Security Now! #950 - 11-28-23

Leo turns 67

This week on Security Now!

Since last week's podcast was titled "Ethernet turned 50" it only seemed right to title this one "Leo turns 67" – I'll have more to say about that at the end. Until then, Ant and I will examine the answers to various interesting questions, including: How many of us still have Adobe Flash Player lurking in our machines? What can you do if you lose your Veracrypt password? Firefox is now at release 120, what did it add? What just happened to give Do Not Track new hope? Why might you need to rename your "ownCloud" to "PwnCloud"? How might using the CrushFTP enterprise suite crush your spirits? Just how safe is biometric fingerprint authentication? How's that going with Apache's MQ vulnerability, and have you locked your credit bureau access yet? Should Passkeys be stored alongside regular passwords? What's the best way to prevent techie youngsters from accessing the Internet?, and is that even possible? What could possibly go wrong with a camera that digitally authenticates and signs its photos? Could we just remove the EU's unwanted country certificates if that happens? What's the best domain registrar, and what was Apple's true motivation for announcing RCS messaging for their iProducts?

Sometimes ya gotta love humanity...

Security News

Adobe Flash Player Updater is (still) desperately trying to update...
"Adobe Flash Player Updater"

I'm still stuck on the task of performing unattended server-side Microsoft authenticode code signing. But I'm managing to inch forward and I've already made one very useful breakthrough, which was programmatically unlocking a PIN-protected hardware token whose key is stored in a new-style KSP key storage provider HSM. I look forward to sharing that with the open source coding community once I come up for air. Last week I discovered an amazing piece of free technology that I would have gladly paid hundreds of dollars for. It's called simply "API Monitor" <http://www.rohitab.com/apimonitor> and I have a link to it here at the top of the show notes. It was once a commercial product but it went free a decade ago – likely because, incredible as this thing is, it's only going to appeal to a relatively small audience. But if this thing is what you need there's nothing else like it. I'd send this guy a donation but there's not even any way to do that. Due to the incredible lack of documentation on Microsoft's next-generation cryptography APIs, I've been reduced to doing a bit of reverse engineering. API Monitor facilitates the creation and exploration of a detailed log of all Windows module API calls. Being an OCD perfectionist myself, I'm not readily impressed but this thing is truly incredible. I won't spend anymore of everyone's time raving about it. If this is the sort of thing that you might find useful, you already know enough to go grab it. It's one of the most utterly stunning pieces of work I've encountered in years.

Here's why I'm talking about it, other than to just give its gifted author more well deserved public praise: One of its process capture modes allows it to be triggered upon any new Windows process that starts. While I was coming up to speed on it, and learning how it works (it also has tutorials) I had not disabled that "trace starting processes" option which I did not need. So I kept seeing pop-ups which I would dismiss. This is not unusual for Windows since there's a bunch of stuff always doing things in the background. But one in particular caught my eye: **Abode's Flash Player Updater kept trying to launch and run!** WHAT?!?!? Since it was interrupting me while I was focused on learning this new tool I kept disabling it. But the damn thing kept popping back up. So, first I clicked the API Monitor's little icon button to disable real-time new process tracing opportunities... then I decided to see what was up with this Flash Player Updater.

Here's the problem with leaving something like this attempting to run in the background: I don't know what URL this thing was constantly querying. Hopefully it was a subdomain of Adobe and **not** some separate domain from Flash Player's legacy Macromedia days. But if whatever domain the updater was querying were to ever become available for any reason, a bazillion PCs around in the world, like mine, would be querying it for an update. Now, hopefully, Adobe also did the right thing and had any updates digitally signed with a "pinned" certificate so that the updater only accept updated code that had been signed with an absolutely specific Adobe certificate. That would, on its face, prevent a malicious actor from injecting their code into these bazillion systems around the world that are all still attempting to update their long since retired copy of Adobe Flash.

Here's what Adobe has to say about their retirement of Flash Player:

Since Adobe no longer supports Flash Player after December 31, 2020 and blocked Flash content from running in Flash Player beginning January 12, 2021, Adobe strongly recommends all users immediately uninstall Flash Player ***to help protect their systems****.*

As we all know, Flash Player was nothing short of a catastrophic security disaster from the moment it appeared. And, of course, Adobe was never seen to be using language like "Adobe strongly recommends all users immediately uninstall Flash Player to help protect their systems" while Flash was a going concern. https://www.adobe.com/products/flashplayer/end-of-life.html

On their Flash Player End-Of-Life FAQ page they ask "Why should I uninstall Flash Player from my system?" and then provide the answer:

Flash Player may remain on your system unless you uninstall it. Uninstalling Flash Player will help secure your system since Adobe will not issue Flash Player updates or security patches after the EOL Date. Adobe blocked Flash content from running in Flash Player beginning January 12, 2021 and the major browser vendors have disabled and will continue to disable Flash Player from running after the EOL Date.

Okay. So first of all, I have no idea why I still had it installed. As we know, at one time, millions of websites and many standalone enterprise applications were dependent upon Flash Player for their operation. I had it installed for research purposes and had been blocking its operation through browsers since early in this podcast. But had I received **any** proactive Adobe suggestion or reminder that Flash Player had gone EOL, I certainly would have clicked their "remove Flash Player" option. I doubt that ever happened. This suggests to me that Adobe may not have been as proactive in promoting Flash Player's removal as they might have been. And even now, when, for the past several years their Flash Player Updater code has been running every hour in my system, probing for any update, why could they not have provided one final update – at any time – which would have caused Flash Player's Updater to remove itself either immediately or the next time my system restarted? This could have been done at any time.

