Security Now! #956 - 01-09-24
The Inside Tracks

This week on Security Now!

I want to start off this week by following-up on last week's podcast about the hardware backdoor discovered in Apple's silicon, to support the conclusion I've reached since then, that this was deliberate on Apple's part, that they always knew about this, and why. Then we're going to wonder whether everyone is as cyber-vulnerable as Ukraine appears to be? And if so, why and just how serious could cyberattacks become? What's the latest on the mess over at 23andMe? How's cryptocurrency been faring, and are things getting better, staying the same, or getting worse? What Google Mandiant account got hacked? Just how seriously, and legally, do we take the term "war" in "cyberwar", and what are the implications of that? LastPass recently announced some policy changes; even if they are about two years late, what lessons should the rest of the 'Net take away? During 2023, how did Windows 11 fare against Windows 10? What happens when users discover that Chrome's Incognito mode is still tracking them? And then, after exploring some questions from our terrific listeners, I want to share the result of some interesting research I conducted last week during the final days of the work on SpinRite 6.1 for this week's podcast, titled: "The Inside Tracks."

Happy 17th Birthday

Security News

Apple’s hardware backdoor

I want to begin this week’s podcast with a follow-up to last week’s blockbuster news of the discovery of a well-protected hardware backdoor existing through the five most recent generations of Apple’s home-grown silicon. The way we left things last week, in the wake of this revelation, was with a large array of possibilities. I’ve settled upon exactly one which I believe is the best fit with every fact we have.

Many people sent notes following-up on last week’s podcast. Many doubted the NSA conspiracy theory because of those other, not easy to effect steps, involving other clearly inadvertent mistakes in Apple’s code, that were needed by this particular malware.

I don’t know why it didn’t occur to me last week, but it has now. As we know and have covered here in great detail in the past, Apple has locked their iPhone down every way from Sunday. I believe from all the evidence and focus that Apple has put into it, that Apple’s iPhones are truly secure. But would Apple actually produce a smartphone handset which **they** absolutely positively truly could not get into even if it meant the end of the world? If Apple believed that they could design and field a truly and totally secure last resort backdoor means of accessing their devices in the event that the world depended upon it, I believe that they would have designed such a backdoor, and I believe that they did – deliberately and purposefully. And I do not think less of them for it. In fact, I think that the case could be made that it would be irresponsible for Apple **not** to have provided such a backdoor. We’ll likely never know whether any external agency may have made them do it, and yes, doing so could hardly be more controversial.

But I can imagine a conversation among just a very few at the very top of Apple, Tim Cook and his head of engineering and security. They **had** to have had a conversation about whether it should be possible, under some set of truly dire circumstances, for them to get into someone else’s locked iPhone. Obviously, the security of such a system would be more critical than anything. But their head of engineering security would have explained, as I did last week, that as long as the secret access details never escaped, because it’s impossible to probe anything that must be accompanied by signature hash, there would truly be no way for this backdoor to be discovered. As I said last week, from everything we’ve seen, it was designed to be released in the field where it would be active and totally safe.

If Tim Cook were told that Apple could design and build-in an utterly secure, emergency prevent the end of the world escape hatch into their otherwise utterly and brutally secure devices, and that this escape hatch could never possibly be opened by anyone else – ever – I imagine Tim would have said, under those conditions, yes. I think that most CEOs who are in the position to understand that with great power comes great responsibility, when assured that it could not possibly be maliciously discovered and abused to damage their users, would say: Yes build it in.

I trust Apple as much as it’s possible to trust any commercial entity operating within the United States. I believe that they absolutely will protect the privacy of their users to the true and absolute limit of their ability. If the FBI were to obtain a locked iPhone known to contain the location and relevant disarming details of a nuclear weapon set on a timer and hidden somewhere in Manhattan, I would be thanking Apple for having the foresight to create a super-secure means for them, and them alone, to gain entry to their device. And I’d argue that in doing so they truly did have the best interests of their customers in mind since, in this scenario, a great many iPhone users' lives would be saved. There are all manner of conspiracy theories possible here. And this one of mine is only one of many. But of all the possible theories, I believe this one fits the facts best and makes the most sense.

Of course, the first thing everyone will say is: "Yeah, but Gibson, they **did** lose control of it, and it **was** being used by malware to hurt some of Apple's users. And that's true. In fact, that's the **only** reason the world learned of it. If this scenario is correct, Apple never divulged this to **any** outside entity, and never would. This would never have been meant for the FBI, CIA, or NSA to have for their own use. If an impossibly high bar were reached, Tim Cook would say: *"Have an agent bring us the phone and we'll see what we can do."* But somewhere within Apple were people who did know. Perhaps someone inside was set up and compromised by a foreign group. Perhaps Apple had a mole. Perhaps it was gambling debt or the threat of some extremely embarrassing personal information being disclosed. One thing we've learned and seen over and over on this podcast is that when all of the security technology is truly bulletproof, the human factor becomes the lowest hanging fruit. Just ask LastPass how they feel about the human factor ... which bit them badly.

So where does this leave us today?

We know that all Apple iPhones containing the A12 through A16 chipsets contain this backdoor and always will. We don't know that it's the only backdoor those chipsets contain. But Apple doesn't need another backdoor since they still have access to this one. They locked a door in front of this one but they can always unlock it. After being contacted by Kaspersky, Apple's iOS updates blocked the memory mapped IO access that was discovered being taken advantage of by malware. But Apple is able to run any software they choose on their own phones. Which means that Apple still has access to this backdoor should they need to use it.

And this means that they've lost plausible deniability. They have the ability to open any iPhone that they absolutely must. So this poses a new problem for Apple when law enforcement now comes knocking with a court order, as it's almost certain to, with way-below-the-bar requests for random iPhone unlocking to "assist" in this or that case. So this is a new mess for Apple.

