Security Now! #924 - 05-23-23

VCaaS – Voice Cloning as a Service

This week on Security Now!

This week, we'll lead off with a tracking device follow-up, then answer some questions including: What happened when I updated my own ASUS router, and what happened when HP attempted to update all of their OfficeJet Pro 9020e-series printers in the field? What did the Supreme Court have to say, if anything, about Section 230? How concerned should KeePass users be about this new master password disclosure vulnerability? What's Apple's position on ChatGPT? What's Google been quietly doing about its "user profiling without tracking" Privacy Sandbox technology? What disappointing news did the Senate Intel Committee just reveal about the FBI, and why did The Python Foundation suddenly close all new registrations of users and packages? Then, after I announce and explain the discovery and fix for a longstanding bug that has always existed in SpinRite 6.0, probably extending as far back as SpinRite 3.1 in the mid 90's, we're going to finish by examining the emergence of new "Voice Cloning as a Service" Dark Web facilities.

Brilliant!

Security News

Tracker Follow-Up

As I mentioned last week, GRC's Security Now newsgroup was holding a great deal of discussion surrounding the AirTag tracking issues. In that vein a person named "Li", posting that they had stumbled across a Youtube video which explained how to, without much trouble, remove the little beeping speaker from an Air Tag. So I asked Li whether he had any impression from the video about WHY the person making the video thought that would be useful? Li replied *"The reason is to prevent alerting a thief to the presence of the AirTag - the Youtuber begs people not to mis-use the hack."* There's been some related discussion in the newsgroup questioning how an AirTag that's beeping can be used for stolen item recovery since the beeping will inherently disclose the hidden presence of the Tag. At which point the attacker might either, as you did last week, Leo, smash the crap out of it with a mallet, or perhaps be more sneaky and tuck it down into the back seat of an Uber driver's car.

The salient point here being that we have two use-cases which are inherently at odds with one another. Someone trying to locate something that is not in the hands of someone nefarious wants help locating the object – so in this context beeping to call attention to the not-stolen tag is desirable. Whereas, someone trying to locate something that has been stolen doesn't want the nefarious thief to know that they have been tagged and are being tracked – so in that context beeping which calls attention to the presence of a tracking tag is undesirable. Since the tag's paired owner might know which of the two contexts is proper – did I leave my car keys somewhere or has my scooter been stolen? – I'm sure that someone at some point must have suggested that it would be possible to give the tag's owner remote control over the beeping since they would know which context applied. But, of course, we cannot do that either, since that control assumes that the tag's paired owner is honest, honorable and ethical... which would not be the case when someone wishes to deploy the tracking tag not to keep track of their own stuff, but to surreptitiously track someone else or their stuff.

So the point I'm hoping to highlight here is that a fundamental tension exists for which there is no single perfect answer. And that this tension has apparently been reflected in the forthcoming IEFT tracking specification. The spec never explains the **why** of any of its requirements; it only lays out the **what**. So we're left to guess. But, for example, the provision that tags must wait a random length of time – from 8 hours to 24 hours – after being physically separated from their owner before they're allowed to start making noise, and then, only when they are jostled, feels like a compromise between the need for stealth in the event of a tagged item's theft and the need for disclosure in the event that someone else's tag is being used for nefarious tracking.

Now we learn that disabling a tag's beeping speaker is easy to do – and using somewhat more surgical precision than Leo did. Clearly, someone who tags their own eBike, for example, would desire to keep their tag quiet since "locating" a tag attached to a large bicycle won't ever require a speaker and preventing a thief from discovering the tag could be extremely useful. In fact, if I were to tag something big like that today, knowing what I know now, I would disable its tiny speaker, too. Unfortunately, the obvious flip side to this is that low budget bad guys, who wish to track others surreptitiously could also disconnect the device's tiny speaker in order to keep the tag from revealing itself. There's no way to win this one. These Bluetooth LE tags are a powerful consumer tracking technology that's inherently prone to abuse.

Next up: Automatic IoT device updating:
Some of it is here already and more is clearly coming. Much of the forthcoming legislation and pending regulation surrounding the behavior of connected IoT devices makes mention of the need for the provision for automatic autonomous firmware updates.

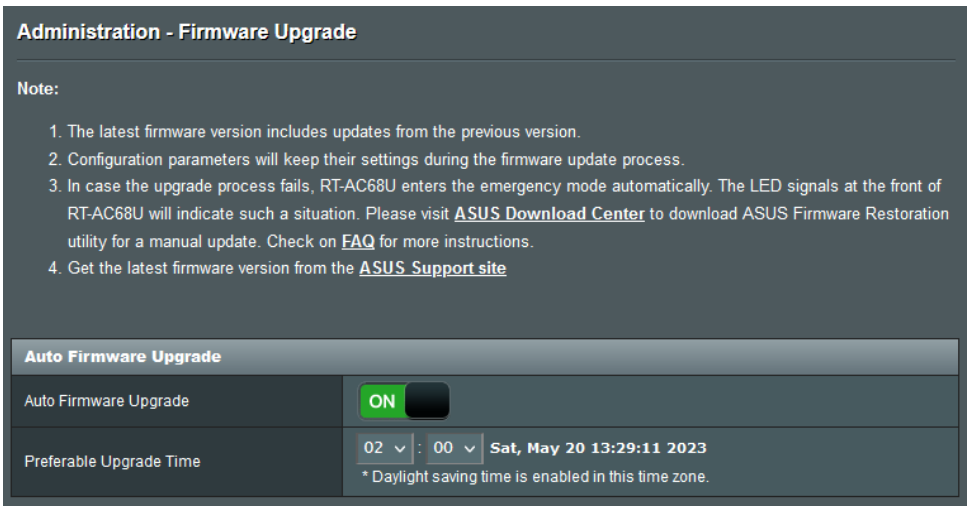
I recently mentioned that when I asked the ASUS RT-AC68U WiFi router at one of my locations to perform a self check for newer firmware, it said that there was an update available. I mentioned this in the context of the problem of a lack of automatic autonomous firmware updating, wondering how many of us were incessantly checking for the availability of updated firmware for our various devices, and presuming that most of us generally had more pressing things to do. And thus the problem that routers may not be getting updated. The next week we learned from a TP-Link using listener that those vulnerable TP-Link routers DID have autonomous update capabilities which was enabled by default. So that was great news. Then I received a Twitter DM from another listener:

