size it had been previously. And, you know, he said: "So my theory is that some, maybe many, maybe most, have logging, caching, and likely other unnecessary, stale data that builds up over time, which they simply don't bother to clean up on their own." So yes. Deleting and reinstalling would likely save you a lot of space. And of course I have always found the same is true with setting up a new version of Windows. He's like, oh, let's just start over again.

Someone whose handle is Mental Calm Today, he said: "Greetings, Steve. Long-time SN 'student,' TWiT Club member, SpinRite user."

Leo: Yay.

Steve: He said: "So excited that you have 6.1 ready for prime time. I'm reaching out to say thanks for your mention of LearnDMARC yesterday." So he was tweeting on Wednesday. He said: "It's really helpful re a confusing protocol."

Leo: Yeah.

Steve: So this serves as a reminder to me to mention that LearnDMARC website that we mentioned, and that's L-E-A-R-N-D-M-A-R-C dot com, we mentioned and took a look at last week. It was a huge hit among our listeners, from all the feedback that I've seen. One person said that the site was offline and suggested maybe that it was because we mentioned it. Well, that would be flattering except that the nature of a podcast is that the listening is well distributed in time. So it's not like a purely live event, where we bring websites down by talking about it. And I guess, what, we used to do that back in the TechTV days, didn't we, Leo.

Leo: Yeah, oh, yeah. They called it "slashdotting" a site because Slashdot used to do it.

Steve: Right, right, right, right.

Leo: Yeah, yeah. It's been a while since we've done that. Internet's gotten more robust, I think.

Steve: Well, and frankly having downloads distributed is a good thing because it's better for everybody. Ron tweeted: "Hi, Steve. This is in regards to Sync. I messaged them after your item on Security Now!, and this is what I received," he said. And then he quoted me what Sync responded, saying: "Hi there, Ronald. Bailey from Sync here. Thanks for reaching out. There was a bug identified within the Sync Mobile App, regarding the iOS Files app integration, which prevented folks from navigating within the Sync folders (Files and Vault) via the Files app. Users were still able to navigate within the Sync Mobile App. This Files app integration bug has now been resolved." There's a link to it. "Let us know if you have any further suggestions. Thanks again," writes Bailey from Sync.

So anyway, just a follow-up to that previous listener who was feeling a little despondent because the reply he got from Sync suggested that, well, yeah, so don't do that. We'll get around to it someday. You know, that put us all off of Sync a little bit. It's like, what? But apparently that was a red herring. Sync did get on it quickly, and fixed it, and it's back up and running. So thank you, everybody.

Johnathan Rouse said: "Hello, Mr. Gibson! Firstly, you have been a role model for me all throughout high school, college, and now as I redirect my career into education."

Leo: Nice.

Steve: "Thank you for the hours of laughs and education, as well as Leo and the rest of the TWIT Team. I figured you might want to see the response Windows Defender gave" - and then he cites the version of Windows Defender - "when downloading the 6.1 Pre-Release. After manually allowing the program, it went along perfectly in creating a USB Boot Drive, but regardless I wanted to show you what I encountered. I'm hoping the new and improved ISO created will work with Ventoy Bootable Drives as well, and I can't wait to try it out. Thanks again for all the years of dedication, and I hope to be half the teacher you seem to be in your sleep."

Leo: You're not sleeping, I want to point out,

Steve: So first of all, Johnathan, I can only say, and I know that you, Leo, feel similarly, that I am so pleased that this podcast and TWiT have been so useful to you.

Leo: You bet, you bet, yeah.

Steve: The good news is that since you're just starting out, you have a lifetime of teaching ahead of you. So I do wish you all the best as you launch into your career. As for Windows Defender's reactions to SpinRite, yes, it continues to be an annoyance. But I noted that he sent his Tweet last Tuesday, and things may have become better since then. Most recent experimentation suggests that Windows Defender is happier. And as for Ventoy, you will likely have discovered that SpinRite 6.1 and Ventoy are not getting along currently. But that will be resolved shortly. I'll have more to say about Ventoy in a minute when I update everybody about SpinRite.

