

# Security Now! #938 - 09-05-23

## Apple Says No

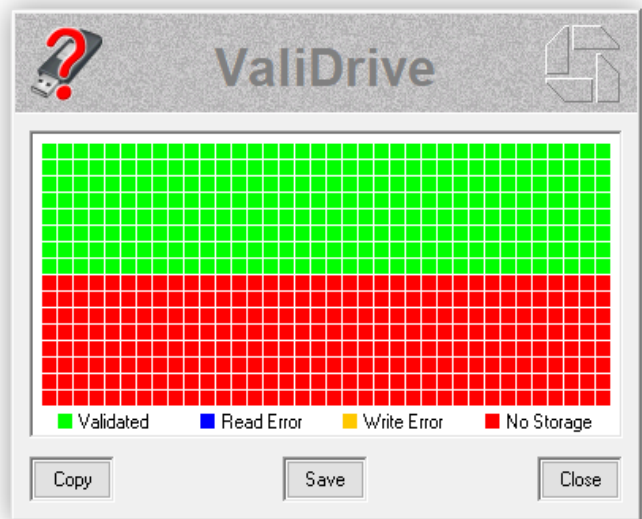
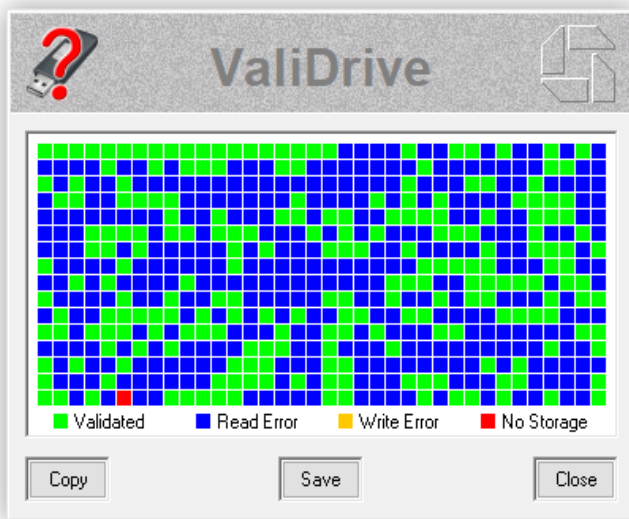
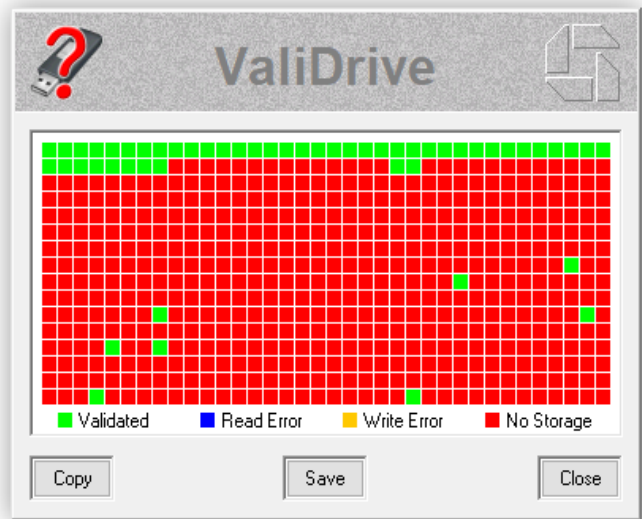
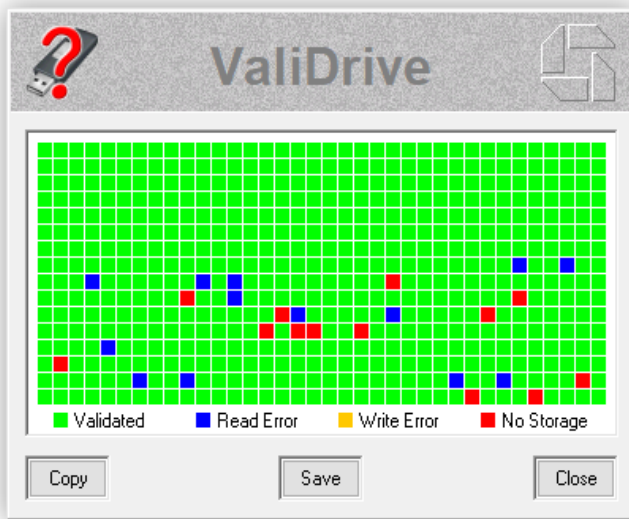
### This week on Security Now!