Shaun Merrigan / @shaun645D

> *Steve: Regarding Auto-update routers. Asus added this feature into this now "old" router (RT-AC68U) a couple of years ago. And they are still updating the firmware regularly.*

That was interesting to me. I knew from having recently checked, new firmware **was** available for that router, though I hadn't taken any action on that. In defense of my apparent laxity, I'll remind everyone that this consumer router is **not** on the frontline with its WAN port facing a hostile Internet. It sits safely behind the front line router which is running pfSense since I play lots of static port translation gymnastics to avoid COX's annoying consumer bandwidth port filtering.

But Shaun's tweet came at a perfect moment so I logged into the ASUS RT-AC68U and asked it to please update itself. It did that and lo and behold! What missing feature do you think I found had been added to this router since its last update?



Yep... my router now has the ability to update itself autonomously at 2am whenever it detects that ASUS has some new firmware for it. So, that's the good news. But this good news comes nicely paired with the most catastrophic-possible outcome of autonomous auto-updating:

HP 9020e - error code 83C0000B

Post on 05-08-2023 07:30 AM
Product: HP OfficeJet Pro 9022e All-in-One Printer

> *"Hi there, my printer is offline with a blue screen and error code 83C0000B. I have read through other posts and tried the power drain reset but to no avail. Can someone send me instructions to perform a semi-full reset please? Many thanks"*

That particular forum posting generated 132 replies such as:

I have the same problem and have just been going around in circles trying to find a solution.

Ditto - no coincidence that we're all having the same issue on the same printer on the same day (software issue). HP please help!

Same problem here too - did anyone get a solution? My printer is only 3 months old and it is used for business so I need it to work ASAP!

Has anyone had any luck resolving this? I have the same issue.

Hi there, did you get any resolution as the exact same thing has happened to me this morning.

I have the exact same problem, brand new printer. I have tried unplugging the power cable and waiting for a minute. The error is still there in a blue screen 83C0000B and a power button sign. HP please find a solution as our work is affected.

Hello, Same problem on my 9022e (4 months old). I switched it off few hours (remove power cable), but still blocked on 83C0000B error message.

I have an Officet Jet 9022 Pro under warranty and with an HP cartridge subscription. I have same error code 83C0000B since yesterday. 08/05/2023. After three calls to HP after-sales service, more than an hour and 30 minutes => no solution. *"The problem is being studied by HP"* **[Yeah, I'll bet!]** I am awaiting instruction! If no return info, I return it and it could end with a refund via distributor (Amazon FR Thank you)

This is insane, and not a word from HP?! I came here to make a post about this, and it seems it's basically a global issue! Can someone from HP provide an update? This is terrible, we cannot print anything, stuck with this error!

So now let's turn to our friends at BleepingComputer for a succinct summary of what's going on: BleepingComputer's headline reads *"HP rushes to fix bricked printers after faulty firmware update"* And note that this was posted just this past Saturday, 12 days after all of their affected printers were turned into expensive bricks, BleepingComputer writes:

> *HP is working to address a bad firmware update that has been bricking HP Office Jet printers*

worldwide since it was released earlier this month.
While HP has yet to issue a public statement regarding these ongoing problems affecting a subset of its customer base, the company told BleepingComputer that it's addressing the blue screen errors seen by a "limited number" of users.

HP told BleepingComputer: "Our teams are working diligently to address the blue screen error affecting a limited number of HP OfficeJet Pro 9020e printers. We are recommending customers experiencing the error to contact our customer support team for assistance: https://support.hp.com."

Impacted printers include HP OfficeJet 902x models, including HP OfficeJet Pro 9022e, 9025e, and the 9020e and 9025e All-in-One Printers.

Affected customers report that their devices display blue screens with "83C0000B" errors on the built-in touchscreen.

Since the issues surfaced, multiple threads have been started by people from the U.S., the U.K., Germany, the Netherlands, Australia, Poland, New Zealand, and France who had their printers bricked, some with more than a dozen pages of reports.

One customer said: "HP has no solution at this time. Hidden service menu is not showing, and the printer is not booting anymore. Only a blue screen."

Another added: "I talked to HP Customer Service and they told me they don't have a solution to fix this firmware issue, at the moment."

Others have said that the only way to address the issue is to send the printer for servicing to HP and that "The firmware doesn't even load partially, it instantly fails... HP remotely bricked our devices! Some users said that HP would be sending out a replacement."

Since the buggy update seems to install automatically onto Internet-connected printers, HP customers are advised to disable their devices' Internet connection and wait for a firmware update to fix the bricking issue.

So, as it happens, it's not that often that we actually get a horrific real life answer to our often posed question: *"What, could possibly go wrong?"* — but, oh boy! Wow. Apparently, the only reason some of these susceptible model printers were **not** blasted into oblivion was that they had not phoned home to check-in for a firmware update. In this instance they had a blessed **lack** of connectivity.

We do need to expand upon our normally rhetorical *"What, could possibly go wrong?"* question to also ask *"How, could this possibly happen?"* Right?? I mean, this would lead anyone to ask *"Did HP not ever actually test the firmware that they sent out to every available printer which then proceeded to kill all of them?"* It must be that HP had working firmware in the lab. So we'd have to conclude that somewhere between the Lab and all those end users, that firmware image file became corrupted. Perhaps it was a corrupted file transfer to the update server, or somehow the file on the update server itself became corrupted. But without information that only HP has, and which we will probably never have, it's impossible to know exactly how and why this blew up. All we can do is examine the debris field.