Leo: Yeah. Huge fan of Ventoy. I really like that. I use it all the time.

Steve: So thank you.

Leo: Good, good, good. Very nice.

Steve: Yeah. Anotherthomas is his handle. He said: "@SGgrc: About crypto signing camera." He said: "It can work if the private key is in a removable HSM assigned to the photographer. She/he will then able to prove that she/he is the author."

Now, okay, that is some nice thinking outside the box, or in this case outside the camera. If this were done, it would make the private key about the owner of the key, not about the camera.

Leo: Right.

Steve: And the key is presumably more easily protected by them than having the key locked inside the camera. You know, you still have to protect the key, but owners would have the incentive to do that since their photographic reputation is on the line. So anyway, I haven't heard anyone talk about that. I think that's a very neat idea.

Leo: It's not the problem that they're trying to solve, though. They're trying to solve the problem of authenticity of the photo.

Steve: Correct.

Leo: And I just - I have been playing with this content, I think you call it, what is it? There's a name for it.

Steve: Right. The content protection stuff.

Leo: Yeah. And I have it turned on on my camera right now. And it associates the serial number, I guess, with the name. I don't, you know - now, you can remove it. You absolutely can remove it because you can remove any Exif information in a photo by just JPEGing it and, you know, saying don't save the - there's lots of ways to strip off Exif. But I guess the point is that this is going to be used by news organizations where they aren't going to remove it, and they can provably say this is created by this camera at this time. And that can't be modified.

Steve: Right.

Leo: You know, I think that's the idea, is that this photo is not a fake, you know, and here's the chain of custody. It even shows in this information, you know, how I edited it and so forth, you know, what program was used to edit. I think it shows that somewhere. Maybe not on this one. But it does have that.

Steve: I know that, if you're using Adobe's tools, which are the only ones that are authorized to do this, then it does absolutely create basically a chain of custody through the editing.

Leo: Exactly, yeah.

Steve: Yes. And you made a really good point because it's not trying to authenticate the reputation of the - it's not trying to authenticate the reputation of the person who took the picture. The reputation is assumed, like an accredited, well-known news agency. Which brings us to the next question that DellAnderson asked. He said: "Grateful you're going past 999. Can't help but ask a basic question about digital camera authentication. What would prevent a very low-tech workaround where the digital camera - Nikon, Leica, et cetera - takes a perfectly authenticated photograph of a digitally manipulated image?"

Leo: Ah. Excellent point. An analog loophole, yup.

Steve: Yup. "How would this fancy Nikon camera know it was photographing a high-resolution 2D image rather than reality?"

Leo: It wouldn't. Very good point. Yup.

Steve: And so I replied to Dell that I had the same thought, as I imagine many of us have. The problem is that the "authentication," and I have that in quotes, does not and cannot extend out to the actual landscape or subject that's being photographed. This signing technology is intended to prevent the manipulation of an image's digital recording after it's been captured optically. But doesn't this beg the question, what's to prevent someone from presenting a fake scene to the camera to capture and then sign. Okay, now, I understand that this is a different problem. This is not the problem this camera was designed to prevent. This camera was designed to prevent undetected post-image-capture manipulation. And what it was designed to prevent is a significant problem. You know?

So anyway, I think that what we have to keep in mind is the threat model and what it is we're trying to say. We are unable to say, Leo, as you instantly got, we're unable to say that the scene that the camera took a picture of was authentic. What we are able to say is to the best of our ability after the camera took the picture we know exactly what was done to it in a verifiable fashion. So again, you know, and what's cool about this is we talk about threat models and what we can and cannot assert in the realm of security. So here's a perfect example of what we can and cannot assert and what we can and cannot protect.

Leo: Which, by the way, I want to thank you, gave me an excuse to buy a new camera. So I appreciate that, Steve.

Steve: Well, Leo, for that research you had to have that.

Leo: I had to. I had to do it.

Steve: Absolutely.

Leo: Exactly.

Steve: Yeah. And if the IRS, you know, ever audits you and says...

Leo: I'll give them this.

Steve: Exactly. You know, absolutely important that you were able to demonstrate that. Slartibartphast...

