Passkeys vs 2FA

Description: What happened with CERT? What headache has VMware been dealing with? What's Microsoft's latest vulnerability disclosure strategy? What's China's "Document 79," and is it any surprise? What long-awaited new feature is in version 7.0 of Signal? How is Meta coping with the EU's new Digital Marketing Act that just went into effect? What's the latest on that devastating ransomware attack on Change Healthcare? And after addressing some interesting feedback from our listeners, I want to clarify something about Passkeys that is not at all obvious.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-965.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-965-lq.mp3

SHOW TEASE: Hey, I'm Mikah Sargent, subbing in for Leo Laporte. Coming up on Security Now!, first we follow-up on what happened with CERT. Yes, the listener who talked about a huge serious security flaw in the website of a major enterprise has some follow-up on speaking to the vulnerability analysis team at CERT. Then we talk about what VMware is dealing with; what Microsoft is choosing to do when it comes to vulnerability disclosure. Here's a hint. They're kind of waiting until the end of the week to tell people what's going on. Plus China ditching America, at least in terms of its technology. And easily my favorite part of the show, Steve Gibson explains why Passkeys are "far more secure" than any super-strong password plus any second factor. All of that coming up on Security Now!.

MIKAH SARGENT: This is Security Now! Episode 965 with Steve Gibson and Mikah Sargent, recorded Tuesday, March 12th, 2024: Passkeys vs 2FA.

Hello and welcome to Security Now!. As you can probably tell by listening, this is not Leo Laporte. No, it is Mikah Sargent who is filling in for Leo Laporte while he is on vacation. But across from me, at least in the way of the Internet, is the tried-and-true Steve Gibson, who is here as he always is. Steve, thank you for always being here.

Steve Gibson: And you know, I've noticed that Leo actually, like you are right now, actually looks at the screen in order to, like, make it look like you're looking at me. So I appreciate that. And sometimes when Leo's heading off from the main, well, the studio that you're in at the end of MacBreak Weekly where he is, heading to his office, I'll deliberately look this way.

MIKAH: Just to watch him go.

Steve: Oh, see you, Leo. That's right.

MIKAH: Oh, I like that. I like that.

Steve: Anyway.

MIKAH: It's good to see you.

Steve: Likewise, and thank you for standing in for Leo. He's on a beach somewhere. He says he will come back with a tan. So that's good. We'll hope no skin cancer.

MIKAH: Yes, exactly, a safe tan.

Steve: At our age, one considers the downside of being too brown. So anyway, I had said, had suspected, or planned, to title today's podcast "Morris II" after the sort of, well, it was named that way by the guys who created this thing. It's a way they found of abusing GenAI to create an Internet worm. And of course the Morris worm was the very first worm on the Internet and is famous for that reason.

Anyway, something came up, as sometimes happens, and so we'll be talking about Morris II next week unless something else comes up that pushes it a little bit further downstream. And this is the result of a listener's question, which caught me a little bit by surprise because I understand this stuff because I live in it, this being across the network authentication. And I thought, okay, I realized that his question was a really good one, and it had a really good answer. And so first I put it as the first of our listener feedback questions. But as I began to evolve the answer, I thought, no, no, no, okay, this just has to be something we really focus on because it's an important point.

So today's podcast number 965 for March 12th is titled "Passkeys vs 2FA." And I would bet that even for people who think they really understand all this, there may be some nuances that have been missed. So I think it's going to be a great and interesting and useful podcast for everyone. But of course that's at the end.

We're going to start with whatever happened with that guy who complained to CERT about a vulnerability he found in a major website. What headache has VMware been dealing with just the last few days? What's Microsoft's latest vulnerability disclosure strategy? And why does it, well, suck? What's China's "Document 79" all about, and is it any surprise? What long-awaited new feature is in version 7.0 of Signal, currently in beta, but coming out soon? How is Meta coping with the EU's new Digital Marketing Act that just went into effect and requires its messaging platforms to be interoperable with others? Whoops.

Also, what's the latest on that devastating ransomware attack on Change Healthcare? Many of our listeners have said, hey, Steve, did I miss you talking about that, or haven't you? I haven't, so it's time because now we have a lot of information about it. And, you know, as I said, after addressing a lot of interesting feedback from our listeners which we're also going to make time for, we're going to talk about a few things about Passkeys that are actually not at all obvious. And of course we always have a great Picture of the Week. So I think overall a great podcast for our listeners.

MIKAH: It is time for the Picture of the Week.

Steve: Okay. So as I said, this relates to one of our more famous earlier pictures of the week where there was a big AC generator sitting on a factory floor somewhere, and next to it was a pail, a large pail of dirt. And a rod was stuck into the dirt to which was attached a big ground wire. And really, it just demonstrates a complete lack of understanding of the nature of grounding something. I mean, it's not like, I mean, the dirt in this first picture, the earlier picture, was in a plastic pail. So it wasn't connected to the earth, which is really what we're wanting for a ground to be.

Anyway, here we have an updated version of the same thing. It looks like some electrical wiring in progress. It's not all finished yet. The construction, whereas most typical electrical high-power wiring are in steel conduit and steel boxes, this is all white plastic. So that suggests that the need for grounding is even more imperative than it would be if this was all in steel, which is probably grounded somewhere anyway, but not here. So for reasons that are unclear, a green wire is coming out of this and going into a plastic bag which has been suspended from a plastic piece of conduit. And this bag, sure enough, it's got about an inch worth of brown dirt at the bottom of it, and the green wire has been certain to, like, get all the way down to the bottom of the bag and bury itself in the dirt. And I gave this thing, I gave this picture the caption "It's certainly a good thing that bag of dirt was labeled with a 'ground' symbol, or the dirt's purpose might have been unclear."

MIKAH: Yeah.

Steve: We see, like, the equivalent of what looks like a brown Post-it note, but somebody went out of their way to draw a very pretty ground symbol.

MIKAH: It's very good, yeah.

Steve: I mean, because electricity and electronics was my first passion, I've drawn my share of ground symbols in my life, and I would have to say this is right up there with the best of them. So yes, we have a - again, it's like, who...

MIKAH: Who, what, when, where, and why?

Steve: Would you trust the wiring of an electrician who did this? I mean, I don't know. It just, it really - we have some pictures that really do beg the question, what happened here? Anyway...

MIKAH: It's just question marks all around. It's fantastic, but awful. Oh, dear.

Steve: Okay. So recall from last week our listener who discovered a serious flaw in the website of a major enterprise whose site entertains millions, hundreds of millions of users. And in fact since then as a consequence of what he forwarded to me, and which I redacted, I now know who that site is and what that enterprise is. And wow. So the problem was that these people were using a very outdated version of the nginx web server which contained a well-known critical remote code execution vulnerability for which working Python proof-of-concept code was readily available.

Which all means that any bad guy who went to any of this company's various websites, and there are several, who then looked at the headers of the response from the website, which would identify the server as nginx and version, could then, as I did, and as our listener did, google that version of nginx and see, whoa, there are some problems with that version from four years ago. And that person would be able to find the proof-of-concept code, as I did, and use it, which neither our listener nor I did, in order to get inside, crawl inside the web server.

Anyway, serious problem with this major enterprise and their website. And I'll just say that this enterprise is - it's not like the website happens to be a side-effect feature of this enterprise. The nature of the enterprise is such that it is the website.

MIKAH: Oh.

Steve: I mean, like, its entire purpose and existence on the planet is this website and others that it also owns.

MIKAH: So to be clear, it's not as if the website is just a place where you go to learn about its features. The service is the site that you're going to. It's not just a pretty little site.

Steve: Yes. It is not Google, but it would be as if - it would be like Google having a problem of this magnitude with its servers. I mean, you know, its purpose is its site, essentially. And so it's a big deal. Now, our listener, as I described last week, gave this company 120 days to, like, deal with it. He contacted them. They replied, "Okay, thank you for notifying us. We'll look into this and get back to you." Oh, and they said, "And it may take four months for that to happen." So he waited until 118 days and then said, uh, knock knock.

MIKAH: You've got two days here.

Steve: What's going on? And they said, oh, oh, right. Well, let's give it a couple more days. So they came back and said, well, we've asked our security people, and they're not really thinking that's a big problem. How about this? We'll give you 350 bucks to just shut up.

MIKAH: What?

Steve: And go away.

MIKAH: What?

Steve: So at that point he tweeted me, and he said, "Steve, what do you think about this?" And I said, you know, you've done everything you can. They're not being very cooperative. Why don't you send a report to CISA? Which is, you know, the U.S.'s front facility for dealing with this kind of stuff. So he thought, okay, that's a good idea, so submitted a report along with his entire email chain, his whole dialogue with these guys, to CISA, which forwards it to CERT.

Okay. Well, he sent me a tweet updating me on what was going on, and I have it in the show notes, after I removed the identification of the site that we're all talking about here. And this comes back from CERT saying: "Greetings. Thank you for your vulnerability report submission. After review, we've decided not to handle the case, for two reasons. First, we typically avoid handling or publishing reports of vulnerabilities in live websites.

"Second, since the vendor is cooperating and communicating with you, there's very little additional action that we can contribute." What? Anyway, "We recommend working directly with the affected vendor before proceeding with public disclosure." Well, okay. He's never going to disclosure this publically. This would be a disaster. "Feel free to contact us again if you run into problems with your coordination effort, or if there is some other need for our involvement. Regards." And in like...

MIKAH: Giant.

Steve: 50-point type, yeah, "Vulnerability Analysis Team, at the CERT Coordination Center at kb.cert.org." Okay, well, that's disappointing. You know, they say: "We typically avoid handling vulnerabilities in live websites." So, what? They limit their handling of vulnerabilities to dead websites? You know, this doesn't inspire confidence.

MIKAH: It feels like they didn't read the evidence that was provided. This feels like when you go and you talk to any tech support person.

Steve: Yes.

MIKAH: And they immediately jump to assumptions, and they don't hear you out first.

Steve: Yes.