But regardless of what happened upstream of these printers, which will almost certainly forever remain unknown, what we **do know today** is that HP's autonomous updating system is not well designed, because this should have been not just unlikely to happen, but impossible to have happen. We're witnessing the obvious downside of poorly designed autonomous updating, which is that if you're not careful, you can potentially hose your entire installed base of devices, whatever and wherever they may be. And I suspect that since HP has not yet been forthcoming with an explanation – though I'm sure by now they know just how deep in the doo-doo they are – we're never going to know exactly what happened. Today's HP is a massive enterprise, which means they have squadrons of well-paid attorneys who immediately clamped down on all communication in an attempt to minimize perceived culpability even though it's going to be difficult to explain this one away.

Today's HP is not the HP I grew up admiring. HP was founded by a pair of electrical engineers working in a small garage in Palo Alto, California. Back in 1939, Bill Hewlett and David Packard correctly divined that what the world really needed was a high quality analog signal generator. So they built and sold some and then they built and sold many more.

The reason for my annoyance is not just HP's sloppy software, although their printer software, at least for Windows, is a true atrocity. But what appears to be the case, is that HP has just bricked all of these printers – rendering them unusable at least for the time being – in order to update the printer's detection of non-HP printer ink cartridges in a move to further thwart the use of significantly less expensive and just as effective 3rd-party ink. ArsTechnica also covered this month's HP printer bricking debacle, and at the end of their coverage they added some additional interesting insight. Ars wrote:

> *It's unclear what exactly prompted the firmware update that broke HP's printers. HP's support pages for the 9020e and 9025e series only emphasize the printers' use of HP's much maligned "Dynamic Security." HP uses Dynamic Security to stop printers from working with non-HP brand ink cartridges. HP's notoriously sudden issuance of Dynamic Security to printers has abruptly rendered piles of ink useless.*
>
> *Officially, HP says it uses "Dynamic Security measures to protect the quality of our customer experience, maintain the integrity of our printing systems, and protect our intellectual property," but since debuting in 2016, it has resulted in class-action lawsuits and irate customers stuck with otherwise functioning ink that HP decided shouldn't work.*
>
> *HP's Dynamic Security page notes that Dynamic Security printers* ***require periodic firmware updates to "maintain Dynamic Security effectiveness"*** [I'll just bet they do] *and that "updates can improve, enhance, or extend the printer's functionality and features, protect against security threats, and serve other purposes."*
>
> *HP's poor handling of printer firmware updates can deter people from future updates that may be important. Meanwhile, HP has traumatized countless customers, and many will now think twice before depending on an HP printer again.*

If we wonder why HP's printers which employ "Dynamic Security" might require periodic security updates, it may be because HP feels that their control over their printers allows them to avoid playing on a level playing field where they need to compete with other suppliers of ink

cartridges. So they're engaged in a game of cat and mouse with the 3rd-party ink vendors who are continually working around HP's "Dynamic Security" measures in order to get their ink to flow. That then necessitates HP again updating their non-HP ink defenses with new firmware. It's a sad game really, and this time it appears that the joke's on HP.

Regardless of why this happened, it's going to be quite interesting to learn whether this can be fixed in the field. The printers do all have a USB port. So if the printer is able to boot from a properly formatted USB thumb drive even with its firmware in this apparently badly-blasted state, HP might be able to avoid replacing all of the destroyed printers in the field by giving their users a recovery image to download onto their own thumb drive or expreSS mailing a ready to boot thumb drive to anyone who requests one. But we don't yet know whether that could be done. Since customers would probably not have shipping boxes, the shortest path might be to send a new printer in a box and ask the user to box up their dead printer and return it in the same box.

It's a mess. The ASUS router upgrade page says: "In case the upgrade process fails, RT-AC68U enters emergency mode automatically. The LED signals at the font of the RT-AC68U will indicate such a situation. Please visit "ASUS Download Center" to download ASUS Firmware Restoration utility for a manual update. Check FAQ for more instructions."

It doesn't appear as though HP has provided for such a contingency. But the observation I'd like to make is that today's flash memory is very inexpensive, so there is **no** excuse for not incorporating a factory backup image of printer firmware that the system can boot to in an emergency simply by pulling the most significant address line of the flash memory high, thus switching to a secondary memory bank containing the backup image – then trigger a reboot and you're back in business. Processors used in embedded environments typically incorporate what's known as a "watchdog timer." It's a sort of a deadman switch. The idea is that running software will periodically reset the watchdog timer, always making sure to do so before the timer expires. This is typically easy to do, since the watchdog generally doesn't need to respond instantly to any problem. It's the response of absolute last resort. So, whereas the code in the device's main execution loop might reset the watchdog timer every second, the timer might wait for 20 seconds without a reset before it expires and declares an emergency.

The first time an emergency occurs, the watchdog might just perform a reset to recover the device from a random glitch – a power-line hiccup or a cosmic ray. But it would remember that that's what it had done. And if it got called back to action soon, or several more times, it would pull that high address line high, switch to the backup firmware, and bring the printer back online.

If HP had done this, a couple of 3rd-party ink cartridges might have slipped past due to not running the latest anti-3rd-party ink firmware, but a major economic and reputational disaster would have been averted.

So the point I want to leave everyone with is that, yes of course, autonomous updating can be done wrong – and we are apparently seeing a glaring example here. But it can also be done in a way that will always fail safe rather than fail dead.