Apple's most recent silicon is the A17, yet Kaspersky told us that this facility had only been seen in the A12 through A16. If the malware didn't contain that initial unique per-chip-generation unlock code for the A17 silicon then this backdoor might still be present in today's iPhone 15 and other A17 devices. That's the most reasonable assumption. But what about the next iteration of silicon for the A18? Another thing we don't know is what policy change this disclosure may have for the future. We don't know how committed Apple was to having this capability. Have they been scared off? Will they move it to a different location within the ARM's 64-bit address space?

As we were, after last week, we're left with a handful of unanswered and unanswerable questions. But my intention here was to hone this down to explore what appears to be the single most likely scenario. Apple designed, and is quite proud of, the GPU section of those chips which contains the backdoor hardware. There's no chance they weren't aware of the little hash-algorithm-protected DMA engine built into one corner. People within Apple knew.

Listeners of this podcast know that I always separate policy decisions from mistakes which unfortunately happen. So I sincerely hope that Apple's **policy** was to guard this as perhaps their most closely held secret and specifically that it was never their policy to disclose it to any outside agency, of any kind, for any reason. Somewhere a mistake happened. And I'd wager that by now, Apple knows where and how that mistake occurred.

Spying eyes among us...

We talked earlier about the interesting abuse of Internet-connected cameras by Russia to obtain attack and improved targeting of their weapons strikes inside Ukraine. I saw a bit of an update on that last week after Russian hackers successfully hijacked at least two security cameras and used their live video feeds to adjust missile strikes targeting the city of Kyiv last week. Once Ukraine's SBU, their Security Service, detected the hacks they took down the cameras to prevent their further abuse. In part of the coverage of that was the information that since the start of Russia's invasion, more than **10,000** security cameras have been taken down across Ukraine.

Hearing this makes me wonder and worry about the situation here in the U.S., in Europe, and with any of our allies. Back when we began this podcast, you know, when loincloths were in fashion, the idea of "cyberwar" was still squarely the stuff of fiction. From everything we hear, there are constant low-level "cyber skirmishes" going on all the time now. We know that the usual suspects of China, North Korea and Russia have talented hackers who are more or less continually poking around in our computer networks. And for our own sake, I hope we're giving at least as well as we're getting, since a cyber standoff is in everyone's best long term interest.

Everyone in the world is pulling from the same common technology pool. So we're probably all about equally vulnerable to each other. There's no reason to believe that the cameras we have everywhere here in the States and throughout Europe are any more secure than those that Ukraine was using. And the same applies to all of our other interconnected technology.

I've often wondered what I would do if I were starting out in the world today. I've always had a strong intellectual curiosity about whether I could hack other people's stuff. But doing so is both unethical and illegal. So I never have. But participating in my country's defense and, if necessary, its offensive operations, does have some appeal while resolving the ethical and legal roadblocks. One thing is very clear today, this is no longer the stuff of Sci-Fi... it's very real. And it appears that our countries need us.

Russian hackers were inside Ukraine telecoms giant for months

I think this topic is important enough for me to spend a bit more time on a specific example: Last Thursday, Reuters news service published an article titled "Russian hackers were inside Ukraine telecoms giant for months." Here's the information that Reuters published:

> *LONDON, Jan 4 (Reuters) - Russian hackers were inside Ukrainian telecoms giant Kyivstar's system from at least May last year in a cyberattack that should serve as a "big warning" to the West, Ukraine's cyber spy chief told Reuters.*

> *The hack, one of the most dramatic since Russia's full-scale invasion nearly two years ago, knocked out services provided by Ukraine's biggest telecoms operator for some 24 million users for days from Dec. 12.*
>
> *In an interview, Illia Vitiuk, head of the Security Service of Ukraine's (SBU) cybersecurity department, disclosed exclusive details about the hack, which he said caused "disastrous" destruction and aimed to land a psychological blow and gather intelligence.*
>
> *He said: “This attack is a big message, a big warning, not only to Ukraine, but for the whole Western world to understand that no one is actually untouchable.” He noted that Kyivstar was a wealthy, private company with heavy investments in cybersecurity.*
>
> *The attack wiped "almost everything", including thousands of virtual servers and PCs, he said, describing it as probably the first example of a destructive cyberattack that “completely destroyed the core of a telecoms operator.”*
>
> *Later, following some investigation, on December 27th SBU’s head of cybersecurity said they found that the hackers probably attempted to penetrate Kyivstar in March or earlier last year. He said: “For now, we can say securely, that they were in the system at least since May of 2023. I cannot say right now, when they had full access: probably at least since November.”*
>
> *The SBU assessed the hackers would have been able to steal personal information, understand the locations of phones, intercept SMS-messages and perhaps steal Telegram accounts with the level of access they gained. A Kyivstar spokesperson said the company was working closely with the SBU to investigate the attack and would take all necessary steps to eliminate future risks and following the major break there were a number of additional attempts aimed at dealing more damage to the operator.*
>
> *Kyivstar is the biggest of Ukraine's three main telecoms operators and there are some 1.1 million Ukrainians who live in small towns and villages where there are no other providers. People rushed to buy other SIM cards because of the attack, creating large queues. ATMs using Kyivstar SIM cards for the Internet ceased to work and the air-raid sirens - used during missile and drone attacks - did not function properly in some regions, he said.*
>
> *Post attack forensics are made more difficult because of the wiping of Kyivstar's infrastructure. But Vitiuk said he was "pretty sure" it was carried out by Sandworm, a Russian military intelligence cyberwarfare unit that has been linked to cyberattacks in Ukraine and elsewhere.*
>
> *A year ago, Sandworm penetrated a Ukrainian telecoms operator, but was detected by Kyiv because the SBU had itself been inside Russian systems. Vitiuk said the pattern of behavior suggested telecoms operators could remain a target of Russian hackers and during 2023, the SBU said that it thwarted over 4,500 major cyberattacks on Ukrainian governmental bodies and critical infrastructure.*