Leo: I love the name. You know where that's from.

Steve: We know where it came from; right.

Leo: Yeah, yeah, yeah.

Steve: "I wonder if Google needs native iOS engine to make the new ad auction stuff work." And the answer is, absolutely and without question. The entire Privacy Sandbox API is a collection of new web browser features, intrinsic to the web browser, that requires a bunch of data storage locally. I'm sure this is why they have been working on a native implementation for iOS, even though it isn't clear to the outside world how they might get it into iOS. There is, you know, there's so much that we don't know yet about how we're going to get to where we are today.

Google wants to move the entire world. And "moving the world" is no exaggeration. Given that advertising supports the Internet, the required size of this change would be difficult to understate. Like everything needs to change. Google already has control of nearly all desktops and Android, which are the majority of smartphones. So I guess my questions are, what are Mozilla and Apple thinking about this? What conversations may be going on among them? Because this is big stuff, and actually this is what we're going to be talking about here as we end today's podcast.

Aeon tweeted - and I know what his first name is. It's not actually Aeon. He said: "Steven, I'm personally inviting you to the Gathering of the Stephvens."

Leo: Note how it's written. I love it.

Steve: Yes.

Leo: It's "phv."

Steve: Yes. He said: "Next year, in 2025, we're going to set a Guinness World Record for the most people named Steven in one area. First goal, gather the Stephvens in this

Discord." And he provided a link. "Next goal, conquer the world." And he said, "You down?"

Leo: You down.

Steve: So I thanked Aeon, whose first name is presumably Stephen, for thinking of me. But I explained that I was pretty sure that traveling to a massive meeting of people with whom I phonetically share a first name, for the sake of contributing with my presence to the setting of a Guinness Book record is not something that, when the time was approaching, I would be glad I was taking the time to do. But I told him that I looked forward to hearing more about how it goes, even in absentia. So, you know, thank you, Stephen.

Leo: We're having fun creating the regular expression for Stephen with a PH or a V in the Discord. I think we've got it, actually.

Steve: Yeah, curly braces and then...

Leo: Yes, exactly.

Steve: ...a couple brackets and...

Leo: And an OR.

Steve: Yup. Yup. Okay. So we've all seen video segments of complex manufacturing facilities where thousands, if not hundreds of thousands, of cans or something, bottles or boxes or whatever, are moving through a complex system that's sorting and spinning and stamping and printing or counting or whatever it's doing. You know, like these crazy-looking manufacturing facilities. Treadmills and gates opening and closing, routing stuff.

Leo: I love that stuff. It's one of the things I love on TikTok is there are a bunch of TikTok videos of how stuff's made. And it's always fascinating.

Steve: Very cool.

Leo: Always fascinating, yeah.

Steve: So just as some of those pre-electronics early computers used banks of mechanical relays, back before the advent of computers, process control engineers, as they're called, would design insanely complex control systems built up from individual mechanical relays. We would call such a system "discrete" as opposed to "integrated." Then, blessedly, integrated electronic solutions became cost effective, and these large process control solutions were replaced by PLC systems, Programmable Logic Controllers.

These PLCs were not very smart because they didn't need to be. Basically they were replacing a bunch of relays. They were essentially, "If A, then B. Wait until C, then do D. And once E, go back to the start." But being solid-state they were at least more reliable. Now, remember that we have the term of a hardware or software "bug" because back in 1947 a dead moth, you know, a bug, was found to be the underlying cause of Harvard's Mark II relay computer not working correctly. Anyway, you know, relays are not as reliable as solid-state because they can actually have bugs.

Anyway, we've talked about these PLCs on this podcast multiple times because attaching them to the Internet has turned out to be a generally really bad idea. They were never designed for that, and it hasn't been turning out well. I'm bringing all this up today because I received a long, insightful, and interesting Direct Message from a listener whose thoughts about the problems with PLCs are worth sharing.

