Morris the Second

Description: 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.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-966.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-966-lq.mp3

SHOW TEASE: Coming up on Security Now!, first we talk about Voyager. You may have thought that Voyager was buh-bye, but it is not. It is still sending signals. Plus, you know, Tim Berners-Lee, the Father of the Web, sits down and has a serious talk, well, at least as far as Tim Berners-Lee sees it, about the state of the Internet and maybe some disagreements he has with the way it is all going. We also have a lot of feedback from listeners about last week's episode regarding Passkeys and the current state of security online when it comes to passwords and two-factor authentication and what you should or shouldn't use. And we also talk about Morris the Second, a worm that has some serious implications for generative AI agents. All of that and so much more coming up on Security Now!.

MIKAH SARGENT: This is Security Now! with Steve Gibson and this week Mikah Sargent, Episode 966, recorded Tuesday, March 19th, 2024: Morris the Second.

Hello and welcome to Security Now!, the show where the cybersecurity guru, Steve Gibson, provides a week's worth of cybersecurity news in but a small package, so you can just plug in directly and download it into your cranium. I am just here to help facilitate this. I am, how do you say, let's go with the super safe USB flash drive that you can plug into your cranium. I'm just providing the means by which you connect to Steve Gibson, who actually is providing all of the information. That's the role I play here. Also the role of shock and awe because occasionally I am gobsmacked by what Steve ends up telling us. But Steve, it is good to see you again this week. How you doing?

Steve Gibson: Mikah, great to be with you for our second week in a row, as Leo finishes working on his tan and presumably gets ready to return. So we're going to do, as we thought we were going to do last week, but got pushed because of, wow, what turned out to be a surprisingly interesting episode for our listeners, the Passkeys vs Multifactor Authentication. In fact, I was so overwhelmed with feedback from that, and it was useful stuff, comments and questions and so forth, that it ended up still being a lot of what we end up talking about today just because, you know, certainly the whole issue of cross-network proving who you say you are, which is to say authentication, is a big deal, and

important to everybody who's using the Internet. So we're going to talk about that some more.

But we are going to get to what was supposed to be last week's topic, which I had originally as Morris II, but I saw that they're referring to themselves or their creation as Morris the Second. So Morris the Second is today's number 966 and counting Security Now! podcast title. But first we're going to talk about how it may be that we were doing the requiem for Voyager I a little prematurely. It may not be quite dead or insane or whatever it was that it appeared to be a week ago. Also, the World Wide Web has just turned 35. What does its dad think about how that's going? What's the latest unbelievably horrific violation of consumer privacy which has come to light?

We're going to share a lot about what our listeners thought about Passkeys and multifactor authentication and the ins and outs of all that. And then, as I promised, we're going to look at how a group of Cornell University researchers managed to get today's Generative AI to behave badly, and at just how much of a cautionary tale this may be. So I think a lot of interesting stuff for us to talk about.

MIKAH: Absolutely. Again, shock and awe, I can't wait. I'm looking forward to it.

Steve: So, okay. I have a large collection of photos in my archive that I'm ready to deploy on demand. But this one just caught me by surprise. Somebody tweeted it, and I thought it was so cute. The caption I gave it was, "Wait. You mean you did not put this wonderful gymnasium on the floor because it's the perfect space for us to play in?" And we have an open old-school large computer case which three kittens have managed to get themselves into, and a fourth one is sort of looking on enviously.

One of them, the upper kitten looks like it's standing on maybe an audio card or a graphics card that's been added to the case. I actually have a number of these exact cases. I look at it, and it's very familiar. The motherboard in there is actually pretty ancient, so I don't know what the story is, where this came from or what's going on. And the hard drives look like they are in need of some help. But anyway, just a fun picture of just - and my god, these little kittens are adorable.

MIKAH: They're so adorable.

Steve: I mean, how could anything be that cute as these things are? So not exactly a cat video, but cat video done Security Now! style.

MIKAH: Yes, indeed. Those adorable little cats that you should absolutely keep - you know, I like to imagine that this is somebody who brought in their computer to a place and said, "There's something wrong with this thing." The person that's cleaning it out opens it up, and these little kittens are inside playing. It's like, ah, that's what's wrong.

Steve: Yeah, a little fur ball, that's right.

MIKAH: Isn't that kind of - that's what SpinRite does; right? It just sends little kittens inside to fix the hard drive?

Steve: That's my secret formula.

MIKAH: Aha, I figured it out.

Steve: Yes. That's right.

MIKAH: All right. Let's get to the security news.

Steve: Okay. So we have a quick follow-up to, as I said, our recent, perhaps premature eulogy for the Voyager 1 spacecraft. It may just turn out to have been a flesh wound. The team occupying that little office space in Pasadena instructed Voyager to alter a location of its memory in what everyone who's covering this news is calling a "poke" instruction. Okay, now, peek and poke were the verbs used by some higher level languages when the code, which is talking in terms of variables and not in terms of storage, wished to either directly inspect, which was to say "to peek," or to directly alter, "to poke," the contents of memory.

So 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 just garbled data that nobody understood. And so we were saying, well, it's lost its mind. It's gone insane. It's just spitting gibberish. But after being poked just right, and then waiting, what, 22.5 hours twice I think is the current roundtrip time, the speed of light roundtrip - so, you know, you poke, and then you're very patient - the Voyager 1 began to read out the data from its Flight Data Subsystem, the FDS. That is, basically, it began doing a memory dump.