Anyway... I first examined my system's registered system services and sure enough, right up there at the top when sorted in alphabetical order, was "Adobe Flash Player Update Service." But the service's run state was set to "manual". So next I went over to the system's Task Scheduler app and, once again, along with scheduled tasks to keep Google Chrome, Microsoft Edge and a few other odds and ends updated was the task to run the Adobe Flash Update service hourly around the clock. My next stop was Windows Programs & Features where Adobe Flash Player was, once again, at the head of the class. I highlighted it and clicked "Uninstall" ... and to its credit it did, indeed, remove every trace of itself from my system. And good riddance.

So this leaves us with two questions: First, how many of this podcast's security minded listeners might also still have Flash Player – and its very persistent updater – present in their systems? Since it doesn't show itself unless you peek into the proper corners it might be worth taking a look at your various Windows machines' "Programs & Features" list to make sure it's not still represented there. And while you're at it, why not scan through that list and remove any of the other cruft that most systems tend to accumulate over time? I'll betcha there's a bunch of stuff there that you're never going to use again.

The second question is why hasn't Adobe at least been proactive in shutting down the millions of Adobe Flash Player Updater instances that must still be running around the world? The idea of them still having their hooks – literally – into all of these systems is more than a bit disturbing. Nearly three years ago they formally stopped any further updating. So, if they were unwilling to proactively remove Flash Player from everyone's machines, at least they could use everyone's hourly query to remotely shutdown all future queries by removing the Task Scheduler entry and the Update service from everyone's machine. Adobe never did seem to be highly responsible with their shepherding of Flash Player. So perhaps even now it's still everyone's individual responsibility to protect themselves from Adobe's irresponsibility.

Not what you want to have happen

Over the weekend I received a note from a desperate person. It's unclear why he wrote to me. Perhaps GRC came up in a Google search for Veracrypt. But in any event, this is what he wrote:

> *Hi, Veracrypt password lost. How can I get into my device? Veracrypt site says it's impossible: <https://www.veracrypt.fr/en/FAQ.html> So, everything on this device is lost? Please, if you can help... Appreciate any/all help! cgs*

Sorry as we might be for this hapless person, the entire reason he, or someone, presumably chose to encrypt his device with Veracrypt is because, assuming a good password, just as Veracrypt's FAQ correctly stated, NO help is possible or available by deliberate design.

And I said "he or someone" just now because we only have his statement to lead is to assume that he has ***any*** legal or ethical right to the data that have been encrypted on that drive. Knowing nothing more, it could just as easily be that a thief has stolen someone else's driving knowing that it contains the password information for someone's cryptocurrency worth millions and dollars, where the only thing protecting that crypto from theft is the device's Veracrypt encryption... and that right now at this very minute the original true owner of this device is thanking his lucky stars that (a) he chose Veracrypt and (b) he also locked that drive up with a password that no one will **ever** be able to brute force. However, in the meantime, and out of an abundance of caution this person whose jewels have been stolen has had plenty of time to relocate his crypto to some other wallet where it will now, again, be safe. :-)

In any event, the lesson here is (a) use Veracrypt; it remains the go-to solution for open source, previously audited, whole drive or partition encryption. (b) no matter what, always use a really really strong difficult to brute force password, and (c) be very careful to create ample backups of the password you assigned.

Firefox moves to 120 with a bunch of very nice new features

Exactly one week ago, last Tuesday, Firefox's release channel began offering version 120.0. And this one is worth taking a moment to examine. As we noted last week, with the impending end of Chrome's support for Manifest v2.0 which will disrupt the operation of some of Chrome's more popular advertising and tracking controlling extensions, we may soon be seeing a welcome resurgence in Firefox's popularity. I run with its new page recommendations by Pocket. If that helps Mozilla to generate some revenue I'm happy to oblige.

Anyway, Firefox's release 120 brings us a few new features that are worth noting. Its right-click pop-up context menu when right-clicking on a link adds the new feature down at the bottom: "Copy Link Without Site Tracking" which, Mozilla says, ensures that ant copied links no longer contain tracking information. I think that's neat.

Firefox now also supports the welcome setting (in Preferences → Privacy & Security) to enable the GPC – Global Privacy Control – beacon for all queries. Though this is opt-in during normal browsing it is enabled in private browsing mode by default. Also in that same Privacy & Security region you'll find the "Do Not Track" request which can and should also be enabled. As I mentioned a few weeks ago after rediscovering the EFF's Privacy Badger, Privacy Badger also adds those beacons to every user's web requests... but there's no harm in adding a belt to go with those suspenders. And I'll have some ***very*** encouraging news about DNT to share in a minute.

Firefox's private windows and its ETP-Strict privacy configuration now also enhance its Canvas APIs with Fingerprinting Protection, thereby continuing to protect our users' online privacy. As we've discussed in the past, allowing websites to probe various very subtle details in how a given browser renders specific pixel illumination to the user's viewing canvas is just one more trick the trackers have developed to follow us around the Internet.

And listen to this one! Firefox is rolling-out Cookie Banner Blocking by default in private windows for users in Germany during the coming weeks. Firefox will now auto-refuse cookies and dismiss annoying cookie banners for supported sites. Furthermore, also only for all users in Germany for the time being, Firefox has enabled URL Tracking Protection by default in private windows. Firefox will remove non-essential URL query parameters that are often used to track users across the web. Again, I'll have more to say about "why Germany" in a minute.

Firefox now imports TLS trust anchors (you know, web certificate authority certificates) from the operating system root store. This will be enabled by default on Windows, macOS, and Android, and if needed, can be turned off in settings (Preferences → Privacy & Security → Certificates). On my Firefox 120 this checkbox is labeled (and was enabled by default)

> [x] Allow Firefox to automatically trust third-party root certificates you install.