MIKAH: And then they're providing advice that you're going, but did you not just hear what I literally just - I don't like this. Sorry. Go ahead.

Steve: No. No, and you're exactly right. Our listener's correspondence, which he had forwarded to CERT with his report, made clear that the company had no intention of doing anything further. So CERT was just passing the buck back to someone who had already demonstrated that he had no leverage. I mean, and that was why I recommended these guys. I figured if Uncle Sam knocks on the door and says, hey, you've got a little problem with your web server, and it would be bad if you got compromised, that then they might listen, because they're not listening to Joe.

So anyway, in my reply to him just now, since he had never disclosed anything publicly, and I know he never would, he's not a bad guy, I suggested that, since he had done all he could reasonably do - and again, never said anything publicly - he should take the money as compensation for his trouble and leave whatever happens up to fate. You know, if the company does eventually get bitten, it will only be their fault. He warned them. He didn't, I mean, he even reminded them after four months. And they said, oh, yeah, you. Hmm.

MIKAH: Not you again. We thought you'd just disappear. Why?

Steve: So, yeah, anyway.

MIKAH: Can I ask you something? I would like to know when is it reasonable and, as you see it, reasonable and the right thing to do to disclose things publicly? Not in this situation in particular. I just mean - because I thought that that was something that some security researchers end up doing is, if the company continues not to behave, or it continues to not work with them, isn't a public disclosure part of the bargaining power that the security researcher has?

Steve: Yeah, famously. Like Google, for example. You know, they find a problem, they notify the vendor, and they say, "We have started a 90-day clock, and you need to fix this sucker." So if you don't, we're going, well, or whether or not you do, we're going to go public with this in 90 days. So take it seriously. Because unfortunately, this policy is the result of experiences just like this one, where companies are like, eh, you know, we don't think it's that big a problem. Okay, great. We're going to let everybody know about it if it's not that big a problem. Whoa, whoa, whoa, whoa. Hold on. Hold on. Wait a minute. So this guy's not Google. He can't do that. And of course Google also has lord knows what kind of a bank of attorneys they have that are able to...

MIKAH: To protect themselves, got it, yeah.

Steve: ...protect themselves. Unlike this guy, who the last thing he wants is to get stomped on by a massive enterprise that just turns one of their attorneys loose and says let's go make this guy's life miserable.

MIKAH: So that's the danger there. That makes sense.

Steve: Yeah. So first of all, I mean, there have been instances where even Google has not followed their own policy when what they have found has been so egregious that, like, they can't in good conscience let this be known publicly, even though they really want to, because it would just collapse the world.

MIKAH: Right.

Steve: So, you know, you can lead the company to water. You can say, "Please fix this, won't you? We all want you to. The world will be a safer place." But ultimately, if they say no, it's like, okay, well, you called our bluff because we're not going to tell everybody because that would be really bad. So good luck. And at least we're going to protect ourselves from you. In fact, this guy was worried that the information this company has about him, because he's a subscriber or a member or whatever it is, that that would be at risk. And he asked them in his correspondence, like, what are you doing about my privacy concerns because you've got an insecure website, and you know a lot about me. And they never ever said anything about that. So again, you know, I guess maybe he could contact Google.

MIKAH: That would be great.

Steve: That might be an idea, actually, is contact their threat people and say, hey, what do you think about this? If you think it's bad, maybe you ought to give them a 90-day countdown and get this thing really fixed.

MIKAH: Yeah, that is...

Steve: Anyway, it was a little - it was a disappointing response, though, from our own government because this is not a company you want to have attacked. And in fact, speaking of which, we'll be talking about Change Healthcare here in a while. Talk about a company you don't want to have attacked. Oh, my lord. One in three of the nation's health records were compromised in this.

MIKAH: Love that. Love that. That's so great.

Steve: What could possibly go wrong? Speaking of which, a week ago, on March 5th, Broadcom, which is VMware's parent company now after they bought them, issued a security advisory that happens to be VMSA-2024-0006. And it's encouraging that it's got a one-digit number on it. You know, 0006, that's nice because now, you know, these days the CVEs have had to expand to six digits. Well, no, actually five. But they used to be four. Just because there are so many problems happening.

Okay. So this security advisory addresses vulnerabilities that have been discovered in VMware ESXi, VMware Workstation Pro and Player, and Fusion. In other words, pretty much everything. If it's got VMware in its name, we've got a problem. This is because of a problem in the ubiquitous USB virtualization drivers, which have been found to contain four critical flaws. I didn't dig in to find out who found them, whether they found them, somebody else found them and told them, or what. But they are so bad that VMware has even issued patches for previous out-of-service-life releases of anything with VMware in its name. An attacker who has privileged access in a guest OS virtual machine, so a root or admin, may exploit these vulnerabilities to break out of the virtualization sandbox, which of course is the whole point of virtualization, to access the underlying VMware hypervisor.

Patching VMware of any kind immediately is the optimal solution. But due to the location of the problem, which is just a specific driver, if anything prevented that from being done, and if your environment might be at risk because it includes untrusted VM users, the removal of VMware's USB controllers from those virtual machines will prevent exploitation until the patches can be applied. VMware said that the prospect of a hypervisor escape warranted an immediate response under the company's, what they call their, for some reason, IT Infrastructure Library, or ITIL. They said: "In ITIL terms, this

situation qualifies as an emergency change, necessitating prompt action from your organization."

So both the UHCI and XHCI USB controller drivers contain exploitable use-after-free vulnerabilities, each having a maximum severity rating of 9.3 for Workstation and Fusion and a base score of 8.4 for ESXi. There's also an information disclosure vulnerability in the UHCI USB controller, with a CVSS of 7.1. Somebody who has admin access to a virtual machine could exploit it, that is, this other vulnerability, to leak memory from the vmx process. Anyway, I've included a link in the show notes to VMware's FAQ about this for anyone who might be affected and interested in more details.

Basically, you know, if you've got VMware in your world, where somebody running in a virtual machine might be hostile, that's to be taken seriously. If you're just a person at home with VMware because you like to run Win7 in a VM or whatever, then this is not to worry about. Update it, you know, when you get around to it. But anyway, VMware is on alert because of course it is in use in many cloud infrastructures where, you know, you don't know who your tenants are. So, important to get that fixed.

So unfortunately, Microsoft's networks, and apparently their customers, remain under attack to this day by that Russian-backed Midnight Blizzard group. And remember that they were originally named Nobelium, and Microsoft said, oh, let's call it Midnight Blizzard because that sounds scarier, I guess. This is the group that successfully infiltrated some of Microsoft's top corporate executives' email by attacking the security of a system kind of that had been forgotten, it was in the back room somewhere, and it had not been updated to the latest state-of-the-art security and authentication standards. And so, you know, they said, whoops, we forgot to fix that.

So the sad thing here, the saddest thing of all, I think, is that Microsoft has apparently, believe it or not, now taken to dropping their public relations bombs late on Fridays, as happened again this past Friday, and not for the first time. This is now their pattern. The Risky Business security newsletter explained things by writing this. They said: "Microsoft says that Russian state-sponsored hackers successfully gained access to some of its internal systems and source code [whoops] repositories. The intrusions are the latest part of a security breach that began in November of last year and which Microsoft first disclosed in mid-January.

"Initially, the company said hackers breached corporate email servers and stole inboxes from the company's senior leadership, legal, and cybersecurity teams. In an update on the same incident posted late Friday afternoon - as is the practice of every respectable corporation..."

MIKAH: That's sarcasm, for those who are...

Steve: That's sarcasm, by the way. "Microsoft says it found new evidence over the past weeks that the Russian hackers were now weaponizing the stolen information."

MIKAH: Great. Love that.

Steve: Uh-huh. "The Redmond-based giant initially attributed the attack to Midnight Blizzard, a group also known as Nobelium" - that's right, but before Microsoft gave them a scarier name - "one of the cyber units inside Russia's Foreign Intelligence Service. Microsoft says that since exposing the intrusion, Midnight Blizzard has increased its activity against its systems." Well, and how. "Per the company's blog post, Midnight Blizzard password sprays increased in February by a factor of 10, compared to the 'already large volume it saw in January of 2024.'

"Furthermore," Risky Business notes, "if we read between the lines, the group is now also targeting Microsoft customers. While Microsoft's legalese makes it clear that no customer-facing systems were compromised, the company weaselly confirmed that customers have been 'targeted.'" Okay, what does that mean? "Emails stolen by the Russian group also contained infrastructure secrets, you know, secret tokens, passwords, API keys, et cetera, which Midnight Blizzard has been attempting to use."

So it's not good that Russians managed to compromise Microsoft executives' email in the first place, and that the information they stole is now being leveraged to facilitate additional intrusions, including apparently the exfiltration of Microsoft's source code repositories. That's not good for anyone. But as we know, I always draw a clear distinction between mistakes and policies. Mistakes happen. Certainly a big one happened here. But the new policy that caught my eye was that, unfortunately, we are now seeing a pattern develop of late Friday afternoon disclosures of information which Microsoft hopes will be picked up and published when fewer people are paying attention. So that's...

MIKAH: So, yeah. Having worked in journalism for ages, that is, well, ages. For a long time.

Steve: Yes, how old are you exactly? How many ages ago?

MIKAH: One age, at least.

Steve: Okay. An age.

MIKAH: But, yes, it is - that's something that we always pay attention to. When is that coming out? And what is the messaging behind there? And the fact that they're waiting until Friday afternoon is a clear signal that they don't want people to see it. And that is wrong when it comes to security disclosures. And luckily you've got your eye on it, and that means we can get to it on Tuesday. You can let it go on Friday, Microsoft, but we'll pick it up on Tuesday.

Steve: Well, yes. And it's also relevant that this was not like any kind of an emergency announcement that had to go out on Friday afternoon. I mean, this was just, like, well, we're updating you on the status, but we'd rather you didn't receive this.

MIKAH: Yeah, exactly.