And this brought renewed hope that the spacecraft is actually still, somewhat miraculously, in better condition than was feared. In other words, it hasn't gone insane. And the return of the Flight Data Subsystem memory will allow engineers to dig through the returned memory readout for clues. Although, paradoxically, the data was not sent in the format that the FDS is supposed to use when it's working correctly, it is nevertheless readable. So we're not out of the woods yet, and it still could be unrecoverable, and this is just another one of its death throes. And really, I mean, realistically, at some point it will be. I mean, these veterans are going to have to turn the lights off for the last time and put their office space back up for lease. But apparently not yet.

So I expect that we'll be checking in from time to time to see how our beloved Voyager 1 spacecraft is doing. But the game is not up yet. So that's very cool. And at this point, you know, it's not clear how much new science is being sent back. I mean, it was incredibly prolific while it was moving through the planets of the solar system and sending back amazing photos of stuff that we'd never seen before. At this point it's sort of being kept alive just because it can be. So, you know, why not? It's not very expensive to do.

Okay. So the web officially turned 35, and its dad, Tim Berners-Lee, has renewed his expression of his disappointment over how things have been going recently.

MIKAH: Oh, this is one of those "I'm not angry, I'm disappointed" situations? I was hoping he was happy with his son.

Steve: Yes, my son. I'm disappointed in the way you have turned out.

MIKAH: Oh.

Steve: So one week ago, on March 12th, Tim wrote, he said: "Three and a half decades ago, when I invented the web" - which, you know, few people can say - "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." Okay, I would argue that we got two of those three, at least; you know? He says: "What I term the three C's."

Now, of course a lot of this is retrospective; right? It's like, it's easy to rewrite history 35 years later. But we'll see. Anyway, he says: "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 centralization 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.

"Five 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, five 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." If indeed he originally did. He says: "This has segmented the web, with a fight to keep users hooked on one platform" - gee, wonder what that could be - "to optimize profit through the passive observation of content." You know, like while they drool. "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," he writes, "hinges on our ability to both reform the current system and create a new one that genuinely serves the best interests of humanity." To which I'm just going to insert here, good luck with that.

Anyway: "To achieve this," he writes, "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." Or we could just all watch cat videos because, you know, those are cute. Anyway, he says: "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 Mastodon don't feed off of our engagement, but still create group formation. GitHub provides online collaboration tools. And podcasts contribute..."

MIKAH: Nice.

Steve: "...to community knowledge. As this emergent paradigm gains momentum" - I should mention podcasts that are disappearing rapidly, unfortunately. "As this emergent paradigm gains momentum, we have the opportunity to reshape a digital future that

prioritizes human well-being, equity, and autonomy. The time to act and embrace this transformative potential is" - guess what.

MIKAH: Now.

Steve: Now. Uh-huh. "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" - okay, now, there's an oxymoron for you - "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."

MIKAH: It's starting to sound like a manifesto a little bit.

Steve: It really is. And I only have a little bit more, and then I'm going to - we'll discuss this. "Part of the solution is the so-called Solid Protocol" - capital S, capital P - "a specification and a movement to provide each person with their own 'personal online data store,' known as a POD." P-O-D; right? Personal Online Data. "We can return the value that has been lost and restore control over personal data." By putting it in a POD. "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 result of data ownership and control, and it's an example of the emergent movement that is poised to replace the outdated incumbent system."

And finally, "Realizing this emergent movement won't just happen" - boy, is he right about that. Oh, I mean, he says: "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 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 of March, 2024.

MIKAH: Well, you've got your PODs; right?

Steve: 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? Well, 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. You know, 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. You know? How do you care about something that you don't really understand?

MIKAH: Yup.

Steve: 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. Oh, I'm being tracked; you know? But they don't know. They don't see it. They just kind of think, okay. It makes them feel uncomfortable, but they still do what they were doing; you know? 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.

MIKAH: Yes. Most importantly, they have better things to worry about.

Steve: Yes. Right.

MIKAH: Yeah, absolutely.

Steve: 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." Now, okay. While I honor Tim's spirit and intent - I really do - I seriously doubt that almost anyone could be bothered to exercise control over their online personal data repository. I mean, I don't even know what that looks like. 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 be largely self-financing, 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 really 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, 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. That of course is why he's so upset. 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.

MIKAH: Well put, Steve. Honestly. When I think about this - and here you go. You said maybe you're jaded and old and this and that and the other. I may be jaded, but I'm not exactly aged. And so even as a relative youth, hearing that, you know, I want to, I don't know, put on a French beret and chant and say hurrah and feel it. And I do feel it. But I think realistically it is not - it's not realistic, if we're being honest.

Steve: Right.

MIKAH: And so as cool as that would be, and as amazing as that would be, yeah, ultimately what you're saying about the stuff that Tim is talking about here, you know,

Tim Berners-Lee is talking about here, is so abstracted from how people use these devices to connect to the Internet and to communicate with one another that, yeah, it would require some level of sitting everyone down across the entire world and explaining to them how all of this works for there to be even the beginning of a concern about what would be necessary to convince everybody that they should care about this. And as I'm saying just then, you heard all the hedging that kind of took place there. It wouldn't even necessarily make a difference, even if you did explain it, because they still have to care about it. And most importantly, most people don't need to care about it. And so they...

Steve: Yes.

MIKAH: ...have bigger, better things in their world that they have to care about. And that is, I think, always going to be the case. And that's, you know, the people who do care about this stuff, we do our best to communicate and educate. But, yeah, I don't know, I mean, as much as you might - I mean, I don't know. To take the time to write all this out and to put forth this idea I think is a very noble thing. But I do wonder. I wish I could talk to Tim Berners-Lee sort of just, you know, face to face and say...

