Security Now! #994 - 10-01-24

Recall’s Re-Rollout

This week on Security Now!

We have the full story about the Linux remote code execution flaw. What bad stuff can happen if a domain escapes control even briefly? What social media platform is now in Russia's Roskomnadzor crosshairs? Update VLC to eliminate a potential remote code execution flaw. Tor merges with Tails for greater efficiency. Telegram announces that it will now obey court orders to disclose information. Interesting info from Bobiverse's author and some early feedback about Peter F. Hamilton's latest novel. How to keep Windows from re-asking to set up an already setup system. And... Microsoft is re-rolling out Recall. Have they actually addressed the valid concerns? Or is this just more lipstick on a pig?

⚡⚡ Electrician Wanted ⚡⚡

(Experience required this time.)

Security News

We have the news of that somewhat controversial unauthenticated Linux remote code execution vulnerability. Simone Margaritelli began his widely anticipated and still clearly annoyed exposé by writing:

> *Hello friends, this is the first of two, possibly three (if and when I have time to finish the Windows research) writeups. We will start with targeting GNU/Linux systems with an RCE. As someone who's directly involved in the CUPS project said: "From a generic security point of view, a whole Linux system as it is nowadays is just an endless and hopeless mess of security holes waiting to be exploited." Well they're not wrong! While this is not the first time I try to more or less responsibly report a vulnerability, it is definitely the weirdest and most frustrating time as some of you might have noticed from my socials, and it is also the last time. More on this later, but first.*

So first of all, the acronym "CUPS" is the abbreviation for Common UNIX Printing System. It's a modular printing subsystem for Unix-like computer operating systems, including Linux. https://www.evilsocket.net/2024/09/26/Attacking-UNIX-systems-via-CUPS-Part-I/

The Hacker News reported on what Simone Margaritelli revealed by writing:

> *A new set of security vulnerabilities has been disclosed in the OpenPrinting Common Unix Printing System (CUPS) on Linux systems that could permit remote command execution under certain conditions. Security researcher Simone Margaritelli said: "A remote unauthenticated attacker can silently replace existing printers' (or install new ones) IPP URLS with a malicious one, resulting in arbitrary command execution (on the computer) when a print job is started (from that computer)."*
>
> *CUPS is a standards-based, open-source printing system for Linux and other Unix-like operating systems, including ArchLinux, Debian, Fedora, Red Hat Enterprise Linux (RHEL), ChromeOS, FreeBSD, NetBSD, OpenBSD, openSUSE, and SUSE Linux. Simone identified four vulnerabilities which have received CVE designations.*
>
> *A net consequence of these shortcomings is that they could be fashioned into an exploit chain that allows an attacker to create a malicious, fake printing device on a network-exposed Linux system running CUPS and trigger remote code execution upon sending a print job.*
>
> *Network security company Ontinue said: "The issue arises due to improper handling of 'New Printer Available' announcements in the 'cups-browsed' component, combined with poor validation by 'cups' of the information provided by a malicious printing resource. The vulnerability stems from inadequate validation of network data, allowing attackers to get the vulnerable system to install a malicious printer driver, and then send a print job to that driver triggering execution of the malicious code. The malicious code is executed with the privileges of the printing user – not the superuser 'root.'"*
>
> *RedHat Enterprise Linux, in an advisory, said all versions of the operating system are affected by the four flaws, but noted that they are not vulnerable in their default configuration. It tagged the issues as Important in severity, given that the real-world impact is likely to be low.*

RedHat writes: "By chaining this group of vulnerabilities together, an attacker could potentially achieve remote code execution which could then lead to theft of sensitive data and/or damage to critical production systems."

Cybersecurity firm Rapid7 pointed out that affected systems are exploitable, either from the public Internet or across network segments, only if UDP port 631 is accessible and the vulnerable service is listening.

Palo Alto Networks has disclosed that none of its products and cloud services contain the aforementioned CUPS-related software packages, and therefore are not impacted by the flaws.

Patches for the vulnerabilities are currently being developed and are expected to be released in the coming days. Until then, it's advisable to disable and remove the cups-browsed service if it's not necessary, and block or restrict traffic to UDP port 631.

Benjamin Harris, CEO of WatchTowr, said in a statement shared with The Hacker News: "It looks like the embargoed Linux unauth RCE vulnerabilities that have been touted as doomsday for Linux systems, may only affect a subset of systems. Given this, while the vulnerabilities in terms of technical impact are serious, it is significantly less likely that desktop machines and workstations running CUPS are exposed to the Internet in the same manner or numbers that typical server editions of Linux would be."

Satnam Narang, senior staff research engineer at Tenable, said these vulnerabilities are not at a level of a Log4Shell or Heartbleed. He said: "The reality is that across a variety of software, be it open or closed source, there are a countless number of vulnerabilities that have yet to be discovered and disclosed. Security research is vital to this process and we can and should demand better of software vendors."

"For organizations that are honing in on these latest vulnerabilities, it's important to highlight that the flaws that are most impactful and concerning are the known vulnerabilities that continue to be exploited by advanced persistent threat groups with ties to nation states, as well as ransomware affiliates that are pilfering corporations for millions of dollars each year."

https://thehackernews.com/2024/09/critical-linux-cups-printing-system.html?_m=3n%2e009a%2e3473%2eze0ao0d6vk%2e2hio

Okay. So this is sorta what we expected, right? If it was a four-alarm fire emergency there wouldn't have been much controversy around it. In this instance, yes, there are problems. And, yes, they need fixing. But we've seen plenty of CVSS 9.8's and this collection doesn't rank up there with those.

And for his part, Simone still appears to be smarting over the backlash from his trying to get everyone's attention when he didn't feel that developers were taking it seriously enough. At the end of Part 1 of his detailed write up he concluded:

You will maybe be thinking now "wow, that's a lot of stuff to read, code, RFCs, PDFs of forgotten standards, this research must have been so tiring", but in reality this was a weekend worth of rabbit holes, this was the fun part. The actual work, the heavy, boring stuff started when on September 5, after confirming my findings, I decided to open a security advisory on

the OpenPrinting cups-browsed repository and do what to me was the right thing to do: responsible disclosure.

