Security Now! #960 - 02-06-24

Unforeseen Consequences

This week on Security Now!

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?

"But this is where you said you wanted the dangerous high voltage terminal box."

Security News

CISA's "Secure by Design" Initiative

Under the headline "CISA and FBI Release Secure by Design Alert Urging Manufacturers to Eliminate Defects in SOHO Routers", last Wednesday, CISA and the FBI published guidance on Security Design Improvements for SOHO Device Manufacturers as a part of the new Secure by Design (SbD) Alert series that focuses on how manufacturers should shift the burden of security away from customers by integrating security into product design and development.

This third publication in CISA's series examines how manufacturers can eliminate the path threat actors are taking to compromise small office/home office (SOHO) routers. And CISA and the FBI specifically called out the Volt Typhoon group which is known to be sponsored by the People's Republic of China (PRC). CISA and the FBI urge manufacturers to: Eliminate exploitable defects in SOHO router web management interfaces during the product design and development phases. Specifically, they implore router manufacturers to:

- Automate update capabilities.
- Remove web management from the WAN interface.
- And require a manual override to remove security settings.

So, first of all, 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. So it's wonderful that this is being repeated and formalized. And, as we know, many consumer routers do now offer automated updates and they are enabled by default.

But the third point they highlight is new, and I think it's brilliant. They say: "require a manual override to remove security settings." In other words, routers should not accept remote 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 China 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 poke around and look at the router's settings. But if the user attempts to change something, the router's UI will say: *"Please press the **Enable Configuration Changes** button on our router to proceed."* Once the button is pressed, the router will allow the user to change its configuration until the user either logs out of the interface or after some period of inactivity of the web interface.

Would this be a pain in the butt? Yep. Especially if the router is in the attic. But 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. The problem being addressed in this case is real. We **are** populating the world with insecure yet increasingly powerful consumer routers which are actually being taken over by malign forces that wish to exploit our traditional lack of security focus. So I, again, give big props to CISA for leading this truly necessary change.

The GNU C Library Flaw

While we were recording last week's podcast, the Qualys Threat Research Unit (TRU) 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 Linux environment. One of these four is a severe vulnerability tracked as CVE-2023-6246. This vulnerability affects major distros such as Debian, Fedora, Red Hat, and Ubuntu. The bug impacts versions going back to August 2022 and is an elevation of privilege flaw that can allow attackers with access to a system to obtain root privilege access.

Here's what Qualys explained about their discoveries:

> *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.*

In other words, it was more than a little bit shocking to Qualys to discover serious exploitable vulnerabilities in a core component of a system that is so 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. Qualys continues:

> *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, 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.*
>
> *For 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 2.37 (August 2022) and subsequently backported to glibc 2.36 while addressing a different, less severe vulnerability (CVE-2022-39046). Major Linux distributions like Debian (versions 12 and 13), Ubuntu (23.04 and 23.10), and Fedora (37, 38 and 39) are confirmed to be vulnerable. This flaw allows local privilege escalation, enabling an unprivileged user to gain full root access, as demonstrated in Fedora 38.*
>
> *In our analysis of the same function affected by CVE-2023-6246, we identified two additional, albeit less severe, vulnerabilities:*
>
> - *CVE-2023-6779 (glibc): This vulnerability involves an off-by-one heap-based buffer overflow in the _vsyslog_internal() function.*
> - *CVE-2023-6780 (glibc): This is an integer overflow issue in the _vsyslog_internal() function.*

Based on our assessment, triggering these vulnerabilities appears more challenging than CVE-2023-6246. Additionally, exploiting them effectively is likely to be more complex.

As for the last vulnerability, a memory corruption issue was found in the GNU C Library's qsort () function, caused by missing bounds check. This vulnerability can be triggered when qsort() is used with a nontransitive comparison function (such as cmp(int a, int b) returning (a – b)) and a large number of elements controlled by an attacker, potentially leading to a malloc() 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. 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 has affected all glibc versions since **1992**. These flaws highlight the critical need for strict security measures in software development, especially for core libraries widely used across many systems and applications.*

Yeah, no kidding. What happens behind the scenes is always interesting. Here's a quick blow-by-blow timeline from the discovery through the coordinated release last Tuesday:

- *2023-11-07: We sent a preliminary draft of our advisory to Red Hat Product Security.*
- *2023-11-15: Red Hat Product Security acknowledged receipt of our email.*
- *2023-11-16: Red Hat Product Security asked us if we could share our exploit with them.*
- *2023-11-17: We sent our exploit to Red Hat Product Security.*
- *2023-11-21: Red Hat Product Security confirmed that our exploit worked, and assigned CVE-2023-6246 to this heap-based buffer overflow in __vsyslog_internal().*
- *2023-12-05: Red Hat Product Security sent us a patch for CVE-2023-6246 (written by the glibc developers), and asked us for our feedback.*
- *2023-12-07: While reviewing this patch, we discovered two more minor vulnerabilities in the same function (an off-by-one buffer overflow and an integer overflow). We immediately sent an analysis, proof of concept, and patch proposal to Red Hat Product Security, and suggested that we directly involve the glibc security team.*
- *2023-12-08: 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.*
- *2023-12-11: The glibc security team suggested that we postpone the coordinated disclosure of all three vulnerabilities until January 2024 (because of the upcoming holiday season). We agreed.*

- *2023-12-13: Red Hat Product Security assigned CVE-2023-6779 to the off-by-one buffer overflow and CVE-2023-6780 to the integer overflow in __vsyslog_internal().*
- ***2024**-01-04: We suggested either January 23 or January 30 for the Coordinated Release Date of these vulnerabilities. The glibc developers agreed on January 30.*
- *2024-01-12: The glibc developers sent us an updated version of the patches for these vulnerabilities.*
- *2024-01-13: We reviewed these patches and sent our feedback to the glibc developers.*
- *2024-01-15: The glibc developers sent us the final version of the patches for these vulnerabilities.*
- *2024-01-16: We sent these patches and a draft of our advisory to the linux-distros@openwall. They immediately acknowledged receipt of our email.*
- *2024-01-30: Coordinated Release Date (18:00 UTC).*

The fact that the biggest worry is a local privilege escalation which enables an unprivileged user to gain full root access, doubtless has many consequences... but this one doesn't feel like the end of the world for the countless – literally countless – Linux-based IoT devices that now litter the landscape. If the exploitation of the privilege escalation requires physical presence at the device, then the world has dodged a big bullet. Let's hope no one figures out how to make remote malicious use out of this one.

Meanwhile, of course, all of the various Linux distros have been updated with everything rebuilt using the patched GNU glibc library. So life goes on.

Fastly CDN switches from OpenSSL to BoringSSL

And speaking of libraries: OpenSSL has lost another big user. The CDN "Fastly" announced that they've decided to switch from OpenSSL to BoringSSL. In their announcement they explained:

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 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*
> - *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*
> - *Similar performance to OpenSSL*
>
> *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.*

It makes sense as a technology is maturing that it's also going to be getting a bit old and creaky. In the case of OpenSSL, it spans decades, having started in 1998. That makes it 26 years old. And, as we know, SSL has evolved dramatically during that time. So Google created BoringSSL and we know that, for example, Amazon's AWS service uses their own homegrown stack.

I'm sure that OpenSSL will remain the bedrock for experimentation and testing 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.

Roskomnadzor asserts itself

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 which is, of course, Russia's telecommunications watchdog agency. So far, 266 web hosting providers have registered with the Roskomnadzor, and all are local companies. Not a single external provider has registered, and those providers are responsible for about one third of all Russian websites. I don't know what's up, but it does seem suspicious that not a single external provider has registered. So this makes me wonder whether this is actually a backhanded way of forcing the remaining one third of Russian sites to switch over to "registered" internal providers. At some point, if Roskomnadzor follows through with its threats, all non-registered providers will be cut off from access to Russian territory.

Google updates Android's Password Manager

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 Password Manager"* Here's what Google announced:

> *Google is working to accelerate passkey adoption. 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.*

Recently, 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 from Pixel 5a) as well as Pixel Tablet. Google Password manager will incorporate these updates for other platforms in the future.

Best of all, 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 as well, to help bring you this easy passkey upgrade experience and usher you into the passwordless future.

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, any website or app, as well as any other password manager, can leverage it to help their users upgrade to passkeys for supporting accounts. It's all part of Google's commitment to help make signing in easier and safer.

