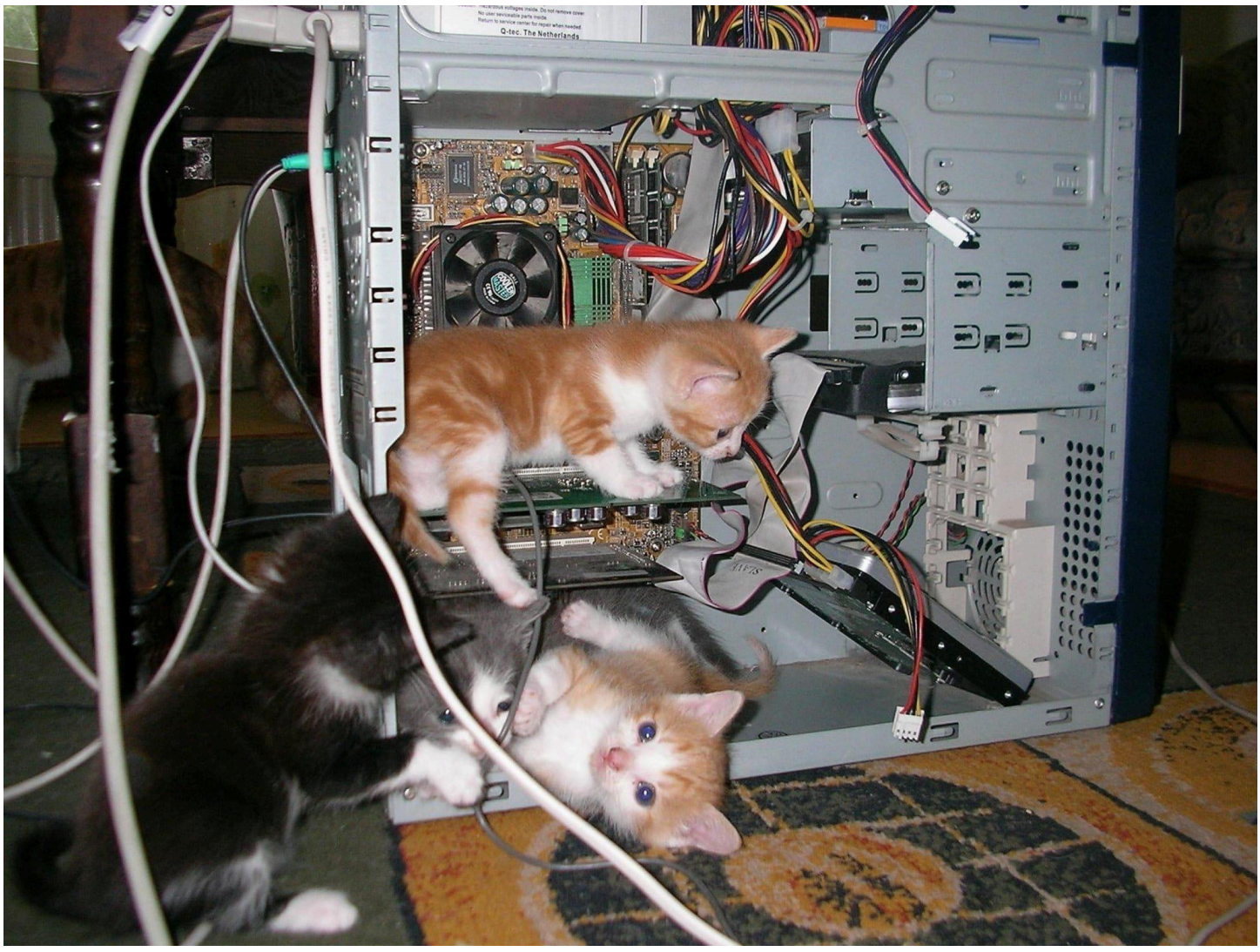
# Security Now! #966 - 03-19-24

## Morris The Second

### This week on Security Now!

Voyager lives! (Maybe). The world wide web just turned 35. What does its Dad think? What's the latest horrific violation of consumer privacy to come to light? Our listeners have been extremely engaged and interested in several of this podcast's recent topics. So we're going to use their feedback to finish off several of those topics. And finally, we look at how a group of Cornell University researchers managed to get today's generative AI models to behave badly and at just how much of a cautionary tale this may be.

"Wait... You mean you did ***not*** put this wonderful gymnasium on the floor because it's the perfect space for us to play in?"



# Security News

**Voyager 1**

A quick follow-up to our recent perhaps premature eulogy for the Voyager 1 spacecraft. It may turn out to have just been a flesh wound. The team occupying the little office space in Pasadena instructed Voyager to alter a location of its memory in what everyone is calling a "poke" instruction. Peek and poke were the verbs used by some higher level languages when that code wished to either directly inspect (to peek) or directly alter (to poke) the content of memory.

For the past several months, there has been a rising fear that the world may need to say farewell to the Voyager 1 spacecraft after it began to send back garbled data. But after being "poked" just right, the Voyager 1 sent back a read-out of the Flight Data Subsystem (FDS) memory. This brought renewed hope that the spacecraft is in a better condition than feared. In other words, it hasn't gone insane as was feared. And the return of the Flight Data Subsystem memory will allow engineers to dig through the returned memory read-out for clues. Although this data was not sent in the format that the FDS is supposed to use when it's working correctly, it's nevertheless readable. So we're not yet out of the woods and this could still be unrecoverable death throes. At some point it will be... but maybe not quite yet.