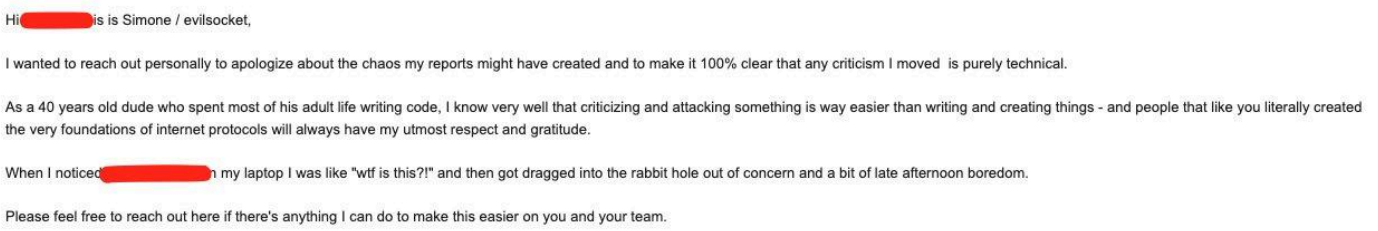
I won't go into the details of the initial conversation, or the ones that followed. You are free to read them (if they will ever open any of the threads and you are willing to read 50+ pages of conversations) or not, and make your own opinion.

While the research only took a couple of days, this part took 22. And this part was not fun. I will only say that to my personal experience, the responsible disclosure process is broken. That a lot is expected and taken for granted from the security researchers by triagers that behave like you have to "prove to be worth listening to" while in reality they barely care to process and understand what you are saying, only to realize you were right all along three weeks later (if at all).

Two days for the research, 249 lines of text for the fully working exploit. Twenty-two days of arguments, condescension, several gaslighting attempts (the things I've read these days … you have no idea), more or less subtle personal attacks, dozens of emails and messages, more than 100 pages of text in total. Hours and hours and hours and hours and f-ing hours. Not to mention somehow being judged by a big chunk of the infosec community with a tendency of talking and judging situations they simply don't know.

Let that sink in for a moment … What The Actual F!!

And we're not talking about time spent on fixes while I was impatient and throwing a tantrum on twitter. The actual fixes (or a part of them) started being pushed much later. The vast majority of the time has been spent arguing whether or not these were issues worth considering. While I was trying to report that there's something bad that should be addressed ASAP, the devs were being dismissive (and pushing other code, also vulnerable, for other functionalities instead of fixing) because I dared to criticize the design of their software. While at the same time I was trying to reach out privately to de-escalate and assure whoever was getting offended that my intent was not adversarial:



Hi [redacted] is is Simone / evilsocket,

I wanted to reach out personally to apologize about the chaos my reports might have created and to make it 100% clear that any criticism I moved is purely technical.

As a 40 years old dude who spent most of his adult life writing code, I know very well that criticizing and attacking something is way easier than writing and creating things - and people that like you literally created the very foundations of internet protocols will always have my utmost respect and gratitude.

When I noticed [redacted] n my laptop I was like "wtf is this?!" and then got dragged into the rabbit hole out of concern and a bit of late afternoon boredom.

Please feel free to reach out here if there's anything I can do to make this easier on you and your team.

To the people that more or less directly questioned my integrity, accused me of spectacularization and of spreading FUD on my socials: I don't do this for a living. I don't need CVEs to get a job or to prove how good my kung-fu is. Or any attention other than what my projects and research already provide. I don't play InfoSec Influencer™ like many. My mission was to interrupt the triagers focus until they re-prioritized. When I saw that what I thought was pretty serious was being dismissed as an annoyance, I used the only platform I had plus a pinch of drama as a tool to have them f-ing re-prioritize. And it worked, wonderfully, more fixes happened after two tweets than with all the arguing and talking. So don't hate me, hate the system that forced me to do that in order to be taken seriously.

Our takeaway here is that some unlikely to be exploitable yet important flaws were found and

they will be fixed in future editions of the Linux and BSD code. Simone did a good thing. And the open source ecosystem is better for his willingness to push. We'll never know whether he might have obtained the same results without making such a fuss. But it is good that this will be fixed.

Risky Business News posted their summary, which adds some interesting bits and additional details. They wrote:

> *Threat actors are scanning the internet for UNIX systems that are exposing their printing ports in an attempt to exploit a set of four vulnerabilities in the CUPS printing component. The vulnerabilities were discovered by Italian security researcher Simone Margaritelli earlier this year and were disclosed at the end of last week. They impact CUPS, the Common UNIX Printing System, an open-source component to allow UNIX systems to function as print servers.*
>
> *The four bugs are part of an exploit chain that can allow an attacker to deploy a malicious printer, have the printer indexed by a victim's CUPS server, plant malicious code on the CUPS server (UNIX system) inside a PPD file, and have the malicious code from the PPD file executed when a user launches a print job via the attacker's (malicious) printer.*
>
> *The exploit chain is in this order: CVE-2024-47176, -47076, -47175, and -47177.*
>
> *Besides Margaritelli's write-up explaining how the four bugs work, other analyses on the four are also available via Akamai, Rapid7, Elastic, Tenable, Qualys, DataDog, and AquaSec. The bugs received a lot of attention and were extremely over-hyped over the past week after Margaritelli posted about them on Twitter before patches were released.*
>
> *Let's just say they are not as bad as they were made out to be. They don't impact all Linux distros (only a few, actually), they're only exploitable in very limited scenarios, and the 9.9 CVSS score should have been lower. Yes, they're bad bugs that are easy to exploit, but they're not the Linux world-ending kind (like Heartbleed, for example).*
>
> *But regardless of their severity and all the weird conditions needed to exploit the bugs, threat actors don't care. After Margaritelli and others published proof-of-concept code at the end of last week, threat actors began scanning the internet for UDP port 631, which is the port on which the CUPS server listens for new printers announcing their presence. If this port is exposed on the internet, then bad things are about to happen to your CUPS server in the coming days.*
>
> *Even if CUPS ships disabled by default on most distros, according to Shodan, there are currently over 75,000 systems running CUPS exposed over the internet, which is quite an attractive piece of pie if you're an attacker. Other scans have these numbers at over 107,000, but they could be even bigger than this.*
>
> *Mitigating the vulnerability should be pretty easy. Just disable, remove, or update CUPS. You shouldn't be running that anyway.*

The CRUCIAL importance of Domain Control Security
The news last week was that Ether.fi, a DeFi – Decentralized Finance platform – was the target of a DNS hijack after threat actors took control of its Gandi account. On September 24th, by abusing Gandi.net's account recovery mechanisms, bad guys managed to switch Ether.Fi's registered nameservers to those that they controlled. Since Ether.Fi received account recovery notification, within three hours the changes had been reverted and Ether.Fi's account had been successfully locked to prevent further tampering.

