Apple Says No

Description: This week we have our first sneak peek at "ValiDrive," the freeware I decided to quickly create to allow any Windows user to check any of their USB-connected drives. There's been another sighting of Google's Topics API; where was that? Has Apple actually decided to open their iPhone to researchers? And what did some quite sobering research reveal about our need to absolutely trust each and every browser extension we install, and why was that sort of obvious in retrospect? We're then going to entertain some great feedback from our amazing listeners before we conclude by looking at the exclusive club which Apple's just-declared membership made complete.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-938.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-938-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. As always, a fact-filled fun tour through the world of security. We'll talk about Steve giving in to pressure, mostly from me, and developing a free app that will test the integrity of your hard drive. The details on ValiDrive coming up in just a little bit. Then we'll also talk about why you've got to be really careful about the browser extensions you use, more careful than anybody ever realized. And finally, Apple's response to the demand that they open up their encryption. Steve has all the details, coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 938, recorded Tuesday, September 5th, 2023: Apple Says No.

It's time for Security Now!, the moment you've been waiting for all week long. Here he is, the man of the hour, the man of the week, Mr. Steve Gibson. Hello, Steve.

Steve Gibson: Hello, Leo. Well, it is great to be with you for the beginning of September. And aren't we about to change our clocks? We had to wait till after...

Leo: I think it's October now; right?

Steve: They keep moving...

Leo: It has to be after Halloween. Actually it's November because the sugar industry...

Steve: Oh, that's right, because we want to keep the...

Leo: Yes.

Steve: That's right, we want to keep the light on for the kiddies while they trick-or-treat.

Leo: The sugar industry said, no, please change it after Halloween, please.

Steve: I don't think anyone trick-or-treats anymore. They go to malls and, like, schools and things to do that.

Leo: Too much stranger danger out there.

Steve: That's right, that's right.

Leo: We don't let them out of the house.

Steve: So we have Security Now! Episode 938 for this beginning of September, the 5th, titled "Apple Says No." And of course you know what we'll be talking about, Leo, because you've been talking about it on your previous podcasts.

Leo: Yep.

Steve: But this week we have our first sneak peak at ValiDrive, which is the freeware I decided to quickly create to allow any Windows user to check any of their USB-connected drives. I'm going to just show some screenshots before we begin deeply. There's been another sighting of Google's Topics API. Where was that? Has Apple actually decided to open their iPhone to researchers? And what did some quite sobering research reveal about our need to absolutely positively trust each and every browser extension we install? And really, why was that sort of obvious in retrospect?

We're then going to entertain some great feedback from our amazing listeners. I got a bunch of great stuff in the past week. And then we're going to conclude by looking at the exclusive club which Apple's just-declared membership has made complete. So I think another great podcast for our listeners. And we have a terrific Picture of the Week, too.

Leo: That's a very Yoda-like riddle you just told. Apple just decided to complete - okay. Well, we'll find out. We'll find out. And now, the moment you've all been waiting for - this is a good one, I did peek - the Picture of the Week.

Steve: So for those who haven't seen it, I am tweeting the pictures now every week to Twitter. So for those 64,000-some followers, you may have seen it. We've got a picture sort of - it looks like sort of a large central park somewhere. And on the left side is a large, wide, paved road sort of running from our foreground well into the distance. To the right side is, you know, the park, the green grass part where you can imagine couples

and their newborns picnicking and frolicking. Well, I presume this is not a busted water main. Maybe it's just been a long rain. But this is all flooded; right? So the road on the right is completely flooded. The water is up over the curb. It's poured into the green grassy area. So this is all flooded. Except in the center. In the center foreground is the drain.

Leo: Which is an island of green in that...

Steve: Which is above the water level, all the water. Just there, it's like the one dry spot in the entire picture is the drain into which no water is flowing because, as you said, it's an island. And so I gave this picture the caption: "Well, those civil engineers were too expensive, so they decided to hire the mayor's nephew."

Leo: I can do a drain. It's easy.

Steve: Yeah. That's not a problem. It's dumb. It's just a park. I'll just put a hole here.

Leo: Simple, yeah.

Steve: Oh, yeah. Yeah, well, you get what you pay for.

Leo: Wow.

Steve: So anyway, another great picture, thanks to our great listeners, who provided this. And, you know, Leo, I was thinking about your reaction during our podcast last week to this problem of bogus mass storage drives and the need for a quick nondestructive test for both new and existing drives.

Leo: When we left the show last week you said, well, it'll be in the next SpinRite. And I thought, well, that's reasonable. But I said maybe you could write something. Talk about a quick reaction. I saw your tweet. And it was like, wow.

Steve: Well, in fact it was - you were on MacBreak, no, you were on This Week in Google. I happened to have it on in the background while I was writing code for this.

Leo: Yeah.

Steve: And so I tweeted that I decided to briefly pause work on SpinRite 6.1 because this just seemed like too big a need, and something that I had all the pieces in place to do. So in the show notes I've got four actual screenshots made by what I'm calling ValiDrive, with a tip of the hat to Paul Holder, who came up with...

Leo: Excellent name. Excellent.

Steve: Came up with the name, yeah. And sure enough, I mean, what we're learning, I released it to my testing gang two days ago, on Sunday. And they immediately jumped on it. Remember the drive I told the story about is in the lower right. That's the one where it was a 256MB drive which SpinRite rejected, and that's how we stumbled on this whole problem was that SpinRite was checking the very end of the drive, just to make sure that its own drivers were working correctly with this drive. And it was being rejected. And the guy, millQ, has god knows how many USB drives, but this was the only one that was having a problem. So we drilled into the problem and discovered that my belief is, because it was so old, and it's only a 2:1 ratio between good and bad size, it only actually has exactly 128MB rather than 256MB. And yes, I'm saying megabytes, not gigabytes. That's how old...

Leo: That is old, yeah.

Steve: ...that this drive is. So I don't think it would behoove any cheater, you know, any fakery, to only cut the drive in half. Typically you see like 2TB that only have 32GB because that's enough to hold the file allocation table at the front of the drive to make the drive look valid when in fact it isn't. You know, it'll store only 32GB. And if you try to use more than that, the drive says, yeah, I'm storing it, no problem. Well, until you try to read it back. So anyway, the little piece of freeware is working. It's being tested. I'll finish it up during this week, and I'll have an announcement I'm sure while you're back away, Leo.

Leo: You know what, I kind of knew that you would do this because I could see you already thinking about what you would have to do to make it work. Well, let's see. We'd have to fill - we'd have to write to every sector. Oh, we've got to worry about maybe being fooled - you were already working on it before the show even ended. I knew you were going to do this. I'm so glad you did. Now, when do you think you'll be able to release it?

Steve: A couple days from now.

Leo: Oh, quick, good. This will be hugely valuable to people.

Steve: Yeah. Actually, some of the people who have been testing it said that they think it's going to be GRC's number one most downloaded freeware. Now, to be fair, though, the DNS Benchmark has more than 8.5 million downloads.

Leo: Wow.

Steve: With 2,000 new downloads every single day. So it's going to be tough for little ValiDrive to catch up to the DNS Benchmark. But anyway, I'm glad I did it. I'll be back to SpinRite by the end of the week. We'll have a very useful piece of freeware that everyone will be - every Windows person will be able to...

Leo: A must-have, yeah.

Steve: And I did tweet these screenshots, and there were some reactions from people saying, hey, what about if you're colorblind because...

Leo: Oh, yeah. Red green.

Steve: Because I'm using color in order to show valid storage, read errors, write errors, and missing storage. I will have a monochrome option that will put different shape black-and-white characters in the cells in order to accommodate people who don't have full normal color discrimination ability. So anyway, it's on its way. And so I'm glad I did it, and we'll have a new gismo here shortly. And yes, Leo, your intuition was correct.

Leo: I'm just, I mean, this is right up your alley. I saw on GitHub there are a few people who've written C++ programs to do this, but who are they? And it's command-line and blah blah blah.

Steve: And they're destructive. They wipe out your data. You're able to run this on an existing drive.

Leo: Well, that's not good.

Steve: No.