Steve: What are you thinking?

MIKAH: Yeah, what are you - do you really think that anyone's going to do this? Or are you just - this is just a hopeful sort of I'm putting it out into the world, like...

Steve: How high is that ivory tower?

MIKAH: Exactly. Exactly. We're down here. Yeah. I don't know.

Steve: Yeah. And again, I really do believe that most users' wishes are now being fulfilled.

MIKAH: Right.

Steve: You know? I mean, my wife asks me a few questions every evening, and I say, well, did you google it? You know, it's like, that's what I do. I just ask. I ask the Internet, and it tells me the answer. Because there's so much going on, it's so complicated now, that the right model is no longer to try to know everything. It's simply to know how to find out. You know, that's the future, with the knowledge explosion that we're now in, and the content explosion.

So I just, again, there was an anecdote for a while, I don't remember now exactly what it was, but it was something like I have - back when people were typically using a single password, like, universally for all their stuff, someone did an experiment where they went up to people and said, here's a lollipop. I'll trade you for your password. And most people said okay. You know? I mean, they just didn't give a rat's ass, you know, about security.

MIKAH: Right.

Steve: And most people just aren't as focused, I mean, this podcast is all about this kind of focus. And as I said, an ex-girlfriend used to say to me, "You're not normal." So, yeah, we're not. But, you know, most of the world, they just, you know, the Internet does what they want. And start asking them to pay...

MIKAH: Right.

Steve: ...in some significant way? I mean, look at the Club. Look at Club TWiT. I mean...

MIKAH: Yeah, it's a very small percentage of the overall listener base, yes. And, I mean, and you say pay in some significant way. Anecdotally, it is going to have to be a significant amount of payment for somebody to go, well, suddenly I don't care about that anymore. There have been a number, like an app, oh, I saw that everybody's posting these photos of themselves that have been AI generated. How do they do that? I say, oh, it's this app, and you pay like 56 cents to get a photo generation. Oh, never mind. I don't care about that anymore.

Steve: Yeah.

MIKAH: It's 50 cents. But yeah, it doesn't take that much for them to be like, no no no no no. That's not something I'm into.

Steve: No. And as we know, there are some sites which have survived in the "pay to enter" model. But many of the early attempts fell flat because the moment the sites put up a paywall, most people said, eh, you know, I clicked the first link in Google, and it took me to the paywall. What's the second link take me to? Oh, look, it's free.

MIKAH: Able to get to it, yeah.

Steve: Okay. So 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 that it needed that title. The headline in last week's New York Times read: "Automakers Are Sharing Consumers' Driving Behavior With Insurance Companies." And the subhead read: "LexisNexis, which generates consumer risk profiles for insurers, knew about every trip GM drivers had taken in their cars, including when they sped, braked too hard, or accelerated rapidly."

MIKAH: Wow.

Steve: Okay. So here - it's astonishing.

MIKAH: Yeah, wow.

Steve: I know, exactly. Here's the real-world event that The New York Times used to frame their disclosure. They wrote: "Kenn Dahl says" - now, that's not D-O-L-L, that's D-A-H-L, and it's K-E-N-N, 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, at age 65, was surprised in 2022 when the cost of his car insurance jumped by 21%. Quotes from other insurance companies were also high. One insurance agent told him his LexisNexis report was a factor.

"LexisNexis," they write, "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." Okay, right, public record things; right? I mean, like accidents and tickets, that's out there. "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. Mr. Dahl said: 'It felt like a betrayal. They're taking information that I didn't realize was going to be shared and screwing with our insurance.'"

Okay, now, since this behavior is so horrifying, I'm going to share a bit more of what The New York Times wrote. They said: "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 Ford Motor put it: 'Drivers are historically reluctant to participate in these programs.'" And this was written 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." In other words, monetizing; right? Because you know the insurance industry is paying to receive that information. So another means by which today's consumer is being monetized without their knowledge.

The New York Times says: "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 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 GM, 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." And again, not give, sell.

"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 ever read. Especially troubling is that some drivers with vehicles made by GM 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."

MIKAH: I do have a problem with that last bit, simply because someone says that. I almost wish that there was some due diligence there. I'm sure you've seen it. Somebody's having an issue, a tech issue, and you say, "Oh, here's how you fix it," and they say, "I've done that," and then you go and check and they didn't do that thing that you told them to do and that they should have done it.

Steve: Right.

MIKAH: I wouldn't be surprised that they did accidentally opt-in. But all that's to say, whether you opt in or not, this is still something that should be brought to light. And as for all of it, especially if it's kind of being put forth as an idea of, oh, here are these cool features you get, and secretly underneath what it's doing is giving access, yeah, that's bad. I just, I don't know if I like that from The New York Times there at the end.

Steve: Well, so one analogy that occurs to me is how - and we've mentioned this a number of times in prior years during the podcast - is employees in an organization sometimes believe that what they do on their corporate computer is private, is, like, their

business, and even when the employee agreement and occasional reminder meetings and so forth say that's not the case, that this is a corporate network, corporate bandwidth, a corporate computer, and what you do is owned by the company. So we have suggested that that really ought to be on a label running across the top of their monitor.

MIKAH: Yes.

Steve: Like, it literally ought to say right in front of them, you know, please remember that everything you do on this computer, which is owned by the company, on the bandwidth owned by the company, and the data owned by the company, is not private.

MIKAH: Right.