My problem with this wording is that it's somewhat misleading. It sounds as though **users** would be the ones to install those 3rd-party certificates. But that's rarely the case. Presumably, Mozilla is attempting to be more compatible with the 3rd-party TLS proxying middle boxes increasingly employed by enterprises to filter their network traffic. The use of any of those requires that the browser trusts the certificates they mint on the fly. Those 3rd-party root certs are typically installed directly into the operating system over the network through active directory and group policies. Firefox is unique in that it has always used its own root store and has not been dependent upon the hosting operating system's root store. So it must be that this move is intended to make the use of Firefox easier in such settings.

The worry is that if the EU gets its way and is able to force browsers and operating systems to install their member countries' web certificates into their root stores, then this mechanism,

which has just now been added to Firefox 120, **would** automatically place it into compliance with that EU effort. However, given Mozilla's clearly and quite strongly publicly stated position on the EU's planned eIDAS 2.0 QWACs certificates, it seem unlikely that pre-compliance with something to which Microsoft quite strongly disagrees what their motivation. Remember that rather than signing onto that large open letter that most others co-signed, Mozilla chose to write one of their own, which the likes of Cloudflare, Fastly, the ISRG, the Linux Foundation and others signed. So my guess is that smoother functioning within the enterprise was the sole motivation. And note that we do have a simple checkbox we can uncheck if we do NOT want to have Firefox's root store supplemented (or polluted) by its underlying host OS root store.

So now let's talk about Firefox and Germany...

Do-Not-Track is back on track!!

Okay. So there were several interesting changes in Firefox which, for now at least, only benefit German users. What's up with that? It turns out that the German courts have been weighing several issues and that their decisions have come down on the side of user privacy and choice.

TechRadar pulled together a nice piece, providing this recent news and also some backstory about the DNT and GPC beacons. With a bit of editing, here's what TechRadar explained:

Germany is perhaps the most proactive country when it comes to protecting its citizens' privacy, something that privacy advocates and enthusiasts have been aware of for a while now, and the country recently reiterated its stance against Microsoft-owned LinkedIn.

***A Berlin Court found in favor** of the Federation of German Consumer Organizations, which filed a lawsuit against LinkedIn for ignoring users who turned on the '**do-not-track**' function on their browsers. According to the German judge, **companies must respect these settings under GDPR**.*

A small victory for privacy, the Do-Not-Track (DNT) ruling might end up reshaping how websites and other online platforms have to handle our data more broadly. Adoption and support of DNT has been in sharp decline over the past years. Now, ad-blocker and VPN service provider, AdGuard, believes this potentially game-changing court decision could exhume the once-abandoned privacy initiative for good.

Do-Not-Track (DNT) headers are beacons sent by web browsers to proactively inform a website not to collect or track that visitor's browsing. DNT was first proposed by researchers Christopher Soghoian, Sid Stamm, and Dan Kaminsky in 2009 to limit web tracking. A year later, the US Federal Trade Commission gave its approval and called for the creation of a universal mechanism to give users more agency over their data.

*The first web browser to support the new initiative was privacy-focused **Mozilla Firefox** which added the feature in March 2011. Other services followed suit, including Microsoft's Internet Explorer, Apple's Safari, and Opera. Google Chrome embraced the industry trend in 2012.*

AdGuard saidL "The early 2010s was perhaps the time when the enthusiasm for the DNT and its potential to improve privacy was at its peak."

Yet, after initial success with browsers, the DNT wave seemed destined to dwindle. The problems started from the lack of similar support among websites and advertisers. Even Google implemented the feature only on Chrome, refusing to "change its behavior" on its websites and web services. In other words, Google declined to honor DNT requests, even from users of its own browser. The final nail in the coffin came in 2019, when the group working on standardizing DNT was dismantled due to a lack of consensus.

Privacy advocacy groups did not want to renounce giving users a way to better protect their personal data and browsing activities, though. AdGuard explained: "While DNT failed to gain much support, the need for a mechanism that would allow people to opt out of having their personal information shared or sold was still strong."

Privacy focused experts believe organizations should allow their customers to decide whether to have their information shared or sold in the first place. It was from this need for an alternative that the ***Global Privacy Control (GPC)*** *was born in 2020.*

Like DNT, the GPC is a signal sent with every web request over HTTP to op-out having browsing data collected or sold. Supporters of this new initiative include many privacy-first browsers and search engines—like DuckDuckGo, Brave, and Firefox—and browser extensions such as Abine's Blur, Disconnect, OptMeowt, and EFF's Privacy Badger.

GPC seems to have gained more traction than DNT was ever capable of—until now at least.

Learning from past mistakes, GPC found a way to add value for publishers and advertisers, too, who recognize that GPC seems to help websites increase trust among users. At the same time, GPC still allows them to sell ad space without having to sell data to third parties.

According to AdGuard, GPC also makes things a bit easier for websites. "GPC is more precise about what exactly the other party is allowed or disallowed to do, this is part of the attempt to make it legally binding. DNT just says "don't track me" without specifying what tracking is. What if the data being collected is actually required for the service to operate? GPC says "do not sell or share my data" which is much easier to understand."