Steve: So here it is. Right. Okay. So the official name for Beijing's secret directive is "Document 79," but it is informally known as the "Delete A" order where "A" is understood to stand for America. Yes, China wants to delete America. The secret directive was issued several years ago, back in September of 2022. It's designed to remove American and other non-Chinese hardware - although apparently since we get the "A," we figure prominently - and so to remove American and other non-Chinese hardware and software from its critical sectors. The directive mandates that state-owned companies replace all U.S. and other non-Chinese equipment and software in their IT systems by 2027. So you've got three years left, folks. And that was a five-year order that began two years ago. So far the affected companies include Cisco, IBM, Dell, Microsoft, and Oracle. And my reaction to this is, yeah, who can blame them?

MIKAH: Yeah. [Crosstalk].

Steve: You know? Right. With tensions on the rise between the U.S. and China, this only makes sense. The U.S. has done the same thing for the manufacturers of Chinese-made security cameras within, like, sensitive areas like military bases and protected corridors

of power. And we know that Russia is working to remove American technological influence from inside its borders. The problem for all the parties involved is that today's systems are so complex that trojan capabilities are readily hidden and can be made impossible to find. No one doubts the influence that Chinese intelligence services are able to exert over the design of Chinese equipment. And there has long been some question about just how much influence U.S. intelligence services might have over the installation of such backdoors in proprietary software. Years and years ago there was a key inside of Windows that purely by coincidence had the initials NSA. And it was like, [gasp], oh, my god. You know, it's like, folks, if it was an NSA key, they would have changed the name.

MIKAH: Yeah. It would not be that obvious.

Steve: But it was like, oh, my goodness, I mean, just everyone was all atwitter, if you'll pardon the choice of words, over that the NSA had some secret key buried in Windows. No. But which is not to say they don't. Like, why wouldn't they? But they wouldn't leave the name the same. So anyway, to my way of thinking, I just wish we could all get along. You know, it's unfortunate to be seeing this pulling apart, you know, and this rising global tension because ultimately it is less economically efficient for us all to have to, like, retrench, and like we have to now be making chips in the U.S. when China was so good at it. And, like, what's the problem? Well, you know, it could be a problem. So ultimately it's hardly surprising, but I just really regard it as unfortunate.

Okay. And I'm keeping my eye on the time, but this is going to be a long podcast because we have a lot of good stuff to get to. So I'm going to do one more before we take our next break.

MIKAH: Sounds good.

Steve: Signal's reliance - Signal, you know, the messaging app, like the best one there is that is cross-platform. I would argue that iMessage is good if you're in iOS world, but Signal if you need other stuff. Their reliance upon physical phone numbers for identifying the communicating parties has been a longstanding annoyance, as well as a concern for privacy and security researchers who have long asked the company to switch from phone numbers to usernames in order to protect users' identities. The just-announced version 7 of Signal will add the creation of temporary username aliases.

So here's how Signal explains it in their announcement. They said: "Signal's mission and sole focus is private communication. For years, Signal has kept your messages private, your profile information like your name and profile photo private, your contacts private, and your groups private" - notice the word "private" figuring prominently in that single sentence - "among much else. Now we're taking that one step further by making your phone number on Signal more private.

"Here's how: If you use Signal, your phone number will no longer be visible to everyone you chat with by default." And props to them for changing this. Many organizations will add a feature, but they're afraid to change the behavior. It's like, oh, well, you know, we don't want to, like, freak everyone out. So we've got it now, but we're going to have it turned off, and you've got to turn it on if you want it. No. Signal says those phone numbers are going to disappear. So get over it.

They said: "People who already have your number saved in their phone's contacts will still see your phone number since they already know it. If you don't want to hand out your phone number to chat with someone on Signal, you can now, as of 7" - which is in beta so it'll be available within a few weeks, they said - "you'll be able to create a unique username that you can use instead." They said: "You'll still need a phone number to sign up for Signal." And they said: "Note that a username is not the profile name that's displayed in chats. It's not a permanent handle." That's why I referred to it earlier as an

alias. And they said: "And not visible to the people you are chatting with in Signal. A username is simply a way to initiate contact on Signal without sharing your phone number.

"If you don't want people to be able to find you by searching for your phone number on Signal, you can now enable a new, optional privacy setting. This means that unless people have your exact unique username, they will not be able to initiate a conversation, or even know that you have a Signal account - even if they do have your phone number.

"These options are in beta," they said, "and will be rolling out to everyone in the coming weeks. And once these features reach everyone, both you and the people you're chatting with on Signal will need to be using the most updated version of the app to take advantage of them. Also, all of this is optional. While we changed the default to hide your phone number from people who don't already have it saved in their phone's contacts, you can change this setting. You're not required to create a username, and you have full control over whether you want people to be able to find you by your phone number or not. Whatever choices work for you and your friends, you'll still be able to communicate with your connections in Signal, past and present."

So that seems all good. They've changed to hiding phone numbers by default, and they've added an optional textual username as a phone number alias. Signal knows the phone number behind the alias, but users do not. I think that's pretty slick. Their blog posting about this contains a great deal more information. Basically it's a complete user guide to how to work with this coming feature. I've put it in this week's show notes since it might be of interest to anyone who uses Signal often. So you can click the link in the show notes, and you'll be taken to their page. It's phone-number-privacy-usernames. So you can also probably google that, if you want to just jump right to it at Signal's site. So anyway, nice improvement coming from Signal.

MIKAH: VV in the Discord says: "I set it up last week, and I love it. This is such welcome news." So we've already got some folks who have enabled these, you know, the new username feature. Or as you pointed out or call it, an alias I think is a good way of putting it, given the - and it's clear that, you know, when you can tell, I think of when I used to post on X, formerly known as Twitter, in the past, and I would have a tweet, and I'd have about three things underneath it that are sort of responses to how all of the pedantic people on Twitter will probably take my post and say, "You didn't say this," or "You forgot to say that."

This feels like that. They're answering every possible question that folks might have. And as you said, there's a full user guide. And I think that that's smart given their user base; right? There are going to be lots of concerns about any change taking place and what it does, what it doesn't do. So good on them, or good on it, I guess, for figuring out what needs to be said.

Steve: Yeah. Basically they've created a mapping on top of the phone numbers, where, you know, a little dictionary where a user can assign themselves an alias and then say to somebody, hey, contact me on Signal at this username. You know. And it's going to be squirrelybumblebee or something, which is not their phone number, but it doesn't need to be. It just needs to be something else. And that allows the - so beneath the covers, Signal knows when somebody says, hey, I want to connect to squirrelybumblebee, that that actually means this phone number. But the user making the connection never sees that.

MIKAH: Yeah. As you said, very slick. I think that's a right way to do it.

Steve: Okay. So last Thursday the 7th, the European Union's DMA, the Digital Markets Act, went into force. Among many important features, it requires interoperability among

specified, probably large, instant messaging systems. But here's what Meta recently explained about their plans. I'll give everybody a quick hit. It amounts to, if you want to talk to us, you've got to use the Signal protocol. But we'll meet you halfway by making it possible for you to connect with our previously closed servers.

So Meta wrote: "To comply with a new EU law, the Digital Markets Act, which comes into force on March 7th, we've made major changes to WhatsApp and Messenger to enable interoperability with third-party messaging services. We're sharing how we enabled third-party interoperability" - which they shortened to "interop" - "while maintaining end-to-end encryption and other privacy guarantees in our services as far as possible.

"On March 7th, a new EU law, the Digital Markets Act, comes into force. One of its requirements is that designated messaging services must let third-party messaging services become interoperable, provided the third party meets a series of eligibility, including technical and security, requirements." And that's of course the key because, if not, Meta is able to say, uh, no no no, not so fast. "This allows," they wrote, "users of third-party providers who choose to enable interoperability to send and receive messages with opted-in users of either Messenger or WhatsApp, both designated by the European Commission as being required to independently provide interoperability to third-party messaging services.

"For nearly two years our team has been working with the European Commission to implement interop in a way that meets the requirements of the law and maximizes the security, privacy and safety of its users. Interoperability is a technical challenge, even when focused on the basic functionalities as required by the DMA. In year one, the requirement is for 1:1 text messaging between individual users and the sharing of images, voice messages, videos, and other attached files between individual end users. In the future, requirements expand to group functionality and calling.

"To interoperate, third-party providers will sign an agreement with Messenger and/or WhatsApp, and we'll work together to enable interoperability. Today we're publishing the WhatsApp Reference Offer for third-party providers which will outline what will be required to interoperate with the service. The Reference Offer for Messenger will follow in due course. While Meta must be ready to enable interoperability with other services within three months of receiving a request, it may take longer before the functionality is ready for public use. We wanted to take this opportunity to set out the technical infrastructure and thinking that sits behind our interop solution.

"Our approach to compliance with the DMA is centered around preserving privacy and security for users as far as is possible. The DMA quite rightly makes it a legal requirement that we should not weaken security provided to Meta's own users. The approach we've taken in terms of implementing interoperability is the best way of meeting DMA requirements, whilst also creating a viable approach for third-party providers interested in becoming interoperable with Meta and maximizing user security and privacy.

"First, we need to protect the underlying security that keeps communication on Meta end-to-end encrypted messaging apps secure, the encryption protocol. WhatsApp and Messenger both use the tried and tested Signal Protocol as a foundational piece for their encryption. Messenger is still rolling out end-to-end encryption by default for personal communication, but on WhatsApp this default has been the case since 2016. In both cases, we are using the Signal Protocol as the foundation for these end-to-end encrypted communications, as it represents the current gold standard for end-to-end encrypted chats.

"In order to maximize user security, we would prefer third-party providers to use the Signal Protocol. Since this has to work for everyone, however, we will allow third-party

providers to use a compatible protocol if they're able to demonstrate it offers the same security guarantees as Signal." And I'll just interrupt here to say that will be a high bar.

MIKAH: Yeah. [Crosstalk]. That's it.