**The Web turned 35 and Dad is disappointed.**

The Web officially turned 35 and its Dad has renewed his expression of his disappointment. Since the open letter that Tim Berners-Lee posted on Medium is not too long, I want to share it, then share a few reactions. One week ago, on March 12th, Tim wrote:

> ***(Original Hope)*** *Three and a half decades ago, when I invented the web, its trajectory was impossible to imagine. There was no roadmap to predict the course of its evolution, it was a captivating odyssey filled with unforeseen opportunities and challenges. Underlying its whole infrastructure was the intention to allow for collaboration, foster compassion and generate creativity — what I term the 3 C's. It was to be a tool to empower humanity.*
>
> *The first decade of the web fulfilled that promise — the web was decentralized with a long-tail of content and options, it created small, more localized communities, provided individual empowerment and fostered huge value. Yet in the past decade, instead of embodying these values, the web has instead played a part in eroding them. The consequences are increasingly far reaching. From the centralisation of platforms to the AI revolution, the web serves as the foundational layer of our online ecosystem — an ecosystem that is now reshaping the geopolitical landscape, driving economic shifts and influencing the lives of people around the World.*
>
> ***(State of Affairs)*** *5 years ago, when the web turned 30, I called out some of the dysfunction caused by the web being dominated by the self-interest of several corporations that have eroded the web's values and led to breakdown and harm. Now, 5 years on as we arrive at the Web's 35th Birthday, the rapid advancement of AI has exacerbated these concerns, proving that issues on the web are not isolated but rather deeply intertwined with emerging technologies.*
>
> *There are two clear, connected issues to address. The first is the extent of power concentration, which contradicts the decentralized spirit I originally envisioned. This has*

*segmented the web, with a fight to keep users hooked on one platform to optimize profit through the passive observation of content. This exploitative business model is particularly grave in this year of elections that could unravel political turmoil. Compounding this issue is the second, the personal data market that has exploited people's time and data with the creation of deep profiles that allow for targeted advertising and ultimately control over the information people are fed.*

*How has this happened? Leadership, hindered by a lack of diversity, has steered away from a tool for public good and one that is instead subject to capitalist forces resulting in monopolization. Governance, which should correct for this, has failed to do so, with regulatory measures being outstripped by the rapid development of innovation, leading to a widening gap between technological advancements and effective oversight.*

*The future hinges on our ability to both reform the current system and create a new one that genuinely serves the best interests of humanity. To achieve this, we must break down data silos to encourage collaboration, create market conditions in which a diversity of options thrive to fuel creativity, and shift away from polarizing content to an environment shaped by a diversity of voices and perspectives that nurture empathy and understanding.*

***(Level Set)** To truly transform the current system, we must simultaneously tackle its existing problems and champion the efforts of those visionary individuals who are actively working to build a new, improved system. A new paradigm is emerging, one that places individuals' intention rather than attention at the heart of business models, freeing us from the constraints of the established order and returning control over our data. Driven by a new generation of pioneers, this movement seeks to create a more human-centered web, aligned with my original vision. These innovators hail from diverse disciplines — research, policy, and product design — united in their pursuit of a web, and related technologies, that serve and empower us all. **Bluesky** and **Mastadon** don't feed off our engagement but still create group formation, **Github** provides online collaboration tools and podcasts contribute to community knowledge. As this emergent paradigm gains momentum, we have the opportunity to re-shape a digital future that prioritizes human well-being, equity, and autonomy. The time to act and embrace this transformative potential is now.*

***(Fundamental Change)** As outlined in the 'Contract for the Web' a multitude of stakeholders must collaborate to reform the web and guide the development of emerging technologies. Innovative market solutions, like those I've highlighted, are essential to this process. Forward-thinking legislation from governments worldwide can facilitate these solutions and help manage the current system more effectively. Finally, we as citizens all over the world need to be engaged, and demand higher standards and greater accountability for our online experiences. The time is now to confront the dominant system's shortcomings while catalyzing transformative solutions that empower individuals. This emergent system, ripe with potential, is rising, and the tools for control are within reach.*

*Part of the solution is the Solid Protocol, a specification and a movement to provide each person with their own 'personal online data store', known as a POD. We can return the value that has been lost and restore control over personal data. With Solid, individuals decide how their data is managed, used, and shared. This approach has already begun to take root, as seen in Flanders, where every citizen now has their own POD after Jan Jambon announced four years ago that all Flanders citizens should have a POD. This is the future of data ownership and control, and it's an example of the emergent movement that is poised to replace the outdated incumbent system.*

***(Call For Action)*** *Realizing this emergent movement won't just happen — it requires support for the people leading the reform, from researchers to inventors to advocates. We must amplify and promote these positive use cases, and work to shift the collective mindset of global citizens. The Web Foundation, that I co-founded with Rosemary Leith, has and will continue to support and accelerate this emergent system and the people behind it. However, there is a need, an urgent need, for others to do the same, to back the morally courageous leadership that is rising, collectivize their solutions, and to overturn the online world being dictated by profit to one that is dictated by the needs of humanity. It is only then that the online ecosystem we all live in will reach its full potential and provide the foundations for creativity, collaboration and compassion.*

*Tim Berners-Lee / 12th March 2024*

Call me jaded. Call me old. But I do not see any way for us to get from where we are today to anything like what Tim envisions. The web has been captured – hook, line and sinker – by commercial interests; and they are never going to let go. Diversity? The one browser most of the world uses is maintained by the world's largest advertiser. And no one forced that to happen. For some reason most people apparently just like that colorful round Chrome browser icon. And Chrome is cleaner looking. Its visual design is appealing. Somehow the word spread that it was a better browser and nothing convinced people otherwise. And what Microsoft has done to their Edge browser would drive anyone anywhere else. But I've wandered away from my point.

People do not truly care about things that they neither see nor understand. The technologies that are being used to track us around the Internet and to collect data on our actions are both unseen and poorly understood. People have some dull sense that they're being tracked, but only because they've heard it said so many times. They don't have any idea what that really means. They certainly have no idea about any of the details and they have better things to worry about.