In August of last year, GPC won its first legal battle in California against commercial retail brand, Sephora. And now, October 30, 2023, may be remembered as a milestone for the DNT initiative as ***the Berlin Regional Court*** *ruled that LinkedIn can no longer ignore its users' Do-Not-Track requests.*

Rosemarie Rodden, a legal officer with the German consumer rights group who brought the lawsuit said: "When consumers activate the 'do-not-track' function of their browser, it sends a clear message: They do not want their surfing behavior to be spied on for advertising and other purposes. Now, website operators ***must*** *respect this signal."*

It turned out that the judge agreed with Rosemarie, ruling that LinkedIn is no longer allowed to warn its users that it will be ignoring their DNT signals. ***That's because, under GDPR, the right to opt out of web tracking and data collection can also be exercised using automated procedures.*** *In other words, the court found that a DNT signal* ***is*** *legally binding. This sets a precedent and revives the all-but-abandoned idea of Do-Not-Track."*

Not everyone is happy with this decision. LinkedIn spokesperson told Cybernews: "We disagree with the court's decision which relates to an outdated version of our platform and intend to appeal the ruling."

Outdated? What's outdated, the LinkedIn platform? If the ruling only applies to an outdated version of the LinkedIn platform then why appeal the ruling? And surely you can see with the growing support for the closely related GPC signal, and with Google own development of a non-tracking means to obtain interest categories for web users, that this is a tide that's finally beginning to shift?

> *AdGuard told TechRadar: "The implications of this ruling are far-reaching and potentially earth-shattering. Websites have been ignoring the signal for years without consequence."*
>
> *According to AdGuard, the fact that the judge agreed with the consumer groups' requests for DNT to be honored highlights the need for companies to prioritize user preferences. At the same time, despite being unclear about what will happen next, AdGuard is also hopeful this could mark the start of a new era for Do-Not-Track which, for the first time, has been recognized as legally binding in Court in favor of the more promising GPC.*
>
> *AdGuard said: "We hope that the [DNT] signal will become binding, but it may still be a long shot. Industry players have become accustomed to doing anything they wish and ignoring such signals. So it's likely to take active steps by regulators to change their behavior. But in general, enforcement is possible."*

And more enforcement is likely on the horizon. One decision in one country doesn't change the world overnight. But it's a step in the right direction. And there are times when having some ambulance chasing attorneys around can come in handy! :-)

"ownCloud" -or- "PwnCloud" ??

It appears that anyone running any instance of the very popular open source "**ownCloud**" file sharing system needs to take immediate action – as in, immediately unplug anything running ownCloud to get it off the Internet. Did I already say ***"immediately!"*** ?? Unfortunately, due to today's ultra-swift nature of the exploitation of any publicly announced vulnerabilities – in this case it's a remotely exploitable CVSS 10 out of 10 – it may already be too late. But even if so, at least closing the unlocked front door and working to clean up any damage still needs to be done. So here's the story...

GreyNoise reported the following in their coverage of CVE-2023-49103, yesterday, they wrote under the title *"ownCloud Critical Vulnerability Quickly Exploited in the Wild"*:

> *On November 21, 2023* [so that's exactly one week ago, today]*, ownCloud publicly disclosed a critical vulnerability with a CVSS severity rating of 10 out of 10. This vulnerability, tracked as CVE-2023-49103, affects the "graphapi" app used in ownCloud. ownCloud is a file server and collaboration platform that enables secure storage, sharing, and synchronization of commonly sensitive files. The vulnerability allows attackers to access admin passwords, mail server credentials, and license keys. GreyNoise has observed mass exploitation of this vulnerability in the wild as early as November 25, 2023.* [so it took four days from announcement for mass exploitation to take off.]
>
> *The vulnerability arises from a flaw in the "graphapi" app, present in ownCloud versions 0.2.0 to 0.3.0. This app utilizes a third-party library that will reveal sensitive PHP environment*

configurations, including passwords and keys. Disabling the app does not entirely resolve the issue, and even non-containerized ownCloud instances are at risk. Docker containers before February 2023 are not affected. Mitigation information listed in the vendor's disclosure includes manual efforts such as deleting a directory and changing any secrets that may have been accessed. In addition to CVE-2023-49103, ownCloud has also disclosed other critical vulnerabilities, including an authentication bypass flaw (CVE-2023-49105) and a critical flaw related to the oauth2 app (CVE-2023-49104). Organizations using ownCloud should address these vulnerabilities immediately.

Okay. So for ownCloud users, we have a potential 4-alarm fire situation. There are three newly disclosed CVEs with ratings of the difficult to attain 10.0, a very critical 9.8 and a still bad 9.0.

The 49103 CVE with a CVSS of 10.0 allows for a disclosure of sensitive credentials and configuration in both containerized and non-containerized deployments. The 49105 CVE is the second worst with a CVSS of 9.8. It's a WebDAV API authentication bypass using Pre-Signed URLs which impacts core versions from 10.6.0 to 10.13.0. And the third 49104 CVE with the CVSS of 9.0 is a subdomain validation bypass impacting OAuth2 prior to version 0.6.1.

In the case of this first worst mistake – and a mistake is what it is, it's not some fancy Log4J vulnerability – anyone who has any experience with PHP knows that you **never** want to expose PHP's phpinfo applet to the public Internet, yet that's exactly what this "graphapi" has done. Located down the path: "owncloud/apps/graphapi/vendor/microsoft/microsoft-graph/tests/" is a "GetPhpInfo" PHP file – and it can be accessed remotely to disgorge all of the system's internal sensitive data including all the environment variables of the web server and in containerized deployments this includes the ownCloud admin password, mail server credentials, and license key. ownCloud recommends deleting that file and administratively disabling the very dangerous phpinfo function. This can be done simply by adding "phpinfo" to the "disable_functions" list in the system's php.ini file – and, sadly, that list is empty by default.

After doing this, do **not** make the mistake of not **also** immediately rotating all of the system's credentials – the admin password and the mail server and database credentials as well as Object-Store S3 access keys if the ownCloud instance was hosted by an S3 cloud provider.