So wait a sec... at launch, this initially works with Adobe, Best Buy, DocuSign, eBay, Kayak, Money Forward, Nintendo, PayPal, Uber, Yahoo! Japan and soon, TikTok. 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 and, for example, notify us when we may have reused a password somewhere. So this is our password managers being proactive about our security. It turns out that there's an open standard means for any website that supports Passkeys to advertise its support in a way that any password manager can check and similarly advise.

I did a bit of digging and I found the page where Google describes this, 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.

Speaking in this instance to app and website developers, Google writes:

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 what's the user's experience. Google explains:

*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.*

What this is about is a standardized and uniform way for any Passkey-supporting site to declare its support in a machine-readable way. So this, more broadly than just Google, 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 is the /.well-known/ web directory located at the root of the domain. There's a "passkey-endpoints" JSON-format file under /.well-known that contains two URLs; one to enroll a new passkey and another to manage existing passkeys:

```
{
  "enroll": "https://passkeys-demo.appspot.com/home",
  "manage": "https://passkeys-demo.appspot.com/home"
}
```

Again, any Passkey-supporting site should take every opportunity to enroll its users when they are logging onto the site with a Passkey-supporting client. That's the primary way we can expect Passkeys to become adopted. But it will also be cool to be able to come at this from the direction of the passkey-enabled password managers to have **them** reveal the sites to which we could enroll and switch over to passkey logon and authentication.

Firefox gets post-quantum crypto

Just a quick note that Mozilla has added support for post-quantum cryptography protections to its developer Firefox Nightly builds. So we'll all be seeing it once the release build is published. It can be enabled by going to **about:config** and enabling **security.tls.enable_kyber**.

Closing the Loop

Jeff Zellen / @JeffZellen

Steve: I've been a listener to SN for quite some time and have really enjoyed and gotten a lot out of your "correspondence school". ;) 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."
The github repo is https://github.com/dmaasland/lastpass-authenticator-export -Jeff

I checked it out and it looks nifty. So I wanted to thank Jeff and to share his discovery in case it might be something that anyone needs. Converting TOTP secret back into QR codes can be handy for transferring them to another device or, as Jeff noted, allowing them to be printed onto hardcopy in their universal QR code format – which is what I've done for all of mine.

brenty / @abrenty

Re: Oddly inflated app data: If you look in i(Pad)OS 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".

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, and since they had my user data in my account in the "cloud", I figured, "What the hell?" delete them and reinstall to see what happens. Sure enough, the "documents and data" size was roughly 1/10 of the size after getting everything setup again.

So my theory is that some (many? most?) have logging, cache, and likely other unnecessary, stale data that builds up over time, which they simply don't bother to clean up on their own. I hope this helps, but, of course, always have a backup!

Mental Calm Today / @MentalCalmToday

Greetings, Steve. Long time SN 'student', TWiT Club member, SpinRite user. So excited that you have 6.1 ready for prime time! I'm reaching out to say thanks for your mention of LearnDmarc yesterday. It's really helpful re a confusing protocol. Cheers! (Until next Tues...)

This serves as a reminder to me to mention that the "LearnDmarc" website that we mentioned and looked at last week has been a huge hit among out listeners from all of the feedback I've seen from everyone. So this is just a reminder about it in case you heard of it last week but it slipped from your mind amid all the other things that are going on. It's a truly lovely and very impressive piece of work.

Ron / @rlesserrl

Hi Steve, this is in regards to Sync. I messaged them after your item on Security Now and this is what I received:

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: https://www.sync.com/blog/sync-3-8-21-ios-mobile-app-released/
Let us know if you have further questions. Thanks again, Bailey

So, just a follow-up to that previous listener note that something broken the Sync mobile app for iOS and that Sync was suggesting that it wasn't a priority that would be fixed soon. It appears that was not the case, so I was pleased to learn that Sync responded promptly.

Johnathan Rouse / @PointAndClickTX

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. 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 (Version 1.403.2286.0) when downloading the 6.1 PreRelease. 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!