Tim writes: *"Part of the solution is the Solid Protocol, a specification and a movement to provide each person with their own 'personal online data store', known as a POD. We can return the value that has been lost and restore control over personal data."* While I honor Tim's spirit and intent, I seriously doubt that almost anyone could be bothered to exercise control over their online personal data repository. The listeners of this podcast would likely be curious to learn more. But as one of my ex-girlfriends used to say "we're not normal."

My feeling is that the web is going to do what the web is going to do. Yes, there are things wrong with it. And, yes, it can be deeply invasive of our privacy. But it also appears to largely finance itself, apparently at least in part thanks to those same privacy invasions. We pay for bandwidth access to the Internet, and the rest is free. Once we're connected we have virtually instantaneous and unfettered access to a truly astonishing breadth of information. And it's mostly free. There are some annoying sites that won't let you in without paying. So most people simply go elsewhere. The reason most of the web is free is that, with a few exceptions such as Wikipedia, for-profit commercial interests see an advantage to them for providing it. Are we being tracked in return? Apparently. But if that means we get everything for free do we care? If having the Internet know whether I wear boxers or briefs means that all of this is opened up to me without needing to pay individually for every site I visit then, yeah, okay... "briefs" have always been my thing.

Tim may have invented the World Wide Web 35 years ago, but he certainly did not invent what the web has become. The web has utterly outgrown its parent and it's finding its own way in the world. It is far beyond discipline and far beyond control and most importantly of all, today it is already giving most people exactly what they want. Good luck changing that.

**Wow... Just Wow.**

In the show notes, I gave the title of this bit of news the title "Wow... Just Wow" because it tells the story of something that's so utterly violating of consumer rights and privacy. The headline in last week's New York Times read: *"Automakers Are Sharing Consumers' Driving Behavior With Insurance Companies"* And the sub-heading read: *"LexisNexis, which generates consumer risk profiles for insurers, knew about every trip G.M. drivers had taken in their cars, including when they sped, braked too hard or accelerated rapidly."* Here's the real world event that The New York Times used to frame this disclosure:

> *Kenn Dahl says he has always been a careful driver. The owner of a software company near Seattle, he drives a leased Chevrolet Bolt. He's never been responsible for an accident.*
>
> *So Mr. Dahl, 65, was surprised in 2022 when the cost of his car insurance jumped by 21 percent. Quotes from other insurance companies were also high. One insurance agent told him his LexisNexis report was a factor.*
>
> *LexisNexis is a New York-based global data broker with a "Risk Solutions" division that caters to the auto insurance industry and has traditionally kept tabs on car accidents and tickets. Upon Mr. Dahl's request, LexisNexis sent him a 258-page "consumer disclosure report," which it must provide per the Fair Credit Reporting Act.*
>
> *What it contained stunned him: more than 130 pages detailing each time he or his wife had driven the Bolt over the previous six months. It included the dates of 640 trips, their start and end times, the distance driven and an accounting of any speeding, hard braking or sharp accelerations. The only thing it didn't have is where they had driven the car.*
>
> *On a Thursday morning in June for example, the car had been driven 7.33 miles in 18 minutes; there had been two rapid accelerations and two incidents of hard braking.*
>
> *According to the report, the trip details had been provided by General Motors — the manufacturer of the Chevy Bolt. LexisNexis analyzed that driving data to create a risk score "for insurers to use as one factor of many to create more personalized insurance coverage," according to a LexisNexis spokesman, Dean Carney. Eight insurance companies had requested information about Mr. Dahl from LexisNexis over the previous month.*
>
> *"It felt like a betrayal," Mr. Dahl said. "They're taking information that I didn't realize was going to be shared and screwing with our insurance."*

Since this behavior is so horrifying I'm going to share a bit more of what The New York Times wrote:

*In recent years, insurance companies have offered incentives to people who install dongles in their cars or download smartphone apps that monitor their driving, including how much they drive, how fast they take corners, how hard they hit the brakes and whether they speed. But "drivers are historically reluctant to participate in these programs," as Ford Motor put it in a patent application that describes what is happening instead: Car companies are collecting information directly from internet-connected vehicles for use by the insurance industry.*

*Sometimes this is happening with a driver's awareness and consent. Car companies have established relationships with insurance companies, so that if drivers want to sign up for what's called usage-based insurance — where rates are set based on monitoring of their driving habits — it's easy to collect that data wirelessly from their cars.*

*But in other instances, something much sneakier has happened. Modern cars are internet-enabled, allowing access to services like navigation, roadside assistance and car apps that drivers can connect to their vehicles to locate them or unlock them remotely. In recent years, automakers, including G.M., Honda, Kia and Hyundai, have started offering optional features in their connected-car apps that rate people's driving. Some drivers may not realize that, if they turn on these features, the car companies then give information about how they drive to data brokers like LexisNexis.*

*Automakers and data brokers that have partnered to collect detailed driving data from millions of Americans say they have drivers' permission to do so. But the existence of these partnerships is nearly invisible to drivers, whose consent is obtained in fine print and murky privacy policies that few read.*

*Especially troubling is that some drivers with vehicles made by G.M. say they were tracked even when they did not turn on the feature — called OnStar Smart Driver — and that their insurance rates went up as a result.*

Stepping back from the specifics of this, add the context of Tim Berners-Lee's unhappiness with what the web has become, and the growing uneasiness over the algorithms being used by social media companies to enhance their own profits, even when those profits come at the cost of the emotional and mental health of their users, we see example after example of amoral aggressive profiteering by major enterprises, where the operative philosophy appears to be "We'll do this to make as much money as we can, no matter who is hurt, until the governments in whose jurisdictions we're operating get around to creating legislation which specifically prohibits our conduct. And until that finally happens, we'll do everything we can to fight against those changes, including, where possible, lobbying those governmental legislators."