The second problem makes it possible to access, modify or delete any file without any authentication if the username of the target is known and they have no signing-key configured, which is the default behavior. That's also obviously quite a potentially serious vulnerability.

And the third 9.0 flaw relates to a case of improper access control that allows an attacker to "pass in a specially crafted redirect-url which bypasses the validation code and thus allows the attacker to redirect callbacks to a TLD controlled by the attacker." Besides adding hardening measures to the validation code in the oauth2 app, ownCloud has suggested that users disable the "Allow Subdomains" option as a workaround.

Everyone using ownCloud should update to the latest builds and make sure that everything else is still okay. 12 unique IPs were found to be scanning the Internet looking for instances of ownCloud.

CrushFTP Critical Vulnerability

While we're on the topic of critical vulnerabilities that will wreck your day or your week or maybe even your month, anyone using the "CrushFTP enterprise suite" (and there are currently somewhere around 10,000 publicly exposed instances of it on the Internet) must **immediately** update to v10.5.2 or later.

Back in August, the security firm Converge Technology Solutions responsibly disclosed a critical unauthenticated 0-day vulnerability affecting the CrushFTP enterprise suite. Having 10,000 of these instances publicly exposed is bad enough but a great many more are known to be residing behind corporate firewalls which malware might arrange to crawl behind. The exploit permits an unauthenticated attacker to access all CrushFTP files, run arbitrary programs on the host server, and acquire plain-text passwords. The vulnerability was fixed in CrushFTP version 10.5.2 and it affects software in the default configuration on **all** operating systems.

What's more, Converge's threat intelligence has discovered that the security patch which resolved this problem **has** been reverse-engineered, and adversaries have developed proofs of concepts. So forthcoming exploitation can be presumed. The attack chain hinges upon an unauthenticated query when CrushFTP parses request headers for a data transfer protocol called AS2. By exploiting the AS2 header parsing logic, the attacker gains partial control over user information Java Properties. This Properties object can then be leveraged to establish an arbitrary file read-and-delete primitive on the host system. Using **that** capability, the attacker can escalate to full system compromise, including root-level remote code execution.

Bypassing fingerprint authentication

An interesting flaw has been found in the fingerprint sensors manufactured by Goodix, Synaptics, and ELAN. The OEMS who purchase and integrate those sensors and whose equipment is therefore vulnerable to fingerprint sensor based authentication bypass include the Dell Inspiron 15, the Lenovo ThinkPad T14, and Microsoft Surface Pro X laptops, just to name a few which are known to contain those popular sensors.

The flaws were discovered by researchers at Blackwing Intelligence. All three of the fingerprint sensors are the "good kind" which perform MoC verification which stands for "match on chip". That's what you want since it integrates the matching and other biometric management functions directly into the sensor's integrated circuit. But the researchers said:

*"While MoC prevents replaying stored fingerprint data to the host for matching, it does not, in itself, prevent a **malicious sensor** from spoofing a legitimate sensor's communication with the host and thus falsely claiming that an authorized user has successfully authenticated."*

To thwart this problem, Microsoft created something known as the Secure Device Connection Protocol (SDCP). It's designed to eliminated this problem by establishing an end-to-end secure channel between the sensor and the machine's motherboard.. But the researchers designed a novel technique that can successfully circumvent these SDCP protections to create adversary-in-the-middle (AitM) attacks.

And the ELAN sensor which interfaces over USB doesn't even offer SDCP. So it's easily spoofed simply by sending cleartext security identifiers (SIDs). This allows any USB device to masquerade as the fingerprint sensor and claim that an authorized user is logging in.

In the case of the Synaptics fingerprint sensors, not only was SDCP found to be turned off by default, the implementation used a known-flawed custom Transport Layer Security (TLS) stack to secure USB communications between the host driver and sensor. So it was possible to defeat the biometric authentication.

The exploitation of the Goodix sensors leverages a fundamental difference in enrollment operations carried out on a machine that's loaded with both Windows and Linux – or a Windows machine that just transiently boots a copy of Linux. This takes advantage of the fact that Linux doesn't support SDCP. So, in a truly lovely hack the following is done:

- Boot Linux
- Enumerate valid IDs
- Enroll the attacker's fingerprint using the same ID as a legitimate Windows user
- Intercept the connection between the host and sensor by leveraging the cleartext USB communication
- Then boot to Windows
- Intercept and rewrite the configuration packet to point to the Linux DB.
- And finally, login as the legitimate user with attacker's print

Essentially, this allows for the installation of an attackers fingerprint and association to the legitimate user's fingerprint.

It's also worth noting that although the Goodix sensor design anticipated this bait-and-switch weakness and therefore uses separate fingerprint template databases for Windows and non-Windows systems, the attack is possible thanks to the fact that the host driver sends an unauthenticated configuration packet to the sensor to specify what database to use during sensor initialization. Whoops!

To mitigate such attacks, the researchers have recommended that OEMs enable SDCP and ensure that the fingerprint sensor implementation is audited by independent qualified experts. So, insert our standard refrain there.

And just for the record, this is not the first time Windows Hello biometrics-based authentication has been successfully defeated. In July 2021, Microsoft issued patches for a medium-severity security flaw (CVE-2021-34466, CVSS score: 6.1) that could permit an adversary to spoof a target's face and get around the login screen. The researchers said that:

"Microsoft did a good job designing SDCP to provide a secure channel between the host and biometric devices, but unfortunately device manufacturers seem to misunderstand some of the objectives. Additionally, SDCP only covers a very narrow scope of a typical device's operation, while most devices have a sizable attack surface exposed that is not covered by SDCP at all."