Steve: And Signal is open source. It's an open protocol. Why in the world would anybody roll their own from scratch and say, oh, well, you know, well, like Telegram, that's got a bizarro protocol that nobody has ever understood, and no one has ever tried to prove was secure because it's just garbage. You know, I mean, it scrambles stuff. So maybe it's secure. But it's nice to have proofs, and these days we can get proofs. Except not for random stuff somebody just made up.

They said: "To send messages, the third-party providers have to construct message protobuf structures, which are then encrypted using the Signal Protocol, and then packaged into message stanzas in XML," you know, the eXtensible Markup Language. "Meta servers push messages to connected clients over a persistent connection. Third-party servers are responsible for hosting any media files their client applications send to Meta clients, such as image or video files. After receiving a media message, Meta clients will subsequently download the encrypted media from the third-party messaging servers using a Meta proxy service.

"It's important to note that the end-to-end encryption promise Meta provides to users of our messaging services requires us to control both the sending and receiving clients. This allows us to ensure that only the sender and the intended recipient can see what's been sent, and that no one can listen to your conversation without both parties knowing." Now, just to decrypt that corporate speak, what they're saying is, if we don't control the sending and receiving clients, which is to say if we are going out of our own - what am I looking for? Not out of band, out of - there's a current term. It was a well-known...

MIKAH: Out of pocket? Out of environment? Out of...

Steve: Yeah, out of environment. Then we can't make any representations about what happens to the message that you send out of our services, or where those messages come from into our services. So that's sort of them just sort of, you know, stepping back, saying only if you stay within Messenger and WhatsApp are we willing and can we take responsibility over the messages that are being transacted. So, you know, beware.

Anyway, they finish, saying: "While we've built a secure foundation for interop that uses the Signal Protocol encryption to protect messages in transit, without ownership of both clients' endpoints we cannot guarantee what a third-party provider does with sent or received messages, and we therefore cannot make the same promise." So again, they just said what I just said. I didn't realize that was coming. So anyway.

MIKAH: With those last two paragraphs, are they talking to - do you think they're talking to the potential services that are trying to connect and explaining their thinking here? Or is this Meta not being absolutely certain how the EU will take this choice to use Signal and kind of push Signal, and so they're almost trying to explain to them, look, we know that it would be easier if we would just open it up to all of them. But, you know, they're sort of going through and explaining why Signal is the best.

I guess what I'm asking is, it kind of runs contrary to what the company said early on about having worked with the EU for so long on figuring this out, if what they're trying to do there is like, come on, EU, don't punish us further, we promise this works. Do you think this is more for those other third-party messaging platforms to go, look, we know that you might want to use something else, but it makes sense why we're doing this and why we have to have control of the sending and receiving.

Steve: So my ignorance is showing here. Can non-Facebook users use WhatsApp? I know Messenger is Facebook only.

MIKAH: That is a really good - I don't think you need a Facebook account to have WhatsApp. But that's a good question.

Steve: I think that's correct. So my feeling is, you know, yes, other Messengers might want to hook up. But, you know, even if you didn't have WhatsApp, the most obvious messaging system to connect would be Signal itself; right? Why wouldn't Signal, that invented the Signal protocol, say hey, yeah, we'll - that sounds great. We'd like to be able to allow Facebook Messenger users and WhatsApp to interact with Signal users. So, you know, Signal themselves would be the obvious interop choice. But again, Telegram's going to have a hard time. I mean, there are lots of other, you know, when I was talking last week about the various levels, Level 0, 1, 2, and 3, that Apple has assigned to messaging, there were some down at Level 0 where encryption was optional that I'd never heard of, like QQ. What's that?

MIKAH: Yeah, don't know.

Steve: But that's apparently some messaging app somewhere. So again, any other company is going to have a heavy lift. And basically I think what this means is, you know, we're using Signal at Meta, and anybody else who wants to talk with us is going to have to do it, too. Because, again, Signal's bar is very high. Well, in fact they just added post-quantum encryption in Signal.

MIKAH: Right.

Steve: So, you know, you're going to - if what Meta is saying is we'll connect to you if you're as good as Signal, well, that now means if you also have post-quantum encryption available.

MIKAH: Yeah. So basically it is saying you're going to need to use Signal because you're not going to be able to prove to us that anything else is as secure.

Steve: It won't be as good if it's not Signal because Signal is the best. Unless you ask Apple, and they think, well, Signal is our Level 2. You know, we're Level 3. It's like, okay, fine. So. Because Apple is rotating their keys every 50 messages or every week, whichever comes first.

MIKAH: Right.

Steve: And they're saying, oh, that gives us an advantage. So it's okay, fine.

MIKAH: Yeah. I don't know - I'm with you in that I don't think that should bump it up to Level 3. But at the same time, when I was reading through the explanation of everything, I did think that was cool because I hadn't considered how having somebody's whole history of messages, even in a post-quantum world...

Steve: Ah, is using the same key.

MIKAH: Yeah, using the same key, that that means the whole thing. But this just means they only get - that's pretty smart. That would be cool to see implemented.

Steve: I agree with you. It's like, well, why not do it if we can? And it was expensive to do it. You know, they had to amortize that key across many messages because the key's 2K, and many messages are like, "Okay, Mom, I'm on my way."

MIKAH: Right.

Steve: You know? So the per-message overhead of 2K, that was insane, you know, when the message is 12 characters. So anyway, okay. So a lot of our listeners have asked me about this big cyberattack, a ransomware attack, on the U.S.'s Change Healthcare service. Nearly three weeks ago, on Wednesday, so three weeks ago tomorrow, on Wednesday, February 21st, the American company Change Healthcare, which is a division of UnitedHealth Group, also known as UHG, was hit by a ransomware attack that was devastating by any measure. That cyberattack shut down the largest healthcare payment system in the U.S.

Then, to give you a sense for this, just this past Sunday, two days ago, the U.S. Department of Health and Human Services addressed an open letter to "Health Care Leaders," writing: "As you know, last month Change Healthcare was the target of a cyberattack that has had significant impacts on much of the nation's healthcare system. The effects of this attack are far-reaching. Change Healthcare, owned by UnitedHealth Group (UHG), processes 15 billion" - with a B - "healthcare transactions every year and is involved in one of every three patient records. The attack has impacted payments to hospitals, physicians, pharmacists, and other healthcare providers across the country. Many of these providers are concerned about their ability to offer care in the absence of timely payments, but providers persist despite the need for numerous onerous workarounds and cash flow uncertainty." So this has really upset a lot.

Okay. So backing up here a bit, the day following the attack, February 22nd, this UnitedHealth Group filed a notice, as they must, with the U.S. Securities and Exchange Commission because they're a publicly traded company, stating that "a suspected nation-state associated cybersecurity threat actor" had gained access to Change Healthcare's networks. Following that UHG filing, CVS Health, Walgreens, Publix, GoodRx, and BlueCross BlueShield of Montana reported disruptions in their insurance claims. Yeah, basically it was all shut down. The cyberattack affected family-owned pharmacies and military pharmacies, including the Naval Hospital at Camp Pendleton. The Healthcare company Athenahealth was affected, as were countless others.

MIKAH: Wow.

Steve: One week later, on the 29th of February, you know, Leap Year Day, UHG confirmed that the ransomware attack was "perpetrated by a cybercrime threat actor who represented itself to Change Healthcare as ALPHV/Blackcat." And I'm just going to refer to them as Blackcat from now on because it's easier. In the same update, the company stated that it was "working closely with law enforcement and leading third-party consultants Mandiant and Palo Alto Networks" to address the matter. And then, four days later, that is, after this disclosure on the 29th, which is eight days ago on March 4th, Reuters confirmed that a bitcoin payment equivalent to nearly $22 million USD has been made into a cryptocurrency wallet "associated with Blackcat."

MIKAH: We think they paid the ransom?

Steve: Yeah.

MIKAH: Oh, my god.

Steve: Yeah. UnitedHealth has not commented on the payment, instead stating that the organization was "focused on the investigation and the recovery." Right. Apparently to the tune of $22 million USD. On the same day, a reporter at Wired stated that the transaction looked "very much like a large ransom payment." What's transpired since then is a bit interesting, since Blackcat is a ransomware-as-a-service group.

MIKAH: Oh. Okay.

Steve: This of course means that they provide the software and the backend infrastructure, while their affiliates are the ones that perpetrate the attacks, penetrate the networks, and in return for that the affiliate receives the lion's share, in this case 70%, of any ransoms paid. That's always been the way it is. However, in this instance it appears that Blackcat is not eager to part with that 70%, which amounts to a cool $15.4 million. So they're claiming that they've shut down and disbanded.

MIKAH: What?

Steve: Nice timing on that. Okay. So exactly one week ago, the HIPAA, the U.S. organization, the HIPAA Journal posted some interesting information. They wrote...

MIKAH: Oh, sorry. Sorry, Steve, can I correct you there? It's H-I-P-A-A, not H-I-P-P-A.

Steve: Oh, good, thank you, yes. Good. Thank you. HIPAA. Thank you. So they said: "The ALPHV/Blackcat ransomware group appears to have shut down its ransomware-as-a-service (RaaS) operation, indicating there may be an imminent rebrand. The group claims to have shut down its servers, its ransomware negotiation sites are offline, and a spokesperson for the group posted a message, 'Everything is off, we decide.'" Probably a Russian speaker. "A status message of 'GG' was later added, and Blackcat claimed that their operation was shut down by law enforcement and said it would be selling its source code.

"However, security experts disagree and say there is clear evidence that this is an exit scam, where the group refuses to pay affiliates their cut of the ransom payments and pockets 100% of the funds. Blackcat is a ransomware-as-a-service operation where affiliates are used to conduct attacks and are paid a percentage of the ransoms they generate. Affiliates typically receive around 70% of any ransoms they generate, and the ransomware group takes the rest.