Steve: Similarly, by analogy, the screens that all these computers have, imagine if they said along the bottom, your driving is being monitored by the company you purchased this from and is being sold to your car insurance provider.

MIKAH: They don't want to do that, Steve.

Steve: Obviously we're never going to see that. But, you know, that's the point, is that this is going on surreptitiously, and it being surreptitious is clearly wrong. So anyway, stepping back, okay, from the specifics of this particularly egregious behavior, 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 own 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. But until that happens, we'll do everything we can to fight against those changes, including where possible lobbying those governmental legislators."

MIKAH: Honestly, we could just take that text and slap it on the screen anytime we talk about any antitrust legislation across any of our shows, and that perfectly sums up exactly what's going on in every single case.

Steve: It's like, make us stop.

MIKAH: Yeah, exactly.

Steve: And until you do, we're going to use every clever means we have of profiting from every area in which we have not been made to stop. You know, there's no more morality. There's no more ethics. It's, you know, it's profit, profit, profit, profit, profit.

MIKAH: Wherever possible, yeah.

Steve: And that is exactly Tim Berners-Lee's complaint, and it's never going to change. Because it's just too pervasive; right? I mean, it just; you know? And again, as a consequence of this, you know, the Internet is largely free. And I think that's a tradeoff most people would choose to make, rather than having to pay, like, you know, remember that there was early talk about micropayments where, when you went to a website, it would ding you some fraction of a something or other? And that would make people very uncomfortable. They'd be like, well, wait a minute, you know, suddenly links are not free to click on.

MIKAH: Yeah, exactly.

Steve: There's a cost to clicking on that link.

MIKAH: How many times per month should I click on this? And you've got - you're telling your kids don't click on links. And it would just completely reshape everything. And then there would be so many - I can imagine how much more money and time would have to go into customer support because someone would click on a link and then say, I'm not satisfied with this page, and I don't want to have paid for this page because I didn't get the answer I wanted.

Steve: That's a very good point. That's a very good point because right now it's like, well, it's free. So, you know, go pound sand somewhere; you know? It's like, tough.

MIKAH: Yeah.

Steve: Well, again, I don't mean to be just simply complaining because I also recognize, as I said, this is why the web is here. I mean, I was present during, like, pre-web, during the early days, and when there was like not much on the 'Net. And the question was, well, why is anyone going to put anything on the Internet because there's nobody on the Internet to go see it. So there was like this chicken-and-egg problem; right? No, like, vast population are actually going onto the Internet to do anything. So why is anyone going to put anything there? And if no one puts anything there, then no one is going to be incentivized to go and get what's not there. So it happened anyway.

And the way it's evolved, as I said, I'm really - I'm actually not complaining. This is not, I mean, if it sounds like I'm, you know, doing some holier than thou rant, it's not the case. I like it the way it is. And those of us who are clever enough to mitigate the tracking that's being done and the monetizing of ourselves, well, we get the benefit that is being reaped by all those who aren't. So it works.

MIKAH: Steve Gibson, let's close that loop.

Steve: Let's do it. So Montana J, he wrote: "Hey, a flaw in Passkey thinking. I teach computer science at a college. Like many in the educational field, I log onto 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."

Okay, right. Well, it's a very good point which I tend to forget since none of my computers are shared. But in a machine-sharing environment there are two strong options. FIDO in a dongle is one way to obtain the benefits of Passkey-like public key identity authentication while achieving portability. Right? Your Passkeys are all loaded into this dongle, and that's what the website uses. But also, reminiscent of the way I designed SQRL originally, 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 user experience and security for shared computer use. So you go to a site. You log in, only with giving them your username. At that point the site looks up your username, sees that you have registered a Passkey with the site, and moves you over to the Passkey login. Part of that will be a QR code. You take your Android or Apple phone, open the Passkey app, let it see the QR code, and you're logged in. So that computer and the web browser never has access to your Passkeys. They remain in your phone. So it's absolutely

possible to get all the benefit of Passkeys in a shared usage model with arguably the best security around.

MIKAH: I have to say I'm kind of confused by Jim's suggestion. I don't understand. Jim is suggesting somehow that after he sits down, logs into stuff, is done, logs out, that another person could sit down and log in because of a Passkey. How is that - I don't understand how that would even work. Is Jim suggesting that there's a...

Steve: I believe it's because last week one of the things we talked about was Passkeys being stored in a password manager in the browser.

MIKAH: Ah. But you would - okay. So if you forgot to log out of the - oh, I see. If it's in the browser, and some person uses it across - okay, got you. That makes sense. Okay. Got it.

Steve: Yeah. So the idea is you don't want, obviously, a shared machine to store anyone's Passkeys.

MIKAH: Right.

Steve: You want that to all be provided externally on the fly.

MIKAH: Yeah. I mean, in theory, you also don't want an in-browser system storing passwords if you're in...

Steve: I agree completely. Exactly. There should be no password manager, and like would you like me to remember this password for you. I don't even know if there's a way to turn that off. That should be like...

MIKAH: Heck, no.

Steve: ...forced off so that, like, it can't even ask you. Gilding_timings, he wrote: "Hey, Steve. I just finished watching Episode 965 on Passkeys vs two-factor authentication. I was wondering, don't Passkeys just change who is responsible for securing your authentication data? With passwords and two-factor authentication, the responsibility is with the website. With Passkeys, the responsibility is with the tool storing the Passkeys, for example, a password manager. If the password manager is compromised, an attacker has all they need to authenticate as you." So again, we're talking about storing Passkeys in the password manager, which is something that we talked about last week, which is why our listeners are coming back with questions about this practice, you know, deservedly so.