Again, there’s no reason to believe that things are any different anywhere else. Ukraine is using the same technology as everyone else. As I’ve said, all of the evidence we have suggests that our actual security is far more soft than we would like. What’s generally and thankfully missing is the motivation to abuse it. The rise of cryptocurrency created the motivation to extort enterprises that smugly believed that their IT security budget was sufficient and that the threats were being overblown. No one thinks that any longer. The last thing we need is an escalation.

23andYou
Things are still a mess at 23andMe. They've been hit with 30 lawsuits since last December's revelation of the breach which disclosed the personal information of 6.9 million of their users. To remind everyone, the story is that 14,000 accounts were first directly compromised using simple credential stuffing – reusing known, previously-used passwords of each victim. I would argue that this had to be detectable right there, but you won't see what you're not looking for. From there, we're told, simply using the API that was available to any logged-on user – since that's all these bad guys apparently were – the attackers were then able to siphon off the personal data of an additional, expanded, 6.9 million un-breached users.

I suppose I'm still skeptical about this explanation because I find it difficult to believe that the designers of 23andMe's architecture could deliberately set things up so that anyone logged into their system could have direct access to, on average, the personal data of 493 other members. 6.9 million divided by 14,000 is 492.857 – which is the average "disclosure reach" of each of 23andMe's logged-on users. In order to believe that this is what actually happened, 23andMe's system had to be very horribly designed from the start, which is quite dispiriting.

And then, in the wake of this catastrophe of their own making, adding insult to injury, 23andMe attempted to change the terms of service for their users, retroactively if you can believe that, to require them to agree to settle any disputes through arbitration in lieu of other legal action. That didn't pass notice and you can imagine that it didn't go over very well.

As we know, anyone can make a mistake... but they are directly responsible for their apparently incredibly crappy system design which, if they are to be believed, allowed any legitimate user to have access to the personal details of, on average, nearly 500 other users, each.

Incentives, Anyone?
And speaking of the incentives created by cryptocurrency, the Estonian cryptocurrency platform CoinsPaid was the victim of another cyberattack, losing an estimated $7.5 million worth of crypto assets. I said "another" since this is the company's second hack after it lost $37.3 million last July. CoinsPaid blamed last year's incident on North Korean hackers. I wonder who you call in North Korea to negotiate a settlement? *"Hey, how 'bout we'll give ya a 10% bounty and no hard feelings if you'll return the rest?"* I wouldn't hold my breath. Meanwhile, the Gamma cryptocurrency platform says it lost $6.1 million worth of assets after a threat actor abused the infrastructure of one of its providers to manipulate exchange prices and another threat actor has stolen nearly $4.5 million worth of crypto-assets from the Radiant Capital crypto-platform. The technique used there was a so-called flash loan attack. I don't know where all of this money is coming from or going to, but I'm sure glad that none of it's mine.

Crypto Hacking in 2023:
Stepping back and taking the longer range view, overall during all of last year, hacking attackers made off with more than $1.8 billion USD worth of crypto assets across as the bounty from 751 individual security incidents. There's some good news here, though, since that number is way down, as in "by half", from the $3.7 billion USD that were lost the year before, in 2022.

According to the blockchain security firm CertiK, last year's top 10 most costly incidents accounted for more than $1.1 of the total $1.8 billion dollars stolen last year. So not all attacks are equally profitable. And peeking behind the curtain, the most costly incidents were linked to leaks or compromises of private keys, with more than $880 million stolen that way last year. According to TRM Labs, North Korean hackers were linked to $600 million dollars of those total stolen assets.

Mandiant Twitter scam

And lest we believe that these things only happen to people with low security awareness, an unknown threat actor recently hijacked the Twitter account of Google's Mandiant division, you know, Google's super high-end security folks. The account takeover was used to promote a (guess what?) cryptocurrency scam. The attack was just one of a number of similar incidents that hit many high-profile Twitter gold badge accounts at the start of the year.
The hacks appear to be linked to an underground market where hacked Twitter business accounts are being offered for sale.

This means war!!

Just how seriously, and legally, do we take the term "war" in "cyber**war**"?? Remember back nearly seven years ago when, in 2017, the monster American pharmaceutical company Merck suffered a serious ransomware cyber breach by the NotPetya group? What stunned us, and I remember that Leo was like "Whaaaaaat???" was that Merck, who was carrying significant cyberattack insurance at the time, was claiming that the attack, which they said affected 40,000 of their PCs, would cost them $1.4 BILLION (with a 'B') dollars to clean up.

Naturally, their cyber insurance carriers were none too pleased by the prospect of having to fork over $1.4 billion dollars to, what, finance Merck's physical replacement of their entire PC inventory? I mean, it's not as if the machines melted.

We covered this at the time, and recall that the three Merck insurers who were on the hook for this, were attempting to get out of their policy obligations by claiming that an exemption applied in the case of "Hostile/Warlike Action" which is a commonly present policy exclusion. So the question that has ever since been working its way through our U.S. legal system was whether or not a "cyberattack" should be considered to fall under this standard "Hostile/Warlike Action" policy exclusion. And, of course, this would be precedent setting since devastating cyberattacks are no longer theoretical and insurance to make enterprises whole in the wake of one have become crucial and ever more costly – both in premium and in reimbursement.