Section 230 Stands

As we know, the bit of law that's become famous for its section number, 230, is what essentially allows Facebook, Twitter, YouTube and anyone else who hosts content provided by others to avoid liability for any consequences which might arise from their hosting of 3rd-party material. So, perhaps the biggest news of last week was that the Supreme Court, which first surprised many Court watchers when they agreed to weigh-in on the appeals made to the 9th circuit court's earlier decisions, produced their conclusion in May rather than waiting until the end of the current term, in June.

And, to the great relief of practically everyone, Section 230 held up under the Supreme Court's review and remains, at least for the time being, in full force and effect.

Both the tech and non-tech press covered the news. I felt that the EFF was a bit snarky in their coverage which they headlined "The Internet Dodges Censorship by the Supreme Court" – yeah, dodged it by 9 to 0. TechDirt's Mike Masnick dug deeply into the details. It's way more than we need here, but I linked to his coverage in the show notes for anyone who wants a deep dive:

https://www.techdirt.com/2023/05/18/supreme-court-leaves-230-alone-for-now-but-justice-thomas-gives-a-pretty-good-explanation-for-why-it-exists-in-the-first-place/

Reuters provided a very good overview of the issues, and since this is of crucial importance to the way the Internet evolves, and since this may not be settled law forever, I think it's worth looking at the way Reuters summarized the Court's involvement. They wrote:

> *The U.S. Supreme Court handed internet and social media companies a pair of victories on Thursday, leaving legal protections for them unscathed and refusing to clear a path for victims of attacks by militant groups to sue these businesses under an anti-terrorism law.*
>
> *The justices in a case involving Google's video-sharing platform YouTube, part of Alphabet Inc., sidestepped making a ruling on a bid to weaken a federal law called Section 230 of the Communications Decency Act that safeguards internet companies from lawsuits for content posted by users.*
>
> *They also shielded Twitter Inc in a separate case from litigation seeking to apply a federal law called the Anti-Terrorism Act that enables Americans to recover damages related to "an act of international terrorism."*
>
> *In both cases, families of people killed by Islamist gunmen overseas had sued to try to hold internet companies liable because of the presence of militant groups on their platforms or for recommending their content.*
>
> *The justices in a 9-0 decision reversed a lower court's ruling that had revived a lawsuit against Twitter by the American relatives of Nawras Alassaf, a Jordanian man killed in a 2017 attack during New Year's celebration in a Istanbul nightclub claimed by the Islamic State militant group.*
>
> *In the case involving YouTube, the justices returned to a lower court a lawsuit by the family of a Nohemi Gonzalez, a college student from California who was fatally shot in an Islamic State attack in Paris in 2015. The justices declined to address the scope of Section 230, concluding*

they did not need to take that step because the family's claims appeared likely to fail given the Twitter case decision.

Section 230 provides safeguards for "interactive computer services" by ensuring they cannot be treated for legal purposes as the "publisher or speaker" of information provided by users.

Calls have come from across the ideological and political spectrum - including Democratic President Joe Biden and his Republican predecessor Donald Trump - for a rethink of Section 230 to ensure that companies can be held accountable for content on their platforms. This case marked the first time the Supreme Court had examined Section 230's reach.

Google's General Counsel said: "Countless companies, scholars, content creators and civil society organizations who joined with us in this case will be reassured by this result. We'll continue our work to safeguard free expression online, combat harmful content and support businesses and creators who benefit from the internet."

Critics have said Section 230 too often prevents platforms from being held accountable for real-world harms. Many liberals have condemned misinformation and hate speech on social media. Many conservatives have said voices on the right are censored by social media companies under the guise of content moderation.

The massacre at Istanbul's Reina nightclub killed Alassaf and 38 others. His relatives accused Twitter of aiding and abetting the Islamic State by failing to police the platform for the group's accounts or posts in violation of the Anti-Terrorism Act.

Gonzalez's family argued that YouTube provided unlawful assistance to the Islamic State by recommending the group's content to users. In their brief ruling, the justices wrote that they "decline to address the application of (Section 230) to a complaint that appears to state little, if any, plausible claim for relief."

Chris Marchese, an attorney with NetChoice, a technology industry group that includes Twitter, Meta and Google as members, said: "Even with the best moderation systems available, a service like Twitter alone cannot screen every single piece of user-generated content with 100% accuracy. Imposing liability on such services for harmful content that unintentionally falls through the crack would have disincentivized them from hosting any user-generated content."

The Twitter case hinged on whether the family's claims sufficiently alleged that the company knowingly provided "substantial assistance" to an "act of international terrorism" that would allow the relatives to maintain their suit and seek damages under the anti-terrorism law.

After a judge dismissed the lawsuit, the San Francisco-based 9th U.S. Circuit Court of Appeals allowed it to proceed, concluding that Twitter had refused to take "meaningful steps" to prevent Islamic State's use of the platform.

Conservative Justice Clarence Thomas, who authored the Supreme Court's ruling, said the allegations made by the plaintiffs were insufficient because they "point to no act of encouraging, soliciting or advising the commission" of the attack.

"These allegations are thus a far cry from the type of pervasive, systemic and culpable assistance to a series of terrorist activities that could be described as aiding and abetting each terrorist act," Thomas added.

In the Twitter case, the 9th Circuit did not consider whether Section 230 barred the family's lawsuit. Google and Facebook, also defendants, did not formally join Twitter's appeal.

So, as with many of the decisions reached by the Supreme Court, the result is not terribly satisfying. The Court presumably obtained the outcome they wanted, which was to leave Section 230 in place, as is, in its entirety. But they did so not by directly addressing the issue, but by focusing on procedural distractions, like complaining that someone had forgotten to tie their shoes.