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 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.

**Those civil engineers were too expensive  
so they decided to hire the mayor's nephew....**



# ValiDrive



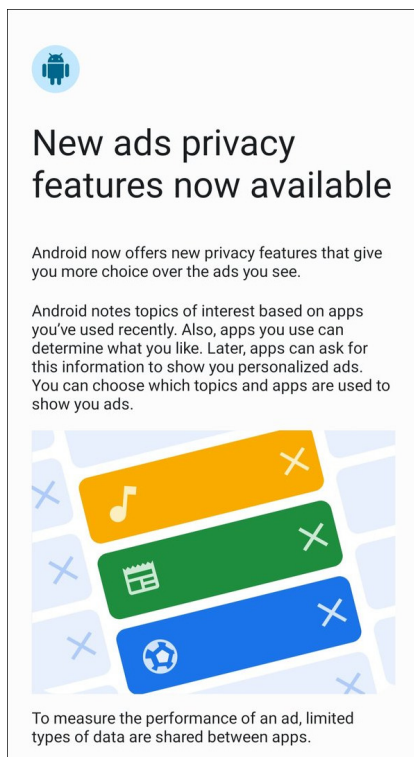
Leo, I was thinking about your reaction during last week's podcast to this problem of bogus mass storage drives, and the need for a quick non-destructive test for both new and existing drives which would always leave their data intact. So, the following day, coincidentally during your *"This Week in Google"* podcast (I watched you discover the Tweet during the show), I tweeted my decision to create a piece of GRC freeware to address the problem. That tweet generated **41** comments, **125** retweets and **779** likes. That's **way** more reaction than my weekly link tweets ever receive.

Yesterday, some of the people who have been testing the first pre-release, which I published Sunday, have said they think it will be the #1 most downloaded freeware from GRC. I'll be surprised if it's able to beat GRC's DNS Benchmark utility, which is, by far, our most downloaded free utility... but we'll see. The DNS Benchmark has more than eight and a half million downloads with 2,000 new downloads per day. So the Benchmark has a good head start!

In any event, "ValiDrive" as it's named (with a tip of the hat to Paul Holder for his suggestion) is indeed providing significant illumination about the USB flash drives everyone is and has been using. Unsuspected problems are being found. The fake drives that the pre-release of SpinRite 6.1 had identified (which put us onto this problem in the first place) have been confirmed with a colorful graphical UI. The difference in performance – and presumably reliability – between inexpensive lower-quality "get what you pay for" drives and those from a reputable brand name manufacturer is somewhat astonishing.

So everyone will be able to play with it shortly. After today's podcast, I'll return to finishing it up and I'm sure I'll be announcing its publication next week with Jason.


## Security News



**New ads privacy features now available**

Android now offers new privacy features that give you more choice over the ads you see.

Android notes topics of interest based on apps 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 apps.

### Google's "Topics" coming to Android **APPS** near you...

Maybe this is a failure of imagination, but it hadn't occurred to me that Google's "Topics" system 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 their user.

One of our listeners was kind enough to share a screen capture of what 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. 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 apps.*

Okay. We know I'm a fan of Google's 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.

I recognize tradeoffs. If websites insist that they receive more revenue when advertisers have some information about their visitors, and advertisers are determined to obtain that information, 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. That's a tradeoff that makes sense to me.

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. I should explain that I personally cannot, and do not, tolerate in-app advertising. Period. If an app is something I want to use, **please** allow me to send a few dollars your way and 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'm unable to silence. But that's just me.

What we see all around us, pervasively, is that advertising works. As you noted last week, Leo, 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. But I'm certain that Google will allow Topics to be completely disabled if that's what its user chooses. So again, props to Google. I am **100%** certain that before offering that full disablement option they thoroughly, and not just once, tested "the tyranny of the default" so that 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, 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, they might as well be as relevant as possible, right?