As I said earlier, the Internet and the World Wide Web is self-funding. Unfortunately, not all of the Internet benefits those who use it.

I'm supremely pessimistic about Tim Berners-Lee's, or anyone's, ability to change this. The forces that have created it are far more powerful and motivated. I believe that the best thing we can be is alert and aware. Since I fully expect to outlive my beloved, Internet-free, 21 year old BMW, when I do purchase a new car I'm sure I'm going to be reading the fine print.

# Closing the Loop

**Montana J / @Montana_Jim777**

*Hey, a flaw in passkey thinking. I teach computer science at a college. Like many in the educational field, I log on to a variety of computers a day that are used by myself, fellow instructors and students. Using a passkey in this environment would allow others to easily gain access to my accounts. NOT a good thing. So turning off passwords is not an option. Just something to think about. –Jim*

Right. That's a very good point which I tend to forget since none of my computers are shared. In a machine sharing environment there are two strong options: FIDO in a dongle is one way to obtain the benefits of passkeys-like public key identity authentication with portability. But also, reminiscent of SQRL, a smartphone can completely replace a FIDO dongle to serve as a passkeys authentication client by using the QR code presented by a passkeys website. And in that model, passkeys probably provides just about the best possible experience and security for shared computer use.

**Gilding_timings / @GildingT72732**

*Hey Steve, I just finished watching episode 965 on passkeys vs. 2FA. I was wondering, don't passkeys just change who is responsible for securing your authentication data? With passwords and 2FA, the responsibility is with the website. With passkeys, the responsibility is with the tool storing the passkeys (ex. password manager). If the password manager is compromised, an attacker has all they need to authenticate as you. I would think that if the website doesn't allow disabling password authentication, then 2FA still has some value if we're talking about password managers being compromised. You can at least store the 2FA data separately from your password manager. I'm loving spinrite 6.1. It's already come in handy multiple times. Thanks so much for continuing the show. I look forward to it every week.*

Okay. First, after three years of work on it, I certainly appreciate the SpinRite feedback and I'm delighted to hear that it has come in handy.

So here's the way to think about authentication security: All of the authentication technology in use today requires the use of secrets that must be kept. The primary difference among the various alternatives is where those secrets are kept, and who is keeping them.

In the username/password model, assuming the use of unique and very strong passwords, the secrets used must be kept at both the client's end, so they can provide the secret, and at the server's end, so that it can verify the secret provided by the client. So we have two separate locations where secrets must be kept.

By comparison, thanks to passkeys' entirely different public-key technology, we've cut the storage of secrets in half. Now, only the client side needs to be keeping secrets since the server side is only able to verify the client's secrets without needing to retain any of them itself. So it's clear that by cutting the storage of secrets in half we already have a much more secure authentication solution. But the actual benefit is far greater than 50%.

Where does history teach us the attacks happen? When the infamous bank robber, Willie Sutton, was asked why he robbed banks his answer was obvious in retrospect. He said: "Because that's where all the money is." For the same reason, websites are attacked much more than individual users because that's where all the authentication secrets are stored. So when the use of passkeys cuts the storage of authentication secrets by half, the half that it's cutting is where nearly all of the theft of those secrets occurs. So the practical security gain is far more than just 50%. Now, our listener said: "*I would think that if the website doesn't allow disabling password authentication, then 2FA still has some value if we're talking about password managers being compromised. You can at least store the 2FA data separately from your password manager.*" That's true. And there's no question that requiring two secrets to be used for a single authentication is better than one; and that storing those secrets separately is better still.

But as we were reminded by the needs of the previous listener who works in a shared machine environment, just like 2FA, passkeys can also be stored in an isolated smartphone and thus kept separate from the browser. Having our browsers' or password manager extensions storing our authentication data is the height of convenience. And we're not hearing about that actually ever being a problem–which is very comforting. But a separate device just feels as though it's going to provide more authentication security, if only in theory. The argument could be made that storing passkeys in a smartphone still presents a single point of authentication failure. But it's difficult to imagine a more secure enclave than what Apple provides, back up by per-use biometric verification.

**Mike Schepers / @mike91788**

> *@SGgrc Hi Steve, I'm a long time listener of securitynow and love the podcast. Thank you so much for all your contributions for making this world a better place and freely giving your expertise to educate many people like myself. I do have a question for you related to Passkeys (episode 965) that I'm hoping that you can help me understand. There are many accounts that my wife and I share for things like banking and health benefits websites where we both both need access to the same accounts. If they were to use only passkeys for authentication is sharing possible? Thank you, Mike*

In a word, yes. Whether passkeys are stored in a browser-side password manager or in your smartphones, the various solutions have recognized this necessity and they provide some means for doing this. For example, in the case of Apple, under Settings > Passwords, it's possible to create a "Shared Group" for which you and your wife would be members. It's then possible for members of the group to select which of their passwords they wish to share with the group, and Apple as seamlessly extended this so that it works identically for passkeys. Apple says: *"Shared password groups are an easy and secure way to share passwords and passkeys with your family and trusted contacts. Anyone in the group can add passwords and passkeys. When a shared password changes, it changes on everyone's device."* So it's a perfect solution.