At the end of his long and interesting analysis, TechDirt's Mike Masnick concluded:

Finally, speaking about money, time, and resources, a shit ton [Mike's words] *of all three were spent on briefs from amici for the Gonzalez case, in which dozens were filed (including one from us). And… the end result was a three page per curiam basically saying "we're not going to deal with this one." The end result is good, and maybe it wouldn't have been without all those briefs. However, that was an incredible amount of effort that had to be spent for the Supreme Court to basically say "eh, we'll deal with this some other time."*

The Supreme Court might not care about all that effort expended for effectively nothing, but it does seem like a wasteful experience for nearly everyone involved.

So, 230 stands unweakened but still likely assailable in the future if the right challenge is brought.

The KeePass Vulnerability.
Last week the entire tech press community lost its collective mind over the news generated by a hacker that KeePass was insecure and that it was possible to obtain the KeePass master password and thus everything else. Infosecurity Magazine's headline was *"KeePass Flaw Exposes Master Passwords"* (that sounds bad). The Hacker News reported under the headline *"KeePass Exploit Allows Attackers to Recover Master Passwords from Memory"* (That can't be good) SC Magazine said *"KeePass bug lets attackers extract the master password from memory"* — and actually, they had by far the coolest graphic which I put into the show notes:

TechTarget wrote *"KeePass vulnerability enables master password theft"*, Anonymania: *"Passwords at Risk – A Major KeePass Flaw Unearthed"*, TechRadar: *"This top password manager apparently has a major security flaw that could spill all your logins"* and HelpNetSecurity: *"KeePass flaw allows retrieval of master password, Proof of Concept is public"* Not good.

So here's the **one and only thing** that this podcast's well-informed listeners need to know: **It's an entirely local attack.**

In other words – ***NEWS FLASH!*** – if your machine has malware running amok, it's not safe to use a password manager – any password manager, not just KeePass. By the very nature of what any password manager is and needs to do, **no password manager is safe** from local attack... and they don't try to be, because it's a fool's errand.

Recall how almost all commercial video DVD's were encrypted to prevent the theft of their contents. But DVD players were fundamentally unable to keep their keys secret because they needed to use them in order to decrypt the DVDs. It was so dumb. In an exactly analogous way, a password manager that is going to fill-in blank username and password fields must have access to that manager's decrypted password database. Even if you set it up in an annoying way to require you to provide the master password each and every time you use it, it would still, briefly, need to decrypt and read the master database to obtain the required password.

The reason we were so upset with LastPass was not over something like this, which, as I said, is an inherent design point for any client-side password manager. We got upset with LastPass because they lost everyone's encrypted data all at once, and then we learned that a lot of it had never been encrypted and that what was encrypted might not have been very well encrypted.

KeePass is a popular, free and open source password manager. I don't use it, but I know from Twitter that some of our listeners do. You have to imagine that KeePass's lead developer, Dominik Reichl (rye-kle), who has recently been attacked from every angle, has got to be asking himself about now, why he's been working so long and hard on this currently thankless project.

Here are some details to give everyone some sense for what's going on in this particular case: The issue affects the software's custom text box control which is used for entering the master password and other passwords during editing.

For the "KeePass 2.X Master Password Dumper" proof of concept tool to work, you need some access to the system RAM through a system RAM snapshot, so on a Windows machine that would be the process dump, the pagefile.sys swap file, the hiberfil.sys hibernation file, or a RAM dump of the entire system. The researcher whose work has generated so much click-bait said: *"The flaw exploited here is that for every character typed, a leftover string is created in memory. Because of how .NET works, it is nearly impossible to get rid of it once it gets created."* So this sounds like a case where a developer, using a very high level language framework (.NET), is being betrayed by the lower-level way some of the language environment functions. In this case the environment's automatic string management. It wasn't really meant for implementing super secure systems – at least not without giving it extra explicit thought.
There's just too much automatic stuff happening in the background.

You know how when entering a password in iOS the most recent character typed can be seen for confirmation while all previous characters are blanked with a round bullet character? It appears that this is what KeePass does, but that the custom control that its author created is discarding each of the intermediate strings which leaves them floating around in RAM.

The exploit's developer said: *"For example, when 'Password' is typed, it will result in these leftover strings: •a, ••s, •••s, ••••w, •••••o, ••••••r, •••••••d. The proof of concept application searches the RAM memory dump for these patterns and offers a likely password character for each position in the password."* – except for the first character, which is not available. The vulnerability affects the KeePass 2.X branch for Windows, and possibly for Linux and macOS. It has been fixed in the test versions of KeePass v2.54 whose official release is expected by July.

Presumably, in this updated release, Dominik is no longer using intermediate strings, or he's blanking the intermediate string before releasing its storage back to the environment so that released strings will not contain any sensitive information.

But steeping back from the details, the broader lesson here is to always keep a secure system's security model in mind. There are things that the model provides and things that it doesn't. The security model for a password manager is security and protection across the network. No one acting remotely should be able to obtain any secret information. In return for using a password manager, impossible-to-remember (or enter!) passwords can be used to provide today's highest generally available level of security for remote network logon. But only from across the network.

Password managers do not protect themselves against local attackers. And they really cannot, because users demand operating systems that are fundamentally insecure. A truly secure operating system is no fun to use because you really can't get anything done. You spend all of your time manually authorizing the system to do anything you want. For a long time, users were running with full root privileges because they had grown tired of always needing to log off and back on as a root administrator to install a program, then switch back again. Microsoft finally developed Windows' split-token UAC (user account control) system which minimizes the grief of normally running as a user who only has safe permissions. It's a terrific compromise.

Apple joins Samsung, Amazon and Verizon in banning ChatGPT

AppleInsider reported that Apple has joined the growing ranks of companies that are banning the use of AI conversational large language model AI inside the company. And we know that Apple runs a tight ship.

Internal documents and anonymous sources have leaked details of Apple's internal ban on ChatGPT and similar technology, as well as some tasty bits regarding its inevitable plans for its own Large Language Model.