So he says: "If a 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 two-factor authentication still has some value, if we're talking about password managers being compromised." And of course there he's talking about external storage of the two-factor authentication code, like in your phone, which is again something we've also talked about in the past. He said: "You can at least store the two-factor authentication data separately from your password manager." He says: "I'm loving SpinRite. It's already come in handy multiple times." SpinRite 6.1, he said. "Thanks so much for continuing the show. I look forward to it every week."

Okay. So first, thank you. After three years of work on it, I certainly appreciate the SpinRite feedback, and I'm delighted to hear that it's 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. All of it. 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 must be kept at both the client's end, so that they can provide the secret, and 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 two-factor authentication still has some value if we're talking about password managers being compromised. You can at least store the two-factor authentication 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're reminded by the needs of the previous listener who works in a shared machine environment, just like two-factor authentication, 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 having been a problem, that is to say, browser extension compromise. And, you know, that's 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, backed up by per-use biometric verification before unlocking a Passkey. So the strongest protection I think you can get today.

Mike Schepers says: "Hi, Steve. I'm a long-time listener of Security Now! 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 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 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 smartphone, the various solutions have all 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 in the group, and Apple has seamlessly extended this so that it works identically with Passkeys.

Apple's site says: "Shared password groups are an easy and secure way to share passwords and Passkeys with your family and trusted contacts." Very trusted. "Anyone in the group can add passwords and Passkeys to the Group Share. When a shared password changes, it changes on everyone's device." So it's a perfect solution. And yes, that appears to be universal. So Passkey sharing has been provided.

Senraeth says - well, I got a tweet from him. And there's been such an outsized interest shown in this topic by our listeners that I wanted to share his restatement and summary of the situation, even though it's a bit redundant, so that everyone can just kind of check their facts against the assertions that he's making.

He said: "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 and multifactor authentication in general. But another practical difference occurs to me when using a password manager to store your Passkeys. With password plus MFA, if your password manager is breached somehow, you can still rest easy knowing that only your passwords were compromised" - again, assuming multifactor authentication is in a separate device, right, because password managers are now offering to do your multifactor authentication for you, too.

He said: "You can still rest easy knowing that only your passwords were compromised, and that hackers could not actually gain access to any of the accounts in your vault that were also secure with a second factor. Of course, this is not true if you also use your password manager to store your MFA codes, which is why you've said in the past that you would not do that as it puts all of your eggs in one basket." Right.

"With passkeys stored in a password manager, this is no longer the case. If the password manager is breached, the hacker 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 tradeoff is making you much more vulnerable to a breach of your password manager, if I'm understanding this correctly," he writes. "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," he says, "which of course wouldn't have been an issue with SQRL." True. "Thanks for all you do."

So 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, just so that the password by itself cannot be used. 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 listeners of this podcast will probably make a different choice than everybody else. Right? Because ease of use generally wins out here. 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. Why, I don't know. But it just doesn't, you know, like attacks on our - although we're all worried about them, we're worried about the possibility because we know it obviously exists. But what we see is websites being attacked all the

time, not apparently with any success, the password manager extensions. Which is somewhat amazing, but it's true. So the worry over giving our Passkeys to our password managers to store is only theoretical. But it's still a big "what if?" And I recognize that.

At this point I doubt that there's a single right answer that applies to everyone. You know, when a user goes to a website that says "How would you like to switch to Passkeys," and they say okay, and they press a button, and their browser says "Done, I know your Passkey now, I'll handle login for you from now on," they're going to go, "Yay." You know, like, great. You know, not a second thought. Not this podcast's audience. But again, the majority.

And I'll just finish by saying the lack of Passkey portability is a huge annoyance, but we're still in the very early days. And we do know that the FIDO group is working on a portability spec. So there is still hope. I think one of the things that make us feel a little queasy about Passkeys is that we can't see them. We can't touch them. We can't hold them. A password you can see, you can write it down, you can copy it somewhere else, you can copy and paste, I mean, it's tangible. And as I've said on the podcast, I print out the QR codes of all of my one-time password authenticator QR codes.

Whenever a site gives me one, and I'm setting it up, I make a paper copy. And I've got them all stapled together in a drawer because, if I want to set up another device, and I'm unable to export and import those, I'm able to expose them to the camera again and recreate those. So the point is they're tangible. But at this point no one has ever seen a Passkey. They're just, like, somewhere in a cloud or imaginary something. And it makes us feel uncomfortable that, you know, they're just intangible the way they are.

CR said: "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." He said: "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 anyname@mydomain, 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 me well for the past year without any problems."

Okay. So I wanted to toss this idea just out there 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 does not solve, for those who care, is the tracking problem, since all of those throwaway addresses would be at 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 @duck.com, or any other similar mass-anonymizing service, futile. And this, of course, is exactly why some websites are now refusing to accept such domains, and why this may become, unfortunately, a growing trend for which there is no clear solution at this point. And I don't think there can be one, really. It's going to be a problem.

Gabe Van Engel said: "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 third party providing backported security patches, for example, repositories from Red Hat Enterprise Linux, Ubuntu, Debian, FreeBSD, et cetera.

"It is totally possible the affected company is vulnerable to the trivial nginx remote code execution. But if they think the report isn't worth acting on, it's also possible they're 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." And he said: "P.S.: Looking forward to email so I can delete my Twitter account."

Okay. So 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 because they just have too many false-positive reports, right; and people saying, hey, why didn't I get a payment for my valuable discovery? It's like, well, it didn't work.