Leo: Yeah. So you read the data, save it, and then write to it, and then write it back.

Steve: Yeah.

Leo: You are good.

Steve: And then I read it again, and then I rewrite it and reread it to make sure that it got written back correctly.

Leo: And I think probably the reason people don't do that is that it must take some time to do that on a 2TB drive or more.

Steve: Yes. And in fact somebody commented, they had one of those destructive things. And I said, well. And he said, you know, yours is taking maybe, I think it was like twice as long. Well, twice as long is a few minutes. Actually, many times it completes in 15 seconds. And what many people are finding - oh, yeah, it's...

Leo: That's fast.

Steve: It's only a few seconds. And many people are noting that what they're seeing, even when they get a field of green, is the speed and the hesitation. When I first started

using it myself on some of mine, I thought, wow, it was like freezing. Well, what is it doing when it's freezing? So there's also little read and write lights that flicker back and forth while it's running as it's counting down the number of tests remaining. So you can really see, like, what's going on. And so even if your drives are good, it helps you spot problems because put in a high-quality drive, it just zips along. Put in a cheesy Walmart drive, and it's like [mimicking bad drive]. It's like, god, what is wrong?

Leo: Well, that tells you something, too. Wow.

Steve: That's, yeah, that's going to be a goody. Okay. So...

Leo: Thank you, Steve, from all of us, thank you very much.

Steve: Well, it's going to be fun. So Google's Topics is coming to Android Apps near you. And maybe this is a failure of imagination, but it hadn't occurred to me that Google's Topics system, which we've talked about now several times, might not only apply to websites. In retrospect, it's so obvious that Google would also be assigning Topics to Android apps; and that advertisers, and apparently other apps, would also be able to query the device's local Topics API to obtain a few bread crumbs of information about, you know, the person who's using the app.

One of our listeners was kind enough to share a screen capture of what had just popped up on his Android phone. Under the headline "New ads privacy features now available," the screen reads: "Android now offers new privacy features that give you more choice over the ads you see." This is very much like the text that the Chrome users got. Android notes topics of interest based on the app you've used recently. Also, apps you use can determine what you like. Later, apps can ask for this information to show you personalized ads. You can choose which topics and apps are used to show you ads. To measure the performance of an ad, limited types of data are shared between ads.

Okay. So, you know, we know I'm a fan of Google Topics. I understand it, and I've tried to carefully explain the way it works, which is admittedly somewhat convoluted and open to misunderstanding because Google is trying to slice this thing very close. Google wants access to some limited information about the users of their Chrome web browser and now their Android phones in an environment where users have become skittish about privacy and tracking.

And, you know, we all recognize the tradeoffs; right? If websites insist that they receive some revenue, more revenue when advertisers have some information about their visitors, and advertisers are determined to obtain that information by any means possible, then Topics is the cleanest tradeoff compromise I can imagine. If, eventually, once legislation catches up, Topics replaces all other forms of tracking and information gathering about me, then I'm all for Topics. You know, that's a tradeoff that makes sense.

So I suppose I shouldn't feel any differently about Topics being extended outside of the browser. If a user wants to use advertising-supported smartphone apps, then I suppose the same logic applies there; right? And, you know, I should explain that I personally cannot, and do not, tolerate in-app advertising. You know, if an app is something I want to use, please allow me to send a few dollars your way to turn off its ads. I will do that happily. Otherwise, I don't care how great it is, nothing is that great. I will delete any app whose advertisements I am unable to silence. But that's just me.

What we see all around us, pervasively, is that advertising works. And Leo, as you noted last week, even if I refuse to click on some advertisement, the brand being advertised has been planted in my brain. That's out of my control. And the fact is we live in an advertisement-supported world. This podcast is underwritten by a few high-quality enterprises that are willing to pay to make our listeners aware of their presence and offerings. That's all they ask.

So Google is extending Topics beyond Chrome and into the underlying Android platform. That only makes sense, really, in retrospect, as I said. But I'm certain that Google will also allow Topics, as they do on Chrome, to be completely disabled if that's what its user chooses. So again, props to Google for that. I am 100% certain that before offering that full disablement option they thoroughly, and not just once, tested what I often call "the tyranny of the default."

So they absolutely know that nearly 100% of Android phone users will never know nor bother to disable their Android device's local Topics feedback. And they also know that by allowing their more knowledgeable Android users - like every listener of this podcast - the option to disable Topics, by giving them the option to disable Topics they're retaining and comforting those users who would be upset by this local, albeit extremely mild, smartphone surveillance. And, you know, if ads in apps are inevitable, if they're supporting the apps that you're using, then you might as well get as relevant an ad as possible, if you're got to have one anyway. So anyway, I thought that was interesting. It just hadn't occurred to me that Topics would be something that Android at large did more than just Chrome. But, you know, it only makes sense.

I've often bemoaned the problem researchers have with helping Apple to find their own platform's security shortcomings because the platform is so thoroughly and utterly locked down. But last week I was reminded that, for the past four years, since 2019, this has not been strictly true. Last Wednesday's blog post from Apple's Security Research was titled: "2024," which is, you know, next year, "2024 Apple Security Research Device Program now accepting applications." And this window is one month, so jump if you're interested. We've talked about this before, but I've been overlooking this truly marvelous exception to Apple's "no one gets in" stance.

In their Security Research's, Apple's Security Research's overview of this, they explain: "iPhone is the world's most secure consumer mobile device" - and I would agree with that completely - "which can make it challenging for even skilled security researchers to get started." Or actually to get anywhere. They said: "We created the Apple Security Research Device Program to help new and experienced researchers accelerate their work with iOS. Now accepting applications through October 31st, 2023. Apply below." And then under "How it works," they remind us.

They said: "The Security Research Device (SRD) is a specially fused iPhone that allows you to perform iOS security research without having to bypass its security features. Shell access is available, and you can run any tools, choose your own entitlements, and even customize the kernel. Using the SRD allows you to confidently report all your findings to Apple without the risk of losing access to the inner layers of iOS security." And I guess that means that the phone won't suddenly lock you out. Anyway, they said: "Plus, any vulnerabilities that you discover with the SRD are automatically considered for Apple Security Bounty." Which, you know, has ranged up to $100,000 in some cases.

Then elsewhere they elaborate this a bit. They said: "iPhone is the most secure consumer mobile device on the market, and the depth and breadth of sophisticated protections that defend users can make it very challenging to get started with iPhone security research. The central feature of SRDP" - which is the program - "is the Security Research Device, the SRD, a specially-built hardware variant of iPhone 14 Pro that's designed exclusively for security research, with tooling and options that allow researchers to configure or

disable many advanced security protections of iOS that cannot be disabled on normal iPhone hardware in the hands of users.

"Among other features, researchers can use a Security Research Device to install and boot custom kernels; run arbitrary code with any entitlements, including as platform and as root outside the sandbox; set Non-Volatile RAM variables; install and boot custom firmware for Secure Page Table Monitor and Trusted Execution Monitor, which are new in iOS 17." And they said: "Even when reported vulnerabilities are patched, the SRD makes it possible to continue security research on an updated device. All SRDP participants are encouraged to ask questions and exchange detailed feedback with Apple security engineers."

And in another place, explaining about eligibility for the program and some constraints, they said: "The SRD is intended for use in a controlled setting for security research only. If your application is approved" - that is, your application to join the program - "we," said Apple, "will provide you an SRD as a 12-month renewable loan. During this time, the device remains the property of Apple." So, you know, you don't have to buy it. They're saying, "Here, but it's still ours." "The SRD is not meant for personal use or daily carry, and must remain on the premises of program participants at all times. Access to and use of the SRD must be limited to people authorized by Apple.

"If you use the SRD to find, test, validate, verify, or confirm a vulnerability, you must promptly report it to us and, if the bug is in third-party code, to the appropriate third party. Our ultimate goal is to protect users, so if you find a vulnerability without using the SRD for any aspect of your work, we'd still like to receive your report. We review all research that's submitted to us and consider all eligible reports for rewards through the Apple Security Bounty.