Until last week, when New Jersey's state Supreme Court was set to hear oral arguments from both sides, a lower New Jersey appeals court had ruled that the Merck was entitled to half of what they were seeking under their policy coverage. In other words, $700 million. The insurers still wanted to pay less and Merck wanted more. But hours before oral arguments were to begin, the parties announced that they had reached a settlement, though the terms of that settlement have not been disclosed. Given that Merck is a publicly traded company, owned by its stockholders, I would imagine that the terms of the settlement will eventually become known.

Interestingly, in amicus briefs filed before the scrapped oral argument, national associations for big business, manufacturers, and corporate insurance litigators had all urged the court to uphold the ruling that cybercrime did not fall under an insurer's "Hostile/Warlike Action" policy exclusion and plant a national flag on this issue to benefit insured businesses. But the decision was not cut and dried. Dueling briefs from international law scholars debated whether foreign-linked hacking against corporations is warlike action. The takeaway for insurers should probably be that they are going to need to stand behind their cyberattack policies, and those paying for coverage by those policies should probably demand some explicit clarification from any policy that contains such potential wiggle room language.

LastPass is making some changes

Last Tuesday, LastPass posted a blog update titled: "LastPass Is Making Account Updates. Here's Why" I'll just share the beginning of what they said since we are all well able to read between these lines. They wrote:

> *You may have noticed that lately we've been asking our customers to make some changes to their LastPass accounts. These changes include requiring customers to update their master password length and complexity to meet recommended best practices and prompting customers to re-enroll their multi-factor authentication (MFA), among others. All of these changes are intended to help make our customers more secure, and we want to share additional context about the evolving cyber threat environment that's driving these requests so customers can better understand WHY these changes are important. To do this, we'll address some of these recent changes, and explain what threats are driving them, and how these updates are designed to help.*

My only complaint, of course, is that it's closing the barn doors after the horses have all run off. This would have been nice to see several years ago.

This effort is clearly an attempt to respond to the theft of the master vault data and to mitigate future disasters. Requiring everyone to "re-enroll" their multi-factor authentication is definitely smart after any breach, and enforcing new 12-character minimum password complexity requirements only makes sense, too.

What should really be happening across the Internet is that users should begin to be forced to increase the security of their logons. It should not just be happening at scattered sites in the wake of devastating attacks. Any service that supports logons where a breach could have devastating consequences for its users should start doing the same. Users really want to reuse "their" personal password everywhere. That's still the typical behavior. Obviously, not among this podcast's listeners, but pretty much everywhere else. Never underestimate the strength of inertia. Users do **not** want to change and they will **not** change unless they're forced to.

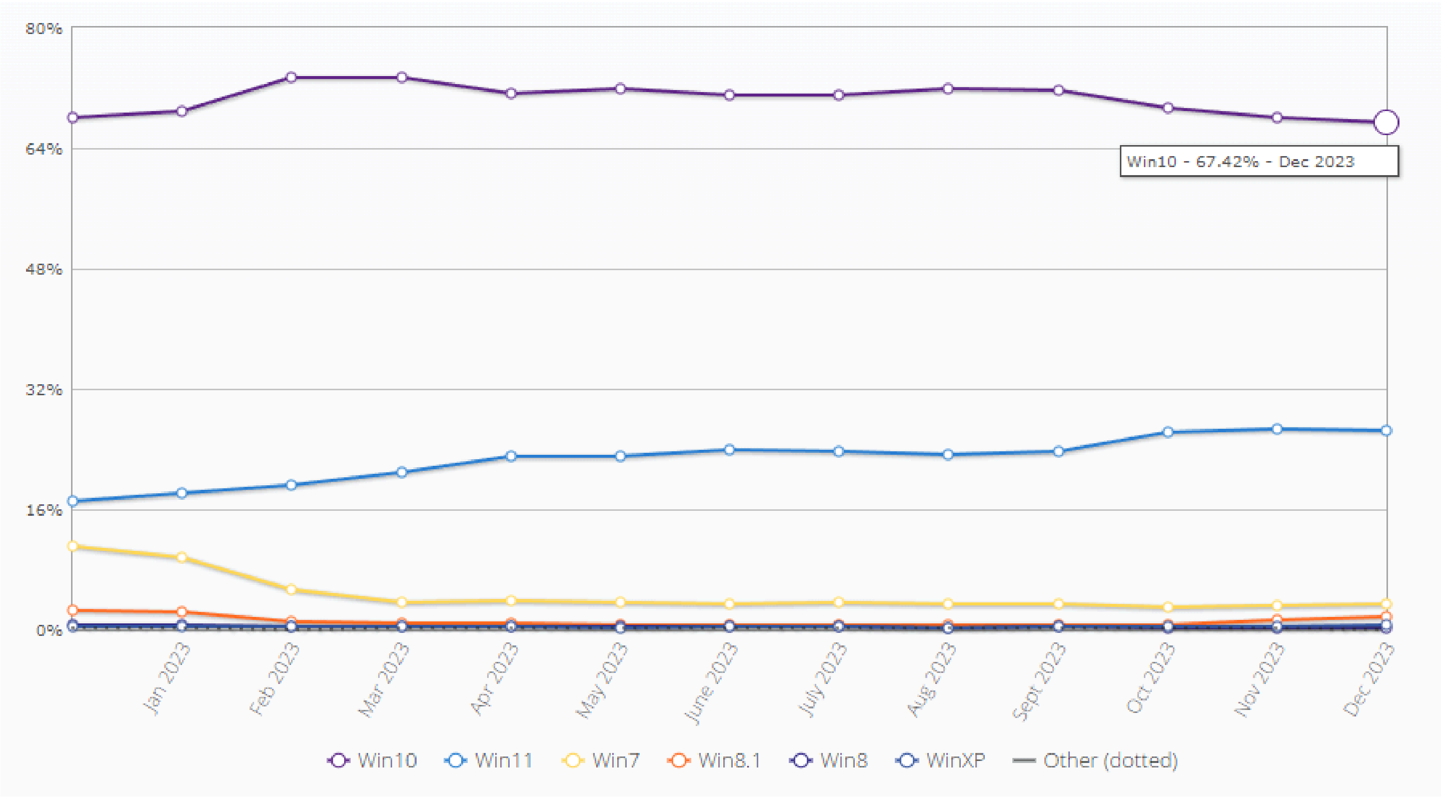
We now have the technology to enforce password complexity rules on the user in their browser thanks to client-side JavaScripting. Users hate password requirements. Why? Because those requirements prevent them from using their favorite universal pet password everywhere. And those requirements mean that they may need to deal with unique passwords per site. The question is whether the Internet should continue to let them? If the Internet continues to allow this past behavior, it will never change. We all know that. Why would it change? Users will need to be forced. But every site is understandably terrified of doing that because they don't want to alienate their users.