"After the earlier disruption of the Blackcat operation by law enforcement last December, Blackcat has been trying to recruit new affiliates and has offered some affiliates an even bigger cut of the ransom. An exit scam is the logical way to wind up the operation, and there would likely be few repercussions, other than making it more difficult to recruit affiliates if the group were to choose to rebrand." Right.

MIKAH: Oh, wow.

Steve: Who's going to do this again if you just screwed the last affiliate that you were working with after they generated a $22 million ransom.

MIKAH: Hello. We are Whitecat, and we have no affiliation to the Blackcats. Would you like to work with us?

Steve: Right. Maybe dark black or dark gray. Anyway, HIPAA wrote: "It's not unusual for a ransomware group to shut down operations and rebrand after a major attack, and Blackcat likely has done this before. Blackcat is believed to be a rebrand of the earlier BlackMatter ransomware operation, which itself was a rebrand of DarkSide."

MIKAH: Okay. If they have something with shadows, dark, black, you know it's the same company, people.

Steve: That's right.

MIKAH: Going forward. Come on.

Steve: That's right. DarkSide was the ransomware group behind the attack on the Colonial Pipeline in 2021 that disrupted fuel supplies on the Eastern Seaboard of the United States. Shortly after the attack, the group lost access - what do you know - to its servers, which they claimed, probably was, due to the actions of their hosting company. They also claimed that funds had been transferred away from their accounts and suggested they were seized by law enforcement. BlackMatter ransomware only lasted for around four months before it was in turn shut down, with the group rebranding in February of 2022 as Blackcat.

On March 3rd of this year, an affiliate - and here it comes - with the moniker Notchy (N-O-T-C-H-Y) posted a message on Ramp Forum claiming they were responsible for the attack on Change Healthcare. The post was found by a threat researcher at Recorded Future. Notchy claimed they were a long-time affiliate of the Blackcat operation, and that could have gone back two years because that's when Blackcat reformed, had the "affiliate plus" status granting them a larger piece of the pie, and that they had been scammed out of their share of the $22 million ransom payment.

They claimed that Optum, which is the actual organization within Change that was hit and paid, Optum paid a 350 Bitcoin ransom to have the stolen data deleted and to obtain the decryption keys. In other words, full-on standard ransomware payment. Notchy shared the payment address which shows a $22 million payment had been made to the wallet address, and the funds have since been withdrawn. The wallet has been tied to Blackcat as it received payments for previous ransomware attacks that have been attributed to the group.

Notchy claimed Blackcat suspended their account following the attack and had been delaying payment before the funds were transferred to Blackcat accounts. Notchy said that Optum paid to have the data deleted, but they have a copy of 6TB, that's how much was stolen, of data in the attack. Notchy claims the data includes sensitive information from Medicare, Tricare, CVS-Caremark, Loomis, Davis Vision, Health Net, MetLife, Teachers Health Trust, tens of other insurance companies, and others. The post finishes with a warning to other affiliates that they should stop working with Blackcat. It's unclear what Notchy plans to do with the stolen data and whether they will attempt to extort Change Healthcare or try to sell or monetize the data.

Fabian Wosar, Emsisoft's CTO, is convinced this is an exit scam. After checking the source code of the law enforcement takedown notice, he said it is clear that Blackcat has recycled it from December's earlier takedown notice. Fabian tweeted: "There is absolutely zero reason why law enforcement would just put a saved version of the previous takedown notice up during a seizure instead of a new takedown notice." He also reached out to contacts at Europol and the NCA, who said they had no involvement in any recent takedown. Currently, neither Change Healthcare nor its parent company UnitedHealth have confirmed that they paid the ransom and then been cheated out of it, and issued a statement that they are currently focused on the investigation.

MIKAH: I'm sure.

Steve: So, yeah. This is all a big mess. It appears that the Blackcat gang has made off with Optum's $22 million, that Notchy affiliate did not receive the $15.4 million or more that they feel they deserve, Optum got neither the decryption keys nor the deletion of their 6TB of data, which they paid for. And no one but the Blackcat guys are smiling at this point. Since rebranding and returning to the ransomware-as-a-service business may be impossible after taking their affiliate's money, and since $22 million is a nice piece of change, they may just go find a nice beach somewhere to lie on.

MIKAH: One hopes.

Steve: You know? Meanwhile, here in the states, the inevitable class action lawsuits have been filed due to the loss of patient healthcare records. At last count at least five lawsuits are now underway.

MIKAH: Oh, my word. This is awful. All the way around this is awful.

Steve: Big mess.

MIKAH: It's a huge mess. And it's, I don't know, it kind of speaks to the dangerous nature maybe of having the entire country's, nearly the entire country's healthcare operating under one company. That that much can be impacted by getting into one company is pretty scary.

Steve: Yes. They are a service provider. And so all kinds of other of their clients use them to provide the insurance processing, and in return the payments back to them. And all of that is shut down now. They don't have their decryption keys. All of their server infrastructure is encrypted. They were willing to pay $22 million to get it back online more quickly. We don't know anything about the state of their backups and so forth. But even so, 6TB of medical record history is now also in the hands of bad guys.

And, you know, this is the unfortunate downside of the U.S.'s free enterprise system, which I'm, you know, has lots of upsides because it allows people to apply their efforts and to be clever and to create companies. Unfortunately, there's a tendency for the big fish to eat the small fish, and for consolidation to happen. And so what we're seeing here is, as you said, what would have otherwise been a much more distributed set of services have all been consolidated.

You know, on one hand, yes, it's more efficient. You're getting to use one larger set of infrastructure instead of lots of smaller infrastructures. But with it comes responsibility. With that consolidation comes responsibility. And what we're now seeing is what happens when one could argue that responsibility was not met.

MIKAH: With great power, they say.

Steve: Yes.

MIKAH: That's so many records. You said, what, for every three, it's two of every three?

Steve: One out of every three medical records in the U.S. is basically in that 6TB of data. Yeah.

MIKAH: Yeah, awful.

Steve: On my side, everything continues to proceed well with SpinRite. The limitations of 6.1, you know, being unable to boot UEFI-only systems and its lack of native high-performance support for drives attached by USB and NVMe media, are as annoying as I expected that they would be. So there just wasn't anything I could do about it. I'm working to get 6.1 solidified so specifically so that I can obsolete it with 7.0 as soon as I possibly can. But that said, I still do want to solve any remaining problems that I can, especially when such a solution will be just as useful and necessary for tomorrow's SpinRite 7 as it is for today's 6.1.

And that's the case for the forthcoming feature I worked on all of last week. It's something I've mentioned before which I've always planned to do, and that's to add the

capability for people without access to Windows, mostly Linux users, who are Linux users and so don't have Windows, to directly download an image file that they can then transfer to any USB drive to boot their licensed copy of SpinRite.

For those who don't know, the single SpinRite executable, which is about 280K because of course I write everything in assembly language, is both a Windows and a DOS app. When it's run from DOS, it is itself SpinRite. But when that same program is run from Windows, it presents a familiar Windows user interface which allows its owner to easily create bootable media which contains itself. So it can, you know, that can either format and prepare a diskette - which there's not much demand for, but since the code was there anyway from 6.0, I left it in. Or it can create an optical disc ISO image file for burning to a disc or loading with an ISO boot utility. It can also create a raw IMG file, which can be put somewhere, or it can prepare a USB thumb drive.

Twenty years ago, back in 2004 when I first created this code, the dependence upon Windows provided a comprehensive solution because Linux was still mostly a curiosity back in 2004 and had not yet matured into the strong alternative OS that it has since become. Today, there are many Linux users who would like to use SpinRite, but who don't have ready access to Windows in order to run SpinRite's boot prep. And this need for non-Windows boot preparation will continue with SpinRite 7 and beyond.

The approach I've developed for use under Windows, which is used by InitDisk, ReadSpeed, and SpinRite, is to start the application, then have its user insert their chosen USB drive while the application watches the machine for the appearance of any new USB drive. This bypasses the need to specify a drive letter, it works with unformatted drives or drives containing foreign file systems so they wouldn't get a drive letter, and it makes it very difficult for the user to inadvertently format the wrong drive, which was my primary motivation for developing this user-friendly and pretty foolproof approach.

Unfortunately, there's a downside. It uses very low-level Windows USB monitoring, which was not implemented in WINE, which is the Windows emulator for Linux. So for Linux users, a ready-to-boot image file is the way to go, and I'm in the process of putting the final pieces of that new facility together. It should be finished later this week, and I'm sure I'll just make a note of it next week that it's, you know, you can go use it.

And let's take our last break, and then we'll do some feedback from our listeners and get to our episode's main topic.

MIKAH: Wonderful. All right. I do have to take a moment here. By the way, Mikah Sargent, subbing in for Leo Laporte, who is on vacation.

Steve: We knew that.

MIKAH: It's time to close the loop with feedback from the listeners.

Steve: Yes, it does. Okay. So John Robinette said: "Hey, Steve. I'm sure I'm not the only one to send you this note about Telegram after listening to SN-964. There's been much chatter about the protocol Telegram uses for end-to-end encryption, but it is a common misunderstanding that they use this by default. Telegram's default uses only TLS to protect the connection between your device and their servers, and does not provide any end-to-end protection. They have an additional feature, 'Secret Chats,' that does use end-to-end encryption on the client device. It is not possible to use end-to-end encryption in group chats, and when used for one-on-one chats it limits the conversation to a specific device. Based on my anecdotal experience using Telegram with a few friends, most people either do not know about 'Secret Chats' or do not use it.

"I found a post from 2017 that explains Telegram's reasoning for this." And he quoted it, saying: "The TL;DR is because other apps, for example WhatsApp, allow you to make unencrypted backups" - this is Telegram speaking, essentially - "actual end-to-end encryption isn't worth being on by default."

MIKAH: What?

Steve: Okay. So they don't turn it on by default, and claim they are more secure because of how they store backups. He says: "Anyway, it's not just Apple marketing speak that Telegram is not end-to-end encrypted. It's by design from Telegram. Thanks."

MIKAH: Wow.