"Participation in the Security Research Device Program is subject to review of your application. To be eligible for the Security Research Device Program, you must have a proven track record of success in finding security issues on Apple platforms, or other modern operating systems and platforms." So, you know, have some pedigree. "Be based in an eligible country or region."* And there's a little asterisk on that we'll get to in a minute. "Be the legal age of majority in the jurisdiction in which you reside," they said, "18 years of age in many countries. And not be employed by Apple currently or in the past 12 months.

"To enroll as a company, university, or other type of organization, you must be authorized to act on your organization's behalf. Additional users must be approved by Apple in advance and will need to individually accept the program terms." Now, where they said "be based in an eligible country or region,"* I looked at the bottom of the page where the asterisk was referring, and there was a very long list of qualifying countries. Notably absent, and they were alphabetized, so it was easy to spot, were China, Russia, and North Korea.

Leo: Yeah, there's a good group to be in.

Steve: So sorry there, Vladimir. You and your minions will not be authorized. You may have some underhanded, surreptitious way of getting your hand on a phone. You know, and Leo, I was thinking, I wouldn't be at all surprised if these things are not geolocked also.

Leo: Oh, I'm sure they are. Oh, hell, yes.

Steve: I'll bet there's some tethering on this thing.

Leo: Apple knows how to do this stuff very well, believe me.

Steve: Yeah, yeah.

Leo: This is good. This is great that they're doing this.

Steve: It is so cool.

Leo: I presume that one of the things is - because this is always the complaint of researchers. They couldn't get into these phones to know whether they were compromised or not. I mean, that complaint continues because a normal phone you still can't get into to know if it's compromised or not. But at least they can research zero-days and so forth.

Steve: Yeah. I'm sure it reflects many prior years of researchers complaining about exactly that; right? That they're just like, well, we'd like to help Apple. There's all this cool tech in there. And oh, by the way, it does seem to be having lots of problems with zero-days. Maybe we could find some of those, but we can't get in. So anyway, I wanted to correct the record of my recent statements that it just wasn't possible to conduct meaningful research into iPhone security. Bravo Apple. Once again, I think they're doing the right thing.

Leo: Has there been any reaction from the security researchers on this? Like is this what they want? Is it enough?

Steve: Oh, yeah, yeah, yeah, yeah, yeah. I mean, the problem is Apple said that there's a limited number of these that they want to have floating around, you know, they aren't going to be able to honor every request. But in some of the text I noted that even universities like security education programs could qualify.

Leo: Ah.

Steve: So students at universities could have access to these special, you know, iPhones.

Leo: I'm sure Matthew Green is applying right now; you know? That's great, yeah.

Steve: Yeah.

Leo: And they do really good work because they're not constrained by commercial necessity. So they can spend months trying to break into this stuff. It's almost always out of universities that the toughest hacks come, like...

Steve: All the research that we talk about. As a matter of fact, we've got some right here.

Leo: Oh.

Steve: This is on the need to REALLY, in all bold caps, trust every web browser extension we install. This sobering research has recently come from researchers at, what do you know, the University of Wisconsin-Madison. As part of their exploration into what a malicious web extension can and might do, even today when operating under the more restrictive Manifest V3 protocol that Chrome introduced which has since been adopted by most browsers, these researchers discovered that their proof-of-concept extension is able to steal plaintext passwords from a website's HTML source. And wait till you hear which websites were vulnerable, found to be vulnerable.

Thanks to the unrestricted access to the DOM tree - that's the web page's Document Object Model which is organized logically as a tree structure. It describes the document. But more recently, I mean, everything that you see on the page is in this Document Object Model. It's just like it's a - it is the web page.

Leo: And stuff that you don't see on the page, more importantly; right?

Steve: Yes.

Leo: Hidden CS, yeah.

Steve: Yes. So that's what the browser uses to drive its renderer and all of the scripting that it also runs. So they demonstrated that the coarse-grained permission model under which we're all now operating, which also covers browsers' text input fields, violates the principles of least privilege. They found that numerous websites with millions of visitors - and I'm not going to stomp on the news of which websites. We'll get there in a minute. But, boy - including some Google and Cloudflare portals, store passwords in plaintext within the HTML source of their web pages - just sloppiness - thus allowing for their ready retrieval by extensions.

So their research paper is titled: "Exposing and Addressing Security Vulnerabilities in Browser Text Input Fields." This is what they explain in their paper's Abstract. They said: "In this work, we perform a comprehensive analysis of the security of input text fields in web browsers. We find that browsers' coarse-grained permission model violates two security design principles: least privilege and complete mediation. We further uncover two vulnerabilities in input fields, including the alarming discovery of passwords in plaintext within the HTML source code of web pages.

"To demonstrate the real-world impact of these vulnerabilities, we design a proof-of-concept extension, leveraging techniques from static and dynamic code injection attacks to bypass the web store review process." In other words, they snuck it in. "Our measurements and case studies reveal that these vulnerabilities are prevalent across various websites, with sensitive user information, such as passwords" - but not restricted to, we're talking social security numbers, credit card numbers, you name it - "exposed in the HTML source code of even high-traffic sites like Google and Cloudflare. We find that a significant percentage (12.5%) of extensions possess the necessary permissions to

exploit these vulnerabilities and identify 190 extensions that directly access password fields.

"Finally, we propose two countermeasures to address these risks: a bolt-on JavaScript package for immediate adoption by website developers, allowing them to protect their sensitive input fields; and a browser-level solution that alerts users when an extension accesses sensitive input fields. Our research highlights the urgent need for improved security measures to protect sensitive user information online."

Okay. Now, the Manifest V3 protocol prohibits extensions from fetching code hosted remotely that could help evade detection, and prevents the use of eval statements that lead to arbitrary code execution. However, as the researchers explained, Manifest V3 does not introduce a security boundary between extensions and web pages, so the problem with content scripts remains.

To test Google's Web Store review process, they created a Chrome extension capable of password-grabbing attacks, and then uploaded it to the extensions repository. Their extension posed as a GPT-based assistant that can capture the HTML source code when the user attempts to login on a page by means of a regex; abuse CSS selectors, you know, the web page CSS, to select target input fields and extract user inputs using the .value function; and perform element substitution to replace Javascript-based obfuscated fields with unsafe password fields, all of which they explain in their research doc.

The extension does not contain obvious malicious code, so it evades static detection and does not fetch code from external sources, which of course would be dynamic injection. So it is fully Manifest V3-compliant. This resulted in the extension passing the review, being accepted on Chrome's Web Store. So the security checks failed to catch the potential threat, which in this case was very real.

Now, of course, the researchers followed strict ethical standards to ensure no actual data was collected or misused. They deactivated the data-receiving server component while only keeping the element-targeting server active. Also, the extension was set to "unpublished" at all times so that it would not gather many downloads. And it was promptly removed from the store following its approval. That is, as soon as they saw that it got in and were able to verify that this thing was able to slip by.

Okay. Subsequent measurements showed that from the top 10,000 websites, roughly 1,100 - that's where that 12.5% figure came from - are storing user passwords in plain text form within the HTML Document Object Model. And extensions script have access to the Document Object Model, thus access to plaintext passwords. So this is a fundamentally insecure design. The designers of those 1,100 websites, that is, 1,100 out of the top 10,000 that these guys looked at, the designers of those websites either wrongly assume that the contents of their page's document object model are inaccessible, which is not true, or they never stopped to consider it.

In addition, another 7,300 websites from that same set of the top 10,000 were found vulnerable to DOM API access and direct extraction of the user's input values. Several of those, including widely used ad blockers and shopping apps, boast millions of installations. So this thing is widespread. Okay. Now, is everybody sitting down? Notable websites lacked the required protection and are thus vulnerable right now. Those include gmail.com.

Leo: Whoops. Oh, nobody uses that, thank goodness. Holy cow.

Steve: Which has plaintext passwords stored in HTML source code.

Leo: What?

Steve: Cloudflare.com, plaintext passwords in HTML source code.

Leo: Passwords to what?

Steve: The users' passwords.

Leo: What?