In the reporting of this everyone appears to be breathing a sigh of relief. But a LOT can be done immediately upon the takeover of a domain. For example, valid web server domain certificates can be immediately obtained from any registrar since proof of domain control is all that's required. And due to the fact that certificate revocation is a myth, those certificates will remain valid throughout their life. Not only can those certificates be used to host a spoofed website if a victim's traffic can be rerouted, but those same certificates can be used to sign spoofed email from the victim domain and it will pass all SPF, DKIM and DMARC validation.

My point is, it is likely that for commercial entities owning valuable domains, security is more important at their domain registrar than anywhere else. I know that many of the listeners of this podcast have their own domains. So, if you were only to use multi-factor authentication in one place, I'd choose authenticating to your domain's registrar and doing anything possible to limit anyone's ability to perform malicious account recovery.

Recall how LastPass suffered that first security event and thought that everything was fine. But then, later, the bad guys were able to use some of the information they gleaned from the first attack to launch a deeper and more destructive attack? That sort of thing might well plague these Ether.Fi folks in the future. They think everything's been buttoned up. But that brief nameserver switcheroo may have provided the bad guys with everything they were really after.

Roskomnadzor strikes a discordant note
The social media platform, Discord, is on the way to being banned in Russia. Our favorite Russian internet watchdog Roskomnadzor just added Discord to its registry, which is the first step in formally blocking access to the service within Russia's borders.

VLC gets a security update:
Just a note for users of the extremely popular VLC VideoLAN player. The project has released a patch to repair an integer overflow vulnerability via a maliciously crafted MMS stream. The update notes that this could be used to crash VLC at a minimum and that although no one had ever created a remote code execution, the possibility cannot be ruled out. VLC media player **3.0.21** addresses the issue.

Tor and Tails Merge
We've had a lot of fun in years past looking at the Tor Project with its unique privacy preserving "Onion Routing" technology. The Tor system wraps an outbound Internet packet in multiple successive layers of encryption where the private key to decrypt each layer is only known to the

specific router to which the "onion packet" is sent. So after wrapping the outbound packet, the sender sends the multiply-wrapped "onion" to the first router, which is only able to remove the outer layer of encryption to reveal the address of the next onion router in the sequence. It cannot determine the packet's final destination nor its contents because that's still hidden by multiple additional layers. Although that first router knows the sender's IP, since it just received a packet from that IP, the second router does not – it only knows the IP of the first router.

When the onion reaches the second router it, and only it, is able to decrypt and remove that layer of the onion, thus revealing the IP of the third router in the sequence. And that second router only knows the IP of the first router.

Once the third router receives the onion, only it is able to remove what is not the outer layer of the onion's encryption to reveal the packet's true destination, and it has no idea whatsoever who originated that packet since that's three hops back.

That clever multi-layered multi-wrapped encryption is the essence of the Onion Routing system whose entire purpose is to give Internet users something that's completely lacking from the Internet's normal point-to-point routing scheme, which is a high degree of anonymity for the sender.

The other privacy-centric OS project is Tails. Tails is an operating system which is bootable from a USB thumb drive. The Tails site bills itself as "Your secure computer anywhere" and explains the OS's purpose:

To use Tails, shut down your computer and re-start it with your Tails USB stick instead of starting the Windows, macOS, or Linux OS. You can temporarily turn your own computer into a secure machine. You can also stay safe while using the computer of somebody else. Tails is a 1.5 GB download and takes ½ hour to install. Tails can be installed on any USB stick of 8 GB minimum. Tails works on most computers less than 10 years old. You can start again on the system's original operating system after you shut down Tails.

You don't have to worry about the computer having viruses because Tails runs independently from the other operating system and never uses the hard disk. But, Tails cannot always protect you if you install it from a computer with viruses or if you use it on a computer with malicious hardware, like keyloggers.

Tails always starts from the same clean state and everything you do disappears automatically when you shut down Tails. Without Tails, almost everything you do can leave traces on the computer:

- *Websites that you visited, even in private mode*
- *Files that you opened, even if you deleted them*
- *Passwords, even if you use a password manager*
- *All the devices and Wi-Fi networks that you used*

On the contrary, Tails never writes anything to the hard disk and only runs from the memory of the computer. The memory is entirely deleted when you shutdown Tails, erasing all possible traces.

We're revisiting these two important projects today because last Thursday under the blog headline *"Uniting for Internet Freedom: Tor Project & Tails Join Forces"* – they announced their merger. The two projects realized that there was a great deal of duplicated effort with managing and fund raising and operational overhead. The Tor blog wrote:

> *Today the Tor Project, a global non-profit developing tools for online privacy and anonymity, and Tails, a portable operating system that uses Tor to protect users from digital surveillance, have joined forces and merged operations. Incorporating Tails into the Tor Project's structure allows for easier collaboration, better sustainability, reduced overhead, and expanded training and outreach programs to counter a larger number of digital threats. In short, coming together will strengthen both organizations' ability to protect people worldwide from surveillance and censorship.*
>
> *Countering the threat of global mass surveillance and censorship to a free Internet, Tor and Tails provide essential tools to help people around the world stay safe online. By joining forces, these two privacy advocates will pool their resources to focus on what matters most: ensuring that activists, journalists, other at-risk and everyday users will have access to improved digital security tools.*
>
> *In late 2023, Tails approached the Tor Project with the idea of merging operations. Tails had outgrown its existing structure. Rather than expanding Tails's operational capacity on their own and putting more stress on Tails workers, merging with the Tor Project, with its already larger and established operational framework, offered a solution. By joining forces, the Tails team can now focus on their core mission of maintaining and improving Tails OS, exploring more and complementary use cases while benefiting from the larger organizational structure of The Tor Project.*
>
> *This solution is a natural outcome of the Tor Project and Tails' shared history of collaboration and solidarity. 15 years ago, Tails' first release was announced on a Tor mailing list, Tor and Tails developers have been collaborating closely since 2015, and more recently Tails has been a sub-grantee of Tor. For Tails, it felt obvious that if they were to approach a bigger organization with the possibility of merging, it would be the Tor Project.*
>
> *The team lead for Tails OS said: "Running Tails as an independent project for 15 years has been a huge effort, but not for the reasons you might expect. The toughest part wasn't the tech — it was handling critical tasks like fundraising, finances, and HR. After trying to manage those in different ways, I'm very relieved that Tails is now under the Tor Project's wing. In a way, it feels like coming home."*

Telegram changes its long-standing "zero cooperation" policy

As we noted a few weeks ago, Telegram's founder and owner, Pavel Durov was first detained then arrested in France after authorities decided to hold him directly responsible for the many abuses known to be flourishing within the totally unmoderated and unfiltered protection of Telegram's service. France's strategy appears to have worked since Telegram recently made some waves by amending its privacy policy and agreeing to comply with court orders requiring it to share its users' phone numbers and IP addresses with law enforcement. So Telegram's cooperation will now extend to various criminal investigations, expanding beyond the previous limit of only terror-related offenses.