**Senraeth / @Senraeth**

There's been such an outsized interest shown in this topic by our listeners, that I wanted to share another listener's restatement and summary of the situation, even though it's somewhat redundant, just so everyone can check their facts against the assertions this person is making:

*Hi Steve, just listened to SN 965 and have a thought about Passkeys security. Completely agree with your assessment of the security advantages of Passkeys vs Passwords/MFA in general, but another practical difference occurs to me when using a password manager to store your Passkeys.*

*With password + MFA, if your password manager is breached somehow, you can still rest easy knowing that only your passwords were compromised and that the hackers could not actually gain access into any of the accounts in your vault that were also secure with a 2nd factor. Of course this is not true if you also use your password manager to store your MFA codes, which is why you have said in the past that you would not do that as it puts all of your eggs in one basket.*

*With passkeys stored in a password manager, this is no longer the case. If the password manager is breached, the hackers can gain access to every account that was secured with the passkeys in your vault. So, while Passkeys most definitely make you less vulnerable to breaches at each individual site, the trade-off is making you much more vulnerable to a breach of your password manager, if I'm understanding this correctly.*

*Like the original listener from last week, Stephan Janssen, this leaves me feeling hesitant to use passkeys with a password manager. I think using passkeys with a hardware device like a Yubikey would be ideal, but then you have to deal with the issue of syncing multiple devices (which of course, wouldn't have been an issue with SQRL....) Thanks for all you do!*

Apple and Android smartphones support cross-device passkey syncing and website logon via QR code. So passkeys remains the winner. No secrets are stored remotely by websites. So the impact of the most common website security breaches is hugely reduced. If you cannot get rid of, or disable, a website's parallel use of passwords then by all means protect the password with MFA if possible. And perhaps remove the password from your password manager if its compromise is a concern.

So that leaves a user transacting with passkeys for their logon, and left with the choice of where they are stored – in a browser or browser extension or on their smartphone? I would suggest that the choice is up to the user. The browser presents such a large attack surface that the quest for maximum security would suggest that storing passkeys in a separate smartphone would be most prudent. But that does create smartphone vendor ecosystem lock-in. And I'll remind everyone that we do not have a history of successful major password manager extension attacks. So the worry over giving our passkeys to our password managers to store is only theoretical. The big "but what if??"

At this point in time I doubt that there's a single right answer that applies to everyone. The lack of passkey portability is an annoyance, but we're still in the very early days and the FIDO group is working on a portability spec. So there's hope.

**CR / @Coder_rotor**

*Hi Steve, on episode 965 a viewer commented on how some sites are blocking anonymous http://duck.com email addresses or stripping out the + symbol. I want to share my approach that gets around these issues.*

*First, I registered a web domain with WHOIS privacy protection to use just for throwaway accounts. I then added the domain to my personal ProtonMail account, which requires a plan upgrade, but I'm sure there are many other email hosting services out there that are cheap or possibly free. Finally, I enabled the catch-all address option. With this in place I can now sign up on websites using any name @ my domain and those emails are delivered to the catch-all in ProtonMail. You can set up filters or real addresses if you want to bypass the catch-all should you want some organization. ProtonMail also makes it really easy to block email senders by right-clicking the email item in your inbox and selecting the block action. So far this setup has been serving well for the last year without any problems!*

I wanted to toss this into the ring as an idea that might work for some of our listeners. And I agree that it solves the problem of creating per-site or just random throwaway eMail addresses. But the problem it doesn't solve for those who care is the tracking problem, since all of those throwaway addresses would be @ the same personalized domain. The reason the @duck.com solution was so appealing is that everyone using @duck.com is indistinguishable from everyone else using @duck.com, making obtaining any useful tracking information from someone's use of an @duck.com or any similar anonymizing service, futile. And this is, of course, exactly why some websites are now refusing to accept such domains, and why this may become a growing trend ... for which there's no clear solution.

**Gabe Van Engel / @gvengel**

*Hey Steve, I wanted to send you a quick note regarding the vulnerability report topic over the last two episodes. I don't know the specifics of the issue the listener reported, but I can provide some additional context as someone who runs an open bounty program on HackerOne.*

*We require that all reports include a working proof of concept to be eligible for bounty. The reason is that many vulnerability scanners flag issues simply by checking version headers; however, most infrastructure these days does not run upstream packages distributed directly by the author, and instead use a version packaged by a 3rd party providing backported security patches. e.g. repositories from RedHat Enterprise Linux, Ubuntu, Debian, FreeBSD, etc. It is totally possible the affected company **is** vulnerable to the trivial nginx RCE, but if they think the report isn't worth acting on, it's also possible they are running a version which isn't actually vulnerable, but still returns a vulnerable looking version string. To be clear, I'm not trying to give the affected company a free pass. Even if they aren't vulnerable, the timeframe over which the issue was handled, and the lack of a clear explanation as to why they chose to take no action is inexcusable. All the best, keep up the good work, Gabe.*

*P.S. Looking forward to email so I can delete my twitter account. :)*

I thought that Gabe's input, as someone who's deep in the weeds of vulnerability disclosures at HackerOne, was very valuable. And it's interesting that they don't entertain any vulnerability submission without a working proof of concept. Given Gabe's explanation that makes sense. And it's clear that a working proof of concept would move our listener's passive observation from a casual case of *"Hey did ya happen to notice that your version of nginx is getting rather old?"* -to- *"Hey! You sure better get that fixed before someone else with fewer scruples happens to notice it, too!"*

As we know, our listener was the former of those two. He only expressed his concern only over the possibility that it might be an issue. He understood that the only thing he was seeing was a server's version headers and that therefore there was only some potential for trouble. And had the company in question clearly stated that they were aware of the potential trouble but that they had taken other steps to prevent its exploitation, the issue would have been settled. It was only their clear absence of focus upon the problem and never addressing his other questions that caused any escalation in the issue beyond an initial casual nudge.

I also wanted to take a moment to talk about Twitter. At the end of his Tweet, Gabe noted that he's looking forward to this podcast having an eMail-based listener feedback option so that he can delete his Twitter account. Many of this podcast's listeners take the time to express similar sentiments. And, at the same time, I receive occasional Tweets from listeners arguing that I'm wrong to be "leaving" Twitter as well as the merits of Twitter and how much Elon has improved it since his purchase. So for the record let me say again that I am entirely agnostic on the topic of Elon and Twitter. In other words, I don't care. More than anything I'm not a big social media user. What we normally think of as "social media" doesn't interest me at all. That said, GRC has been running quiet backwater NNTP-style text-only newsgroups since long before social media existed. And we have very useful web forums. But Twitter has never really been "social media" for. I check-in with Twitter once a week to catch up on listener feedback, to post the podcast's weekly summary, a link to the show notes and recently, our picture of the week.