Johnathan, I can only say, and I know that Leo feels similarly, that I'm so pleased that this podcast and TWiT, have been so useful to you. The good news is that since you're just starting out you have a lifetime of teaching ahead of you. So I wish you all the best as you launch into your career. As for Windows Defender, yes, it continues to be an annoyance. But you sent your Tweet a week ago, last Tuesday, and things may have become better since then. 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 and I'll have more to say about Ventoy in a minute when I update about SpinRite.

anotherthomas / aanotherthomas

@SGgrc: about crypto signing camera.: 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 THAT is some nice outside the box thinking – or in this case, outside the camera! This makes the private key about the owner, not the camera, and the key is presumably more easily protected by them than having the key locked inside the camera. You still have to protect the key, but owners would have the incentive to do that since their photographic reputation is on the line. Very neat idea, Thomas.

DellAnderson / @DellAnderson

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) takes a perfectly authenticated photograph....of a digitally manipulated image? How would this fancy Nikon camera know it was photographing a high resolution 2D image rather than reality?

I replied to Dell that I had the same thought as I imagine many of us have. The problem is that the "authentication" does not and can not 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. 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. But what we actually WANT is to prevent the display of faked scenes as being real? ... and even perfectly protecting the digital chain cannot keep that from happening. It's another case of the lowest hanging fruit. It's easiest to "photoshop" a digital image. But if that becomes far more difficult that someone who is committed to image spoofing can simply spoof the camera itself, which will then attest to having absolutely and positively "seen" the image that it signed.

Slartibartphast /@slartibartphast

I wonder if Google needs native iOS engine to make the new ad auction stuff work.

Absolutely and without question, Yes. The entire Privacy Sandbox API is a collection of entirely new web browser features with a bunch of data storage. 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 so much that we don't yet know about how we get to where we are today and where Google wants to move the 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. Google already has control of nearly all desktops and Android, which are the majority of smartphones. So my questions are, what are Mozilla and Apple thinking? And what conversations must be going on among them?

Aeon | cVk / @AeonFMC

Stephen, I'm personally inviting you to the Gathering of the Stephvens. Next year, in 2025 we're going to set a Guinness World Record for the most people named Stephen/Steven in one area. First goal: gather the Stephvens in this Discord: https://discord.gg/VctfJ2AqPy
Next goal: Conquer the world! You down?

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 World 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.

Dylan / @dpmanthei
We've all seen video segments of complex manufacturing facilities where thousands, if not hundreds of thousands of cans of something, or bottles, or boxes or whatever are moving through a system that's sorting or spinning or stamping or printing or counting or whatever. Just as some of the pre-electronics early computers used banks of mechanical relays, back before the advent of computers, process control engineers would design insanely complex control systems built up from individual 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 with PLC – Programmable Logic Controllers. These PLCs were not very smart. 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.

Remember that we have the term software or hardware "bug" because back in 1947, a dead moth (a bug) was found to be the underlying cause of Harvard's Mark II relay computer not working correctly.

Anyway, we've talked about these PLCs here because attaching them to the Internet has turned out to generally be a 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 & interesting DM from a listener whose thoughts about the problems with PLCs are worth sharing. Here's what Dylan wrote:

Good Day. I'm an engineer and occasionally work with Programmable Logic Controllers (PLCs) 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.

1) Increased demand for 'real-time' data. Just like the CANbus protocol in the automotive industry, PLC's were invented and took hold in manufacturing when security was not a concern. As time went on, protocols were developed to have PLC's talk to each other and to advanced peripherals like motor controllers, touchscreens, printers, or even a SCADA (Supervisory Control and Data Acquisition) computer. 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 in use. I have experienced this. Management wants to know how many widgets were produced, how fast, how many passed QC, was there downtime, was it planned, are there idle hours, is one shift of operators more efficient than another, and on and on. 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 & 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 webservers, SQL Lite databases, TCP/IP stacks, and a lot of things that have NO BUSINESS being attached to a device based on 1960's technology that has no provisions for security.

Again, going back to the automotive comparison, the inventors of CANbus at Robert Bosch company couldn't 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.

2) Security-conscious staff are not involved with PLCs. Even though many consider PLC's 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' of 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-polarity on your Arduino and you're going to Amazon for another one...reverse polarity on a PLC and not a darn thing happens...you'll realize you made a stupid mistake, flip it back, and everything works. Anyway, 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 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 PLC's, and PLC programmers have no incentive or instruction to make the devices secure. IT staff don't consult with the programmer 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 and configuration. 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 700-1000 pages long and various communication features are scattered throughout the PDF. 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 = 1? Well, I don't know what counts and what doesn't so lets make that 10. Set admin password? Better make sure that's blank or default. Oh, and don't change the port number! That other device over there might be assuming the default port is used, and I don't want to break something that works now and lose ground.