Steve: Are available in plaintext in the HTML source. Facebook.com, user inputs can be extracted via the DOM API. Citibank.com, user inputs can be extracted via the DOM API. Irs.gov, social security numbers are visible in plaintext form on the web page source code. Capitalone.com, SSNs are visible in plaintext form on the web page source code. Usenix.org, same thing, social security numbers. Amazon.com, credit card details including the security code and ZIP code are visible in plaintext form on the page's source code, available to all extensions. Yes, it is that bad.

Leo: Holy cow.

Steve: The V3 Manifest was a tradeoff. Due to the way the industry's existing websites and popular extensions had been coded, further limiting extension use would have broken too much existing code, so it wasn't done. When a Google spokesperson was asked about this, they confirmed that they're looking into the matter - you think? - and pointed to Chrome's Extensions Security FAQ that does not consider access to password fields to be a security problem "as long as the relevant permissions are properly obtained." Right. Let's hope this gets fixed soon.

Leo: I can understand, though, why you need access to the DOM if you're an extension. I mean, that's kind of what an extension does.

Steve: That's what you do.

Leo: And Gorhill has complained about Manifest 3 making it impossible to do uBlock Origin because even the little restrictions that it offers make it hard to do adblocking. So I understand, boy, this is a problem.

Steve: It is a Catch-22, Leo, yes.

Leo: Yeah. The web was really not designed to be secure. I mean, that's what we're fundamentally seeing.

Steve: No, no. And we tried to turn web into web apps as if they're the same. And you know, we stretched our browsers mightily in order to do that. In the Takeaways section 5.3 of their paper, they write: "This is a systemic issue." They said: "Our measurement studies on the top 10K websites show that we could extract passwords from all the login pages with passwords. The widespread presence of these vulnerabilities indicates a systemic issue in the design and implementation of password fields."

And they talk specifically about password managers. Now, think about that. We take it for granted; right? But any and all password managers must by design be a third-party extension which has direct access to any website's password fields. They said under "Role of Password Managers: The widespread use of password managers may partially explain the prevalence of vulnerabilities, where password values are obscured, but can be accessed via JavaScript. These tools enhance the user's experience by automating the process of entering passwords, storing the encrypted passwords, and later auto-filling these fields when required. This functionality reduces the cognitive load on users and encourages the use of complex, unique passwords for each site, thereby enhancing overall security.

"However, for password managers to function effectively, they require access to password fields via JavaScript. This necessity creates an inherent security vulnerability. While the password fields may appear obscured to users, any JavaScript code running on the page, including potentially malicious scripts, can access these fields and read their values. This interaction between password managers and these vulnerabilities presents a tradeoff between usability and security. While password managers improve usability and promote better password practices, their operation necessitates JavaScript access to password fields that inherently creates a security risk."

Essentially, these guys just demonstrated that you don't even have to be a password manager to obtain password manager level access. And they were able to successfully sneak their universal password extraction extension code past Google's incoming filters without any trouble. So their 26-page paper is marvelously clear, and none of this stuff is fancy or complex. Its content would be entirely accessible to anyone familiar with modern web page construction and operation. Any of our listeners who are responsible for the design of their organization's secret-accepting web pages might well benefit from making sure their own sites are protected. I've included the link to the research PDF for anyone who is interested. And to improve its availability, it's also this week's GRC shortcut, so you can find it at grc.sc/938, grc.sc/938. And the PDF link is also in the show notes.

So this is important. Again, kind of in retrospect, it's like, well, duh. Of course password managers need to be able to do this. It turns out it's not just password managers that can. They found 190 existing extensions that had this capability. And I don't think there's 190 password managers out there. So a lot of apps are doing it, and they created one and slipped it past Google and could do it, too.

Leo: Wow.

Steve: Yeah.

Leo: Amazing. All right. Let's close the loop.

Steve: So, yeah. Peter Gowdy tweeted. He said: "Hi, Steve. I took note of your Global Privacy Control episode, and just added the Privacy Badger extension to Vivaldi. There

doesn't seem to be a solution for mobile that I could find, even in Firefox mobile. Is there a mobile Global Privacy Control solution that you know of?"

And I'm not surprised that support is still lagging since, as we know, change always comes much more slowly than we expect or hope or wish. But I'm pretty sure that, once additional legislation appears, and we know that it exists in three states beginning with the letter "C," and it is spreading both here in the U.S. and also in Europe, I think we can assume that the Global Privacy Control switch will become a universal feature of browsers. Again, it'll just take some time, but we'll get there.

Longstanding friend of the show Alex Neihaus, he tweeted: "Hi. Re: Microsoft's 'doesn't care' about the STS issue" - remember that's the secure time setting, whatever it was that that stood for, that we've talked about several times, you know, using time in the TLS handshakes in order to set the clocks. He said: "Despite it being known for decades as being unreliable." He said, this is Alex saying: "I don't think they deliberately or maliciously mis-engineered the feature. I think they just didn't do the research. Most people think that Microsoft developers are first-rate. But management there has reduced costs, which has encouraged use of offshore and lower-experienced engineers. Unlike us Boomers, devs today rarely go as deep as you did to understand the issue. The engineer was simply and probably impatient, saw the field in the hello message, and went for it.

"You're most likely correct that they don't want to admit they're wrong because it raises the question I am posing here about their engineering prowess. So it was most likely a combo of poor engineering and design, coupled with hubris today, that prevents them from recognizing the deeper issue."

Leo: I also want to point out that - you know who Alex Neihaus is; right?

Steve: Yeah.

Leo: He was a guy who - he was the first club member, bought the first ads on this show with Astaro. And we thank him so much because he has put us on the map, thanks to Alex. Well, you put us on the map, but Alex helped do it.

Steve: And I don't disagree with anything Alex wrote.

Leo: He's sharp.

Steve: Yes. Everyone here knows how infinitely tolerant I am of mistakes. You know? They happen, and anyone can make them. And I'm sure one of these days I'll make a big one, so I'll be glad that I've always been saying this. You know, there are many...

Leo: Getting defensive, I understand.

Steve: You know, I'm crossing my fingers. I'm as careful as I can be. But there are many adages that begin with, "If you're not making mistakes, dot dot dot." You know? And typical endings for that are "then you're not trying hard enough." Or "then you're not making decisions." But the most famous one appears to be, "If you're not making mistakes, then you're not doing anything." The point of all of these is the clear

acknowledgement that mistakes are a natural and unavoidable consequence of our interaction with the world. You know? You do something. The feedback from what you did, which was presumably not what you expected, informs you that a mistake was made somewhere. So with the benefit of the new information, you correct the mistake.

My entire problem with Microsoft is that we see example after example, this being just the latest, where this feedback system appears to be completely absent. Whether it's well-meaning security researchers informing Microsoft of serious problems they've found, or their high-end enterprise customers for seven years telling them: "Hey, my Windows server clocks are getting messed up, and it's really screwing up everything." Microsoft no longer appears to care. And to Alex's point, though coming at this from a different angle, I think this all boils down to simple economics: Caring costs money. And Microsoft no longer needs to care because not caring costs them nothing.

That's really the crux of it today. There's no longer any sign of ethics for ethics' sake. You know, that's long gone. It's simply about profit. We're all aware of the expression "Don't fix it if it's not broken." Microsoft has extended this to "Don't fix it even if it is broken." And what we get is a system that, you know, it mostly works. It could be better. But I guess it's good enough. And, you know, again, I always want to add the caveat, I'm sitting in front of Windows. I love my Windows. I don't ever want to have it taken away from me. So, you know, I want it to be as good as it can be. But darn it, it could be better.

Leo: I'm coming over, and I'm taking it away from you. No more.

Steve: Listener Michael, he tweeted: "Dear Steve. Just listened to another awesome Security Now! from you. I have a question about VirusTotal, if I'm not bugging you. What's the probability that it could have false positives? I'm asking specifically because of a program I've used since Windows 7 called Winaero Tweaker, which lets me customize Windows so that it's more usable and easier. It's not flagged by Windows Defender, nor by Malwarebytes. I guess what I am asking is, in your opinion, is Winaero Tweaker okay to use, and is VirusTotal ever wrong? Thank you. Michael."