What caught my attention and brought me out of my complacency, was Elon's statement that he was considering charging a subscription for **everyone's** participation, thus turning Twitter into a subscription-only service. That brought me up short and caused me to realize that what was currently a valuable and workable communications facility might come to a sudden end, because it was clear that charging everyone to subscribe to use Twitter would end it as a means for most of our current Twitter users to send feedback.

We don't all have Twitter, but we do all have eMail. So it makes more sense for me to be relying upon a stable and common denominator that will work for everyone. And since I proposed this plan to switch to eMail, many people, like Gabe, have indicated–to me through Twitter–that not needing to use Twitter would be a benefit for them, too.

**markzip @Markzip@twit.social / @markzip**

> *@SGgrc — Just catching the update about the guy who found the flaw in the big site and the unsatisfactory response from CISA/CERT. I think he should NOT take the money. I think that he should tell @briankrebs or another high profile security reporter. They can often get responses.*

This is another interesting possible avenue. My first concern, however, is for our listener's safety. And by that I don't mean his physical safety, I mean his safety from the annoying tendency of bullying corporations to launch meritless lawsuits just because they easily can. Our listener is on this company's radar now and that company might not take kindly to someone like Brian Krebs using his influential position to exert greater pressure. This was why my recommendation was to disclose to CISA and CERT. Being US government bodies, disclosing to them seems much safer than disclosing to an influential journalist.

Recall from earlier, Gabe from HackerOne. I had shared my reply with him and he later Tweeted:

> *"This is one of the benefits of running a program via HackerOne or other. By having a hacker register and agree to the program terms, it both lets us require higher quality reports and to also indemnify them against otherwise risky behavior like actually trying to run RCE's against a target system."*

That indemnification could turn out to be a big deal. And, of course, when working through a formal bug bounty program like HackerOne, it's not the hacker who interfaces with the target organization, it's HackerOne who is out in front – so not nearly as easy to ignore or silence with an implied threat.

**hescominsoon / @hescominsoon** (publicly tweeted)

> *This website with this big vulnerability should be publicly named. You are doing a disservice to everyone who uses that site by keeping it hidden. To quote you in your own words. Security by obscurity is not security. Let us know which site it is so that we can take action.*

Wouldn't it be nice if things were so simple. In the first place, this is not my information to disclose, so it's not up to me. This was shared with me in confidence. The information is owned by the person who discovered it, and he has already shared it with government authorities whose job, we could argue, it actually is to deal with such matters of importance to major national corporations. The failure to act is theirs, not his, nor mine.

The **really** interesting question all of this conjures is whose responsibility is it? Where does the responsibility fall? Some of our listeners have suggested that bringing more pressure to bear on the company is the way to **make** them act. But what gives anyone the right to do that? Publicly naming the company, as this listener asks, would very likely focus malign intent upon them. And based upon what I've previously shared about their use of an old version of nginx, the cat, as they say, would be out of the bag. At this point it's only the fact that the identity of the company is unknown that might be keeping it, and its many millions of users, safe. Security by obscurity might not provide much security, but there are situations where a bit of obscurity is all you've got.

This is a very large and publicly traded company. So it's owned by its shareholders and its board of directors who have been appointed by those shareholders are responsible to them for the company's proper, safe and profitable operation. So the most proper and ideal course of action at this point would likely be to contact the members of the board and privately inform them of the reasonable belief that the executives they have hired to run the company on behalf of its shareholders have been ignoring, and apparently intend to continue ignoring, a potentially significant and quite widespread vulnerability in their web-facing business properties. While some minion who receives anonymous eMail can easily ignore incoming vulnerability reports, if the members of the company's board were to do so, any resulting damage to the company, its millions of customers and its reputation would then be on them.

Stepping back from this a bit, I think that the lesson here is that at no point should it be necessary for untoward pressure to be used to force anyone to do anything, because doing the right thing should be in everyone's best interest. The real problem we have is that it's unclear whether the right person within the company has been made aware of the problem. At this point it's not clear that has happened, through no fault of our original listener who may have stumbled upon a serious problem and has acted responsibly at every step. If the right person had been made aware of the problem we would have to believe that it would be resolved.

So my thought experiment about reaching out to the company's board of directors amounts to "going over the heads" of the company's executives who do not appear to be getting the message. And that has the advantage of keeping the potential vulnerability secret while probably resulting in action being taken. I'm not suggesting that our listener should go to all that trouble, since that would be a great deal of thankless effort. The point I'm hoping to make is that there are probably still things that could be done short of a reckless public disclosure which could result in serious and unneeded damage to users and company alike.

**Marshall / @Marshall_Macro**

*Hi Steve, a quick follow up question to the last security now episode on MFA v Passkeys. Does the invention of Passkeys invalidate the "something you have", "something you know" , and "something you are" paradigm? Or does Passkeys provide a better instantiation of those three concepts? Because the idea with multi-factors is that you'd add another factor for greater security, but with Passkeys do you still consider those factors? Thanks for everything you do!*

I think this was a terrific question. The way to think of it is that the "something you know" is a secret that you're able to directly share. The use of "something you have" – like a one-time password generator – is actually you sharing the result of another secret you have, where the result is based upon the time of day. And the "something you are" is some biometric being used to unlock and provide a third secret. In all three instances, a local secret is made available through some means.