The rational solution is for sites not to pretend that their users have security that doesn't exist. If a site is not going to enforce a sufficiently high level of password complexity then it should not assume that its users have any actual logon protection and it should act accordingly. Or perhaps the client-side JavaScript, which can see the user's plaintext password before it is locally hashed, should examine the password's complexity and send along a ranking of the password strength. Then a site that does offer some sensitive services could explain to its logged on user that the password they are using is fine for logging on, but for their protection, a better password will be required before they are allowed to do anything sensitive that they would not want hackers to be able to do in their name.

So I suppose I'm saying that the industry has clearly been dragging its heels because it hasn't been forced to change, and this has allowed users to, in turn, drag their heels and continue with habits that no longer serve them. Web portal designers would be well served to keep this in mind.

Windows Watch:

As of the beginning of 2024, Windows 10 is holding onto 2/3rds of the desktop while Windows 11 has been gradually creeping upward from about 16% to 26% across 2023. People generally like what they have and don't see any reason to switch. The stories I hear are about new machines all coming with Windows 11, which their new owners don't want, but they have no choice.



Google settles $5 billion lawsuit

The following bit of Google Tracking news made a lot of headlines recently, so I thought I would mention it, too. Remember back in 2020 when Google was found to be tracking users in “incognito mode” and this resulted in a ridiculously large class action lawsuit? The news is that the lawyers on each side of this dispute have reached an agreement, as happens more often than not on the eve of such cases moving forward to trial.

When you’re big, you tend to be a target of attack since the presumption, at least among scummy attorneys, is that it’s worth some money just to make the nuisance lawsuit go away. At the same time, unfortunately, being big also increases the tendency of companies to throw their weight around, bully others, and imagine that they can get away with whatever they want.

Thursday before last, US District Judge Yvonne Gonzalez Rogers put the trial that had been scheduled for this case on hold in California after lawyers said they had reached a preliminary settlement. Judge Rogers had previously rejected Google's bid to have the case dismissed, saying she could not agree that users consented to allowing Google to collect information on their browsing activity.

The class action, which was filed in 2020 by law firm Boies Schiller Flexner (“Boies” as in David Boies), claimed that Google had tracked users' activity even when they set the Google Chrome browser to “Incognito” mode. It said this had turned Google into an "unaccountable trove of information" on user preferences and "potentially embarrassing things". It added that Google could not *“continue to engage in the covert and unauthorized data collection from virtually every American with a computer or phone”*.

Google said it had been upfront about the data it collected when users viewed in private mode, even if many users assumed otherwise. The search engine said the collection of search history, even in private viewing mode, helped site owners “better evaluate the performance of their content, products, marketing and more.” Even when in Incognito mode, websites visited can use tools such as Google Analytics to track usage.

I think my take on this is that it’s a case of the fine print coming back to bite ya. Google claims that the users of their incognito mode were duly informed and knew that tracking was still occurring even though the post-incognito mode residues from their browsing such as history and cookies were not retained. Apparently some of their users disagreed and felt betrayed.

This was a rather thin news week, but I think I may be onto an interesting independent analysis of the privacy protections created by Google’s TOPICS API and other components of Chrome’s privacy sandbox. If it pans out, I’ll have that for next week.

Meanwhile, we have some closing the loop feedback from our terrific listeners after which I want to share some very interesting things I learned last week relating to spinning disk drives...

Closing the Loop

Carl Smith /@CarlRoeSmith

@SGgrc How has Operation Triangulation not received more press coverage? This is huge!!!!

I agree with Carl. I suspect that Apple is benefiting from the fact that while what's really going on here, which everyone listening to this podcast understands, is truly monumental, it was also "patched" with yet another iOS update and the public at large has no way of discerning that this one is any different from any that preceded it. And, really, some Russian security analysts found something they presented during the Chaos Communications Conference in Hamburg? That's not going to make the nightly news. The popular news media cannot begin to explain this to the average consumer. So I'd bet the news producer just says: "talk about the weather" while Apple breathes a huge sigh of relief in the knowledge that they didn't take any P.R. hit from what might have been a disaster for them.

Vjirasek / @vladjirasek

Hi Steve. Great work on SN955. I am wondering why Apple has not implemented ROP attack protection similar to what Intel has done. Would this break the chain of this sophisticated attack? Also concerning to see that Apple has left the back door in the SoC to get in. Thank you for your hard work.