### **The important exception to Apple's absolutely closed smartphone environment**

I've often bemoaned the problem researchers have with helping Apple to find their own platform's security shortcomings due to that platform being 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 Apple Security Research **Device Program now accepting applications"*** We talked about this before, but I've been overlooking this truly marvelous exception to Apple's *"no one gets in"* stance. In Security Research's overview of this, they explain: *"iPhone is the world's most secure consumer mobile device, which can make it challenging for even skilled security researchers to get started. We created the Apple Security Research Device Program to help new and experienced researchers accelerate their work with iOS. Now accepting applications through October 31, 2023. Apply below."* Under *"How it works."* they remind us:

*"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. [I guess that means that the phone won't suddenly lock you out.] Plus, any vulnerabilities that you discover with the SRD are automatically considered for Apple Security Bounty."*



Elsewhere they elaborate a bit, writing:

*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 [that's the program] is the Security Research Device — 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 (SRD) 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 (SPTM) and Trusted Execution Monitor (TXM), new in iOS 17.*

*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:

*The SRD is intended for use in a controlled setting for security research only. If your application is approved, we will provide you an SRD as a 12-month renewable loan. During this time, the device remains the property of Apple. 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 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.*
- *Be based in an eligible country or region.\**
- *Be the legal age of majority in the jurisdiction in which you reside (18 years of age in many countries).*
- *Not be employed by Apple currently or in the last 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.*

Where they said "*Be based in an eligible country or region.\**" there was an asterisk and down at the bottom of the page was a very long list of qualifying countries. Notably absent were China, Russia and North Korea.

So, this is extremely cool and I'm sure it reflects many prior years of researchers complaining that the damn things are just too locked down for them to be able to conduct any meaningful research other than at very long arm's length. So 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 have done the right thing.

### **On the need to REALLY trust every web browser extension we install**

Some sobering research has recently come from researchers at 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 been adopted by most browsers, they discovered that their proof-of-concept extension is able to steal plaintext passwords from a website's HTML source.

Thanks to the unrestricted access to the DOM tree (the web page's Document Object Model) the researchers demonstrated that the coarse-grained permission model which covers web browsers' text input fields violates the principles of least privilege.

The researchers found that numerous websites with millions of visitors, including some Google and Cloudflare portals, store passwords in plaintext within the HTML source of their web pages, thus allowing for their ready retrieval by extensions.

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:

*In this work, we perform a comprehensive analysis of the security of text input 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 the web page. 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. Our measurements and case studies reveal that these vulnerabilities are prevalent across various websites, with sensitive user information, such as passwords, 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 counter-measures 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.*

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, the researchers 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 to select target input fields and extract user inputs using the '.value' function.

Perform element substitution to replace JS-based obfuscated fields with unsafe password fields.

The extension does not contain obvious malicious code, so it evades static detection and does not fetch code from external sources (which would be dynamic injection), so it is 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 was very real.

Of course, the researchers followed strict ethical standards to ensure no actual data was collected or misused. They deactivated the data-receiving server 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.

Subsequent measurements showed that from the top 10 thousand websites, roughly 1,100 (that's where that 12.5% figure came from ) **are** storing user passwords in **plain text form** within the HTML DOM. So this is a fundamentally insecure design. The designers of those 1,100 websites either wrongly assume that the contents of their page's document object model are inaccessible 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 value. Several of those, including widely used ad blockers and shopping apps, boast millions of installations. Is everyone sitting down? Notable websites lacking the required protection and thus vulnerable right now, include:

- gmail.com – plaintext passwords stored in HTML source code
- cloudflare.com – plaintext passwords in HTML source code
- facebook.com – user inputs can be extracted via the DOM API
- citibank.com – user inputs can be extracted via the DOM API
- irs.gov – SSNs 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 – SSNs are visible in plaintext form on the web page source code
- amazon.com – credit card details (including security code) and ZIP code are visible in plaintext form on the page's source code

Yes. It's that bad.

The V3 Manifest was a tradeoff. Due to the way the industry's existing websites and popular extensions had been coded, limiting extension use further would have broken too much existing code.

When a Google spokesperson was asked about this they confirmed that they're looking into the matter, 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.

In the "Takeaways" section 5.3 of their paper they write:

*Systemic Issue. 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. Think about that. We take it for granted. But any and all password managers must by design be a 3rd-party extension which has direct access to any website's password fields.

*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 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 trade-off 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.*

And these guys just demonstrated that they were able to successfully sneak their universal password extension code past Google's incoming filters without any trouble.

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 & operation. Any of our listeners who are responsible for the design of their organization's secret-accepting web pages might 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 it's accessible at <http://grc.sc/938>  
<https://arxiv.org/pdf/2308.16321.pdf>

Okay... Let's see what thoughts and observations our listeners have to contribute...



## Closing the Loop

Peter Gowdy / @PeteGowdy

*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 SPC solution that you know of?*

I'm not surprised that support is still lagging since, as we know, change always comes slowly. But once additional legislation appears, both in the US and in Europe, I think we can assume that a GPC switch will become a universal feature of browsers.