Here's what Dylan wrote. He said: "Good day. I'm an engineer and occasionally work with Programmable Logic Controllers. And I have some thoughts on why these sadly make the news in a bad way sometimes. I believe most of the problems boil down to two root causes. Number one, increased demand for 'real-time' data. Just like the CANbus protocol in the automotive industry, PLCs were invented and took hold in manufacturing when security was not a concern. As time went on, protocols were developed to have PLCs talk to each other and to advanced peripherals like motor controllers, touchscreens, printers, or even SCADA, Supervisory Control and Data Acquisition computers.

"I believe the demand for telemetry and data aggregation is the real reason most PLCs get exposed, not because remote WAN-side control is needed or used. I have experienced this. Management wants to know how many widgets were produced, how fast they were produced, how many passed QC, was there downtime, was it planned, are there idle shift hours, is one shift of operators more efficient than another, and on and on and on."

He says: "I don't need or want to remotely access a PLC in a machine to change anything about it. It has done the same job, over and over and over, correctly, for a decade. But the data the PLC can store and transmit is the reason it's connected to a network and polled every 15 minutes for new numbers. To satisfy this need, PLC manufacturers are building in web servers, SQL Lite databases, TCP/IP stacks, and a lot of things that have no business being attached to a device based on 1960s technology that has no provision for security. Again, going back to the automotive comparison, the inventors of CANbus at Robert Bosch company could not have imagined cars would be driving down the road with IP addresses connected to a global network all the time, and would have security flaws that let anyone observe and change CANbus communications inside the vehicle."

And then he says: "Number two, security-conscious staff are not involved with PLCs. Even though many consider PLCs to be outdated, at the end of the day they are exactly like an Arduino or similar microcontroller. They store a program that is executed in a loop at high speed, and the code is evaluated every scan through the ladder logic. And just a quick plug: They do this for decades, in terrible environments, with noisy electrical signals, and with fantastic circuit protections. Reverse the polarity on your Arduino, and you're going to Amazon to shop for another one. Reverse the polarity on a PLC, not a darn thing happens. You'll realize you made a stupid mistake, flip the polarity back, and everything works.

"Anyway," he says, "the people who program these are aging out, and I suspect globally fewer people know how to program ladder logic than did 20 years ago. I'm 36, and I learned to program them 15 years ago; but it seems I'm in the minority in my age group amongst peers in my industry. My observation is this: IT people don't understand or want to understand PLCs, and PLC programmers have no incentive or instruction to make the devices secure. IT staff doesn't consult with the programmers to tell them what

security practices they should follow, or review the final configuration of the PLC. Conversely, the programmer just needs the machine to work; and they are probably fighting numerous mechanical, electrical, and pneumatic problems while completing the programming..."

Leo: Those pneumatic problems, you know...

Steve: Yeah, we had a pneumatic problem. That's why I didn't get the code working.

Leo: Do not underestimate those. They can be a nightmare.

Steve: You do not want a problem with your air pressure, no. "Any extra changes could break the house of cards they've been building. Imagine everything seems to be working, but all that remains is a communication problem. Some PLCs have manuals 700 to 1,000 pages long, and various communication features are scattered throughout the PDF. No organization there. An inexperienced programmer/engineer who's under pressure to complete the already-late project might just start turning everything on, even if they don't know what it is or what the risks are.

"Require authentication? Nah, uncheck that box, that could be the problem. Max number of connections equal one? Well, I don't know what counts and what doesn't, so let's just set it to 10. Set admin password? Better make sure that's blank or default. Don't want to keep something from connecting. Oh, and don't change the port number. That other device over there might be assuming the default port is used, and we don't want to break something that works now and lose ground."

He says: "Honestly, I don't even think we ever are going to fix this. Either industries will eventually move to more advanced systems, which is already happening in some cases, like PC-based control with National Instruments LabVIEW or their competitors; or existing older PLCs just need to be kept in a DMZ or well-guarded network segment. The trouble is, when things aren't broke, they don't get fixed. So already exposed or at-risk PLCs are just going to be sitting there, connected to networks to harvest data, waiting to be leveraged for attacks. And these are the things that keep massive swaths of our public utilities functioning."

So Dylan, I think you got all of that exactly right. And I've said it before, I'm sure this won't be the last time I say it, this podcast has amazing listeners.