I think our takeaway from this should be to not over rely upon the convenience offered by biometric authentication. This is why Apple, whose biometric authentication has been very

tightly designed by security-crazed engineers, still requires the "something you know" to initially unlock their devices following any restart of the device. If I had any device whose security was truly critical to me, I'd encrypt its drive and supply the key with an outboard USB dongle. That's what I did when I was in Europe with a laptop during the SQRL tour.

ApacheMQ

Though I don't have any big news on the ApacheMQ message queue vulnerability that we first talked about late last month, I wanted to mention that it remains under very active exploitation. A proof of concept exploit was initially posted on Github. It was later updated to add an English language version, and then it was further improved two weeks ago to improve its TLS support. So, by now, it's pretty much the case that any Apache server that has been left unattended will be spinning its fans as fast as possible since crypto-miners have recently been observed being installed into any still-vulnerable servers.

TransUnion & Experian both hacked

Two of our major credit reporting bureaus, TransUnion and Experian, were both just hacked with their super-sensitive consumer data exfiltrated. The hacking group named "N4ughtySecTU" which of course is "hackerese" for "naughty sectu", is asking for $30 million from each firm, threatening to release its customers' data online. And this is the second time the N4ughtysecTU group has hacked TransUnion, having previously done so back in March of 2022.

So I'll just take this opportunity to once again remind everyone that all four of the major credit reporting bureaus support credit locking and that everyone should be taking advantage of this feature. Given today's cybercrime environment, and the fact that those who are holding and aggregating our private information without our permission have been proven unable to keep it private, we need to minimize the chance that our private information will be leveraged against us for identity theft. Identity theft is one of the most debilitating and difficult to recover from things that can happen to an individual.

A few years ago I decided that since I was such a large customer of Amazon, I would route my purchases through their credit card to obtain an additional several percent savings. To apply for their card, I needed to briefly drop my credit reporting agency shields to allow Amazon's credit folks to verify my credit worthiness. What I learned at the time was that it is now possible to ask the bureaus to temporarily drop our shields for a specified duration, after which time they will automatically snap back up. So that really removes the last barrier of inconvenience from having one's credit reporting blocked by default. Everyone listening and everyone you care about should be running with their shields up all the time.

Closing the Loop

Christian Rutrecht / @chrisrutr

Hi Steve, Not sure if you have managed to catch up to passkey support in Bitwarden. I have not heard it mentioned lately in Security Now. https://bitwarden.com/help/storing-passkeys/

I have just started testing it on selective services and it works flawlessly across my various devices. I am very impressed, I must say.

What I would like to know is the view you have on adding all your passkeys to a combined password vault? I know that you have a standpoint that TFA verification apps or physical token devices should be separated. But what about passkeys?

Personally, I prefer combining everything for the sake of conveyance. I have family members and colleagues that I try to nudge toward using a PW manager. For them to be able to use it, it must be easy going; even the concept of having "some thing" remembering their password is complicated to comprehend for some.

In my research I have found out that the best way of keeping my PW security posture up to date and ready available for access, is to have a single vault / APP for everything. I chose Bitwarden for that purpose at the conclusion of my research 2 years ago, as it was the best hardened platform available, including the support for authentication tokens – and now passkeys. Keep up the good work.

So, Christian, I would classify Passkeys exactly as I would passwords. Passkeys are just superior passwords because, by using public key asymmetric crypto instead of secret key symmetric crypto, Passkeys are inherently immune to a great many of the attacks and failure modes that have always beset passwords. In other words, I think it's entirely acceptable to keep Passkeys in the same vault, managed right alongside your traditional passwords.

And as you noted, I do feel strongly that the entire point of multi-factor authentication is to create a clean and clear security boundary for use when remotely authenticating to a higher than usual security facility. For that reason, the idea of having a password manager also able to fill-in the time varying 6-digit MFA token makes me shake my head. Why bother at all. It's true that some benefit will be derived from the inherent time-varying nature of the token. So simple replay attacks will be thwarted. But if you're going to go to the trouble of using some form of multi- factor authentication, why not get as much benefit from it as you can?

A listener named Victor wrote...

Howdy Steve, Long time IT guy here, but recent (~1y) listener as I didn't really do podcasts until SecurityNow.

I have a question for you that may be something of a rabbit hole, but I am seeking opinions on parental controls. I consider myself a well enough accomplished IT guy but I am facing a problem in that one of my kids is about step into the realm of getting a phone.

We, my wife and I, have held them off until now citing COPPA laws but we are out of excuses at this point. The issue is that for all of my IT experience, this kid (my third) is exceedingly

more tech savvy than any of my other kids, having already proven their ability to circumvent restrictions on school operated technology and continuing to do so without any repercussion as the school can't seem to collect any evidence of their wrong doing.

I have done my best to protect the homefront (piHole, pfSense router with static routes to nowhere for undesired sites, etc) but once on the phone, the kid will be able to connect when/where-ever they please and I have yet to find a truly secure parental control app which will do all the standard watchdog things AND self protect from deletion on Apple and/or Android. Any advice is welcome. Here's to 999 and beyond. -Victor

Though I never had my own kids, during my late 20's, 30's and 40's I participated in raising several long term girlfriend's kids from pre-teen through their teens. And although I was never more than "mom's boyfriend", I was around during some important years, so we bonded and I've remained in touch with several of them who are now married and with their own kids. So I'm not a total newbie on this front.

Victor didn't share the age of his youngest and most technically savvy of the three, so this might not apply if this individual is too young. But if this person is this tech savvy then perhaps they're not very young. What occurs to me is to wonder whether this particular problem has a technical solution. I think that perhaps the solution lies in parenting rather than in technology. I'm horrified by what is now available on the Internet. And I completely get it that age appropriateness is a real thing. There are many salacious adult depravities that young minds should not be exposed to until they have obtained sufficient context and maturity to understand them for what they are.