Okay. So if we were to use majority voting, then I have never seen VirusTotal make a mistake. But if you require zero detections in order to be comfortable, you know, out of the 66 different AV scanners that VirusTotal polls, then that's actually somewhat rare. When using any modern AV scanner it's important to understand today's context. The original AV scanners operated by spotting specific code that had been previously identified as being part of a piece of specific known malware. This was quite effective and rarely generated false positive alarms because the AV was actually finding the malware that used some specific code. But then of course malware evolved to avoid direct code recognition by scrambling itself, encrypting itself, compressing itself, and even becoming "polymorphic." Remember the polymorphic viruses which self-rearranged in order to appear as unique as possible from one instance of itself to another.

Then later, as a consequence of this back-and-forth cat-and-mouse game that malware was playing with AV scanners, the scanners began looking at the operating system functions that a program was using and judging whether some things a program like the OS API calls a program might request fall outside of some arbitrary norm. For example, maybe a DNS lookup. There's nothing malicious about a program doing a DNS lookup. But most programs that want to connect to a remote resource just issue an HTTPS request, and the operating system performs the DNS lookup itself to obtain the connection's IP address. So in this example, any program that wanders away from an arbitrary tightly defined norm might trigger a false positive alert, not because it did

anything wrong, but simply because it was found to be doing things that someone judged as unusual.

The final outcome of decades of this back-and-forth contest between AV and malware is the increasing use of a specific program's reputation. The way things have turned out, reputation is the ultimate source of trust. So in that sense things are the same in cyberspace as they are in the real world. You know, we trust people who have earned a reputation. We have the ability to easily obtain unspoofable cryptographic signatures of specific code. This means that for all intents and purposes it's impossible to change the code in any deliberate way without also changing the code's resulting cryptographic signature. So without actually knowing anything about a program, the persistent connectivity provided by the Internet allows a program's use, and its signature, to be tracked over time. If a program is out and about for a few months without anyone complaining or it causing anyone any trouble, then that code, as identified by its unique signature, will have established a good reputation and will therefore become trusted.

The trouble is that any newly created code will have an unknown signature that won't have had any chance to earn a reputation. And as a creator of new utilities, this is a problem I run into all the time. Two days ago, last Sunday, the first people to download the completely harmless and freshly assembled ValiDrive Windows application had Windows 10 immediately quarantining the download, complaining that it was "not commonly downloaded." Well, yeah, no kidding.

Leo: It's brand new.

Steve: It had never been downloaded before. So here we had this brand new, never-before-seen cryptographic signature, and Windows said, whoa, what's this? Where did this come from? Off with its head.

So to make matters worse, actually, part of it was my fault. I was in a hurry to get the code into everyone's hands, so I hadn't stopped to digitally sign the executable file with GRC's code-signing certificate. As soon as the first several complaints came in, I did that. And now things appear to have calmed down since. GRC has a spotless reputation since we've never had an incident of any kind. But even so, code-signing certificates do get stolen. You know, mine are all locked up in hardware security modules, so at least they can't be stolen easily. You know, you have to have physical proximity, and that's unlikely to happen because Level 3 is behind multiple barriers and guards and cameras and alarms and everything.

So anyway, but the point is, in general, certificates do get stolen. So just being signed by someone, even with a perfect reputation, isn't 100% assurance. I did just check ValiDrive with VirusTotal, and there were three false positive "detections" after querying 66 antiviral systems. Cybereason, Cylance, and Trapmine didn't like it, but no one else had any complaints. So as I said, the majority voting, I've never seen any of my stuff objected to by more than a handful. But anyway. So it certainly can happen.

As for Winaero Tweaker, I just grabbed a copy of the setup executable from Winaero.com, which is the publisher, and I dropped it onto VirusTotal. And I received a 100% clean bill of health, with VirusTotal saying that zero out of 43 scanners found it to be suspicious. Now, I noted, however, that the executable program, the setup program, was not signed, which would make me suspicious and uncomfortable since it's almost becoming required these days. A digital signature on executable content is something I always check for now. And needless to say, I always and only obtain such programs, especially if they are unsigned, from the original website source. You don't want to get it from somebody who's like, oh, yeah, we also offer it for download.

But for what it's worth, it's version 1.55, which I just downloaded directly from their site. Other than it being not signed, it looks fine. I did note that it was published in June, so if Michael had grabbed it shortly after its publication, also not being signed, then he could have found that VirusTotal or whatever hadn't had a chance yet to get to know it, to have other people upload it and say what do you think, scan this for me. At this point it looks like it's fine. But anyway, the point is, can it false positive? Yes, unfortunately, happens to me all the time. You ought to see that anything that's been around for a few months will have acquired a reputation, and that reputation now is really the only shield that a program has. So that's the current status of AV scanning.

Rick said: "Steve, on Acceptable Ads and uBlock Origin, how are you doing it? I looked around, but only found this old thread." He provided a link in his tweet. He said: "While it's true that the list it points to is current, Gorhill himself slammed it, though what he's saying about that particular list doesn't seem to apply anymore."

Okay. So I did some digging and refreshing of my memory, and it turns out that I was wrong about uBlock Origin and Acceptable Ads. We discussed all of this after 2014 when it was all happening, but I'd forgotten the details. uBlock was initially developed by a guy named Raymond Hill, better known by his handle of Gorhill, and it was released in June of 2014. The extension relied upon community-maintained blocklists, while also adding some extra features. Not long after that, Raymond transferred the project's official repository to a guy named Chris Aljoudi since he was frustrated with all of the incoming requests. That's sort of not Gorhill's style. Curmudgeon is, you know, we remember him sort of fondly along with John Dvorak.

It turns out that Chris, the guy who obtained the repository, was somewhat less than honest and respectful. He immediately removed all credit from Raymond Hill, portraying himself as the uBlock's original developer. And he started accepting donations showcasing overblown expenses to turn the project into a profit center. Rather than development, Chris was focused more on the business and advertisement side, wanting to milk uBlock for all it was worth. Consequently, Gorhill decided to simply continue working on his extension. But that unfortunately resulted in a naming collision where Chrome saw Chris' uBlock as being the original, and Gorhill's as being the interloper. So Gorhill lost, and Chrome yanked his from the extension repository.

Thus was born uBlock Origin, and here comes the difference that matters: The original uBlock worked with the Acceptable Ads policy and still does. But Gorhill, being Gorhill, wasn't interested in making any exceptions to his extension's ad blocking, especially when exceptions to the Acceptable Ads policy had the reputation of being available to the highest bidder. That's not his style at all. Having watched all this drama unfold, at the time we all went with the original extension's original author, since no one felt any particular sympathy for Chris, whose conduct did not appear to be very honorable. And choosing uBlock Origin also meant no longer being able to allow Acceptable Ads, which I would otherwise have no problem doing.

So that's the story. I'm still disinclined to move away from uBlock Origin since I have the strong sense that curmudgeonly Raymond Hill will always have our backs. I feel much less sure of that from Chris Aljoudi, who is the guy behind uBlock, based on his conduct after getting it. So anyway, Rick, you're right. I was wrong about Acceptable Ads. So I'm glad you brought it up and I was able to correct the record.

Somebody whose Twitter handle is #LoveThyNeighbor, he said: "Steve, can the ReadSpeed utility analyze a drive connected via a USB port? It appears that I can only see drives connected and enumerated on the internal IDE, SATA, or SCSI busses of the computer. Is there a way to have ReadSpeed analyze a USB-connected drive? I faithfully listen to Security Now!, so hope to hear a response there as I am not on Elon's

repugnant X site very often. SpinRite user since Version 1 and listener to Security Now since Episode 1. Thanks for all you do."

Okay, LoveThyNeighbor. Unfortunately, the short answer is no. The ReadSpeed DOS utility was a natural offshoot of the early work on SpinRite 6.1. Specifically, I believed that I had nailed down the operation of what would become 6.1's new native IDE, ATA, and AHCI drivers with parallel and serial ATA drives. And we had discovered the surprising slow performance at the highly-used frontend of many SSD devices. Since I thought that ReadSpeed might be broadly useful, it was spun out sort of as an offshoot along the way. So it won't be until we get to SpinRite 7 that USB and NVMe devices will be added to that collection.