Vjirasek is referring to the use of Return Oriented Programming, which we mentioned and talked a bit about last week. It's a living off the land practice of using bits of code that's already present in the target device to obtain the effects that are needed for the attack. I'm certain Apple has ROP attack prevention in place, as must any highly secure attack-prone operating environment these days. But while ROP makes attacks far more difficult by scrambling and randomizing the memory locations of code, that code is still present in memory, it's just been moved at load time into initially unknown locations. One of the things the Kasperski guys noted was that a huge amount of the malware was spent examining the system's memory. So that would likely have been code designed to locate the bits of executable code that they needed. While ROP can make attacks much more difficult, it is also not a perfect solution.

Robin Ramaekers / @robinramaekers

On SN955 you had Ethan Stone giving you a quick note that he had problems closing the Edge browser. While it is true that when you close the window, the edge processes keep running, there is an easy way to close the browser. If you click the ellipsis on the right you find the option to close Microsoft edge all the way down at the bottom. This is in contrast to clicking the "close" X at the top which may only close the user interface.

— AND —

Warwagon (posting into GRC's newsgroups wrote)

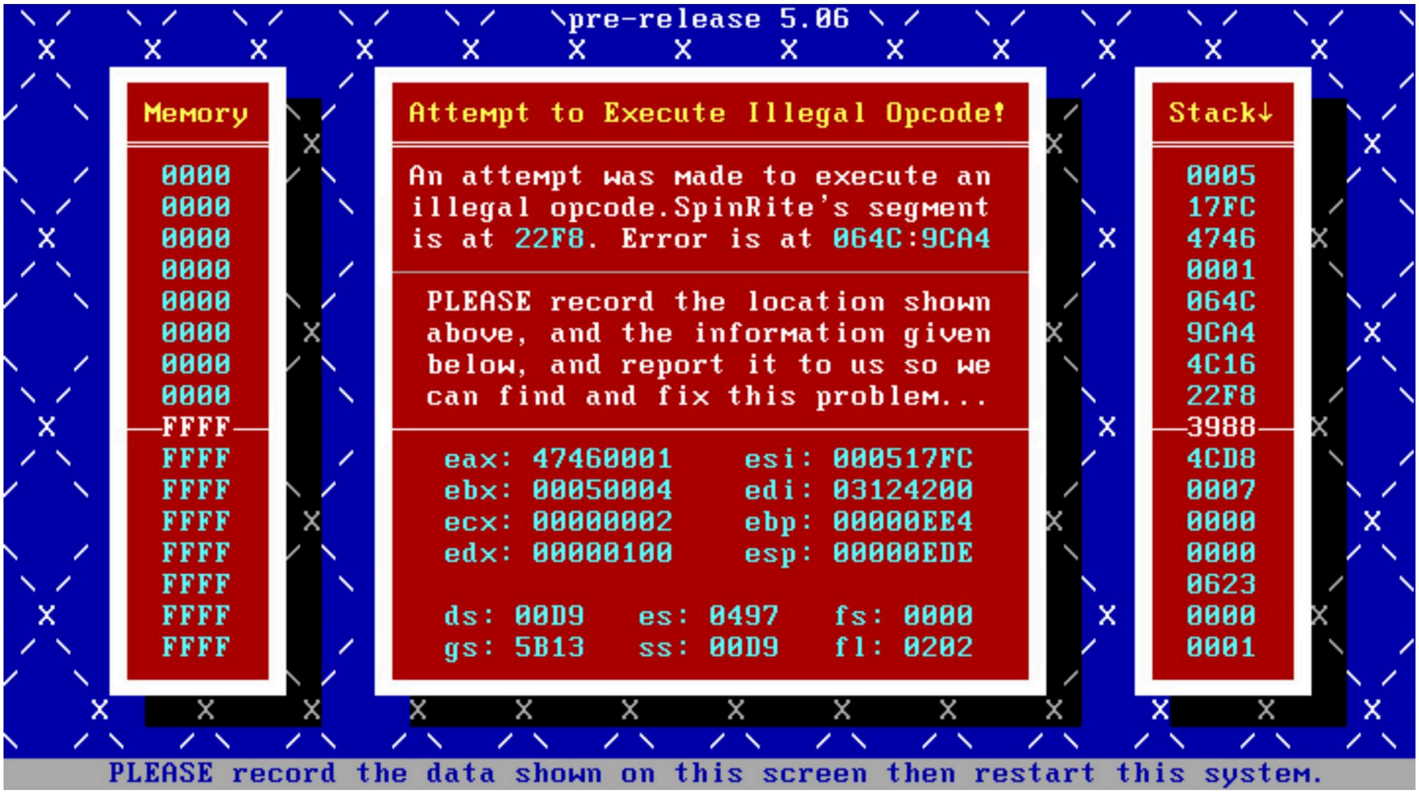
Here is a fix for edge running in the background: Open Edge / Click the 3 dots in the top right *and click "Settings". In the top left do a search for "Startup". On the right you'll see 2 options "Startup boost" and "Continue running background extensions and apps when Microsoft Edge is closed" Turn both of those off. That should remove edge from the task manager when it's closed unless you also have some website notifications enabled.*

Those are some great suggestions. Here’s what I found: When I closed my instance of Edge it did not continue running anything in the background. Following Warwagon’s advice, I found that I had the “Startup boost” option turned off. And that’s what made the difference. With “Startup boost” turned on, Edge doesn’t close UNLESS you open the ellipsis menu and choose “Close” down at the bottom. Just clicking the UI’s [X] box only closes the UI and it definitely leaves a bunch of processes (10 in my case) running.

So, anyone who wants to truly close Edge will need to either turn off “Startup boost” or use the ellipsis menu and select close at the bottom. If you have ample memory and would rather have Edge pop onto the screen quickly (because it’s always actually running) turn on Startup boost.