Steve: So John, thank you. And I should mention several other of our listeners who actually use Telegram also shared their experience. What they showed was that, unlike the way I presumed it would be when I talked about it last week, Telegram really is probably mostly used in its insecure messaging mode since not only is it not enabled by default, enabling it is not even a global setting. It must be explicitly enabled on a chat-by-chat basis. So that makes the use of encryption hostile in Telegram, which is certainly not what anyone hoping for privacy wants. And its fundamental inability to provide end-to-end protection for multiparty group chats strongly suggests that it's being left behind in the secure messaging evolution. So I presume that they're aware of this, and they're hopefully working behind the scenes to bring Telegram up to speed, since otherwise it's just going to become an historical footnote.

Given these facts, I certainly reverse myself and agree with Apple's placing Telegram down at Level 0, along with QQ, whatever that is, where it certainly belongs. So thank you very much, John and everybody else, for educating me about Telegram, which I did not take the time to - and I'm amazed, for all the talk of Telegram, that it is as insecure as it obviously is.

Someone who's tweeted before, whose handle is a hacker version of anomaly, it's 3n0m41y, he said: "Steve, KeePassXC latest release is also supporting Passkeys now. Great news." Okay. I'm not a Passkey user, so I don't know what communication they may provide to their users, so I wanted to pass along that welcome news. Back when Passkeys first appeared, supported only by Apple, Google, and Microsoft, they each created their own individual and well-isolated walled gardens to manage Passkeys only on their devices, and only within their own ecosystems. There was no cross-ecosystem portability.

At that time we hoped that our password managers, which are already an inherently cross-platform, would be stepping up and getting in on the act because they would be able to offer fully cross-platform Passkey synchronization. And as we know, Bitwarden, a TWiT network sponsor, has done so. And given competitive pressures it will soon become incumbent upon any and all other password managers to offer Passkey support. And we'll be talking about what that means when we get to this week's main topic. But for now, for anyone who's using KeePassXC and didn't know, it's now got Passkey support. So that's all for the good. A user whose actual...

MIKAH: Good luck with this one.

Steve: Exactly. His Twitter handle, he later tells us, was created by asking Bitwarden for a random gibberish string. He said: "Episode 964." He said: "I'm on the go and won't have time to write an eloquently worded message like the ones you read on the show." Actually, I think his was, but okay. "So feel free to simply summarize this if you find it

relevant. In 964" - last week - "you mentioned feedback from a listener who mentioned the Taco Bell app using email passwords 'for convenience.'

"Before my main comment, I'd like to mention that they are not the only ones doing this. I suspect the most popular service out there today that does this is Substack, the blogging service that has exploded in popularity since early/mid COVID. When I signed up there for the first time, I wasn't aware, and the whole process was very confusing and counterintuitive given my previously learned username/password behavior. Beyond being confusing, it truly is much more of a pain for those of us who use browser-based password managers, which have made the traditional login process rather seamless."

Okay. So I'll just interrupt to say, right, you know. And you probably haven't heard me say this, Mikah. The more we explore the topic, given that passwords are entirely optional, when every site includes an "I forgot my password" recovery link, I think that viewing passwords as login accelerators, which are handled now by our password managers, is quite apropos.

MIKAH: I like that, yeah. That's a good way to think of it because, yeah. Anecdotally speaking I know many people who are not as techie as we are whose password really is that "I forgot my password button" every time, that's the password because they forget it right after they've changed it, and then they go to their email. So in a way that - because I have always hated this, the I try to log in, and they send me what they call a "magic link" to my email. Slack has this functionality. They kind of suggest you use that first and foremost whenever you log back into Slack. And mentioned here Substack, there are so many that do this. And I don't like it. But when I think about now anecdotally the people I know in my life who are already doing that, going to their email every time to reset their password again, yeah, password or a login accelerator. That's clever.

Steve: That's really what it is. And in fact we began this dialogue several weeks back. We did a podcast titled "Unintended Consequences." And what became clear was that with Google's deprecation of third-party tracking cookies by midpoint of this year, the advertisers are freaked out that they're going to lose the ability to aggregate information about their visitors. So they're putting pressure on websites to add a "give us your email to join the site."

And so the unintended consequence of third-party tracking being blocked robustly in Chrome, as it's going to be, is that websites are being incentivized to ask their visitors, not necessarily to create an account, but to "join." And so they're bringing up a request for your email address, then sending you a link which you have to click on so you just can't use gibberish, in order to continue using the site.

So essentially this is going to get worse. And when we just go to a site, if your browser doesn't already have a persistent cookie for the site, then the site will ask you for an email address just to get it because they want to be able to provide that email address to the advertisers who are on the site because the advertisers will pay more money to the site if they receive an email address which has been confirmed in return. So we're entering a world of pain here. But so that brought us to the whole idea of using email in lieu of logon and what that means. And so that's what this guy's talking about.

So he says: "Anyway, my main comment is that it may very well be possible that these implementations are not for user convenience," meaning implementations of logon with email. He says: "They are for liability." He says: "By requiring an email to log in, the host company hoists account breach liability off of their own shoulders onto the email providers. Your account got hacked? That's a Gmail breach. Not our problem."

He says: "On a side note, you also mentioned in the episode a listener in the financial sector who commented on password length, complexity, and rules being limited by old legacy mainframes. I just wanted to share that I worked for a federal agency about a decade ago, and thought I discovered a bug when I realized I had just been let into my sole enterprise account despite missing a character on the end of my password."

MIKAH: What?

Steve: "I tested that again and discovered the system was only checking the first eight characters of the password I thought I was using. I grabbed a fellow developer to show them, thinking I'd blown the lid off some big bug, and he shrugged and said, 'Yeah, it only checks the first eight.'" He said: "I was stunned."

MIKAH: Good god.

Steve: "Thanks for the wonderful content. Cheers." He says: "P.S.: Yes, this account username was generated by my Bitwarden password generator."

MIKAH: Got it.

Steve: Okay. So we first took up the subject of email-only login with our "Unintended Consequences" episode, since it was becoming clear that websites were planning to react to the loss of third-party tracking with the establishment of first-party identity relationships with their visitors. And in every instance the company's so-called "privacy policy" clearly stated that any and all information provided by their visitors, including their name and their email address, not only could be, but would be, shared with their "business partners," which we know are the advertising networks providing the advertising for their site.

But I thought this listener's idea about limiting liability was interesting, and it had not occurred to me. When you think about it, what's the benefit to a website of holding onto the hashes of all of its users' passwords? The only benefit to the site is that passwords allow its users to log in more quickly and easily, when and if they're using a password manager. The problem is the site cannot know whether the user is using a password manager. How big an issue is that?

Today, only one out of every three people, 34%, are using a password manager. And while this is a big improvement over just two years ago in 2022 when the figure was one in five, or 21%, this means that today two out of every three Internet users are not using password management. And we know this suggests, exactly as you were saying, Mikah, of your friends, that the quality of those two-thirds of all passwords may not be very high. They may no longer be using "monkey123," but their chosen passwords are likely not much better.

Now look at what happened with the 23andMe disaster. A shocking number of 23andMe user accounts were apparently compromised using some form of password spray. That could never have succeeded if 23andMe users were logging in with highly complex unique-per-site passwords, which of course are encouraged by the use of any password manager. But with two-thirds of Internet users still today not using a password manager, that suggests that two-thirds of 23andMe users were not being well protected from the abuse of their poor password hygiene. If you're not someone who understands the value of password management, "monkey123" looks pretty good, and it sure is easy to remember.

My point is the quite successful attack on 23andMe demonstrates that the use of traditional username and password login creates a single point of failure which facilitates automated attack. A widely distributed network of bots can create authentic-appearing

web sessions and attempt to login as legitimate users over and over and over under the cover of darkness, succeeding eventually and incrementally with no involvement on any user's part. And that makes a crucial difference. By comparison, if a user's email inbox were to be flooded with 23andMe login requests, every such user - and quite soon 23andMe - would know that someone or something was up to no good. But would an attacker even bother? Doesn't the email loop completely thwart such attacks? Logins are no longer autonomous. And unless the attacker obtains access to every user's email account, the attacker cannot login using user authentication data contained in the email. This eliminates all generic remote password spray brute force attacks.

So the point I wanted to make is that this listener's comment about liability is very interesting. We're constantly hearing about site breaches, with Troy Hunt's "Have I Been Pwned" service constantly collecting massive new troves of breached user login credentials. All of that disappears if a website no longer offers to store its users' password hashes. Why bother? It's just a potential nightmare. It's a liability, a secret they're not good at keeping. With two-thirds of users today still not bothering to use a password manager, they are likely using crappy passwords, or the same password at all their different sites. This renders the site vulnerable to attack, as 23andMe was, through no fault of theirs because their users cannot be bothered to secure themselves better.

So these attacks go unnoticed and unobserved because they can be conducted by autonomous distributed networks of bots. But that cannot happen if the user's email account is dynamically included in the login process. This completely changes the attack dynamics. And don't get me wrong, I'm not suggesting that I think this is a good idea, except that it kind of is, because it definitely represents a nightmare for the other one-third of us who are using password managers stuffed with long passwords we could never begin to memorize or even correctly enter into a password field.

The problem for our future is that email loop login makes a horrible kind of sense for any website that gets to choose whether or not they want the liability of storing the hashes of their users' potentially crappy passwords. It's easy to imagine websites deciding that they'd prefer to ask their users to obtain a per-login token from their own email.

MIKAH: Dang. This makes sense, Steve.

Steve: Yeah.

MIKAH: Sorry, but it's my responsibility to make sure that this part of the episode never makes it to the light of day so that every - no, I'm just kidding. Now I'm worried that people are going to be listening to this and going, you know what? Because you just convinced me. This whole time I've been annoyed by that email login magic link nonsense. But it's not nonsense, especially because it requires so much extra effort by these folks who are right now not using much effort at all. That is so very true. Huh.