That said, I do expect to be dropping some similar freeware in the early days of SpinRite 7's development, since I'll be anxious to get feedback about this emerging software's, you know, SpinRite 7's dual booting over BIOS and UEFI. And as we know I'll be writing it under a new, that RTOS32 operating system. So at that point we will finally be able to talk natively to all drives, which will be, you know, the real benefit going forward for 7 and beyond.

Austin Wise. He tweeted: "Re man-in-the-middle attacks and HTTPS on Security Now! 937." Boy, and it's going to be a long time, Leo, before I am able to live down my comment that I don't think there's anything that wrong with HTTP. There's still some places for it. I don't think any of our listeners agree. But...

> **Leo:** Well, you know, Dave Winer, who is a father of RSS, wrote a very nice piece on his scripting blog that said HTTP was critical to the development of the web because it's easy to implement, and it should still be used in cases where it's safe to use, and there are plenty. And there's a lot to be said for simplicity, especially so that anybody can create a site.

Steve: Yeah, in fact, somewhere I just saw, I don't know where it was, but it was talking about the nature of security. And it said keep it simple, keep it simple, keep it simple. I mean, and we know, how many times have we talked about complexity being the enemy of security?

> **Leo:** It's not hard to generate a Let's Encrypt certificate and make your site HTTPS. I've done it on a number of places.

Steve: Yup.

> **Leo:** But still, you know, there are places where there is no login, there's no passwords in the DOM, there's no reason that you should worry about a man in the middle. And I think those sites should be allowed to continue HTTP. But Google doesn't, that's for sure.

Steve: Well, and Leo, you just raised a perfect example. You could have TLS 1.3 with the fancy longest bit encryption key. And if you've got a funky extension in your web browser, doesn't matter if it's HTTPS, it's still sucking all your passwords and social security numbers and credit card details right off to wherever.

Leo: Holy cow.

Steve: Lower Mongolia. No, I think actually Lower Mongolia I think they're able to get iPhones. But not Russia and China.

Leo: Good, okay. For all the security researchers there.

Steve: We have a listener from Google. He said: "If an attacker is on a local network, like a coffee shop WiFi, they might not need a privileged position" - which of course was what I was talking about last week - "to modify traffic. See ARP spoofing."

Leo: Right. Right. And the widespread availability of things like the WiFi Pineapple make that something even not so sophisticated people can do.

Steve: Yes. He said: "Also the integrity features of TLS are useful even if you trust the network. Random bit flips in packets, like from a misbehaving router, will be detected and cause the connection to terminate."

Leo: That's a good point. That's a good point.

Steve: "This prevents downloading of corrupted data." That's true. Although the underlying protocols do have checksums, which also catch that. And that's why retransmissions are happening often. And he said: "And regarding Leo's mention of companies using HTTP services for internal sites. This is true of Google, which pervasively uses sites like http://go/" - no dot in it, just go/ - "for short links and http://b/" - just the letter b/ - "for bugs and more. But we have a proxy auto config file in our browsers that make sure all such services are sent over a HTTPS proxy to prevent man-in-the-middle attacks. All that said," he wrote, "I hope browsers continue to support HTTP for years to come. It is such a versatile protocol, it would be a shame to lose it completely. Love the show, Austin."

Leo: There you go. That's sensible. That's a sensible answer. I like it.

Steve: Yup. Austin sounds like a Google engineer, and he makes some great points. Way back in the early days of this podcast we spent a good deal of time, Leo, you'll remember fondly, exploring the details of low-level hacks and attacks such as ARP spoofing. For those who don't know, ARP stands for the Address Resolution Protocol. It's the protocol glue that links the 48-bit physical Ethernet MAC addresses of everyone's Ethernet hardware to the 32-bit logical Internet Protocol (IP) addresses which the Internet uses. When Internet Protocol data needs to go to someone, it's addressed to them by their IP address, but sent to their Ethernet MAC address. The ARP table provides the mapping, the lookup between their current IP address and their device's physical Ethernet MAC address. And it's the ARP protocol that's used to populate and maintain all ARP tables that are on the local Ethernet network.

So here's the point that Austin was making: If someone can arrange to interfere with the proper operation of this Address Resolution Protocol, it's possible to confuse the data in the network's ARP tables to misdirect and redirect traffic to the wrong Ethernet address;

and ARP spoofing is able to cause exactly such misdirection. So Austin is 100% correct that in an open WiFi setting it would be possible for an attacker to arrange to intercept traffic by, for example, causing clients of a router to believe that the attacker's MAC address was the address of the network's gateway. And thus they would send all their traffic there instead of to the router.

And indeed the use of TLS and HTTPS would completely prevent any such attacks. Well, actually it would prevent their success. The traffic would still get routed to him, but he couldn't do anything with it because it would be encrypted, and he wouldn't have the ability to intercept it. I mean, it would be much more complex to do so. He'd have to get a certificate that was trusted into the victim's browser somehow and then perform a full proxying of HTTPS and TLS. So again, I think we've pretty much established where things stand. HTTPS, definitely the way to be secure. But it's not enough because you could have a hinky browser extension and still be in trouble. But definitely better than not. Yet HTTP still useful.

JediHagrid tweeted: "My IT Director at work suggested I message you. I found a SQL file containing user and employee information on a website, as well as social media secure tokens. I've tried calling the company, I signed up for LinkedIn premium for the free month in order to message the COO, and I've tried telling Brian Krebs. Maybe I'm thinking too much into this. Maybe it's not that big a deal. You're the last person I'm going to notify, and if nothing happens, then I guess nothing happens. The file is still on their server, and you can see it here."

This was a DM, so it was a private connection between the two of us. He sent me the link. I've redacted the domain name in the show notes. And he said: "The file is" - and again, I redacted the name - "dot sql. You can search it using Notepad++. There are .gov customer emails, people who are applying for jobs' addresses, everything in plaintext. I'm not sure what else to do."

Okay, now, as I said, since the URL he provided which was not redacted was definitely a going national concern, I suggested that he shoot a report off to CISA. They have a web facility for receiving reports of things like this that people find, at https://www.cisa.gov/report. And they also accept email directly at: report@cisa.gov. So for Jedi Hagrid and for all of our listeners, I thought that was useful to share, that you could just send email to report@cisa.gov if you stumble across something that you think that is big enough and worthy of coming to our cybersecurity, information security agencies' attention. And they'll certainly know how to contact the right people and get their attention.

Leo: Having said that, though, he may have just found a pastebin with personal information in it that's part of a hacker site, you know, I mean, this stuff must be all over; right?

Steve: No, it was on their site.

Leo: Oh, it was on the site of the company?

Steve: It's an active running open SQL file that they left exposed.

Leo: Oh.

Steve: Yes.

Leo: That's, yes. And the company didn't want to hear about it. That's...

Steve: They, you know, the receptionist said, "A what file?"

Leo: I don't know what you're talking about, yeah.

Steve: Yeah, uh-huh.

Leo: We get - there is a scam we get. Fairly frequently emails from people say I found an issue on your website, and I would like a bug bounty. Please don't ignore this because it's a hazard. And usually they say something that shows that this is just a generic email, things like, you know, the logins on your website or something are compromised.

Steve: Yeah.

Leo: So, you know, companies get this stuff all the time, and it's a scam. It's a known scam. So maybe that's why they're ignoring it.

Steve: Yeah. In this case they shouldn't because it's an...

Leo: No, it's serious.

Steve: An actual live SQL file on their - you know. And Leo, if I told you the name of the company, you'd be like, what?

Leo: Oh, okay, got it, okay.

Steve: Yeah, yeah.

Leo: I hope CISA pays attention.

Steve: Oh, CISA will pay attention if they get this email.

Leo: Oh, good.