Honestly, I don't even think we ever fix this. Either industries will eventually move to more advanced systems (already happening in some cases) like PC-based control with National Instruments LabView or their competitors, or existing, older PLC's just need to be kept in a DMZ or well-guarded network segments. The trouble is, when things aren't broke they don't get fixed...so already exposed or at-risk PLC's are just going to keep 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.

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

SpinRite

On the SpinRite front, last week I rewrote GRC's code signing system. My original design, which built the codesigning system into GRC's server code, had not proven to be 100% reliable and it needs to be. 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. So far its operation has been flawless and this feels like exactly the correct solution. I was also able to switch the signing from using an SHA1 hash to SHA256. So that feels better, too.

SpinRite's paint continues to dry nicely. One popular tool for carrying around and booting ISO image files is something called Ventoy. When I initially heard someone report that SpinRite 6's ISO files worked fine with Ventoy but the various 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.

It's a very slick open source project and tool. After it's installed onto a USB thumb drive, 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.

Anyway, it turns out that the DOS environment Ventoy creates doesn't have the HMA – the High Memory Area. The HMA is that clever 64K memory segment that starts at FFFF – the last 16-byte paragraph of the machine's first one megabyte of RAM. Since memory in a segmented memory model is referenced by the positive offset from the start of a segment, starting a segment at FFFF allows for accessing 64K (minus 16 bytes) past the one megabyte point. In other words, this allowed PCs running in Real Mode to access an additional 64K of RAM. It's a neat hack that the PC industry came up with and adopted in the later DOS years 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 doesn't provide that, DOS loads low and it turns out there is just barely insufficient RAM left over for SpinRite to run. And I mean just barely. It turns out that the slightly smaller size of an unsigned version of SpinRite does run. As does the much smaller DOS-only SpinRite executable.

So after today's podcast, I'm going to tweak the Windows component of SpinRite 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 & Windows executable. That smaller SpinRite for DOS should then run without trouble under Ventoy and a bootable ISO has no need for the larger Windows version.

Nothing new bug wise has appeared in the past several weeks despite the fact that more than 1,000 people have downloaded and have been using SpinRite 6.1's latest release candidate. So I'm going to continue to let its paint dry while I work to get this new SpinRite documented... then on to bringing up GRC's eMail system.

Unforeseen Consequences

The Unforeseen Consequences of Google's 3rd-party Cookie Cutoff

Everyone knows how bullish and excited I am about Google's Privacy Sandbox. 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 GPC (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 watching 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, they have found a solution that really works.

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 and it's not good.

A recent Financial Times headline reads: *"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 phase out of cross-website trackers"*. So, 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 US 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 material.

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 publishers is that their advertising offer becomes much less valuable at a time when they are 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 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 with this signing up business?

It occurred to me that one way of thinking about the traditional presence of 3rd 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; cookies do that 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, beacons, 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 a significant, perhaps even a drastic, reduction in advertising revenue since we know that 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 that viewer more valuable and thus generates more revenue for the website 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 3rd party domains. It's good for business and it increases the site's revenue. And besides, visitors don't see that any of that is 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 de-anonymize 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 it is apparently on the horizon: The web site 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, and chief executive of advertising at S4 Capital, saying that "*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".*

These websites won't be charging any money for this sign up. It's not money they want from their visitors, it's the identities of their visitors that, for the first time, they need to obtain from that 1st-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.

"Oh, while you're here creating your 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."

Right. Such form-fill will likely be a one-time event per browser, since a persistent 1st-party "logon" cookie will then be given to our browser to hold and return. 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 may 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, password, etc. 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 to before.

Historically, we only ever needed to logon to a site when we had some need to create an enduring relationship with that site. This 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. 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 happens. 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 visited 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 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 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 to get behind Google's initiative which is far more privacy preserving than this "end around" that appears to be looming. It almost makes 3rd-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 using the Internet feels. It's going to be interesting to see how all this shakes out... and I'm 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.