As we know, despite their popularity, chatting with Large Language Models can be unpredictable and tend to leak the data that's being fed to them. Even if dialog logging is disabled, this information is still leaving the local facility to travel out over the Internet. It's not a friend you're sitting in front of, or in the box you're holding in your hand. It's a massive remote cloud compute farm. A report from The Wall Street Journal details an internal Apple document which restricts

Apple employees from using ChatGPT, Bard, or any similar Large Language Models (LLMs). Anonymous sources also shared that Apple is working on its own version of the technology, though no other details were provided.

We previously noted that Samsung had publicly banned their employee's use of ChatGPT after three incidents where employees uploaded proprietary corporate information in order to get ChatGPT's opinion. So Apple has also joined Amazon, Verizon and likely many other lesser-known organizations in telling their employees that there will be no “Chatting with the Bots” during work.

As for Apple's own internal Large Language Model work... I haven't heard of anyone assembling a fully interactive speech interface for one of these textual chatterboxes, but it sure does seem like the next obvious thing to do. Being able to have a verbal conversation with this sort of simulated intelligence would take a voice assistant to an entirely new level.

Google's Privacy Sandbox moves forward

Although we covered Google's next-generation user interest profiling technology in full detail when its detailed mechanism of operation was first announced, we haven't focused upon it for some time. Yet Google has been diligently and quietly at work getting the technology ready to rollout.

Google– will begin rolling out this new Privacy Sandbox technology in the third quarter of this year, with the release of Chrome 115. This rollout will take the form of a set of new built-in browser APIs that JavaScript code running in ads hosted on websites will be able to use to obtain categories of interest for the website's visitors. The idea is that online advertisers will be able to deliver ads without collecting any personal information about users and without tracking their movements across the web with third-party cookies. Google has said that as the rollout proceeds, it will also be removing support for third-party cookies from Chrome next year. This will be done incrementally and through careful testing. Initially, in the first quarter of 2024, support for 3rd-party cookies will be removed from 1% of the Chrome user base. And by the 3rd quarter, Chrome and its users will be 100% 3rd-party cookie free.

The FBI heavily misused FISA powers.

I really want to believe that the agents of the FBI are good, law abiding people. I really do. And for the most part, I’m sure they are. (I wonder how many of our listeners know the name “Efrem Zimbalist Jr.” He was the lead in “The F.B.I.", which was a television series I grew up watching every Sunday night from 1965 through 1974.) In any event, recently declassified documents have revealed that the **real** U.S. Federal Bureau of Investigation improperly searched the personal communications of Americans more than -- is everyone sitting down -- more than 300,000 times between 2020 and early 2021, according to newly declassified documents which were released by the Senate Intel Committee. I don't know any details, but that seems disappointing... and I’m quite sure that Efrem Zimbalist Jr. would have had no part in that.

The inappropriate searches occurred under FISA, our Foreign Intelligence Surveillance Act, which may have been what made them inappropriate for US Citizens – you know, the "foreign" part. FISA is currently up for renewal by Congress. The documents reveal that the FBI used the massive trove of communications data gathered using FISA powers to search for information on participants who attended the George Floyd protests (ouch) and the January 6 storming of the US Capitol. The FBI also searched the names of more than 19,000 donors to political campaigns for plausible connections to foreign governments.

Hearing this causes me to think that even though my own communications are quite boring, it's my right as a U.S. citizen for them to remain truly private unless and until a court orders otherwise. So... utterly unbreakable end-to-end encryption? Yes, please.

Supply Chain Nightmare

Over in the increasingly fraught open-source registry "supply chain" world, things have grown so bad that the Python Foundation was finally forced to suspend the creation of new accounts and new packages. The Foundation has suspended all new user registrations as well as new package uploads to PyPI, the official Python Package Index. The organization's security team says PyPI saw another wave of malicious packages being uploaded on the platform over the past week. PyPI staff says the incident occurred while some of its members were on leave, and that the smaller group of personnel on duty were quickly overwhelmed by the large number of reports and malicious packages they had to remove. So they threw the switch and said "No more or anything" for the time being. Existing users can still access and update their projects. It's just new accounts and projects that are barred.

What was that about why we can't have nice things? I'll have something to say about these disturbing trends in this week's closing comment.

But first, I have some interesting news about
a longstanding SpinRite 6.0 bug to share...

SpinRite

It turns out that my claim that SpinRite 6.0 has had no known bugs for the past 19 years has not been correct. Though I suppose it would depend a bit upon how you define "known". Certainly it hasn't had any "appreciated" bugs, but it definitely has had a bug. And thanks to the work of an independent coder named Paul Farrer, GRC is now offering a pair of patch utilities which fix this bug that SpinRite has had, perhaps since the mid 90's with SpinRite 3.1.

Since it only occurs on drives larger than 549 gigabytes, when SpinRite's DynaStat system kicks in to perform data recovery and repair, and since this behavior has been present since at least SpinRite 3.1, what is now seen as an overflow was very likely my deliberate design decision at the time, since drives of that size were not even a dream back when 50 megabytes was a large drive. So what likely happened was that as I evolved SpinRite through the decades, I never revisited the parameters surrounding this one division operation to notice that modern drives might cause it to overflow.

Through the years, we've had reports of SpinRite halting with a division overflow error. Somehow, I got it into my head that the location reported by SpinRite was the segment where the error was occurring, not the offset. So "B04E" would not be in program space. It's in the region of memory that was once set aside for the monochrome display adapter. So I assumed that this error was occurring in a chunk of code that the system's BIOS had mapped into that unused region. And this belief was supported by the fact that GRC's tech support guy, Greg, has developed a collection of workarounds for SpinRite's users who encounter this error, things like "Try running SpinRite with that drive on another machine" – which he says often works. In fact, over the weekend, when I wrote to him to tell him that we had a patch for this long standing problem he replied: *"Since we are getting closer to 6.1, I'll probably use this as the last thing to try as all the other "fixes" we have in place are much less technical."* So my point is, this hasn't been a big deal or issue for us. But I know for certain that it has been so for some users – and that's not okay. I also now understand why moving to a different machine may have helped, since part of the issue surrounded the BIOS's mapping of cylinders, heads and sectors to a drive's linear sector number, and that mapping is one of the many things that different BIOSes might do differently.