Steve: The CEO will be getting a call from the government. So anyway, two last things, a quickie. Simoncroft08 said: "Hi, Steve. On SN-936 you talked about multilevel cells in SSD storage. This is also a concern with USB thumb drives and SD cards. When used in

industrial applications, SD media may be storing programs controlling machines where errors cannot be tolerated. Industrial environments will have voltage spikes and transients which can flip bits. Consequently, vendors are now selling specifically SLC, single-level cell storage SD cards for this market. The capacities are much smaller because of this, typically around 2GB, but that is plenty for most controllers. Cheers, Simon." He said: "PS, the heuristics story was an ouch. Glad I'm now retired from sysadmin life."

So anyway, I thought this was an interesting angle. The inherently lower reliability of multilevel cell storage is well understood, and in environments where endurance and reliability trumps maximum storage density, SLC has a much better chance of remaining solid. And finally, Martin Biggs brings us the "Duh, why didn't I think of that?" Head Smacker of the Week.

Martin writes: "Hi, Steve. I'm listening to this week's episode." And that was last week, Episode 937 of Security Now!. "You've just described that you can get VirusTotal to check a file before you download it. The problem with this, though, is that if a malicious site recognizes that VirusTotal is downloading the file" - duh - "then the site can serve VirusTotal a safe version. Then once you're secure in the knowledge that VirusTotal says the file is safe, the malicious site can happily serve you the malicious file.

"As I cannot find a way of downloading the file directly from VirusTotal" - that is, once they've received it - he says: "I think the better option is to download the file onto your computer, then upload it to VirusTotal yourself for checking. This way you can be certain that it is checking the same file that you have. Thank you for the podcast, and I am glad that you've decided to continue past those dreaded three nines. Regards, Martin."

And, as I said, that's a head smacker. You know, I don't know for certain what a download query from VirusTotal looks like and whether they may have taken any precautions to mask their downloading. But Martin is 100% correct. As long as the possibility exists that VirusTotal would be receiving and checking a different file than the user downloads, there is no choice but to get it first and provide it to VirusTotal. So nice catch, Martin, and thank you.

Leo: Now, let's talk applesauce with Mr. Gibson.

Steve: So in a rare occurrence, Apple chose to publicly share a mildly threatening private letter it received last Wednesday which was addressed to Apple's CEO, of course, Tim Cook. The letter was from a CSAM (Child Sexual Abuse Material) activist by the name of Sarah Gardner. And Apple must have decided that their best strategy was to get out ahead of this, since they shared Sarah's letter as the preface to theirs, which they also shared in full. And, I mean, shared publicly.

In terms of the way the future is going to take shape, the biggest thing happening today, I think, in the public policy sphere is the debate and struggle over the tradeoff between privacy and surveillance. The devices we all now carry with us 24/7 are capable of providing more of either - privacy or surveillance - than anything ever before. Since this is a significant move on Apple's part, representing a definitive change of stance and policy, I want to share first Sarah's unsubtle letter, followed by Apple's response.

So this is Sarah Gardner. In the letter that was published, it was redacted, but I'm sure it was written to Tim Apple. She said: "Dear Tim. This exact time two years ago we were so excited that Apple, the most valuable and prestigious tech company in the world, acknowledged that child sexual abuse images and videos have no place in iCloud. It was

an announcement that took bravery and vision - we can live in a world where user privacy and child safety can coexist.

"That is why it was so disappointing when you paused, and then quietly killed this plan in December of 2022. We firmly believe that the solution you unveiled not only positioned Apple as a global leader in user privacy, but also promised to eradicate millions of child sexual abuse images and videos from iCloud. The detection of these images and videos respects the privacy of survivors who have endured these abhorrent crimes - a privilege they undeniably deserve.

"I'm writing to let you know that I am part of a developing initiative involving concerned child safety experts and advocates who intend to engage with you and your company, Apple, on your continued delay in implementing critical technology that can detect child sexual abuse images and videos in iCloud. We are asking you to honor your original intention to detect, report, and remove child sexual abuse images and videos from iCloud; create a robust reporting mechanism for users to report child sexual abuse images and videos to Apple.

"We want to alert you to our presence and our intention to take our very reasonable requests public in one week's time. Should you want to discuss our campaign over the course of the next week, or after we've launched, I can be reached at this email address. We welcome the opportunity to discuss these important issues with you and hear what Apple plans to do in order to address these concerns. Child sexual abuse is a difficult issue that no one wants to talk about, which is why it gets silenced and left behind. We are here to make sure that doesn't happen. Kind regards, Sarah Gardner, CEO of the Heat Initiative."

Okay. So in other words, Tim Cook, you have one week to intercede in our intention to start making loud noises about your refusal to do your duty, she says, or else.

Okay. So this Sarah Gardner is the former Vice President of External Affairs for the nonprofit organization Thorn, which apparently they want to be one in Apple's side, which works to use new technologies to combat child exploitation online and sex trafficking. Two years ago, in 2021, Thorn loudly applauded Apple's plan to develop an iCloud CSAM scanning feature. In a statement to WIRED over this recent event, Sarah Gardner wrote: "Apple is one of the most successful companies in the world with an army of world-class engineers. It is their responsibility to design a safe, privacy-forward environment that allows for the detection of known child sexual abuse images and videos. For as long as people can still share and store a known image of a child being raped in iCloud," she wrote, "we will demand that they do better."

Okay. So the following day, last Thursday, August 31st, Erik Neuenschwander, Apple's Director of User Privacy and Child Safety, replied on behalf of Tim Cook to Sarah Gardner. He addressed the letter to Ms. Sarah Gardner, CEO, Heat Initiative: "Dear Ms. Gardner. Thank you for your recent letter inquiring about the ways Apple helps keep children safe." Right. "We're grateful for the tireless efforts of the child safety community and believe that there is much good that we can do by working together. Child sexual abuse material is abhorrent, and we are committed to breaking the chain of coercion and influence that makes children susceptible to it. We're proud of the contributions we've made so far and intend to continue working collaboratively with child safety organizations, technologists, and governments on enduring solutions that help protect the most vulnerable members of our society.

"Our goal has been and always will be to create technology that empowers and enriches people's lives, while helping to keep them safe. With respect to helping kids stay safe, we have made meaningful contributions toward this goal by developing a number of innovative technologies. We've deepened our commitment to the Communication Safety

feature that we first made available in December 2021. Communication Safety is designed to intervene and offer helpful resources to children when they receive or attempt to send messages that contain nudity. The goal is to disrupt grooming of children by making it harder for predators to normalize this behavior.

"In our latest releases, we've expanded the feature to more easily and more broadly protect children. First, the feature is on by default for all child accounts. Second, it is expanded to also cover video content in addition to still images. And we have expanded these protections in more areas across the system including AirDrop, the Photo picker, FaceTime messages, and Contact Posters in the Phone app. In addition, a new Sensitive Content Warning feature helps all users avoid seeing unwanted nude images and videos when receiving them in Messages, an AirDrop, a FaceTime video message, and the Phone app when receiving a Contact Poster.

"To expand these protections beyond our built-in capabilities, we have also made them available to third parties. Developers of communication apps are actively incorporating this advanced technology into their products. These features all use privacy-preserving technology all image and video processing occurs on device, meaning Apple does not get access to the content. We intend to continue investing in these kinds of innovative technologies because we believe it's the right thing to do. As you note, we decided not to proceed with the proposal for a hybrid client-server approach to CSAM detection for iCloud Photos from a few years ago, for a number of good reasons."

And I'll just interrupt here to note that this is the technology that Apple had proposed and the public immediately rejected. There was an almost audible nationwide gasp at the idea of having users' phones containing those hashed libraries of known CSAM images. The idea of that creeped everyone out, and it became clear that it was a non-starter. It was also clear that Apple was truly trying to innovate within the bounds of user privacy.

What's clear is that, if the entire task of somehow recognizing such content with high accuracy cannot be done locally on each user's device, then every single image and video and text conversation must be filtered through some external central authority. And that is clearly far beyond anything that Apple will consider.

So Erik's note continues: "Having consulted extensively with child safety advocates, human rights organizations, privacy and security technologists, and academics, and having considered scanning technology from virtually every angle, we concluded it was not practically possible to implement without ultimately imperiling the security and privacy of our users."