Steve: And unfortunately it transfers the effort to the user who wants to log in. So it distributes it. Now, we're going to be talking by the end of the podcast about Passkeys versus not. And an interesting compromise suggests itself here, which is, if you use Passkeys, then you get instant logon. If you don't want to use Passkeys, you get email loop login. Because as I'm going to demonstrate, usernames and passwords, they're not equivalent, even close. So more to come.

Mark Jones tweeted: "Another in the 'it was nice while it lasted' category." He said: "I have run into two sites that won't accept anything @duck.com emails." As we know, DuckDuckGo offers an email anonymizing forwarding service, right, where it's something @duck.com. We've talked about this for several weeks as a solution to the problem of every website you visit wanting your email address just for the privilege of seeing the content because they want to provide that to their advertisers. Then the solution is, oh,

let's create, you know, anonymous fake email accounts. Well, Mark is noting that he's run into two sites that will not accept @duck.com emails. He said: "A previous solution, using a '+' in a Gmail address, also has been thwarted.

He said: "That allowed me to" - that, using a "+" - "allowed me to filter easily and reject sites of no interest or that got spammy. One service stripped the plus sign and what was after it out. The other trend you haven't mentioned is the insistence on a cell number. Those are unique, difficult to share, and are harder to make throwaway. I was trying to set up services for a community organization, only to have a couple of vendors balk because my cell was already associated with a different active account. Love the podcast, look forward to it every week, and I'm happy it will continue. Tested 6.1 and am loving that, too."

Okay. So as Mark wrote: "It was nice while it lasted." Unfortunately, while it is possible to hide our real email address behind an identity-protecting email-forwarding service, it's not possible to hide the fact that we're using an identity-protecting email-forwarding service. And the fact that a site is stripping the plus sign and what follows from a user's email address is nothing short of rude. You know, it's not their business to alter someone's email address. That really, you know, wow, that seems so wrong. The only leverage website visitors have is to choose not to play.

Normally, I would suggest that such email address discrimination would not become widespread. That is, you're only going to see it at a few sites. But if websites are asked for their visitors' email as a replacement for third-party tracking, with the incentive of being paid more for ads served to identifiable Internet users, which is the case, then those advertising interests who are definitely paying do not want to pay more for anyone using an anonymizing email-forwarding service. So you can bet that a site's advertisers will be telling those websites that they won't pay for anonymized email. And then, in turn, those sites will be telling their visitors exactly what our listener Mark Jones was told: "Please provide your primary email address, not a forwarding service."

MIKAH: Wow. That - so I have not had that issue. And I use Fastmail, who is a sponsor on the network, along with I use the password manager 1Password. And so when it automatically generates it, it creates a masked email for me.

Steve: Yup.

MIKAH: And that right now, knock on wood, has not been an issue. But I have had the issue with the plus sign. I always - I never thought of it, because this has been a long time ago, and I always - I didn't think of it in the nefarious way that we're thinking of it here. I just thought it was somebody who didn't know what they were doing coding-wise, and so they couldn't, like, parse the plus and what came after it. And I thought, oh, maybe they just don't know how to accept that and think that it's a - the rules for what you can type in, it's telling the system that it's a fake email, or that it's not an email address because of the rules that are set up. But now I can see, too, how in that case it could be somebody going, no, we know that that's just adding on. Because I did, I had my name plus spam @gmail.com.

Steve: Right.

MIKAH: And I could filter it out that way. And I had to - I ran into that issue too many times, so I had to stop doing that.

Steve: Yup, yup. Well, let's hope that Fastmail continues to be available to you.

MIKAH: Yes.

Steve: It'd be interesting to find out if it is not at some point.

MIKAH: Definitely.

Steve: Rob Powell wrote: "Hi, Steve. Following recent discussions on the podcast, I thought other listeners might also appreciate the below. It's a tool to detect when Chrome extensions change owners." And this is really interesting. I got a kick out of the name of the tool that Rob pointed to. It's posted on GitHub under the account of "classvsoftware," all one word, classvsoftware, with the name "Under New Management."

A couple of weeks ago we were noting that the authors and maintainers of Android freeware which had accumulated large install bases over time and earned some implicit trust from their users have effectively been cashing out their installed bases to less than scrupulous buyers, who then take advantage of that installed base, you know, to stick ads in the apps that never used to have them. You know, like dumb apps, you know, free things, like measuring sticks and things. I mean, just nothing.

Since the same thing could - well, actually, specifically it was an app that allowed you to change the brightness of the flashlight. And, you know, it became adware because the author who, you know, wasn't getting paid anything for it, got tired of maintaining it and finally decided to accept an offer. Since the same thing could happen with browser extensions whose special position in our browsers make this an even more pressing and worrisome problem, I was glad that Rob brought this up. I went over to the GitHub site to see what the authors of this extension had to say. That is, this "Under New Management" extension.

They explain. They said: "Intermittently checks your installed extensions to see if the developer information listed on the Chrome Web Store has changed. If anything is different, the extension icon will display a red badge, alerting you to the change." And as to why this is an issue, they write: "Extension developers are constantly getting offers to buy their extensions. In nearly every case, the people buying these extensions want to rip off the existing users of those extensions. The users of these extensions have no idea an installed extension has changed hands and may now be compromised. 'Under New Management' gives users notice of the change of ownership, giving them a chance to make an informed decision about the software they're using."

Now, in the text the phrase "constantly getting offers to buy their extensions" was a link. So I wondered what the author of this "Under New Management" extension might be linking to, and holy crap. The link was to another GitHub page titled "Temptations of an open-source browser extension developer." But before I describe what's there, let me back up a bit. The extension in question is called "Hover Zoom+," and its author explains.

"Hover Zoom zooms images and videos on" - so this is the extension, Hover Zoom. "Zooms images and videos on all your favorite websites - Facebook, Amazon, et cetera. Hover your mouse over any image on the supported website, and the extension will automatically enlarge the image to its full size, making sure that it still fits into the browser window. This is an open-source version of the original Hover Zoom extension, which is now overrun by malware and deleted from the store.

"In this version, all spyware has been removed, many bugs were fixed, and new features were added. It doesn't" - does not - "collect any statistics whatsoever. The only permission it needs is to access data on all websites to extract full images, and optional permissions to access browser history, download/save images, or get tab URLs for per-site configuration. This extension will never be sold out and will never compromise users' privacy. As a proof, please see the list of all takeover offers I've received over the past years."

Okay. And this brings us back to the "Temptations of an open-source browser extension developer" page. It is so astonishing that I've made it the GRC shortcut of the week. If you're at all curious, and please be curious because it is something, visit grc.sc/965. So that's today's episode number, 965, grc.sc, you know, short for shortcut, grc.sc/965. The author starts off, the same author who we just heard from, who fixed this malware-ridden previous Hover Zoom extension, and now vows that he will never sell it.

He said: "Over the years I have received many proposals to monetize this extension. So I think I'll just start posting them here for fun, but not for profit. The main reason I continue to maintain this extension is because I can hardly trust others not to fall for one of these offers. I'm fortunate to have a job that pays well enough to allow me to keep my moral compass and ignore all of these propositions. I realize that not everyone has the same financial security. So hopefully this thread will shed some light on what kind of pressure is put on extension developers."

And what follows really serves to put this into perspective. The first offer he posts is dated September 28th, 2015. It reads: "Hope this message finds you well. I'm a Strategic Partnerships Manager at a monetization platform for browser extensions." In other words, there is such a thing. "I'm contacting you since I came across the extension 'Hover Zoom+' at the Chrome Store. I consider your product can help you bring profit by means of collaborating together. I would like to suggest a potential partnership between our companies that will significantly increase your revenues. Are you interested in discussing our offer in more details? Hope for positive feedback and ongoing cooperation." Okay, that was the first one he listed.

The most recent offer the author posted four weeks ago on February 14th, 2024, read: "I trust you're doing well! Currently I'm exploring opportunities to grow my business by investing in Chrome extensions. Your extension Hover Zoom+ has caught my attention, and I am genuinely interested in discussing the possibility of acquiring it. We can discuss the price and complete the transaction securely through a reputable escrow service, escrow.com or cryptoexchange.com. Google supports a smooth transfer of extension ownership from one account to another, ensuring your Gmail account remains unaffected. If you have any inquiries, or if this aligns with your plans, feel free to reach out to us via this email. Looking forward to hearing from you."

Now, in between that first and last offer are more than - I counted them - 165 other offers this guy has received from parties interested in taking "Hover Zoom+" off his hands. I say "more than" because for a while I was not counting the follow-ups as being separate offers. But there were so many of them that after a while I decided a more accurate and fair count should include them. So 165 and counting.

The danger to the users of extensions that are sold to unscrupulous buyers is that an originally benign extension might be altered to begin harvesting its users' browsing data as the U.S. Federal Trade Commission has formally accused the Avast browser extension of doing. Since the purchasers of popular extensions with a large installed base have no interest in ongoing support of that extension and simply wish to maximize their return on investment while they can, they will have a plan for somehow monetizing the extension's user base.

Given that, the value of the "Under New Management" extension becomes clear. So thanks for bringing it to our attention, Rob. Appreciate it.

And lastly, Max Feinleib said: "I'm sure I'm not the first person who's mentioned this, but SN-905 will be hard to beat for the shortest title." And he says that was just the numeral, the digit "1." And I made a comment to Leo last week that "PQ3," which was last week's title, might be the shortest podcast title we've ever had. Max was actually the second of

two sharp listeners who pointed out what amounted to "Not so fast there, Gibson. What about podcast number 905? That was titled with the single digit '1'."

That frighteningly low number was a reference to the password hashing iteration count some unfortunate LastPass users found when they went looking. So anyway, thank you for catching that, Max and the other person who mentioned it. You're correct. I will never have a shorter podcast title than "1." I can't imagine a null title, so yeah.