Leo: No kidding.

Steve: So thank you, Dylan.

Leo: There's something cool about PLCs. Is it kind of writing in assembly language to write to one?

Steve: Yeah, it's a very low-level tree logic. So it's literally if a, then b. If not, or wait this long, then trigger this. I mean, it is the thing that moves the arms back and forth in those assembly lines.

Leo: I'm sure there are high-level interfaces, though, to see or, you know, Forth was originally designed to do that, to program those things.

Steve: Well, Forth was designed to aim a radio telescope, yes.

Leo: That's right, yeah. And I imagine the aiming mechanism was something like a PLC.

Steve: It was definitely, you know, turn motor on, wait till star moves to center, turn motor off.

Leo: Exactly, yeah, Charles Moore, yeah.

Steve: Yup.

Leo: I love this stuff. There's something cool about putting your code in a hardware device.

Steve: Well, Leo, it's a robot. Robots are cool.

Leo: Yeah, very cool.

Steve: So it is cool. It's cool, I mean, like, the way to motivate grade-schoolers is, remember Logo was the original, you know...

Leo: Yup, a little turtle logic.

Steve: Exactly.

Leo: Yeah, yeah. And of course Start is a great way for high school students to get into robotics, the Start competition. That's, yeah, you're right, that's cool.

Steve: Yeah. I think the idea - and I think also that's where, what is that world that you create, oh, Lego blocks thing?

Leo: Yeah, yeah, Roblox, yeah, Roblox. They're absolutely learning that kind of logic in Roblox. Exactly what they're learning, yeah. Man, I wish I, you know, I wish I had another 50 or 60 years. I'd like to really get into some of this stuff. Very cool. Very cool.

Steve: Okay. So lastly, just quickly on the SpinRite front, last week I rewrote GRC's code signing system.

Leo: Oh. You just rewrote it in a week. No bigs.

Steve: Well, I knew how it worked by then. It took me a month to get it working the first time. But yeah, I did rewrite it because the way I had done it, which was to build the code signing into GRC's server code, had not proven to be 100% reliable, and it needs to be. It turned out that when I was restarting the server, the code signing system did not like that restart. So that was a problem.

Anyway, so I redesigned the system under a client/server model, where we now have code signing as a service. The code signing service runs in the background, with the web server being the service's client, sending it files to be signed. And so far I'm feeling really good about it. It came up. It worked the first time. And it has been flawless ever since. It has never stumbled or had a problem. So this feels like exactly the right solution. Oh, and in the process I was able to switch the signing from using an SHA-1 over to SHA-256. So that feels better, too.

Now, SpinRite's paint continues to dry nicely. One popular tool - which I think is the right way to put it.

Leo: I like it, yeah.

Steve: One popular tool for carrying around and booting ISO image files is something called Ventoy; which, Leo, you obviously are a fan of.

Leo: Yes.

Steve: When I initially heard someone report that SpinRite 6.0's ISO files worked fine with Ventoy, but the various pre-releases of SpinRite 6.1 did not, I planned to eventually get around to looking into what was going on with that. That's the sort of thing one does while the paint is drying. So once I got the signing system redesigned and apparently finally working perfectly, I took a look at Ventoy, which I've never used since I don't do a lot of portable ISO image booting.

Leo: Yeah, it's widely used for things like having 20 Linux distros on a single USB key, that kind of thing.

Steve: Which you are welcome to, yes.

Leo: Well, here's a good example. I would love to have SpinRite plus the Windows installer on a single USB key and be able to switch between the two; right?

Steve: Right. So I brought myself up to speed on Sunday. It is a very slick open source project and tool. It's installed onto a USB thumb drive. Then you simply drop ISO files into its directory. When that drive is then booted, it presents a list of the ISO files it found and allows its user to select any of them to be booted. So I certainly understand its appeal for anyone who wants to carry a toolkit around on a thumb drive.

Leo: Right.

Steve: Okay. Anyway, it turns out that the DOS environment Ventoy creates does not have - or the PC machine environment that DOS boots into doesn't have the HMA. That's the High Memory Area. Now, okay. The High Memory Area is one of the cleverest hacks ever invented.