He said: "Scanning of personal data in the cloud is regularly used by companies to monetize the information of their users. While some companies have justified those practices, we've chosen a very different path, one that prioritizes the security and privacy of our users. Scanning every user's privately stored iCloud content would in our estimation pose serious unintended consequences for our users. Threats to user data are undeniably growing globally the total number of data breaches more than tripled between 2013 and 2021, exposing 1.1 billion personal records in 2021 alone. As threats become increasingly sophisticated, we are committed to providing our users with the best data security in the world, and we constantly identify and mitigate emerging threats to users' personal data, on device and in the cloud. Scanning every user's privately stored iCloud data would create new threat vectors for data thieves to find and exploit.

"It would also inject the potential for a slippery slope of unintended consequences. Scanning for one type of content, for instance, opens the door for bulk surveillance and could create a desire to search other encrypted messaging systems across content types (such as images, videos, text, or audio) and content categories. How can users be assured that a tool for one type of surveillance has not been reconfigured to surveil for

other content such as political activity or religious persecution? Tools for mass surveillance have widespread negative implications for freedom of speech and, by extension, democracy as a whole. Also, designing this technology for one government could require applications for other countries across new data types.

"Scanning systems are also not foolproof, and there is documented evidence from other platforms that innocent parties have been swept into dystopian dragnets that have made them victims when they have done nothing more than share perfectly normal and appropriate pictures of their babies.

"We firmly believe that there is much good that we can do when we work together and collaboratively. As we've done in the past, we would be happy to meet with you to continue our conversation about these important issues and how to balance the different equities we've outlined above. We remain interested, for instance, in working with the child safety community on efforts finding ways we can help streamline user reports to law enforcement, growing the adoption of child safety tools, and developing new shared resources between companies to fight grooming and exploitation. We look forward to continuing the discussion. Sincerely, Erik Neuenschwander, Director, User Privacy and Child Safety."

So I think that one statement from Apple entirely explains their, by now, extremely well-considered position, where they said: "...and having considered scanning technology from virtually every angle, we concluded it was not practically possible to implement without ultimately imperiling the security and privacy of our users." In other words, we want to do it. We tried to do it. If we could do it, we would do it. But ultimately we're not willing to compromise our users' privacy and security to make what is ultimately a tradeoff.

Now, the significance of Apple's position, stated as clearly and emphatically as Apple just has, is that it runs directly afoul of the legislation that is currently pending and working its way through the European Union's lengthy ratifying process. Recall that a little over a year ago, when the updated final draft legislation leaked, the Johns Hopkins cryptography professor Matthew Green tweeted: "This document" - meaning the EU proposed draft legislation, which was then finalized. "This document is the most terrifying thing I've ever seen. It describes the most sophisticated mass surveillance machinery ever deployed outside of China and the USSR. Not an exaggeration." And Jan Penfrat of the European Digital Rights advocacy group echoed Matthew's concern. She wrote: "This looks like a shameful general surveillance law entirely unfitting any free democracy."

So in our ongoing coverage of this, we were previously able to quote the official positions of the many various third-party messaging systems. And at the time we noted and commented that Apple was missing from the fray. I think it's safe to say that they are missing no longer, and that collectively the entire now mobile messaging industry has formed a single unified front. The question is, what happens next? What happens when the EU puts their shiny new communications regulations into effect, expected around the end of this year? Who will be the first to blink? Will the regulations be present but unenforced? Will the government or some upstart third party offer surveillance messaging separately? And if they did, would it matter? Would anyone use it? As I noted at the top, this will determine the shape of the future. What'll that shape be?

Leo: Yeah. And of course the question I asked the MacBreak Weekly panel was can Apple continue on its principled stand, or are they going to be forced into some sort of compromise? And I just don't see them surviving. I don't know.

Steve: It's going to be made illegal, Leo. And the question is, what happens when "illegal" hits, like, the iceberg? Does it sink, or does the iceberg melt?

Leo: It's so hard not to become discouraged these days. That's just all I can say. It's just so hard. There are so many areas in which we aren't doing what is obviously the right thing. No one supports child pornography or CSAM.

Steve: No. No.

Leo: That's not the issue.

Steve: I mean, and even hearing her talk about photos of children being raped, you know, it's just gut-wrenching.

Leo: It's intentionally, though, gut-wrenching.

Steve: Yes. Yes.

Leo: And this was what really bothers me about these people is they are, I mean, maybe in their hearts they're doing what they believe the right thing. But they're using really emotional language, and this is a much more complicated thing. Yes, no one's in favor of CSAM. Well, obviously the predators are, but no one else. No normal person is in favor of it. But also I don't think Saudi Arabia should be able to ask for pictures of gay men from Apple. And I don't think China should be able to ask for pictures of Winnie the Pooh from Apple and have their content scanned. And that's what happens. You start doing this, that's that slippery slope.

Steve: And the other slope is, yeah, in the U.K. legislation, or actually it's the comments from the legislators, they invariably slip in, oh, you know, and terrorism.

Leo: Oh, yeah, throw that in.

Steve: So, you know, and those other people that we're worried about, we want to keep an eye on them, too.

Leo: Yeah. That's the problem is one man's terrorism is another person's freedom. Yeah. I mean...

Steve: So bravo to Apple. I mean, I am so glad. And clearly Erik's letter was meant to be, like, to lay it down to preempt this campaign that what's-her-face is threatening to - I just, I just...

Leo: It's already happening. If you go to their website, they're already...

Steve: I just chuckle at it. It's like, my god.

Leo: I think Apple's doing everything they can, but I think it's in the face of a firestorm, and it's going to be very difficult to...

Steve: It's going to be law. It's going to be legislation. The only thing that will prevent tracking is legislation. The only thing that could cause any of these companies to buckle would be legislation. And Leo, what shape will the technology take then?

Leo: Right.

Steve: I mean, we've gone over and over this. I mean, it's...

Leo: Well, you know, if you're listening to this show, this just argues that you should probably get an open source computer and phone and make sure you've established and downloaded all the encryption technologies you might think you'd ever want and use it. But any commercial device is going to be encumbered by these laws. So you're going to have to roll your own.

Steve: I don't think, I can't imagine that open source could be endangered. I mean...

Leo: How?

Steve: Famously, our government tried to limit the export of crypto to only 40-bit keys because they had some hardware that was able to crack that.

Leo: Right. I mean, forget your phone with Android on it or your phone from Apple because these companies have to obey the law. It turns us all into outlaws. But we've been there before. And encryption is, I think, and privacy are pretty darned important. Oh, well.

Steve: And what politician wants to use a phone which is not safe for them? You know? Isn't that two-faced? I mean, you know, oh, yeah, all of my messages are going to go through some third-party filter? Wait a minute. No, no. Not mine, just everybody else's.

Leo: Just everybody else's. Thank you very much. Hey, our intrepid staff, JammerB and Burke, have apparently for the first time ever watched the show, and they're wondering. Burke came in and said, "That thing over Steve's left shoulder, that white thing, is that new?" And I said, well, I actually did the research. I said it appeared between episodes 568 and 569, back in July of 2016. But I didn't know either. I said, no, I think it's been there. And I actually did the work. What is that thing?

Steve: What are we pointing at?

Leo: Next to the tape, the mag tape, over your left shoulder. No, left.

Steve: Oh, other left.

Leo: Other left. The white-faced box next to the mag tape.

Steve: Ah. That is a field test device for old-school mainframe hard drives.

Leo: Wow.

Steve: You would unplug the hard drive from the mainframe, plug it into this box, and it would do things like exercise the heads and move it back and forth and perform read and write tests.

Leo: I know that you received it in the week of July 12th, 2016, and it has been in the shot ever since. Did somebody send you that? Where did you get it?

Steve: I don't remember. It's got blinky lights. I probably intended to wire them up so they'd be flashing at one point. But then, you know...

Leo: It's only been seven years. I mean, honestly, let's not go crazy here.

Steve: Yeah.

Leo: Before Episode 999 I want them wired up. Pretty soon his whole wall will be blinking lights. Steve, you're a treasure. You really are an international treasure. We're very grateful for the work you do.

Steve: Bye.