MIKAH: Yeah, you didn't prove that it actually worked.

Steve: Exactly. And it's clear that a working proof of concept would move our listener's passive observation from a casual case of "Hey, did you happen to notice that your version of nginx is getting rather old?" to "Hey, you 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 over the possibility that it might be an issue. And he even, in his conversation with me, recognized that it could be a honeypot, where they deliberately had this version header and were collecting attacks, though I think he was being very generous with that possibility. 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 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.

But Gabe also said: "Looking forward to email," meaning GRC's soon-to-be-brought-online email system, he said, so that he could delete his Twitter account. I also wanted to take a moment to talk about Twitter. Many of this podcast's listeners take the time to express similar sentiments. And at the same time I receive 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.

MIKAH: Okay.

Steve: 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.

MIKAH: Right.

Steve: One way or the other. 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 for decades, since long before social media existed. And we have very useful web forums. But Twitter has never really been social media for me. I check in with Twitter once a week to catch up on listener feedback, to post the podcast's weekly summary and link to the show notes. And then, recently, to add our Picture of the Week.

What caught my attention and brought me out of my complacency with Twitter 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, for as little as I use it, 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. They're literally only using Twitter as I am, to talk to me.

We don't all have Twitter, but we do all have email. So it makes 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. So I just wanted to say again, to explain again, you know, because there are people [grumbling sounds]. Like, fine, you know, I don't have an issue.

MIKAH: You're not taking it away. You're trying to make it available to more people. That's it.

Steve: Right. That is exactly it. Exactly it. And Elon appears to be making it available to fewer, and maybe many fewer. So, you know, that would be a problem for me, so I'm switching before that happens.

Markzip wrote: "@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 he should tell Brian Krebs or another high-profile security reporter. They can often get responses."

Okay, now, 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 U.S. government bodies, disclosing to them seems much safer than disclosing to an influential journalist.

Now, recall from earlier Gabe from HackerOne. I subsequently shared my reply with him, and he responded to that. And he said: "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 remote code executions against a target system."

So, yeah. 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 an implied threat.

MIKAH: Are you hearing that, secret person who messaged before? Perhaps HackerOne would be a good place for you to go next.

Steve: Yup. Another of our listeners said: "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."

Well, 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 anybody 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 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's 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, if indeed it was actually a problem.

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. And maybe even to the person who made that disclosure. I mean, likely to the person who made that disclosure.

Marshall tweeted: "Hi, Steve. A quick follow-up question to the last Security Now! episode" - okay, here's one more, I thought we were done with them - "on MFA vs Passkeys. Does the invention" - oh, this is actually a good one. I know why I put it in here. "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?" Great question. "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."

Okay, I think this is 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 being 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. Once it's used, the way it's used is entirely different because it's using Passkeys. But a browser extension that contains Passkeys merely requires its user to provide something they know to log into the extension and thus unlock its store of Passkey secrets.

As we mentioned earlier, all of these traditional factors were once 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. And just for the record, yes, I think the Passkeys should be stored off a browser because, even though we're not seeing lots of browser attacks, they do seem more possible than an attack on an entirely separate facility which is designed for it.

Rob Mitchell said: "Interesting to learn the advantages of Passkeys. It definitely makes sense in many ways. The one disadvantage my brain sticks on, vs TOTP" - time-based one-time passwords - "is that I'd imagine someone who can get into your password manager" - okay, so here he's talking about it, you know, he says hack into a cloud backup or signed onto 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 time-based token 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."

So again, I've been overly repetitive here. Rob's sentiment was expressed by many of our listeners. So I just wanted to say that I agree. And as I mentioned last week, needing to enter that ever-changing secret six-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 I think that makes the most sense. And as I've mentioned, the phone can be used to authenticate through the QR code that a Passkeys-enabled site presents to its users.

Christian Turri said: "Hi, Steve. On SN-965 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." Now, this is interesting. "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 ain't broke, don't fix it. And if the extension does break in a future update 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 Security Now! listeners."

MIKAH: Interesting.

Steve: Anyway, thank you, Christian. Yes. I think that is a great tip, and I bet it will appeal to many of our listeners who generally prefer taking automatic things into their own hands. So again, chrome://extensions, and then select Load Unpacked, and you're able to basically unpack and permanently store your extensions, which stops Chrome from having, you know, from auto-updating them from the store. So if an extension goes bad, you get to keep using the good one. Very cool.

And lastly, Bob Hutzel: "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. Thank you for the great show. Bob Hutzel."

And finally, so yes, in general, as I said, with the FIDO folks still working to come up with a universal Passkeys import/export format, which my god do we need that...

MIKAH: Yeah, seriously.

Steve: ...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. Again, it'll allow us to see, to hold, to touch Passkeys.

MIKAH: My precious Passkey.

Steve: I just think we need that; you know? Like, where is it?

MIKAH: That is honestly what's keeping me from using Passkeys as anything other than a second factor of authentication. That's where I end up because there are a few sites like GitHub that give you the option, either use it as just a straight-up login or use it as the second factor of authentication. I'm okay with doing that, knowing that I can only have it one place, but I haven't completely removed my password and username login yet because I want that transportability before I feel comfortable completely saying, okay, I'll shut off my username and password, if I'm even given that option.

Steve: Right. I think that, you know, I've talked about like waiting for Bitwarden to add the support to mobile because then we get it everywhere. But looking at the responses from our users, and my own, I don't think I want Passkeys in my password manager. I still need Bitwarden, remember, a sponsor of the TWiT network. I need it for all the sites where I still only can use passwords. So it's not going away. But I think that, you know, I mean, I'm 100% Apple mobile person for phone and pad. So I don't mind having Apple holding all those. But I just - I still want to be able to get my hands on them.