Alex Neihaus 🐦 and fediverse @alex@air11.social / @yobyot

*Hi. Re: MSFT "doesn't care" about the STS issue using Unix time despite it being known for decades it is unreliable. I don't think they're deliberately or maliciously mis-engineered the feature. I think they just didn't do the research. Most people think that MSFT developers are first-rate. But management there has reduced costs which has encouraged use of off-shore 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 are 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.*

I don't disagree with anything Alex wrote. Everyone here knows how infinitely tolerant I am of mistakes. They happen and anyone can make them. There are many adages which begin with "If you're not making mistakes..." Typical endings for that are "... then you're not trying hard enough." or "... then you're not making decisions." But the most famous 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 do something. The feedback from what you did, which was presumably not what was expected, informs you that a mistake was made somewhere. So, with the benefit of that 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. 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."

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

If we use majority voting, then I've never seen VirusTotal make a mistake. But if you require zero detections out of the 66 different A/V scanners that it polls, that's actually somewhat rare.

When using any modern A/V scanner it's important to understand the context. The original A/V scanners operated by spotting specific code that had been previously identified as part of a piece of known malware. This was quite effective and rarely generated false positive alarms. But then malware evolved to avoid direct code recognition by scrambling itself, encrypting itself, compressing itself and even becoming "polymorphic" by self-rearranging to appear as unique code from one instance to another.

Later, as a consequence of this back and forth cat and mouse game that malware was playing with A/V scanners, the scanners began looking at the operating system functions a program might use and judging whether some things a program might request fall outside of some arbitrary norm. An example might be 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 was unusual.

The final outcome of decades of this back and forth contest between A/V and malware is the 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. 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 any trouble, then that code, as identified by its signature, will have established a good reputation and will become trusted.

The trouble is that any newly created code will have an unknown signature and won't have had any chance to earn a reputation. And as a creator of new utilities, this is a problem I run into every time. Two days ago, on Sunday, the first people to download the completely harmless and freshly assembled ValiDrive Windows application had Windows 10 immediately quarantine the download, complaining that it was "not commonly downloaded." Yeah, no kidding. It had never been downloaded before. It was brand new with a never-before-seen cryptographic signature.

To make matters worse, I was in a hurry to get it into everyone's hands, so I didn't stop 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 things appear to have calmed down since then. GRC has a spotless reputation since we've never had an incident of any kind. But even so, code signing certificates do get stolen, so just being signed by someone with a perfect reputation isn't 100% assurance. I just checked ValiDrive with VirusTotal and there are three false positive "detections" after querying 66 A/V systems. Cybereason, Cylance and Trapmine didn't like it but no one else complained.

As for WinAero Tweaker, I just grabbed a copy of the setup executable from WinAero.com and dropped it onto VirusTotal. I received a 100% clean bill of health with VT saying that 0 out of 43 scanners found it to be suspicious. I noted that the executable 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. And needless to say, always and only obtain such programs, especially if they are unsigned, from their original web site source. But, for what it's worth, the v1.55 I just downloaded directly from their website, other than not being signed, looks fine.

## Rick / @rpodric

*Steve, on Acceptable Ads in uBO, how are you doing it? I looked around but only found this old thread. 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.*

[https://www.reddit.com/r/uBlockOrigin/comments/ddwisu/can\\_i\\_voluntarily\\_choose\\_to\\_allow\\_acceptable\\_ads](https://www.reddit.com/r/uBlockOrigin/comments/ddwisu/can_i_voluntarily_choose_to_allow_acceptable_ads)

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 happening, but I had 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 2014. The extension relied upon community-maintained blocklists while also adding some extra features. Not long after, Raymond transferred the project's official repository to a guy named Chris Aljoudi since "gorhill" he was frustrated with all the incoming requests.

It turns out that Chris was somewhat less than honest and respectful. He immediately removed all credit to 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 it for anything he could. 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 of 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 behind uBlock.

**#LoveThyNeighbor** 🇺🇸🇲🇾🇲🇪 / **@CenterLeft2020**

*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!*

Unfortunately the short answer is no. The ReadSpeed DOS utility was a natural offshoot of the early work on SpinRite v6.1. 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 front of many SSD devices. Since I thought that ReadSpeed might be broadly useful it was spun out 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, since I'll be anxious to get feedback about this emerging software's dual booting over BIOS and UEFI, and its ability to finally talk natively to all drive types.