Thomas Tomchak / @tomchak

I'm guessing this is out of scope for how spinrite should be used, but I tried to boot my windows VM into spinrite because I wanted to run it on the internal drive of the VM. I first tried this with 6.0 and it booted up, so I then downloaded the PR windows exe and created a new ISO. I uploaded that to the vSphere host, attached it as a CD and told the VM to boot using bios. This time it went right to the attached screen.



```
                              pre-release 5.06

  Memory      Attempt to Execute Illegal Opcode!        Stack↓

   0000      An attempt was made to execute an          0005
   0000      illegal opcode.SpinRite's segment          17FC
   0000      is at 22F8. Error is at 064C:9CA4          4746
   0000                                                 0001
   0000       PLEASE record the location shown          064C
   0000       above, and the information given          9CA4
   0000       below, and report it to us so we          4C16
   0000       can find and fix this problem...          22F8
  -FFFF-                                               -3988-
   FFFF         eax: 47460001     esi: 000517FC         4CD8
   FFFF         ebx: 00050004     edi: 03124200         0007
   FFFF         ecx: 00000002     ebp: 00000EE4         0000
   FFFF         edx: 00000100     esp: 00000EDE         0000
   FFFF                                                 0623
   FFFF         ds: 00D9    es: 0497    fs: 0000        0000
   FFFF         gs: 5B13    ss: 00D9    fl: 0202        0001

    PLEASE record the data shown on this screen then restart this system.
```

I'm sending it in case it's of any help to you but understand I'm using the software in a way it wasn't intended to be used. Hopefully it helps in some way.

When I saw Thomas' screen capture showing that SpinRite intercepted the processor's attempt to execute an illegal instruction – and he was running the latest release 5.06 which is believed to have no such remaining loose ends – my first thought was "oh no, what now?" But then I was greatly relieved to read that this was the result of attempting to run SpinRite 6.1 within a virtual machine. Whew!

I decided to share this question because there has been a lot of interest in running SpinRite in virtual machines for various reasons. So I need to discourage and disabuse everyone of that idea. SpinRite **6.0** was and is a very tame and well-behaved "generic" DOS application – by comparison, SpinRite **6.1** really is NOT. SpinRite now assumes that it has access to true physical hardware and it does things like briefly switch the processor into protected mode, then directly alters its memory management segmentation registers to remove real mode's traditional 64K segment limitations. Due to an original oversight on Intel's part, the processor mistakenly leaves the segment sizes as they are when going into real mode. So SpinRite then returns the processor to real mode and obtains access to the first 4 gigabytes (32-bits) of the system's memory from within DOS, and through direct addressing without the use of any memory manager. This creates a hacked but quite reliable pseudo-mode known as "flat real mode" this allows SpinRite v6.1 to talk to the AHCI driver memory mapped IO up at the high end of the 32-bit address space, and to have access to 16 megabyte or even larger buffers.

So it's very safe to say that SpinRite v6.1 and an emulated environment are not going to be seeing eye to eye.

Guillermo García / @gmogarciag

Hi Steve, just listened to SN955 and your description of the certificate discovery tool. As I consider using it, I'm wondering how to reinstall a certificate that I might erase and later realize that I need?

There's actually a cool solution to this. The Windows Certificates Snap-In that Leo demonstrated last week has a number of pre-existing folders and it's possible to simply drag and drop certificates between them. There is an "Untrusted Certificates" folder which on my Win10 machine contains a "Certificate Trust List" subfolder. But if you drag a certificate from the "Trusted Root Certificate Authorities" folder onto the "Untrusted Certificates" folder it will spontaneously create a nice "Certificates" folder underneath the "Untrusted Certificates" folder which can contain, and document, any certificates that you have chosen not to trust.

You can experiment with this using any of the expired CA certificates in the "Trusted Root Certificate Authorities" since they would not be trusted anyway. If you sort by expiration date you'll see that there are a bunch there that could never be valid.

So, dragging these certificates back and forth is simple and should be error free. And if you should discover that you need one you dragged into the Untrusted Certificates folder, you can just drag it back.

A.j.druda / @ajdruda

> *Steve, do you list on https://GRC.com how to lock credit at the 4 bureaus? All I keep finding are paid sites that will do it for me.*

Believe it or not, this is all so messed up that the terms “lock” and “freeze” have important and different meanings. This listener used the term “lock” but a “freeze” is what everyone wants. And be careful not to go for a “Lock” since some of the services charge a fee for Locking. But freezing, as I said, is what you want and it’s now completely free. I don’t have a page at GRC, but Investopedia has a terrifically clear page which explains the details and provides very good links to each of the credit reporting bureaus. I have the full, long *“How to freeze and unfreeze your credit”* Investopedia link in the show notes, and it’s also this week’s GRC shortcut of the week, so you can get there with: https://grc.sc/956

https://www.investopedia.com/how-to-freeze-and-unfreeze-your-credit-5075527

Andre Couture / @nomade1999

> *Hello Steve, regarding the picture of the week for episode #955. Well I remember having to do something very similar many years ago while traveling to Europe for a presentation I had to give. I had forgotten the European power adapter. So, I used what I had on hand and in luggage to establish a connection. Do what you have to do, right? LoL*

Yikes! I suppose if there was no other choice then, yeah, one does what one must. And I can imagine that it would be something well remembered!

Peter G. Chase / @PchaseG

> *Re: Ad Hoc Adapter: I can't possibly be the first one to point out that while different countries may vary, the EU voltage is usually 240 while of course the US and Canada are 120. So... whatever appliance was on the other end of that cord very likely got fried almost immediately.*

Right. Again, everyone, don’t try this at home. (Or perhaps that should be “don’t fry this at home.”)