It's what's **done** with that secret where the difference between traditional authentication and public key authentication occurs. With traditional authentication the resulting secret is simply compared against a previously stored copy of the same secret to see whether they match. But with public key authentication such as passkeys, the secret that the user obtains at their end is used to sign a unique challenge provided by the other end. And then that signature is verified by the sender to prove that the signer is in possession of the secret private key.

Therefore, the answer, as Marshall suggested, is that passkeys provides a better instantiation of those original three concepts. For example, Apple's passkeys system requires that the user provides a biometric face or thumbprint to unlock the secret before it can be used. But a browser extension that contains passkeys merely requires its user to provide something they know to login to the extension and thus unlock its store of passkey secrets.

And as we mentioned recently, all of these additional traditional factors were layered upon each other in an attempt to shore each other up, since storing and passing secrets back and forth had turned out to be so problematic. We don't have this with passkeys because the presumption is

that a public key system is fundamentally so much more secure that a single very strong factor will provide all the security that's needed.

**Rob Mitchell / @TheBTCGame**

*Interesting to learn the advantages of passkeys, it definitely makes sense in many ways. The one disadvantage my brain sticks on, vs TOTP, is that I'd imagine someone who can get into your password manager (hack into cloud backup or signed-in on your computer) now can access your account with passkeys. Like if passkeys were a thing when people were having their LastPass accounts accessed. But if your TOTP is only on your phone, someone who gets into your password manager still can't access a site because they don't have the TOTP key stored on your phone. Maybe passkeys are still better, but I can't help but see that weakness.*

Rob's sentiment was expressed by a number of our listeners. So I just wanted to say that I agree. As I mentioned last week, needing to enter that ever-changing secret 6-digit code from the authenticator on our phone really does make everything seem much more secure. Nothing that's entirely automatic can seem as secure. So storing passkeys in a smartphone is a choice that might make sense. And as I've mentioned, the phone can be used to authenticate through the QR code that a passkey-enabled site presents. And perhaps some sites will still offer good old 6-digit MFA for those who like a belt to go with their suspenders.

**Christian Turri / @cdturri**

*Hi Steve on SN965 you discussed the issue with Chrome extensions changing owner and how Devs are being tempted to sell their extensions. There is a way to be safe when using extensions in Chrome or FireFox. Download the extension, expand it and inspect it. Once you are sure it's safe you can install it on Chrome by enabling Developer Mode under chrome://extensions/ and selecting Load Unpacked. The extension will now be locally installed which means it will never update from the store or change, it's frozen in time ("if it aint broke dont fix it"). When and if the extension breaks in the future due to Chrome changes you can get the update and perform the same process again. While using these steps requires some expertise it should be fine for most SN listeners.*

Thanks, Christian. That's a great tip which I'll bet will appeal to many of our listeners who generally prefer taking automatic things into their own hands.

**Bob Hutzel / @amoebob**

*Hi Steve, Before embracing Bitwarden's passkey support, it is important to note that it is still a work in progress. Mobile app support is still being developed. Also, passkeys are not yet included in exports. So, even if someone maintains offline vault backups, a loss of access to or corruption of the cloud vault means passkeys are gone.*
*https://bitwarden.com/help/storing-passkeys/#passkey-management-faq*
*Thank you for the great show! Bob Hutzel*

Yes… And in general, with the FIDO folks still working to come up with a universal passkeys import/export format, it doesn't feel right to have them stuck in anyone's walled garden. The eventual addition of passkey transportability should make a huge difference. As it is, they just don't seem like something tangible that we can get our hands on... and that feels creepy.

# Morris The Second

Since Ben Nassi first reached out to me a couple of weeks ago via Twitter (and added his voice to those who are looking forward to having a non-Twitter means of doing so), the work that he and his team have done has garnered a huge amount of attention. It's been picked up by Wired, PCMag, Ars Technica, The Verge, and many more outlets. In thinking about how to characterize this, I'm reminded of our early observations of conversational AI here. We talked about how the creators of these services had tried to erect barriers around certain AI responses and behaviors, but that clever hackers quickly discovered that it was possible to essentially seduce the AIs into ignoring their own rules. By asking nicely, or by being more demanding, it was often possible to get the AI to capitulate to those demands.

What Ben and his team have managed to do here can be thought of the exploitation of that essential weakness – on steroids. To quickly create some foundation for understanding this, I want to run through the Q&A that they provided since it establishes some terms and sets the stage for their far more detailed 26-page academic paper. So. . .

*Q: What is the objective of this study?*
This research is intended to serve as a whistleblower to the possibility of creating GenAI worms in order to prevent their appearance

*Q: What is a computer worm?*
A computer worm is malware with the ability to replicate itself and propagate/spread by compromising new machines while exploiting the sources of the machines to conduct malicious activity (payload).

*Q: Why did you name the worm Morris-II?*
Because like the famous 1988 Morris worm, that was developed by a Cornell student, Morris-II was also developed by two Cornell Tech students (Stav and Ben).

*Q: What is a GenAI ecosystem?*
An interconnected network consisting of GenAI-powered agents.

*Q: What is a GenAI-powered application/client/agent?*
A GenAI-powered agent is any kind of application that interfaces with (1) GenAI services to process the inputs sent to the agent, and (2) other GenAI-powered agents in the ecosystem. The agent uses the GenAI service to process an input it receives from other agents.

*Q: Where is the GenAI service deployed?*
The GenAI service that is used by the agent can be based on a local model (i.e., the GenAI model is installed on the physical device of the agent) or remote model (i.e., the GenAI model is installed on a cloud server and the agent interfaces with it via an API).

*Q: Which type of GenAI-powered applications may be vulnerable to the worm?*

Two classes of GenAI-powered applications might be at risk:

- GenAI-powered applications whose execution flow is dependent upon the output of the GenAI service. This class of applications is vulnerable to application-flow-steering GenAI worms.
- GenAI-powered applications that use RAG to enrich their GenAI queries. This class of applications is vulnerable to RAG-based GenAI worms

*Q: What is a zero-click malware?*
Malware that does not require the user to click on something (e.g., a hyperlink, a file) to trigger its malicious execution.