Closing the Loop

Hello. I'm a long time listener and a much longer time developer. Currently I write mostly for mobile and have apps on the Android play store. From time to time I receive emails from "companies" that want to buy my app and my [not many] users. But yesterday I received something new. This guy wants to "rent" my account to publish his own junk. As you can see, he doesn't value my reputation much… Email below:

From: ***George*** *<george######@gmail.com>*
Date: Sat, Sep 28, 2024 at 10:21 PM
Subject:
To: <#########@gmail.com>

Good day GreenSpot, this is Bytom Gaming Hub. We are reaching out to partner with Google Play Console Account owners for a lasting collaboration to publish our app.

Our compensation plan includes:

$70 for each app upload
$10 for each app update
$50 every 7 days while the app is on your account

If you are interested in collaborating with us, please contact us via WhatsApp at +###########.

Yours Sincerely, Bytom Gaming Hub.

Two years ago, in 2022, Cory Doctorow brilliantly coined the term "enshitification". His use was intended to be aimed at a single company's decline in product quality over time. As Wikipedia describes the term: *"Enshittification (alternately, crapification and platform decay) is a pattern in which online products and services decline in quality. Initially, vendors create high-quality offerings to attract users, then they degrade those offerings to better serve business customers, and finally degrade their services to users and business customers to maximize profits for shareholders."* So Cory didn't define the term to be used more broadly. But it's so tempting to also use the term to describe what we're all feeling, overall, about the decline in the quality of the Internet's service as a whole. So that term comes to mind when we see low- quality apps attempting to pay their way into the accounts of higher-quality apps as a means of riding their reputations. The only reason someone would pay to have their app offered within someone else's account would be because the value derived from advertising would be more than the cost of doing so. The overall result, of course, is the gradual "enshitification" of the platform as the valuable reputation of developers is cashed out and watered down to no longer carry the value it once did.

Our listener who shared this was clearly unmoved by the offer. But it's foreseeable that many others would jump at the chance to obtain some additional income from monetizing whatever loyalty their name may have earned.

Marv & The Bobiverse

Hi Steve, I wanted to give some feedback on the availability of "Not Till We Are Lost: Bobiverse, Book 5". After hearing you mention it was published this month, I've been waiting for the Kindle edition on Amazon ...and waiting ... and waiting. It turns out we Kindle readers will have to wait a few months due to the author Dennis Taylor's agreement with Audible:
http://dennisetaylor.org/wheres-the-whatever-version/
***Marvin Rhoads** / Senior Network Security Engineer*

From Dennis' FAQ:

Question: Where's the Kindle version?

***Answer:** Audible likes to have an exclusivity deal with authors. During negotiations, they will try for up to a six month gap before the text versions are produced. The inducements to the author are: Audible pays for the narrator, Audible pays for the cover, Audible does marketing, Audible offers a much larger advance. Audible is also responsible for about 2/3 of my total income, so they are by definition my primary publisher.*

Fortunately my agent, who is a bit of a pit bull, has kept the exclusive period down to four months. So the text version (for the current contracts, anyway) will always come out 4 months after the Audible version.

While I was there I read the rest of Dennis' FAQ and his irreverent personality, which so many of us have enjoyed in his novels, shows through clearly. Two additional FAQ entries which also provide some additional interesting background are:

Question: Where's the epub or other version?

***Answer:** Amazon only lists your work in Kindle Unlimited if you go exclusive with Amazon for the electronic version. That means no epub or Kobo or Google Play version. Before you ask, KU is probably about 25% of my non-Audible revenue, and that's still a serious chunk of change. See below for discussion of fiduciary greed.*

When I originally self-published Outland, I initially went wide (Kobo, epub, Google Play, etc). If I made so much as a penny from any of those other channels, I don't remember it. When I switched to Amazon exclusivity and KU, my Amazon revenue went up about 20%. So there is literally no inducement for me to consider going wide with my novels.

Question: So it's all about money?

***The answer is oh hell yes.** This writing thing isn't a hobby, and I'm not independently wealthy–I have to pay a mortgage, me and my family have grown accustomed to eating regularly, and I'd like the bank to not take my car back. I literally quit my day job so I could write full-time, which means I can produce books a lot more quickly, but also means I have to be concerned about the financial aspects of my 'job'. So when they wave a wad of bills under my nose, I pay attention. Sorry, that's just the way it is.*

So the good news is that by creating a hit series, Dennis has been able to, as he put it, literally quit his day job, which will result in a higher rate of novels in the future.

Listener Ben shared a welcome tip:

Hey Steve! I recall multiple complaints of Windows 10 asking to be backed up every time there's some sort of update, when many of us already have our own backup solution. Today I decided to see if there was a way to disable this. Turns out, there is!

Settings > System > Notifications & actions: Uncheck the option "Suggest ways I can finish setting up my device to get the most out of Windows."

Ben: Thank you, thank you, thank you! I had never looked, and I had no idea that such an option was available. I, also, am plagued by Windows incessantly promoting its own solutions and asking me to "re-setup" that stuff long after Windows has been configured the way I want it to be. And I have no doubt that many other listeners will be similarly relieved to learn that there's a way to shut that off once and for all. And since it occurred to me that it might be a nice addition to a next release of GRC's "InControl" freeware, I've made a note of that in that project so I can see about adding that.

Listener Matt wrote: "More Experian Woes"

Hi Steve, Because you mentioned some questionable security practice with Experian, I thought I'd mention that I am inadvertently the email-of-record for someone else's Experian account.

I was an early Gmail adopter and have the Gmail address of first initial, lastname at Gmail dot com. I routinely get messages to others in the world who share my first initial and same last name. Occasionally, someone will sign up for services and enter my email address by mistake (forgetting to add whichever qualifiers distinguish their email from mine). It's usually harmless, but someone recently signed up for an Experian account with their information and my email address.