But at the same time, "blocking" feels like a losing uphill battle. The Internet is truly pervasive. If this youngster wants access to the Internet he or she is going to obtain that access. If not at home where IT security is strict, then at a friend's home whose parents never considered this to be a problem; or by breaking through the school's security. And erecting technical blockades might just present a challenge to make what's hiding behind them seem all the more intriguing.

Given what's out there, I understand the dilemma that today's parents face. And I would **not** want to be in that position today. But I also believe that there's a very real limit to a parent's ability to control what a free ranging young person is exposed to. I think that if I were in this place, I would sit down with all of my kids as a group and talk to them honestly and openly about what's on the Internet, and why. About how a great deal of what's there does not represent what most people think and feel. About how it's often deliberately extreme. About how behind a lot of it is a profit motive, trying to separate people from their money one way or the other. And I would also take some time to explain about predation on the Internet. About how there are truly dangerous people hiding behind fake names, photos and identities. That these people are often not who they claim to be. They may well be in a far off country and not be at all nice people. And that the only thing that you ever really know are the people you've met in the real physical world. I wouldn't pull any punches. I'd tell them that I'm terrified by the idea of them being exposed to what's out there on the Internet. And that the only thing that will keep them safe is their own common sense and keeping lines of communication open with their true friends in the real world, and with their parents.

alphageek / @ageekfood

I am interested in your take of this from a security standpoint. Thanks for the years of helping keep my brain sharp. As an EE, your podcast has helped me look smart at important times.

What AlphaGeek was curious about was an interesting solution to the problem with deep fake photos. The IEEE Spectrum Magazine carried an interesting story about a new Leica camera that binds authenticating metadata into the photos it takes then digitally signs them as they are taken. And there's more. Here's what the article explained:

Is that photo real?

There's a new way to answer that question. Leica's M11-P, announced in late October, is the world's first camera with support for content credentials, an encryption technology that protects the authenticity of photos taken by the camera. The metadata system can track a photo from shutter snap to publication, logging every change made along the way.

Award-winning photographer David Butow said: "In the last few years it's become easier to manipulate pictures digitally. Photographers can do it, and when the photos are out on the Web, other people can do it. I think that puts in jeopardy the strength of photography, the sense that it's a true representation of what someone saw."

In November of 2019, Adobe, The New York Times, and Twitter partnered to solve this problem by founding the Content Authority Initiative (CAI). Twitter left the CAI after Elon Musk purchased the company, but CAI, which now boasts over 200 partners, gave itself the difficult task of finding a "long-term, holistic solution" for verifying the authenticity of photos. In 2021 it joined with another initiative called Project Origin to form the Coalition for Content Provenance and Authenticity (C2PA).

Leica's M11-P is the first hardware embodiment of its solution. The camera has a toggle to flip on content credentials, which is based on the C2PA's open technical standard. The M11-P then embeds identifying metadata—such as the camera, lens, date, time, and location—in an encrypted C2PA "manifest." The M11-P digitally signs the manifest with a secure chipset that has a stored private key. The manifest is attached to the image and can be edited only by C2PA compatible software which, in turn, leaves its own signature in the manifest.

Once published, the image can display a small interactive icon that reveals details about the photo, including the device used to take the photo, the programs used to edit it, and whether the image is wholly or partially AI generated.

It's still early days for content credentials, however, so support is slim. Adobe's software is the only popular image-editing suite to support the standard so far. The presentation of the data is also an issue: The interactive icon isn't visible unless an app or program is programmed to present it.

David Butow said: "The way this technology is integrated in Photoshop and Lightroom, which is what I use, is still a bit beta-ish." David used the Leica M11-P for several weeks prior to its release but he says these early problems are countered by one key win: The standard is easy for photographers to use. "You shoot normally, right? There's nothing that you see, nothing that you're aware of when you're taking the picture."

The Leica M11-P's support for content credentials wasn't the only reason it made headlines. It arrived with an intimidating price tag of US $9,195. That's a high price for authenticity, but Leica says the camera's cost has more to do with Leica's heritage. Kiran Karnani, Leica's vice president of marketing said: "If you look at the price points for our M cameras, there's absolutely no added cost to have the content credentials feature in the M11-P."

And the M11-P is just the tip of the iceberg. Canon and Nikon already have prototype cameras with content credentialing support. Smartphones will also get in on the action. Truepic, a startup that builds "authenticity infrastructure," has partnered with Qualcomm to make Qualcomm's Snapdragon 8 Gen 3 chips support content credentials; those chips will power flagship Android smartphones next year.

No news organization currently requires photographers use content credentials, but the C2PA standard's influence is beginning to be felt. Karnani points out that The New York Times and BBC are members of the CAI (as are The Wall Street Journal, The Washington Post, the AP, Reuters, and Gannett). Karnani notes that "Adoption is certainly a goal."

To answer alphageek's question, this all sounds great on the surface. A digital camera contains a digital representation of an image which can be digitally signed by the camera itself. The way this would be done is that metadata would be added to the image. Then a cryptographic hash would be taken of the file. That hash would then be encrypted using the camera's private key.

Then, at any later time, it would be possible to verify that not a single pixel of the image had been tampered with by rehashing the image, and using Leica's published public key to decrypt and verify that the hash bound to the image matches the one that was just made.

But from everything we know of crypto there would appear to be one large glaring problem with this entire concept. A web server's private key is secure **only** because no unauthorized people are able to obtain its key. If that key is in a hardware HSM then that key won't ever even exist in the machine's memory, making it even less accessible. Although asymmetric encryption offers many cool features and powers, it **does** still rely upon a secret being kept. Its private key **must** remain private. And that's the Achilles heel that I fear any digitally signing camera will face.