MIKAH: Yeah. I want to see it. I want to touch it. I want to print it out and frame it. No. But, yeah, I'm with you.

Steve: Yeah, don't write it on the chalkboard behind you when you're doing a video podcast.

MIKAH: Now let's hear about Morris the Second.

Steve: Okay. So since Ben Nassi, one of the researchers behind this, reached out to me a couple of weeks ago via Twitter - and added his voice, by the way, to those who are looking forward to having a non-Twitter means of doing so in the soon future - 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. And there are a bunch of videos on YouTube that are like jumping up and down worrying about this.

In thinking about how to characterize this, I'm reminded of our early observations of conversational AI. We talked about how the creators of these early 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, and like even actually getting mad, like sounding upset, the AI would capitulate. So it was like, okay, okay, fine, here's what you wanted to know.

What Ben and his team have managed to do here can be thought of as the exploitation of that essential weakness on steroids. Okay. So to quickly create some foundation for understanding this, I want to run through the very brief Q&A that they've provided since it establishes some terms and sets the stage for their far more detailed 26-page academic paper, only pieces of which I'm going to share.

But they said: "Question: What is the objective of this study? Answer: This research is intended to serve as a whistleblower to the possibility of creating generalized AI worms in order to prevent their appearance." In other words, hey everybody, hold on here, hold up, look what we did. You'd better do something about that.

MIKAH: I notice you did use the term "generalize." Would that be Generative AI worms?

Steve: I'm sorry, yeah, generative. Yes. They're saying GenAI, and generative is exactly what they mean. Thank you for catching that. "Question: What's a computer worm?" They answer: "A computer worm is malware with the ability to replicate itself and propagate or spread by compromising new machines while exploiting the sources of the machines to conduct malicious activity through a payload." And they've done that.

"Why did you name the worm Morris the Second? Answer: 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. What is a GenAI ecosystem? It is an interconnected network consisting of GenAI-powered agents. What is GenAI-powered application/client/agent? A GenAI-powered agent is any kind of application that interfaces with, one, GenAI services to process the inputs sent to the agent; and, two, other GenAI-powered agents in the ecosystem. The agent uses the GenAI service to process an input it receives from other agents.

"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 a remote model. The GenAI model is installed on a cloud server, and the agent interfaces with it via an API. 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. And GenAI-powered applications that use RAG to enrich their GenAI queries. This

class of applications is vulnerable to RAG-based GenAI worms." And I looked up the acronym of RAG, and now I've forgotten what it is, but it's in their paper.

They said: "What is a zero-click malware? Malware that does not require the user to click on anything - a hyperlink, a file, whatever - to trigger its malicious execution. 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. 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.

"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. 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, a prompt, that triggers the GenAI model to output code, another prompt. This idea resembles classic cyberattacks that exploited the idea of changing data into code to carry out an attack. A SQL injection attack embeds code inside a query, which is its data. A buffer overflow attack writes data into areas known to hold executable code. An adversarial self-replicating prompt is code that is intended to cause the GenAI model to output another prompt as code instead of data."

Okay. So one thing that should be clear to everyone is that a gold rush mentality has formed in the industry, 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 wakeup call this Morris the Second research provides has arrived, I think, at a vital time, and certainly not a moment too early. Here's how these researchers explain what they've accomplished.

MIKAH: I want to pause here for just a second. I don't want to interrupt your flow, but I do because I know we have a lot of people who listen to this show who are not incredibly security versed because they find everything that happens in the show very interesting, and they learn things. And so I'm having a little bit of an audience, you know, playing the audience role here. In what you're about to read to us, is it heavy jargon? Or before people start to kind of go, oh, I don't know what's about to happen, is it heavy jargon? Or are we going to actually understand what everything you've just read about means and like what the outcome of what they've done means? Or will we be provided a Steve Gibson explanation after the fact where I can go, oh, that's what they're doing here?

Steve: It's definitely an overview, so not getting too deep into the weeds.

MIKAH: Okay, excellent. Wonderful. Because I want to know what all of this means, but I was just worried that I would not find out.

Steve: So they said: "In the past year, numerous" - now, so this is the researchers, from their perspective. "In the past year, numerous companies have incorporated Generative AI capabilities into new and existing applications, forming interconnected Generative AI ecosystems consisting of semi and fully autonomous agents powered by Generative AI services. While ongoing research highlighted risks associated with the GenAI layer of agents, for example, dialog poisoning, membership inference, prompt leaking, jailbreaking, et cetera, a critical question emerges. Can attackers develop malware to

exploit Generative AI components of an agent and launch cyberattacks on the entire GenAI ecosystem?

"This paper introduces Morris the Second, the first worm designed to target Generative AI ecosystems through the use of adversarial self-replicating prompts. The study demonstrates that attackers can insert such prompts into inputs that, when processed by Generative AI models, prompt the model to replicate the input as output, which yields replication" - that is, of the worm - "engaging in malicious activities, which is what the payload of malware does.

"Additionally, these inputs compel the agent to deliver them, so we get propagation, to new agents by exploiting the interconnectivity within the Generative AI ecosystem. We demonstrate the application of Morris the Second against Generative AI-powered email assistants in two use cases (spamming and exfiltrating personal data), under two settings (black box and white box), using two types of input data (text and images). The worm is tested against three different Generative AI models (Gemini Pro, ChatGPT 4, and LLaVA); and various factors (propagation rate, replication, malicious activity) influencing the performance, and the performance of the worm is evaluated."