Now I receive email messages every time they have a credit alert. Conscious organizations have a single click opt-out for messages, but for me to turn this off I have to log in to Experian as the user. This wouldn't be a problem because I could easily reset the account password, as I own the email address behind it, but I don't want to be exposed to any more of their personal details than I already am. It seems that Experian doesn't bother with an email verification loop when setting up accounts, or at least they didn't when this person set theirs up.

What a mess. At this point it's not even clear how they could go about untangling it. We've looked extensively at how, due to the universal presence of "I forgot my password" links, the security of our email is really what **all** of our logon security comes down to. Usernames and passwords and even multi-factor authentication are all only logon accelerators since everything falls back to email.

So in this case, what happens when this account owner forgets their password and attempts to use the "I forgot my password" loop, but that confirmation mail lands in Matt's inbox because they always had it wrong on their account, due to a lack, as Ben said, of any email confirmation loop? Now what?

And I thought the other thing Matt noted was interesting: Because he was an early adopter of gmail, he was able to obtain a short and convenient – but as it turns out, not very specific or unique – email address. So now, as he explained, he's receiving many more mistaken emails than he would if he had, for example, taken that early opportunity to obtain his full first and last name. Of course, there was no way of knowing back then what Google's mail service would become. One of the things I've learned during my recent dip into the land of email is just what gmail has become: Nearly one quarter (23%) of all of the email addresses GRC has had contact with over the past 20 years has been gmail (this doesn't count non-gmail or googlemail domains that gmail is handling). I had no idea, until recently, just how prevalent gmail is. But it makes sense for savvy Internet users who are smart enough not to get locked into their own ISP's email services. As we know those are beginning to be shut down or pawned off to another service, as Cox recently did with Yahoo.

Edward McDonald

Hello Steve. I recently updated to iOS 18 and saw were Apple how has an app for their password manager. I wondered your thoughts on it versus some of the other password manager software (like 1Password). Thanks, Ed

Since I'm hanging back with older Apple hardware, iOS 18 is not an option for me. But when we talked about this back at announcement time, what I recall was that what Apple was doing was mostly pulling together what they already had for password management, which was buried and scattered around in iOS, all in one place and giving it a more formal UI presence. The presence of Passkeys, and the need to manage them, increased the need for iOS's password management to be more explicit. So the value of any 3rd party password manager, whose primary benefit would be much wider cross-platform, cross-ecosystem credential synchronization, is neither changed nor diminished with these changes to iOS 18.

A listener named "E" asked about our mailing solution:

Hi Steve, We run a self-coded email system for doing weekly mailshots. We would like to shift to 3rd party code (but remain self hosted). Recently, when you described your modernization effort on your email system, I seem to recall that you bought/licensed a system and wrote your own code around that. I looked back at security now transcripts but I seem to have missed it? Could you put me straight on this point and if you did license, point me in the right direction? (happy to take my answer "on the air" or by email whatever suits you best). Great show and so happy you will go beyond 999! E.

I was glad to see this question because the more I use the solution I found, the more impressed I have become. When John Dvorak recently asked what I had found, since he wanted to move away from MailChimp in order to obtain more control, I was delighted to tell him. And the same goes for our listener 'E' and everyone else. The system is https://www.nuevomailer.com/. As with anything new and sophisticated, it took a while for me to fully "grok" the way it works. But the more I've used it, the more I've grown to appreciate its power. It can be used in a simple production capacity, but also as a powerful emailing workstation, which is the way I've been using it as I've been shepherding GRC's creaky old email list though today's hyper spam-focused

emailing climate. As 'E' noted, I wrote my own sign-up front end for it, but it offers a fully working form based, multiple list double opt-in sign-up system as well. And it has been worth so much more than the $139 that its Greek author, Panos, asks. So, again, nuevoMailer receives my highest recommendation.

And speaking of email...

SpinRite

DJ

I just wanted to let you know that I received the Spinrite 6.1 upgrade email earlier today in my AOL/Verizon inbox—it didn't end up in my spam folder! I've been a dedicated listener of Security Now since episode #3—NAT Routers as Firewalls. I'm also a proud Spinrite owner and user, and I've successfully recovered priceless files for friends and family. Even as an avid listener, I shouldn't admit this, but I've recovered some of my own files that weren't backed up! Now, after the latest use of Spinrite, my SSD is transferring data like it's new again!

Naturally, all of that is music to my ears. Over the past couple of weeks, but primarily last week, I've been working to get 20 years of past SpinRite 6 owners notified of the availability of a no-charge upgrade to v6.1. I finished that on Thursday. Everything went well, but Microsoft appeared to be unhappy with the level of spam complaints emailing to these very old email addresses was generating. I have a test list of 53 people from GRC's newsgroups who have volunteered to receive various test mailings while I've been working to bring up GRC's email system. On Saturday, a test mailing to the list bounced back all 6 of those people whose domains were handled by Microsoft: outlook.com, hotmail.com and live.ca. So I found their postmaster tools and asked about the block on our sending domain. Their reply on Sunday was that they had no record of any block. So I did another test mailing and, sure enough, none of those emails bounced.

However ... and this is the reason I'm mentioning it today ... while the email was not flatly rejected, it was still routed into outlook's JUNK folder. So I would not be surprised, and in fact I expect ... that our listeners who have subscribed to the weekly Security Now! mailings may need to dig them out of their JUNK folders for a few weeks. And it would again be appreciated if those listeners would mark them as "not junk" to help me apologize to Microsoft for mailing to so many old and dead addresses and to reestablish our reputation as a non-spammer.

Sci-Fi (John Slaina aka JammerB)

Hi Steve! I understand not wanting to start it until the series is complete. That's what I do with Frontiers Saga. I like to read all 15 back to back. (3 more "episodes" and I can devour Part 3) ***But it's Peter F. Hamilton!*** *I'm half way through; looking forward to rereading it before part two comes out. Enjoying it immensely. I will say it is great to have a book that is hard to put down. I have many things I can be doing with all my free time... I can tell you that getting back into this book is always at the top of the list. It's a little weird looking in from the outside of TWiT, but I will continue to enjoy all the content TWiT produces. Take care, John*

Recall's Re-Rollout

Listener "Mike" wrote:

> *Leo and Steve, While I can see some value in having a personal AI, running from a user created and selected database, I see far, far, more danger in this both currently and in the future. I believe that it would require a redesigned PC and OS. It could involve partitions or multiple memory devices. It could also involve multiple data incompatible OS's, and CPU's. Perhaps running in some kind of sandbox. It must be air-gapped from the Internet and it certainly cannot be connected to a MS account. Authentication would be local only, perhaps with a Yubikey, when querying the AI. It would be independent of any Windows, lacking security, functions. An application running alongside but not actually in Windows. Most likely different application for storing and retrieving data, as well. Just doodling some ideas, Mike.*

Way back when the Internet happened, Microsoft had ramped up to compete with AOL, CompuServe, The Source and other dial-up services with what they were calling MSN – their Microsoft Network. The sudden surging interest in the Internet appeared to take Bill Gates and Company by surprise. Windows at the time had local area networking with Microsoft's own LAN Manager and with 3rd-parties such as Novell. But there was really no sort of WAN networking.