In any event, Paul had just finished developing a different patch for those buggy AMI BIOSes which we discovered were blasting main memory when anything attempted to access sectors past 137 gigabytes on USB-connected drives. Out of an abundance of caution, which I feel is warranted, SpinRite v6.1 will refuse to go any further than the first 137 gigabytes on any USB drive. But Paul had access to our newsgroup which was full of people who had machines with these buggy AMI BIOSes. So working with them, he's created a tiny utility that can be run before SpinRite. If it finds an AMI BIOS that it recognizes, he'll patch it. And then, with my blessing – we've discussed how this should be done to be stable – his utility will remain in RAM and cause SpinRite 6.1 to believe that USB devices are SCSI devices, thus lifting 6.1's cautionary clamp on USB drive size. So that little utility will be made available for users to use at their own risk if they choose. SpinRite 7 will not use any BIOS, so all of this will be going away as soon as we move there.

After doing that work, Paul became curious about that B04E error. Without my bias of assuming that this was the segment of the problem, and therefore in the BIOS and not in SpinRite, he

assumed that I had been reporting the offset – as I was. So he looked into SpinRite's running, in-memory code and sure enough, he found a division instruction at that offset in SpinRite. He then proceeded to reverse engineer that region of SpinRite's code to figure out what was going on and why. At one point I provided him with the relevant chunk of SpinRite's source code so that he could be sufficiently confident that he knew what was going on. He had it exactly right. Drives had become larger, and the math that I had not revisited for decades, which decomposed a linear sector number into cylinders, heads and sectors for the BIOS was no longer able to handle today's larger drives.

So Paul has produced another patch utility which fixes this problem for SpinRite 6.0. He created both a DOS driver that can be loaded through CONFIG.SYS and a DOS TSR that can be run before running the current SpinRite 6. After testing it thoroughly he provided me with his source code to review, and it was immaculate. So, thanks to his efforts, we have a patch for this bug that's always been in SpinRite since its very early days. Out of curiosity, I checked SpinRite's source code for 5.0 dated February 11th, 1996 and it's the same code that SpinRite 6.0 is still using. So I never changed it for SpinRite 6.0 since, at the time, it was not a bug. But it is today. Hitting this error, which can only occur on drives over 549 gigabytes – and only when DynaStat engages – does not endanger or damage any of a user's data, but it does mean that SpinRite cannot proceed with its recovery and repair. So my advice to all SpinRite users listening would be to grab Paul's patch and to add it to SpinRite's boot media. The file is called MDFYSR60.ZIP and it's in GRC's freeware collection. It's also now referred to near the top of the SpinRite FAQ and it has a menu entry under GRC's main menu under SpinRite and "Knowledgebase: B04E".

As I was writing this up for today's podcast I suddenly became curious about what the code for this looks like now, since we already know that SpinRite 6.1 doesn't contain this bug. We've all been running lots of DynaStat recoveries on lots of multi-terabyte drives without any trouble. So I was pleased to see that I had completely replaced that old code with a new routine which is capable of handling a full 64-bits worth of sectors. That's 18,446,744 terabytes... so, even though releases of SpinRite tend to live for a long time, we should be good for a while.

Since this bug is already long gone from SpinRite 6.1, since 6.1 is so vastly superior to 6.0, is nearly finished, will be free to all 6.0 owners, and since this is never destructive to any SpinRite user's data, I plan to get 6.1 finished, rather than delay it in an attempt to announce this patch to v6.0's current owners. After I put all this online Sunday night, I updated Greg so it's what he'll be pointing anyone to who encounters this problem to the patch. And when I do announce 6.1, I'll also inform all 6.0 users of this patch so that they'll have it for 6.0, even though it will largely become obsolete, as will 6.0. So, a big public thanks to Paul Farrer for his terrific work on this.

One final point: Something else I had forgotten from 19 years ago which recently came to light was that for some weeks after SpinRite's initial public availability I was still finding and fixing some final bits of debris. And I was updating SpinRite's downloadable code on the fly. I didn't have the mature version-stamping system that all of my recent work carries, and which SpiNRite 6.1 will, so all of those early editions just say 6.0 without any indication of any sub version or build number. Nothing has changed in 19 years. But if you believe that you may have downloaded SpinRite 6 within week's of its first release in 2004, and never again since then, you might want to update your copy until 6.1 is ready.

VCaaS – Voice Cloning as a Service

I suppose it was inevitable if we follow the "anything that can be done will be done" rule. My original title for today's podcast was "News, Views and no Snooze." Which I felt was not particularly inspired, but it had the attribute of accuracy. That was, right up until I hit upon the news that the barrier to entry for malicious Voice Cloning had just dropped with the, inevitable in retrospect, appearance of Voice Cloning as a Service. So although today's coverage of this will not be a deep dive, I thought that merited some special attention.