Leo: Underscore "hack," however.

Steve: It is a hack. It is a 64K memory segment that starts at FFFF, the last 16-byte paragraph of the machine's first 1MB of RAM. Since memory in a segmented memory model is referenced by a positive offset from the start of a segment, starting a segment at FFFF allows for accessing 64K, minus 16 bytes, past the 1MB megabyte point. In other words, this allowed PCs still running in Real Mode to access an additional 64K of RAM, when they were only supposed to be able to access a megabyte. It's actually a megabyte plus 64K minus 16 bytes. Anyway, it is a neat hack that the PC industry came up with and adopted in the later years of DOS, and all recent DOSes have been able to load themselves and their buffers into that region in order to leave more conventional memory available for their programs to run.

Since the DOS execution environment created by Ventoy does not provide that, it forces DOS to load low, and it turns out that there is just barely insufficient RAM left over for SpinRite 6.1 to run. And I mean just barely. It turns out that the slightly smaller size of an unsigned version of SpinRite, which is a few K smaller, does run, as easily does the much smaller DOS-only SpinRite executable.

So after today's podcast, I'm going to tweak the Windows component of SpinRite, which is why we let paint dry, just a bit so that the bootable ISO image it builds will contain SpinRite's 81K DOS executable, rather than the full 250K hybrid DOS and Windows executable. That smaller SpinRite for DOS should then run without any trouble under Ventoy. And a bootable ISO has no need for the full larger Windows version anyway.

In the meantime, nothing new, not one new bug has appeared in the last several weeks, despite the fact that more than 1,000 people have downloaded and have been using the pre-release, this release candidate 6 of 6.1. So I'm going to continue to let its paint dry while I work to get this new SpinRite documented online, then on bringing up GRC's email system. And at that point we'll start letting everyone know that it is ready for primetime.

Leo: Very good. How exciting.

Steve: That is very exciting. And Leo, let's tell our listeners about the advertiser they're excited to hear about.

Leo: I will.

Steve: Then we're going to...

Leo: Do something exciting and fun.

Steve: Or something, look at something very disturbing.

Leo: Uh-oh. Well, you want something disturbing? I got a real story just came in. Three million malware-infected smart toothbrushes have been used in Swiss DDoS attacks. These toothbrushes, I have one at home, have RAM. They have a processor. And apparently they're hackable and have been enslaved, that's a little bit of an inappropriate word, into botnets. Conscripted, how about that, into botnets and used in DDoS attacks. Can you believe that? This is from Tom's Hardware. Thank you, Tom's Hardware, for that...

Steve: I do.

Leo: ...dystopian vision. You might want to secure your toothbrush. I don't know how you would do that. You can't take - I guess they're online.

Steve: They're online, Leo.

Leo: I don't know what you can do to - you know? On we go with the show. And the scary part is now...

Steve: Okay.

Leo: This is for grownups, this part. Yes?

Steve: So, yeah. Everybody knows how bullish and excited I am about Google's Privacy Sandbox.

Leo: Yes.

Steve: We all know I'm a bit of a fanboy for technology. And this is a bunch of very interesting new technology that solves some very old problems. Google clearly understands that their economic model is endangered due to the fundamental tension that exists between advertisers, primarily themselves, who demand to know everything possible about the viewers of their ads; and those viewers, along with their governments, who are becoming increasingly concerned about privacy and anonymity. The emergence of Global Privacy Control and the return of DNT, Do Not Track, has not gone unnoticed by anyone whose cash flow depends upon knowing something about the visitors to their websites.

As we've been covering this through the years, we've watched Google iterate on a solution to this very thorny problem. And I believe, though the final solution was to transfer the entire problem into the user's browser, that they've found a solution that really can work. But, and this is a huge "but" that informs today's title topic, it appears that the rest of the world does not plan to go down without a fight. Not everyone is convinced. Apparently not everyone believes that they're going to need to follow Google. And it turns out that there is a workaround that is not good.