Okay. So I think people are going to be a little surprised. As I mentioned last week, I intended to title this podcast "Morris II." That'll be next week. This is a result of a tweet from Stephan Janssen. He said: "Hi, Steve. I'm an avid SN listener and, in part thanks to your information, moved from LastPass to Bitwarden. Now that Passkeys are becoming more prevalent, I'm noticing a Bitwarden popup when I try to enroll my YubiKey as my two-factor authentication token.

"I'm tempted to give Passkeys in Bitwarden a try, in part due to your enthusiasm about it, but I'm also hesitant since it feels like I'm losing and giving up my second factor. With Google, for example, using a Passkey allows me to log in without providing anything else, which makes it feel like I no longer have MFA for my account. Am I right in thinking that Passkeys will reduce my security if I'm now using a random password with a second factor, or am I missing something in my thinking?"

Okay. I'm going to give everyone the short TL;DR first. But then because, as they say, "it's complicated," I'll explain more. So here it is: In a properly operating world, Passkeys, all by its lonesome, provides far more security than any super-strong password plus any second factor, period.

Let's start working our way through what has become a confusing authentication minefield. First off, why does anyone use a second factor? And I'm going to pointedly ignore the primary reason we, who listen to this podcast, use second login factors, which is mostly because they're cool.

MIKAH: Fair.

Steve: Right? You know, it feels a little secret agent-like to have to go look up that time-varying code to complete an authentication. It definitely provides the impression of having much greater security, even if there are still ways around it. And we've talked about the ways around it. Multifactor authentication can be and is actively compromised when there's some strong need to do so. There are two ways this can be accomplished.

The first is arranging to insert a man in the middle. This is done in today's world by arranging to trick an unwitting user into going to a fake login page. They provide their username and password, and are then prompted for the second factor. They provide that to the man in the middle, who turns around and immediately logs in as them before that ephemeral six-digit code has a chance to expire. This is not easily done because it requires a higher end attacker and some setup. But it is being done. Normally this only intercepts a user's username and password, which does give the bad guys more permanent account access until the password is changed. But when you think about it, once you have that man-in-the-middle proxy in place to intercept the username and password, the addition of an ephemeral second factor adds very little to the user's actual security.

The second way the MFA system can be defeated is the same way any site breach, with a loss of their authentication database, you know, their password hashes, can defeat the security for their users. Just as the authentication database contains a hashed password which is used to verify the user's freshly- provided login attempt password, that authentication database must also contain the shared MFA secret which keys the user's

TOTP, their time-based one-time password. So if that MFA secret can be stolen along with the username and password hash, then bad guys have potentially acquired all of the secrets they need to impersonate the site's users anytime they wish.

The key here is my use of the term "secrets" because today's time-based one-time passwords are based upon shared secrets. The site knows the secret key, and your authenticator app has the same secret key. That's what allows the authenticator to generate a six-digit code based upon the time of day; and for the server, knowing the same time of day, to generate the same code and compare them.

And "storing secrets" is the crucial difference in the Passkeys system. This was also true with SQRL's technology which I articulated on this podcast more than 10 years ago. What I used to say was that "SQRL gives websites no secrets to keep." And exactly the same is true of today's Passkeys.

The crucial difference is secret versus public key technology. That's what makes Passkeys so very different from everything that came before it, except of course SQRL, which works similarly. With Passkeys, the user's private key never leaves the authentication device end, and the website only ever receives the user's matching public key. And the only thing that the website's public key can be used for is verifying the signature of a unique challenge.

The website sends the user a never-before-used large random number to sign. Since the large random number has never before been used, its signature will also never have been used before, so no form of replay attack is possible. The user's Passkey agent signs the challenge received from the server and sends it back to the website, signed. The website then uses its matching public key to verify the user agent's Passkey signature. And crucially, verifying signatures is all it can do with that public key. That's the only power it has. So if that key were to be stolen, no one cares. Truly, no one cares.

The reason I felt that Stephan's question deserved our attention is that none of this crucial difference between traditional username and password, and even including multifactor authentication, versus Passkeys is in any way apparent to users. Users just see yet another thing, another way to log on. What's impossible to see and to appreciate from the user-facing side is how completely and importantly different Passkeys is from everything that has come before. So to respond to Stephan's uneasiness about whether he's losing anything, sacrificing anything, or giving up anything by choosing to use a Passkey instead of having traditional login - even when it's augmented with multifactor authentication - the bright flashing neon answer is "No."

The best way to think about multifactor authentication is that it was a last-ditch temporary Band-Aid that was added as an emergency measure in an attempt to do something to shore up the existing ecosystem of fundamentally weak and creaky server-side secret-based password-only authentication. You know, passwords are not that great. Let's add another factor. Unfortunately, it's not that great either. But it's a second one, and it works differently, so that's better than nothing; right? Everything about Passkeys is superior to everything that has come before it. Well, okay, except for SQRL, but that argument is now academic.

However, while everything I just said is true in theory, it may not, interestingly enough, be completely true in practice. As we know, security is always about the weakest link in the chain, and that's every bit as true for authentication. If a Passkey is set up for website login, that system of authentication will be the most bulletproof solution mankind knows how to design today. But if that website also still supports username and password authentication, with or without any second factors, then that username and password fallback will reduce the entire system's effective security, the fact that it's still there.

The problem is that, while Passkeys are actually vastly more secure than username and password login, they're primarily seen as being more convenient. That is, Passkeys; right? They're vastly more secure than username and password login, but most people just see them as more convenient.

MIKAH: Yeah.

Steve: In other words, it's not necessarily the case that enabling the use of a Passkey will simultaneously and robustly disable all use of any preceding username and password. And that is what we really want. If Passkeys are enabled, we want usernames and passwords to be disabled. If you've never used a website which asks you right off the bat to create an account for the first time ever, and offers to let you use a Passkey with no - never giving it a username or password - you're golden since you won't have ever given that website any secrets to keep because Passkey doesn't. It cannot disclose what it doesn't know.

The question is, will websites that start allowing the use of Passkeys also allow their users to then erase all traces of, or firmly disable the use of, their previous username and password for login? Because if they don't, the use of Passkeys is only giving them, the user, more convenience. It also is more secure in the transaction than a username and password because it can't be intercepted in the same way that a username and password can on the fly. But it's not the absolute security that it could be if it's still possible to compromise the old-school login.

Okay, now, this problem was something that my design of SQRL anticipated. After its user had become comfortable with the way SQRL operated and had printed out the necessary identity backups, they were able to turn on a setting that requested every site they subsequently visited to please disable any and all alternative means for logging in, including specifically the email loop fallback. You know, because I've been doing security for a while, I understood that this is a matter of security only being as good as the weakest link.

Okay. So here are our takeaways from this: The crypto technology which is well hidden inside Passkeys totally blows away both username and password and even multifactor authentication, which all rely upon server-side secrets being kept. That's their universal weakness, which is what makes Passkeys not only more convenient, but also far more secure. So whenever given the choice, set up a Passkey without a second thought and without looking back.

Secondly, look around to see whether there's any way to disable any other previous less secure logon fallback. If you cannot disable username and password login, but you are able to strengthen it with MFA, then go ahead and do that. Adding MFA is still the best way to make username and password login safer against attack, even if you're not using your username and password any longer because you've switched over to Passkeys.

But one thing you can definitely do is remove your old username and password from your password manager. And if for some reason you cannot remove it, change it to something bogus, like a long string of pound signs. That will be your sign that this account uses Passkeys. And by removing your password manager's storage of your account's true password, you're protected from any compromise of that password manager or its cloud backup provider.

The big takeaway here is that, even though they may not look all that different from the outside, the technology underlying Passkeys makes them an unreserved win over everything else that has come before. Now that our password managers are fully supporting Passkeys natively, which is what we've been waiting for, whenever possible,

make the switch, and then follow up by doing anything you can to remove the use of the username and password login that preceded it.

And Stephan, thank you for the question. This has obviously been an issue that's needed some time and clarification, and I'm delighted that we were able to give it. Passkeys really do rock.

MIKAH: Wow. Okay. So I have to ask you, how does this compare then - because I was also thinking another thing that sites will often provide you with are those special-use one-time codes that you're supposed to save, that they also are storing obviously, that could then be grabbed off of the site if it was hashed and somehow they got access to it or whatever. But what about if I stick my little key into the side of - my hardware key. Is that somewhere in between where Passkeys exist, as the multifactor authentication option, I mean? I take my little YubiKey and put it into the side and touch it. Does that provide a little bit more security than using those one-time codes?

Steve: Yes, much. Assuming that it's a FIDO key, and that's the key. That's the key, so to speak. Assuming it's a FIDO key, then it is essentially doing what Passkeys is doing. What Passkeys is, is technically FIDO2. And the reason FIDO2 happened was that the uptake of the hardware dongles to do FIDO1 was just too slow. People, you know, again, only one third of the Internet are using password managers? Two thirds are still using monkey123, or maybe 456? I mean, that just shows you that this is a heavy lift.

So people just don't really give that much concern to their login security. So they're not going to go out and buy, you know, spend any money to buy something that they just think, well, you know, what the heck? I'll just use monkey. Why do I need a key? So as long as that dongle is doing FIDO login, and if it's at all recent, that's why you have it, that's what it's doing, then it is performing the same sort of public key transaction that Passkeys does.

MIKAH: Got it. And you brought up a really good point, and then we'll say goodbye. But I just, I really liked what you said earlier about how we view it as so convenient, and that alone makes us maybe go, oh, I don't want to use this, it's too easy. That must mean that it's not as good.

Steve: Right. We don't get to be a secret agent.

MIKAH: Yeah. It sort of sits, like, is it secure? Because all I did, I just did it this morning, I needed to relog into a Google account that the session had expired, and I popped up to say use my Passkey, and it just went right in. I'm like, man, this is just so easy. It's so much simpler. But yeah, we think, oh, but how come I didn't have to whisper something into the phone and scan my retina and do all this other stuff? It can't be as...

Steve: Tap your heels three times.

MIKAH: Yeah, exactly, exactly. Steve Gibson, I want to thank you for an information-packed episode. I certainly learned a lot today.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/