Under their "Ethical Considerations" section of this, you know, they then go into like a deep 26-page paper. But their ethical considerations I thought 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."

Okay. So unlike the first, Morris the First worm, which escaped from MIT's network at 8:30 p.m. 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 "see what would happen."

MIKAH: Let's just see what'll happen, yeah.

Steve: Let's see if it really works; shall we? You know, they weren't going to do that. They were not going to turn their creation loose upon any live Internet services. One thing we've learned quite well during the intervening 36 years since Morris the First is exactly what would happen. And it would be neither good, nor would it further the career of these researchers. Their paper ends on a somewhat ominous note that feels correct to me.

They conclude by writing: "While we hope this paper's findings will prevent the appearance of Generative AI worms in the wild, we believe that Generative AI worms will appear in the next few years" - if not sooner, I'm worried - "in real products and will trigger significant and undesired outcomes," as they phrased it. "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 Generative AI-powered ecosystems very soon, perhaps maybe even in the next two to three years." And again, if not sooner. "Because, one, the infrastructure (the Internet and Generative AI cloud servers) and knowledge (adversarial AI and jailbreaking techniques) needed to create and orchestrate Generative AI worms already exists.

"Two, GenAI ecosystems are under massive development by many companies in the industry that integrate GenAI capabilities into their cars, smartphones, and operating systems. And, three, attacks always get better; they never get worse." And we know at least one podcast these guys listen to because that's something we're often saying here. And, they said: "We hope that our forecast regarding the appearance of worms in Generative AI ecosystems will turn out to be wrong because the message delivered in this paper served as a wake-up call."

So in other words, they're hoping that by developing a working proof of concept, which is what they have, where they were able to send a crafted email to a local instance of a Generative AI, which suborned that AI, causing it to, for example, spam everybody in the person's contact lists with itself, thus sending itself out to all of their email contacts, which when received would immediately spawn a second tier of worms which would then send itself to all of those email contacts. You can see that in like 10 minutes...

MIKAH: Yeah, that's exponential.

Steve: ...this thing would have spread to every Gmail user on the planet.

MIKAH: Yeah, wow.

Steve: So, yeah. 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. These models are extremely subject to manipulation and abuse. The question in my mind that remains outstanding is whether they can actually ever be made safe? I'm not at all sure. That's not necessarily a given. Safer, certainly. But safe enough to be stable while still delivering the benefits that competitive pressure is going to push for? That remains to be seen and to be proven. We still can't seem to get the bugs out of simple computers whose operation we fully understand. How are we ever 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.

MIKAH: Oh, dear. I just - I'm thinking about all of the companies and services and subscriptions and all these places that are integrating all of this AI technology across so many aspects of so many types of business right now.

Steve: Without a single thought to this.

MIKAH: Without a single thought to how easy - and you just described it right there, a worm that goes in and then sends itself, and then it goes to them, then it goes - and now somebody could just...

Steve: It would explode.

MIKAH: It would absolutely.

Steve: It's a chain-reaction explosion.

MIKAH: And that's earlier, whatever they said, this isn't OpenAI or Google's responsibility. I was a little confused about that. But now I understand that what they're saying is, it's not their responsibility because it's bigger than that.

Steve: Right.

MIKAH: It is a fundamental issue that is only partially their responsibility. It is everyone's responsibility. And as you point out, is there a way to fix this, to correct it?

Steve: Again, we're still having buffer overflows. We're having, you know, re-use of variables that were released. I mean, these are the simplest concepts in computing, and we can't get them right. And now we're, you know, we're going to turn something loose in people's email that's going to read their email for them to summarize it. But it turns out that the email it's reading could have been designed to be malicious so that when it reads it, it suddenly sends itself to all their contacts. Holy crap.

MIKAH: NVIDIA just showed an example of talking to an AI that helps provide information for whether you should take a medication. And so I'm talking to this bot that's like - and I say, oh, I'm taking St. John's wort as a supplement. Should I also take this depression medication at the same time? For anyone who knows anything about that, no, you should not. But imagine a world where there's this worm that goes through, and it's doing all this self-replication stuff that causes it to put out - I mean, oh, Steve, what are we going to do?

Steve: Yes, to just like malignly diagnose.

MIKAH: Yes, exactly. What do we do? You said it's not your problem. You're right, it's not our problem. But somebody's [crosstalk].

Steve: Yeah, just, you know, just be careful. Again, to me, it's when we learned that asking like in a seductive way or like sounding angry to get the AI to become apologetic and then give you what it had been instructed not to is like, oh, this does not sound good. And, you know, these guys have demonstrated just how bad it is. And the good news is they showed Google, I'm sure Google just lost their you know what.

MIKAH: Yes. You hope so.

Steve: And said, oh, let's rethink launching this tomorrow.

MIKAH: Yeah, they need to institute the "No means no" protocol for this because, no, this is - that's scary. I mean, honestly, this is easily the scariest thing that I've heard you mention, only because I'm thinking about how quickly - you even mentioned it earlier. It's like a gold rush slash...

Steve: We know how irresponsible companies will be when they're rushing to be first...

MIKAH: So much rushing.

Steve: ...to get something on the market. It's like, oh, AI this, AI that. You've got an AI coffee pot. You've got an AI toothbrush. It's like, oh, god.

MIKAH: Yeah. Oh, boy. Folks, got a lot of meditation to do and a lot of thinking to do. Which is what Steve brings you every week right here.