So they found a TCP/IP stack somewhere, hung it onto Windows and put Windows onto the Internet. The only trouble was, the phrase "Windows Network Security" at the time was an oxymoron. And that was the motivation for my own initial entry into the world of online security, with the creation of ShieldsUP! to show people that if they had previously shared their "C" drive on the private LAN, then the entire world could now see and browse around inside their machine's "C" drive.

The sudden appearance of the Internet represented a **discontinuity** in the use of Windows. Microsoft was caught off guard without a good solution, so they shipped what they had, despite the fact that it was a total security disaster. In the beginning, when ShieldsUP! was born, my web server was showing its visitors the contents of their hard drives in a browsable tree.

I was reminded of this, by analogy, because Recall represents a similar discontinuity in the use of Windows. This is due to the fact that having an agent locally storing its user's entire computer usage history in machine-accessible form is not something that has ever been done before, and it represents a massive change in the system's security profile. It's not sufficient to say *"oh, we'll just encrypt it"* or *"don't worry, it's protected by Windows Hello."* Anyone trained in security knows that none of that is anything but feel-good nonsense. It's a salve for the masses.

Just as once upon a time Windows had never needed to have the kind of network security that was required to safely attach it to the Internet, Windows has never needed to have the kind of local desktop security that's required to allow it to safely accumulate and protect all of its user's past activity over time.

The good news is, these fundamental truths were self-evident to anyone and everyone trained in security and pretty much all of them started screaming and posting when Microsoft blithely dropped a functioning Recall beta into Windows CoPilot+ PCs without any sort of protection... exactly as, back in Windows 95, they hooked Windows to the Internet without any preparation.

What's different between now and then, is that we've lost our innocence. Today, the world has 30 years of experience with security – and with Windows – and even if Microsoft tends to forget that major new features really do need some peer review, the rest of the world is here to remind them. And thanks to the Internet, the rest of the world has a microphone.

So last Friday, David Weston, Microsoft's Vice President for Enterprise and OS Security, posted a comprehensive update on the state of Recall under the title: *"Update on Recall security and privacy architecture"*. My first reaction to what they have done is to judge this as extremely impressive. Microsoft clearly has some big guns who could not have been involved in Recall's initial design. There was no sign of them, then. But they are now, and any reading of Recall's new protection system design would have to be prefixed with the statement to the rest of the security industry: "Message received".

Now, please don't read this and assume that I will now be running Recall on my machines. In the first place, only Windows 11 will offer that option and I'm only now feeling really good about Windows 10. So I'll be Recall-free for the foreseeable future. But I know that many of our listeners, their friends, families and others whose security they care about will be running Recall. So it's definitely worth updating ourselves on what Microsoft has wrought.

First off, on the "all or nothing" front, it appears that the option to remove Recall entirely, which someone at Microsoft said was a bug, not a feature, David is not saying is, in fact, a feature. Under the heading "The user is always in control." David's posting says:

- *Recall is an opt-in experience. During the set-up experience for Copilot+ PCs, users are given a clear option whether to opt-in to saving snapshots using Recall. If a user doesn't proactively choose to turn it on, it will be off, and snapshots will not be taken or saved. Users can also remove Recall entirely by using the optional features settings in Windows.*

Now, that said, Microsoft clearly wants everyone to turn this on. David's posting shows a screenshot of the Recall offer, at least as it stands now. And, of course, it's all glowing happiness.

The screen that comes up has the catchy offer: "Unlock your photographic memory with Recall", which reads:

If you allow Recall to save snapshots, an image of your screen will be saved every few seconds. This will create a photographic memory for you of the apps, websites, documents and images you've seen on your PC.

Then we have three benefits described:

- *Easily find what you need: Scroll through a timeline of your snapshots or describe what you're looking for – even text or images within a snapshot.*
- *Pick up where you left off: From a snapshot, you can seamlessly return to documents, images, emails, and webpages as you left them.*
- *You're always in control: You choose if and when snapshots are saved and only you can access them. In Settings, you can also choose which apps and websites to filter out of snapshots, delete snapshots, or change Recall settings anytime.*

This page concludes with the question: "Start saving snapshots of your screen on Recall?" with options to "Learn more", "Yes, save" or "No, don't save"

That all sounds great and we didn't expect Microsoft to laden their invitation with any concern over the security of the system's stored snapshots. After all, it says under the 3rd benefit that "only you can access them." So, okay, then.

But here's were we get into the part that impressed me and which made it clear that what is now being presented came from some other place entirely than the initial entirely lame first Recall beta preview. Here's what Microsoft has engineered after clearly awakening to the fact that there really IS an awesome responsibility associated with gathering and locally storing all of this potentially very personal and private data. They highlight three features:

Sensitive data in Recall is always encrypted and keys are protected: *[Easy to claim, but they elaborate] Snapshots and any associated information in the vector database are always encrypted. The encryption keys are protected via the Trusted Platform Module (TPM), tied to a user's Windows Hello Enhanced Sign-in Security identity, and can only be used by operations within a secure environment called a Virtualization-Based Security Enclave (VBS Enclave). This means that other users cannot access these keys and thus cannot decrypt this information.*

Recall services that operate on snapshots and associated data are isolated: *Within Recall, the services that operate on screenshots and associated data or perform decryption operations reside within a secure VBS Enclave. The only information that leaves the VBS Enclave is what is requested by the user when actively using Recall.*

Users are present and intentional about the use of Recall: *Recall leverages Windows Hello Enhanced Sign-in Security to authorize Recall-related operations. This includes actions like changing Recall settings and run-time authorization of access to the Recall user interface (UI). Recall also protects against malware through rate-limiting and anti-hammering measures. Recall currently supports PIN as a fallback method only after Recall is configured, and this is to avoid data loss if a secure sensor is damaged.*