*Q: Why do you consider the worm a zero-click worm?*
Due to the automatic inference performed by the GenAI service (which automatically triggers the worm), the user does not have to click on anything to trigger the malicious activity of the worm or to cause it to propagate.

*Q: Does the attacker need to compromise an application in advance?*
No. In the two demonstrations we showed, the applications were not compromised ahead of time. They were compromised when they received the email.

*Q: Did you disclose the paper with OpenAI and Google?*
Yes, although, this is not OpenAI's or Google's responsibility. The worm exploits bad architecture design for the GenAI ecosystem and is not a vulnerability in the GenAI service.

*Q: Are there any similarities between adversarial self-replicating prompts and buffer overflow or SQL injection attacks?*
Yes. While a regular prompt is essentially code that triggers the GenAI model to output data, an adversarial self-replicating prompt is a code (prompt) that triggers the GenAI model to output code (prompt). This idea resembles classic cyber-attacks that exploited the idea of changing data into code to carry out their attack.

- an SQL injection attack embeds code inside a query (data).
- a buffer overflow attack writes data into areas known to hold executable code (code).
- an adversarial self-replicating prompt is a code that is intended to cause the GenAI model to output prompt (code) instead of data.

It’s clear that a gold rush mentality has taken shape, with everyone rushing to stake out their claim over what appears to be a huge new world of online services that can be made available by leveraging this groundbreaking new capability. But as always, when we rush ahead mistakes are inevitably made and some stumbles, perhaps even large ones, can occur.

The wake up call this “Morris the Second” research provides has arrived at a vital time and certainly not a moment too early.

Here’s how these researchers explain what they have accomplished:

*In the past year, numerous companies have incorporated Generative AI (GenAI) capabilities into new and existing applications, forming interconnected Generative AI (GenAI) ecosystems consisting of semi/fully autonomous agents powered by GenAI services.*

*While ongoing research highlighted risks associated with the GenAI layer of agents (e.g., dialog poisoning, membership inference, prompt leaking, jailbreaking), a critical question emerges: Can attackers develop malware to exploit the GenAI component of an agent and launch cyber-attacks on the entire GenAI ecosystem?*

*This paper introduces Morris II, the first worm designed to target GenAI ecosystems through the use of adversarial self-replicating prompts. The study demonstrates that attackers can insert such prompts into inputs that, when processed by GenAI models, prompt the model to replicate the input as output (which yields replication), engaging in malicious activities (thus carrying a payload).*

*Additionally, these inputs compel the agent to deliver them (so we get propagation) to new agents by exploiting the inter-connectivity within the GenAI ecosystem. We demonstrate the application of Morris II against GenAI-powered email assistants in two use cases (spamming and exfiltrating personal data), under two settings (black-box and white-box accesses), using two types of input data (text and images). The worm is tested against three different GenAI models (Gemini Pro, ChatGPT 4.0, and LLaVA), and various factors (e.g., propagation rate, replication, malicious activity) influencing the performance of the worm are evaluated.*

The "Ethical Considerations" section of their 26-page paper was interesting. They wrote:

*The entire experiments conducted in this research were done in a lab environment. The machines used as victims of the worm (the "hosts") were virtual machines that we ran on our laptops. We did not demonstrate the application of the worm against existing applications to avoid unleashing a worm into the wild. Instead, we showcased the worm against an application that we developed running on real data consisting of real emails received and sent by the authors of the paper and were given by the authors of their free will to demonstrate the worm using real data. We also disclosed our findings to OpenAI and Google using their bug bounty systems.*

So, unlike the first Morris worm, which escaped from MIT's network at 8:30 pm on November 2nd, 1988 after having been created by Cornell University graduate student Robert Morris, today's Cornell University researchers were extremely careful not to <quote> "see what would happen" <unquote> if they were to turn their creation loose upon any live Internet services. One thing we've learned quite well during the intervening 36 years is exactly what would happen... and it would neither be good, nor would it further the careers of these researchers.

Their paper ends on a somewhat ominous note that feels correct to me. They conclude, writing:

*While we hope this paper's findings will prevent the appearance of GenAI worms in the wild, we believe that GenAI worms **will** appear in the next few years in real products and will trigger significant and undesired outcomes. Unlike the famous paper on ransomware that was authored in 1996 and preceded its time by a few decades (until the Internet became widespread in 2000 and Bitcoin was developed in 2009), we expect to see the application of*

*worms against GenAI-powered ecosystems very soon (perhaps maybe even in the next two-three years) because (1) the infrastructure (Internet and GenAI cloud servers) and knowledge (adversarial AI and jailbreaking techniques) needed to create and orchestrate GenAI worms already exists, (2) GenAI ecosystems are under massive development by many companies in the industry that integrate GenAI capabilities into their cars, smartphones, and operating systems, and (3) attacks always get better, they never get worse.* [We know at least one podcast these guys listen to.] *We hope that our forecast regarding the appearance of worms in GenAI ecosystems will turn out to be wrong because the message delivered in this paper served as a wake-up call.*

Okay. What these guys have done is crucial. They have vividly shown – by demonstration that cannot be denied – just how very immature, unstable and inherently dangerous today's first-generation open-ended interactive generative AI models are. They are extremely subject to manipulation and abuse. The question that remains outstanding for me is whether they can ever be made safe? That's not necessarily a given. Safe-**ER,** certainly. But safe enough to be stable while still delivering benefits? That remains to be seen – and proven. We still can't seem to get the bugs out of simple computers whose operation we fully understand. How are we going to do so for systems whose behavior is emergent and whose complexity literally boggles the mind?

I'm glad it's ***not*** my problem!