The Inside Tracks

I’m feeling very good about where SpinRite is today. No new significant problems have arisen for several weeks despite significant continual testing. And those people whose drives SpinRite was recently having trouble with have all reported back in that SpinRite’s latest prerelease managed to plow through their known sticky spots while effecting recovery and repair. Several have publicly stated that they’ve been amazed and impressed. So it very much feels as though SpinRite is back.

To share some sense for where my recent focus has been, I spent the past few days exploring whether I could improve SpinRite’s remaining time to work prediction. Back when SpinRite was born in the late 80’s, drives were sectored like pie slices with radials stretching out from the center which described the region of each “sector” around the circumference. In fact, we’ve grown so used to the term “sector” that it has completely lost its original meaning. The term was born when these were literally angular sectors of a disc.

The problem with this simple sectoring was that the tracks at the outside of the disc were physically longer than tracks nearer to the center; yet back then, all tracks contained the same amount of data. If all tracks contain the same amount of data, and the outer tracks have a longer circumference and the inner tracks have a shorter circumference, that meant that the individual bits were being written with reduced density around the outer tracks and increased density around the inner tracks.

Disk drive read and write electronics were originally separated from the drive in an outboard controller and drives had no intelligence at all. But IDE drives – where IDE stands for Integrated Drive Electronics – changed that by placing a drive’s read and write electronics onto each drive. Once that was done the drive was able to become something of a black box. It could simply declare how many sectors-worth of storage it contained and everything about how it worked in detail could be kept internally.

Drive designers very quickly saw that this meant they could dispense with the whole original notion of sectoring as it once was. If the outer tracks had a larger circumference, they could take advantage of that to store more data around those longer tracks. And this also allowed them to push tracks further inward toward the center of the drive by reducing the storage bitrate so as not to be cramming too many bits into too small a circumference. This allowed them to squeeze every last bit of storage into each drive and to make more complete use out of each physical disk surface.

There is one cost to that which is often overlooked, which is that the data transfer rate drops as we move inward toward the inner tracks. If we think of the beginning of the drive’s storage as the outer tracks then the end of the drive is the inner tracks where things are slower.

The reason this matters to us today is that SpinRite’s original remaining time estimation system assumed a uniform data rate across the entire drive. In other words, it performs a linear estimation. It continually monitors the total elapsed time required to get wherever it is and projects its completion time assuming that the rest of the drive will be the same as the average of everything it has seen so far. That was accurate and worked well for SpinRite versions 1, 2 and 3, but it has become less and less true as the end of drives have become slower and slower as advancing technology has pushed more data closer to the disk's center where tracks are the shortest.

The result is that for today's spinning drives, SpinRite's estimation will always underestimate the total time it will require. So, as I said, I've spent the last few days looking closely into this to see what I might be able to improve. I've learned some interesting things that I thought I'd share while they're fresh and on my mind.

What I found, after examining a handful of different multi-terabyte spinning drives, is that the ends of those drives have half the performance of their beginning tracks. Now, at first blush that sounds awful. But the decrease in performance is not linear. What we're really looking at is area rather than circumference, and as we know, area changes with the square of a circle's radius. What his means is that while a drive's data transfer performance does steadily decrease as we move inward toward its end, the decrease is very gradual until we get much closer to the end of the drive, where it begins to drop significantly.

So let's put some numbers to this. In general, SpinRite's current linear estimator takes about one minute to stabilize. Which is to say, it needs 60 seconds of operation to have established a sufficient baseline of work, in time and distance, to settle into a prediction that does not vary. And what I found through lots of experimentation on many different contemporary spinning drives is that it's necessary to add an additional 30% to SpinRite's initial front-of-drive only linear estimation. To make the math easy, say that SpinRite predicts a 10 hour run for a drive. The actual running time, due to the very end of the drive being much slower, will be 13 hours.

Here's another interesting factoid that falls out of the math and which I've verified multiple times experimentally: The first 60% of the drive requires exactly half of the total running time. So the last 40% of the drive requires the second half of the total running time. Or, expressed another way: Whatever length of time is required for SpinRite to get 60% of the way through the entire drive is the amount of time that will be required for it to finish.

I'm not certain, yet, exactly what I'm going to do with this information, but I needed to gather it to know what I was dealing with. This obviously doesn't apply to solid state storage since it's not spinning, or even to shingled magnetic storage (SMR) drives, since both or those technologies track when memory has been written to and so don't read anything from their media when SpinRite checks to see what's there. So, as we've seen, the later portions of those drives appear to perform much faster than their fronts where they are storing data. They don't slow down as we go along, they speed up.

And since drives are supposed to be "black boxes" which we just trust with our data, there's no requirement for any drive to declare what technology it's using – and many, if not most, do not give SpinRite any indication of what lies behind their interface.

The 30% rule could just be a common rule of thumb for SpinRite's users. At least initially, they know better than SpinRite whether they're testing a spinner or a solid state drive. So the rule of thumb for spinning media would be to start SpinRite, give its predictor a minute to settle down, see how long it expects to be running, then add 30%.

The other thing I'm considering is changing SpinRite's label, which currently just says "time" to "est. time" – as in estimated time. Then, once SpinRite gets to the 60% point it will have acquired sufficient awareness of the drive, having seen a gradual decrease in performance over that time, to determine that the drive **is** spinning and that as much time remains as has been spent so far. So at that point, it would adjust its timer and change the "est." to "true time" so that any time after 60% complete, its user would see a much better estimate of the time remaining. In any event, after we're finished here today I plan to make those final changes after which I believe I'll finally be content to declare SpinRite 6.1 finished and ready for the world.