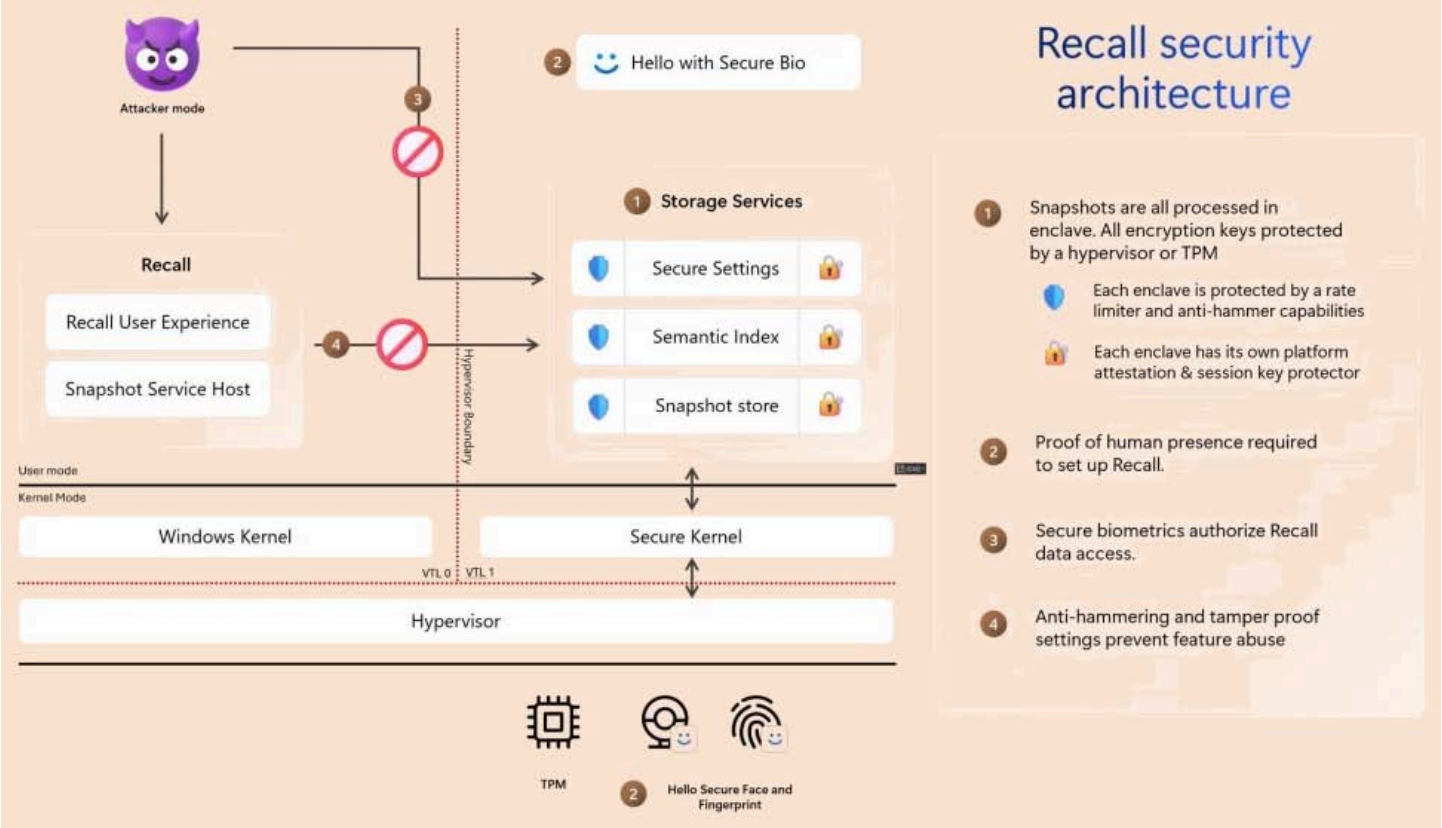
So Microsoft is using its hypervisor-based machine virtualization to create as isolated a container for this information as is possible without requiring an entirely new hardware design.

Microsoft explains what this means under "Recall security model" by writing:

> *Recall snapshots and associated data are protected by secure VBS Enclaves. VBS Enclaves use the same hypervisor as Azure to segment the computer's memory into a special protected area where information can be processed. Using Zero Trust principles, code in these enclaves can use cryptographic attestation protocols to safeguard that the environment is secure before performing sensitive operations, such as snapshot processing. This area acts like a locked box that can only be accessed after permission is granted by the user through Windows Hello. VBS Enclaves offer an isolation boundary from both kernel and administrative users.*
>
> *Recall snapshots are available only after you authenticate using Windows Hello credentials. Specifically, Windows Hello Enhanced Sign-in Security biometric credentials protect your privacy and actively authenticate you to query your semantic indices and view associated snapshots.*

Biometric credentials must be enrolled to search Recall content. Using VBS Enclaves with Windows Hello Enhanced Sign-in Security allows data to be briefly decrypted while you use the Recall feature to search. Authorization will time-out and require the user to authorize access for future sessions. This restricts attempts by latent malware trying to 'ride along' with a user authentication to steal data.

I'll just interrupt to note that none of this was present before. What they released earlier wasn't even half baked – it wasn't even warm. They then repeat the various UI features as privacy controls: Snapshot saving can be stopped and resumed, snapshots can be deleted, private browsing will never be recorded, etc. But what really makes the difference here is Recall's security architecture. And this is properly where most of their effort has been invested:



The core components of the Recall architecture are the following:

Secure Settings: *A protected data store used within the VBS Enclave, which stores security configuration data for Recall. To make any changes to security-sensitive settings a user must authorize the actions taken within the enclave to prevent malicious tampering. In addition, the settings are secure by default, meaning if tampering is detected they will revert to secure defaults.*

Semantic Index: *The semantic index converts images and text into vectors for later search. These vectors may reference private information extracted from snapshots, so these vectors are encrypted by keys protected within the VBS Enclave. All query operations are performed within the VBS Enclave.*

Snapshot Store: *Contains the saved snapshots and associated metadata, including any launch URIs provided by apps integrating with* ***Recall User Activity API****, as well as data like the time of the snapshot, title bar string, app dwell times, etc. Each snapshot is encrypted by individual keys and those keys are protected within the VBS Enclave.*

Recall User Experience: *The UI experience that users leverage to find things they have done on their PC, including timeline, search and viewing specific snapshots.*

Snapshot Service: *Background process that provides the run time for saving new snapshots, as well as querying and processing data returned by the VBS Enclave. Recall's storage services reside in a VBS Enclave to protect data, keys and tampering from malware or attackers operating on the machine. Recall components such as the Recall UI operate outside the VBS Enclaves and are untrusted in this architecture.*

Because the Snapshot Service must release information requested by a user by design, a key tenet of the design is to reduce the potential for exfiltration of data outside the normal use of the Recall system.