Last Thursday, the security firm Recorded Future published a 16-page Cyber Threat Analysis report titled "I have no mouth, and I must do crime." A link to this full PDF is in the show notes for anyone who wants more than the summary that I'm going to share.

https://go.recordedfuture.com/hubfs/reports/cta-2023-0518.pdf

The full report leads with an Executive Summary which I will share. It says:

Deepfake voice cloning technology is an emerging risk to organizations, which represents an evolution in the convergence of artificial intelligence (AI) threats. When leveraged in conjunction with other AI technologies — such as deepfake video technology, text-based large language models (such as GPT), generative art, and others — the potential for impact increases. Voice cloning technology ***is currently*** *being abused by threat actors in the wild. It has been shown to be capable of defeating voice-based multi-factor authentication (MFA), enabling the spread of misinformation and disinformation, and increasing the effectiveness of social engineering. We are continuously monitoring the emergence of deepfake technologies and their use in cybercrime, as detailed in our April 29, 2021 report "The Business of Fraud: Deepfakes – Fraud's Next Frontier".*

As outlined in our January 26, 2023, report "I, Chatbot", open-source or "freemium" AI platforms lower the barrier to entry for low-skilled and inexperienced threat actors seeking to break into cybercrime. These platforms' ease-of-use and "out-of-the-box" functionality enable threat actors to streamline and automate cybercriminal tasks that they may not be equipped to act upon otherwise. Many of the voice cloning platforms referenced in this report are free-to-use with a registered account, thus lowering any financial barrier to entry for threat actors. For those that are not free-to-use, premium prices are negligible — rarely more expensive than $5 per month.

Voice cloning samples that surface on social media, messaging platforms, and dark web sources often leverage the voices of public figures — such as celebrities, politicians, and internet personalities ("influencers") — and are intended to create either comedic or malicious content. This content, which is often racist, discriminatory, or violent in nature, enables the spread of disinformation, as users on social media are sometimes deceived by the high quality of the voice cloning sample. This "proof-of-concept" (POC) work shared by threat actors has inspired a trend on dark web and special-access sources, with threat actors speculating about the emergence of voice cloning as an attack vector. Conversations among threat actors often reference executive impersonation, callback scams, voice phishing ("vishing"), and other attacks that rely on the human voice.

One of the most popular voice cloning platforms on the market is ElevenLabs's Prime Voice AI

(https://beta.elevenlabs.io/), a browser-based text-to-speech (T2S; TTS) software that allows users to upload "custom" voice samples for a premium fee. While there are a number of voice cloning platforms referenced in this report (such as MetaVoice, Speechify, and so on), ElevenLabs is one of the most accessible, popular, and well-documented, and thus serves as the case study for this research.

Then they wrap up their Summary with five bulleted Key Findings:

1. *Voice cloning technologies, such as ElevenLabs, lower the barrier to entry for inexperienced English-speaking cybercriminals seeking to engage in low-risk impersonation schemes and provide opportunities for more sophisticated actors to undertake high-impact fraudulent schemes.*

2. *Currently, the most effective use of voice cloning technologies is in generating one-time samples that can be used in extortion scams, disinformation, or executive impersonation. Limitations to the use of voice cloning technologies, especially for enabling real-time, extended conversations and generating prompts in languages other than English, mean that extensive planning is required for fraudulent operations with a higher impact.*

3. *Threat actors have begun to monetize voice cloning services, including developing their own cloning tools that are available for purchase on Telegram, and the emergence of* ***voice-cloning-as-a-service (VCaaS)****.*

4. *Public interest in AI, including voice cloning technology, has prompted an interest on dark web and special-access sources in AI platforms' potential for abuse. Threat actors are also interested in leveraging multiple AI platforms in concert, thus enabling the convergence of AI threats. However, not all threat actors are confident in their ability to leverage such platforms in their current state — threat actors that do not have an expert grasp of the English language may have the greatest hesitation about using these new technologies.*

5. *In order to mitigate current and future threats, organizations* ***must*** *address the risks associated with voice cloning while such technologies are in their infancy. As these technologies will only get better over time, an industry-wide approach is required immediately in order to preempt further threats from future advances in voice cloning technology.*

So I went over and took a look at ElevenLabs.io: https://beta.elevenlabs.io/ They call their web-based service *"Prime Voice AI" – "The most realistic and versatile AI speech software, ever. Eleven brings the most compelling, rich and lifelike voices to creators and publishers seeking the ultimate tools for storytelling."* So we need to be clear that there are, and will be, all manner of public-benefiting wonderful applications for this emerging technology but that, once again and as always, along with the good comes the bad.

They mention applications for storytelling, audiobooks and news articles. They boast having indistinguishable speech quality. They claim: *"Our AI model is built to grasp the logic and emotions behind words. And rather than generate sentences one-by-one, it's always mindful of how each utterance ties to preceding and succeeding text. This zoomed-out perspective allows it to intonate longer fragments convincingly and with purpose. And finally you can do this with any voice you want."*

I played a bit with their online tool and I have to agree that it was producing far better, more believably human and less robotic speech than we normally hear from text to speech. But I wouldn't say that it's at the point where it's going to fool anyone who might be at all suspicious.

As for cloning well known voices, under their "Voice Lab" they say: *"Voice Lab is your creative AI toolkit. Clone voices from samples or clone your own voice. Or design entirely new synthetic voices from scratch. Our cloning model learns any speech profile based on just a minute of audio, without [additional] training. And our generative model lets you create completely new voices that never spoke before."*

At the bottom of this page they do have a statement under the topic *"Ethical AI"* they write: *"At Eleven, we believe that we should strive to make the most of new technologies, but not at all cost. As we develop them, we make every effort to implement appropriate safeguards which minimize the risk of harmful abuse. With this in mind, we're fully committed both to respecting intellectual property rights and to actioning misuse."*

I want to end with this observation:

A macro trend that we appear to be seeing with increasing frequency, is that bad guys are now jumping onto and exploiting new technologies every bit as quickly as good guys. Over the lifetime of this podcast we have watched *"Doing bad on the Internet"* evolve from being a curiosity, perhaps a basement hobby, into truly full-blown industries. Under our watch, The Dark Web went from being a fanciful theory to a widely inhabited, active and quite busy underground.

It's going to be interesting to see what comes next.