A web server's private keys are safe only because no one has unauthorized ***physical*** access to its hardware. If you can get to the hardware all bets are off. Just ask the folks that thought that encrypting DVD discs was a great idea. They thought: *"No problem! We'll just embed all of the decryption keys into every consumer DVD player so that they'll be able to decrypt the discs."* Right. Back in the day, my copy of "DVD Decryptor" was one of my favorite tools. It was and is entirely legal to decrypt one's own DVDs; and I appreciated the freedom that afforded.

In order for this Leica, or any other camera, to digitally sign anything, it ***must*** carry a secret. It's the camera's secret that makes its signature mean something. But the camera is obviously not locked up in some data center somewhere. Just like a DVD player, it must be out in the open to do its job. And everything history has taught us is that these secrets cannot be kept. Not under these conditions. And if that's true, it creates another new problem that we never had before: **Digitally verified deep fakes.** Once a camera's secret signing key escapes, deep fakes will be signed and digitally authenticated, making the problem worse than it was before. So, it'll be interesting to see how this all turns out. Mark me down as skeptical and a bit worried.

Andrew Drapper / @adrapper

If the EU demand their certificates are in our root store, could we not just remove them or have a script, or extension that does this?

So many questions about this still remain unanswered. For example, would the EU's certs be counter-signing traditional certificate authority certs? If not, then removing those trust roots would prevent access to those EU web services. Would these EU certs be trusted all by themselves? If not, then we really don't have anything to worry about. So long as a traditional CA also needs to sign a website's certificate, the EU's signing would simply be adding additional information. But if this were the case, everyone would not be all up in arms over this. So it appears that the EU wants their certs to be able to stand alone. Would these EU certs carry some distinguishing mark that would allow an automated cert sweeper to uniquely identify and remove them? I suspect that the CA Browser forum **would** require some form of clear designation and the good news is that certs have all manner of means for carrying such markings. This would make a cert cleaner entirely safe. One potential problem is that users of affected machines, such as in the enterprise, may have limited access to their machine's certificate root stores. But the biggest problem is that while those listening to this podcast and other in-the-know techies might know enough to clean their root stores, most of the world would not. So, yeah, even if some of us were to keep our machine's clean, that doesn't help everyone else.

Mike / @bigmike613

Hey Steve, what company do you recommend for a domain registrar? I currently have all my domains with google domains and they are moving to square space. I only need a place to store the domains as my name servers are with various other providers. Thanks, Mike.

Without any hesitation I would and do always recommend **Hover**... and I cannot imagine why I would ever move. I did move once, and that was away from Network Solutions. They were the original primary registrar of domain names but let's just say, they did not age well. I became so tired of Network Solutions constant up-selling attempts. When I did anything I would be forced to decline one "special limited time offer" after another, endlessly, just to renew a domain. I'm inherently loyal, so I stuck with them as long as I could. But finally it was too much. So I went looking for an alternative. A good ultra-techie friend of mine, Mark Thompson at AnalogX, has all of his domains with GoDaddy. But GoDaddy's style doesn't appeal to me either. They just don't seem serious, and the one thing you want in a domain registrar is seriousness. They've also had security problems in the past with some of their services, though I don't think with their domain registrar business. By comparison, Hover is just a clean and simple domain registrar. They do offer some other services, but they are never pushed. For a long while they were advertisers here on TWiT, but it was one of those situations where I had switched to Hover and was already singing their praises every chance I got long before they began advertising here. And I still am for the same reason. So anyone who is looking for a clean and simple no-frills, no annoying upselling, domain registrar will find that in Hover. And I know that Leo feels exactly the same.

Glenn F / @GlennFields

Hi Steve. I was just listening to SN949 and wanted to let you know you may have been a bit overly charitable when describing Apple's motives around RCS. Looks to me like Google got creative and used the EU as a cudgel to "encourage" Apple to adopt RCS. From what I can tell, Apple's RCS announcement appears to coincide with the deadline for their response to the EU. Love the show and just recently joined Club TWIT due to all the great content. -Glenn

Glenn's Tweet linked to an interesting article at The Verge. The Verge's headline reads: *"Google turns to regulators to make Apple open up iMessage"* and their tag line is *"In addition to shaming Apple for not supporting RCS, the search giant has reportedly co-signed a letter arguing that iMessage should be designated a core platform service under the EU's Digital Markets Act."* I read the entire piece and I agree with Glenn's assessment. What Google really appears to want is to force Apple to open iMessage, since today's green bubbles are lame by comparison. I have a text messaging group where one of its five members is an Android user. As a consequence, the entire group is forced out of iMessage into SMS, thus being reduced to this lowest common denominator due to the presence of this one individual. So if Apple were to upgrade the rest of us iPhone people to RCS then the green bubbles would be at parity with iMessage's blue bubbles. But as for opening iMessage? From a technical standpoint I can't see how that's really possible due to the closed security ecosystem iMessage lives within. So the addition of RCS does seem like a clever counter measure designed to take the pressure off of Apple in this regard. And I think that Google should be happy with it. I know I will be.

Leo turns 67

Last week's podcast was titled "Ethernet turned 50"... and since I didn't have anything better to name this week's podcast, I decided to go with "Leo turns 67" since that's happening tomorrow, November 29th. And even though he's currently sequestered in some far off cave with no Internet or other technology, doubtless contemplating the nature of life, the universe, and everything, you might want to send him birthday wishes which he'll likely discover once he emerges and rejoins the rest of humanity... much as he'll be rejoining us this time next week.