Processes outside the VBS Enclaves never directly receive access to snapshots or encryption keys and only receive data returned from the enclave after authorization. The authorization period has a timeout and anti-hammering protection that limit the impact of malicious queries. The Snapshot Service is a protected process further limiting malicious access to memory containing the data returned from the query outside the VBS Enclave. Protected processes are the same technology used to protect anti-malware and the Windows LSA host from attacks.

Lastly, the Recall VBS Enclave leverages concurrency protection and monotonic counters to prevent malicious users from overloading the system by making too many requests.

Okay. So it should be completely clear that what we have today is no longer a set of SQL database files containing the user's history snapshots that were found lying around in a user's private directory in that initial public preview release. This is an entirely new ballgame.

One thing there that caught my eye was the mention of a ***Recall User Activity API***. Huh? What's that? Some poking around discovered that this is the means by which it's possible to have Recall return to some past situation and allow the user to pick up from there. As a developer, I was extremely skeptical about Windows' ability to do that, since it would have required snapshotting not just the system's screen but its entire running context, and that's not possible. It turns out that this *"Recall User Activity API"* is the means by which apps which have been modified to be "Recall Ready" can cooperate with Recall to make that sort of rewind-to-a-past-state possible.

For developers, under the heading *"Use Recall in your Windows app"* Microsoft explains:

> *For those who opt-in by enabling "Recall & snapshots" in Settings, Windows will regularly save snapshots of the customer's screen and store them locally. Using screen segmentation and image recognition, Windows provides the power to gain insight into what is visible on the screen.*

As a Windows application developer, you will now be able to offer your app users the ability to semantically search these saved snapshots and find content related to your app. Each snapshot has a UserActivity associated that enables the user to relaunch the content.

A UserActivity refers to something specific the user was working on within your app. For example, when a user is writing a document, a UserActivity could refer to the specific place in the document where the user left off writing. When listening to a music app, the UserActivity could be the playlist that the user last listened to. When drawing on a canvas, the UserActivity could be where the user last made a mark. In summary, a UserActivity represents a destination within your Windows app that a user can return to so that they can resume what they were doing.

To engage with a UserActivity your Windows app would call: UserActivity.CreateSession. The Windows operating system responds by creating a history record indicating the start and end time for that UserActivity. Re-engaging with that same UserActivity over time will result in multiple history records being stored for it.

So that explains a lot about how this can possibly work: In short: It doesn't, until and unless the apps the user is running explicitly add support for it. I'm sure users will be able to turn back the clock to look at what they were doing and to read saved screens. But jumping back into an app at that point in time will require explicit support from the app. I'm sure that Edge and Office and Microsoft's Windows apps will offer this. And it might become a competitive feature that other apps will need to add to remain at feature parity. We'll see how that goes.

There's a bit more that I don't want to skip over in the interest of presenting the whole story. Under "Additional architectural properties that are key to security for Recall:" Microsoft adds:

Bound and verified VBS Enclaves: Encryption keys used by Recall are cryptographically bound to the identity of the end user, sealed by a key derived from the TPM of the hardware platform and are performed entirely within the trusted boundary of Virtual Trust Level 1 (VTL1).

Virtualization Based Security (VBS): The hypervisor provides the secure enclave environment, which loads integrity-verified code into a confidential and isolated TEE (Trusted Execution Environment).

Recall only operates on Copilot+ PCs that meet the Secured-core standard and include the following capabilities by default, which are verified by Recall:

- *BitLocker (on Windows 11 Pro) and Device Encryption (on Windows 11 Home) TPM (Trusted Platform Module) 2.0: Tthe TPM provides root of trust for the secure platform, management of keys used by the Secure Enclave TEE and additional platform hardening primitives, such as un-forgeable monotonic counters.*
- *Virtualization-based security and hypervisor enforced code integrity.*
- *Measured Boot and System Guard Secure Launch – If a machine is not booted securely, it cannot attest to the system's security state and release keys, which can unseal content previously protected, thus mitigating early boot attacks.*

- *Kernel DMA Protection against peripheral attacks.*

And finally and significantly, under **"Recall security reviews"**

> *In addition to designing and architecting Recall with security, privacy and responsible AI in mind, we have also conducted a set of thorough security assessments of the feature. This includes the following efforts to ensure a thoughtful and secure approach:*
>
> - *The Microsoft Offensive Research & Security Engineering team (MORSE) has conducted months of design reviews and penetration testing on the Recall.*
> - *A third-party security vendor was engaged to perform an independent security design review and penetration test.*
> - *A Responsible AI Impact Assessment (RAI) was completed, which covered risks, harms and mitigations analysis across our six RAI principles (Fairness, Reliability & Safety, Privacy & Security, Inclusiveness, Transparency, Accountability). A cohesive RAI Learn and Support document was developed for increasing awareness internally, and external facing RAI content was published to drive trust and transparency with our customers.*

This is so much more than that original collection of SQL files stored in a user's private directory that it should be abundantly clear that today's Recall implementation bears no resemblance whatsoever to the disaster waiting to happen that Microsoft originally proposed.

This is why I felt it necessary to give Recall's Re-Rollout an entire podcast topic of its own: It would not be fair to Microsoft for our option of Recall to remain colored by that first impression.

The difference between then and now is so stark that I have no explanation for what **that** was, where it came from, or who did it. It feels like it was nothing more than an innocent first pass proof of concept that some **idiot** in marketing insisted upon shipping immediately because it was so amazing. Yes! ... it's amazing. But it could also **never** be safe unless it was also implemented correctly.

Fortunately, the entire security industry rose up and collectively said "What The Actual F" and got Microsoft's attention. What we have now, what is protecting Recall's aggregated user data, is a seriously well thought out state-of-the-art security architecture. And they're even had it reviewed by outsiders.

But even given all that, the number one lesson we've learned about security, and there are no exceptions to this, is that **only time will tell** whether this will be enough — but at least it looks like it now stands a chance.