So a recent Financial Times headline read "Amazon strikes ad data deal with Reach as Google kills off cookies," which was followed by the subhead "Media sector scrambles to deal with fallout from phaseout of cross-website trackers." So with a little bit of editing for the content for our listeners, the Financial Times writes: "Tech giant Amazon has struck a deal with the UK's largest publisher, Reach, to obtain customer data to target online advertising, as the media industry scrambles to respond to Google's move to axe 'cookies.' In one of the first such agreements in Europe, Amazon and Reach unveiled a partnership on Monday designed to compensate for the loss of 'third-party' cookies that help gather information about users by tracking their activity across websites to help target advertising.

"Google said this month that it had started to remove cookies on its Chrome browser, following a similar move by Apple to block them over Safari, aiming to switch off all third-party cookies by the end of the year. Reach said it will partner with Amazon on sharing 'contextual' first-party data, for example, allowing advertisers to know what articles people are looking at, with the U.S. tech group using the information to sell more targeted advertising on the UK publisher's sites. The companies said the deal comes 'as the advertising world tackles deprecation of third-party cookies, a long-anticipated industry milestone that Google kick-started in early January.' Financial details for the arrangement were not revealed.

"The partnership involves the contextual advertising of Mantis, originally a brand safety tool that could ensure that brands were not being presented next to potentially harmful or inappropriate content. The tool is also now used to place ads next to content users may want to see, helping to better target specific audiences with relevant advertising. Other publishers also use Mantis.

"Amazon Ads director of EU adtech sales Frazer Locke said that: 'As the industry shifts towards an environment where cookies are not available, first-party contextual signals are critical in helping us develop actionable insights that enable our advertisers to reach relevant audiences without sacrificing reach, relevancy, or ad performance.'

"The loss of cookies means that almost all Internet users will become close to unidentifiable for advertisers. The risk for advertisers is that their advertising offer becomes much less valuable at a time when they're already losing ad revenues, which has led to thousands of job cuts in the past year. Reach last year announced 450 roles would be axed.

"Other media groups are also looking at deals involving their customer data, according to industry executives. Some publishers are experimenting more with registration pages or paywalls that mean people first give first-party information that they can use, such as email addresses and logins. Reach is already seeking to harvest more such data from readers.

"Jon Steinberg, chief executive of Future, said that the 'elimination of third-party cookies is one of the biggest changes to the advertising market in the digital age.' He added that 'advertisers and agencies will be looking to publishers that have high-quality editorial, scale, and rich first-party data,' and predicted that 'advertisers, agencies, and quality publishers will work even more closely together to reach audiences that drive outcomes for brands.'

"Sir Martin Sorrell, chief executive of advertising firm S4 Capital, said that some clients that did not have access to first-party data on their customers were panicking. He said that there would be more focus on getting customers to sign up to websites with their information as companies attempted to boost their stores of 'consented data.'"

Okay. So let's think about this for a minute. This notion of requiring more user sign-ups is interesting, and it's not something that had occurred to me before. This article makes it clear that the advertising industry is not going to let go, and go down without a fight. They don't want to change. They don't want to adopt Google's strongly anonymous interest-based solution. No. They want to continue to know everything they possibly can about everyone, which is something Google's dominant Chrome browser will begin actively working to prevent, at least using the traditional tracking methodology. So what are they going to do? And what's up with this signing into sites business?

It occurred to me that one way of thinking about the traditional presence of third-party tracking cookies is that because they effectively identify who is going from site to site on the Internet, there's no need for us to explicitly sign up when we arrive somewhere for the purpose of identifying ourselves to the site and its advertisers. Cookies do that for us, silently and unseen, on our behalf.

Who we are when we visit a website is already known from all of the cookies our browsers transmit in response to all of the transparent pixels and beacons and scripts and ads that laden today's typical website. But soon, all of that traditional, silent, continuous, background identification tracking is going to be prevented, and the advertising industry is finally waking up to that reality.