**Austin Wise / @AustinWise**

*RE: man-in-middle attacks and HTTPS on SN 937. If an attacker is on a local network, like a coffee shop Wi-Fi, they might not need a privileged position to modify traffic. See ARP spoofing [https://en.m.wikipedia.org/wiki/ARP\\_spoofing](https://en.m.wikipedia.org/wiki/ARP_spoofing).*

*Also, the integrity features of TLS are also 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. This prevents downloading corrupted data.*

*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/> for short links and <http://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, 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*

So, Austin – who sounds like a Google engineer – makes some great points.

Way back in the early days of this podcast we spent a good deal of time exploring the details of low-level hacks and attacks such as ARP spoofing. For those who don't know, ARP stands for 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 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 on the local Ethernet network.

So here's the point that Austin was making: If something can 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 IP traffic to the wrong Ethernet address; and ARP spoofing is able to cause exactly such misdirection. So Austin is 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, indeed, the use of TLS and HTTPS would completely prevent any such attacks. So he's correct that a man-in-the-middle position could be obtained using a successful ARP spoofing attack.

### **JediHagrid he/him / @JediHagrid**

*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 of 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 the server you can see it here: <https://www.REDACTED.com/application/files/?C=S;O=D> it's called: REDACTED.sql you can search it using notepad++. There are .gov customer emails, people who are applying for jobs' addresses, everything in plain text. I'm not sure what else to do.*

Since the site 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](mailto:report@cisa.gov)

### **Simoncroft08 / @simoncroft08**

*Hi Steve, On SN936 you talked about multi level 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** 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 - PS the heuristics story was an ouch! Glad I'm now retired from sysadmin life.*

I thought this was an interesting angle. The inherently lower reliability of MLC 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 Biggs / @kasamhor**

*Hi Steve, I am listening to this week's episode (episode 937) of Security Now. You have just described that you can get VirusTotal to check a file before you download it. The problem with this, though, is that if the malicious site recognises that VirusTotal is downloading the file, 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 can not find a way of downloading the file directly from VirusTotal, I think that the better option is to download the file onto your computer, then upload it to VirusTotal 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 have decided to continue past those dreaded three '9's. Regards, Martin*

And, as I said, that's a head smacker. 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, and as long as the possibility exists that VT would be receiving and checking a different file than the user downloads, there's no choice but to get it first and provide it to VirusTotal. Nice catch, Martin!

# Apple Says No

In a rare occurrence, Apple chose to publicly share a mildly threatening private letter it received last Wednesday addressed to Apple's CEO, 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.

In terms of the way the future is going to take shape, the biggest thing happening today 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, representing a definitive change of stance and policy on Apple's part, I want to share Sarah's unsubtle letter followed by Apple's response:

*From: Sarah Gardner  
Date: August 30, 2023 at 1:24:37 AM GMT+2  
To: [ REDACTED but presumably "Tim Apple" ]  
Subject: Detect CSAM in iCloud: Incoming Campaign*

*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 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 a 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 wanted to alert you to our presence and our intention to take our very reasonable requests public in a week's time. Should you want to discuss our campaign over the course of the next week, or after we have 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 Heat Initiative*

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

This Sarah Gardner is the former vice president of external affairs for the nonprofit organization Thorn, 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, 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 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:

*August 31, 2023*

*Ms. Sarah Gardner CEO,  
Heat Initiative*

*Dear Ms. Gardner,*

*Thank you for your recent letter inquiring about the ways Apple helps keep children safe. 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 have 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 them stay safe. With respect to helping kids stay safe, we have made meaningful contributions toward this goal by developing a number of innovative technologies. We have 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 to not 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.*

I'll 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 user's phones containing that hashed library of known CSAM images. The idea of that crept 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 textual 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:

*After 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.***

*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 of 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 have 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 have outlined above. We remain interested, for instance, in working with the child safety community on efforts like 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.*

I think that one statement from Apple entirely explains their, by now, extremely well considered position: *"... 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 user's 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 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 (EDRi) advocacy group echoed Matthew's concern, writing, *"This looks like a shameful general surveillance law entirely unfitting any free democracy."*



In our ongoing coverage of this, we were previously able to quote the official positions of the many various 3rd 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 mobile messaging industry has now formed a united front.

Now 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 they be present but unenforced? Will the government or some upstart 3rd party offer surveillance messaging? 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 shape will it be?