What this means for a website itself is significant, perhaps even drastic, a reduction in advertising revenue, since as we know advertisers will pay much more for an advertisement that's shown to someone whose interests and history they know. That allows them to choose the most relevant ads from their inventory, which makes the presentation of the ad that the viewer sees more valuable, and thus generates more revenue for the website that's hosting the ad. And that's, of course, been the whole point of all this tracking. That's why websites themselves have never been anti-tracking, and it's the reason so many websites cause their visitors' browsers to contact so many third-party domains. It's good for business, from the website's perspective, and it increases the site's revenue. And besides, visitors don't see any of that happening.

So tomorrow, when visitors swing by a website with Chrome, which no longer allows tracking, and those visitors are therefore anonymous and far less valuable to that site's advertisers, how does a website itself deanonymize its visitors to know who they are for the purpose of identifying them to its advertisers so that those advertisers will pay that site as much money as possible?

The answer is horrible and is apparently on the horizon. The website will require its visitors to register and sign up before its content - and its ads - can be viewed. At the end of that Financial Times piece, they quoted Sir Martin Sorrell, the chief executive of advertising at S4 Capital, saying "some clients that did not have access to first-party data on their customers were panicking, and that there would be more focus on getting customers to sign up to websites with their information as companies attempt to boost their stores of consented data."

Now, these websites won't be charging any money for this signup. It's not money from their visitors they want. It's the identities of those visitors that, for the first time, they need to obtain from that first-party relationship in order to share that information with their advertisers so that they can be paid top dollar for the ads displayed on their websites. And you can be 100% certain that the fine print of every such site's publicly posted privacy policy will state that any information they obtain may be shared with their business partners and affiliates, meaning the advertisers on their sites.

We thought those cookie permission pop-ups were bad, but things might soon be getting much worse. And those "signup to create an account" forms may also attempt to obtain as much demographic information as possible about their visitors. You know, "Oh, while

you're here creating an account, please tell us a bit more about yourself by filling out the form below so that we can better tailor our content to your needs and interests."

Uh-huh. Right. Such form-fill will likely be a one-time event per browser, since a persistent first-party logon cookie will then be given to our browser to hold and return to the site. So it will only be a brief hassle once. But the result of filling out a form to create an account at every site which might begin to require one will be that our visits to that site will no longer even have the pretense of anonymity. We will be known to that site, and thus we will in turn be known to every one of that site's advertisers.

We may forget that we have an account there, or we may find our name shown in the upper right-hand corner of the screen with a menu allowing us to logout, change our email address, our password, et cetera. And password managers are likely going to become even more important because typical Internet users will be juggling many more Internet login accounts than they've ever needed before. Historically, we only ever needed to logon to a site when we had some need to create an enduring relationship with that site. That is what promises to change. Sites with which we have no interest or need to be known will begin insisting that we tell them who we are in exchange for access to their content, even though it'll be free. And the reason for their insistence will be that we become a much more valuable visitor once they're able, in turn, to tell their advertisers exactly who we are.

And it's all perfectly legal because no tracking is happening. We sign up and implicitly grant our permission for our real-world identities to be shared with any and all of that site's business associates. Most people will have no idea what's going on. Maybe it won't actually be that big a deal. It won't be obvious why sites they've been visiting for years are suddenly asking them to create an account. They already have lots of other accounts everywhere else, and the site won't be asking for money, just for their identities, which most people are not concerned about divulging.

One thing we can be certain of is that a trend of forced identification before the content of an advertising-supported website can be viewed will cause the EFF to have a conniption. Nothing could ever be more antithetical to their principles. The EFF wants nothing short of absolute and complete anonymity for all users of the Internet. So this represents a massive step directly away from that goal. The EFF would be well served, in fact, to get behind Google's initiative, which is far more privacy-preserving than this end-around that appears to be looming.

It almost makes third-party cookie tracking look attractive by comparison. I don't want to be forced to create accounts for every low-value website I might visit briefly. If this happens, it's going to change the way the Internet feels. It's going to be interesting to see how all this shakes out. And yes, I am more glad than ever to be going past Episode 999 since it's going to be very interesting to be observing and sharing what comes next.

Leo: Completely agree. Our mission has really just begun. For a long time, the last five years I thought, well, we've kind of done it all, you know. How much fun is there in the newest iPhone or whatever. But no. I think times are getting very interesting, actually.

Steve: Yeah.