Credential Exchange Protocol

Description: Did Chinese researchers really break RSA encryption? What did they do? What next-level terror extortion is being powered by the NPD breach data? The EU to hold software companies liable for software security? Microsoft lost weeks of security logs. How hard did they try to fix the problem? The Chinese drone company DJI has sued the DoJ over its ban on DJI's drones. The DoJ wishes to acquire deepfake technology to create fake people. Microsoft has bots pretending to fall for phishing campaigns, then leading the bad guys to their honeypots. It's diabolical and brilliant. A bit of BIMI logo follow-up, then a look at the operation of the FIDO Alliance's forthcoming Credential Exchange Protocol which promises to create passkey collection portability.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-997.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-997-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We have things to talk about. Did Chinese researchers really crack RSA? This might be a problem of headline confusion. The DoD is being sued by DJI over their drone ban. And a look at the new plan to allow you to move your passkeys from one place to another. All of that's coming up and a lot more, next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 997, recorded Tuesday, October 22nd, 2024: Credential Exchange Protocol.

It's time for Security Now!, the show where we cover the latest security news, privacy issues, breach news, exploits, CVEs, and science fiction, not necessarily in that order, with this guy...

Steve Gibson: And a little bit of health...

Leo: A little health thrown in...

Steve: Yes, just for good measure.

Leo: With this guy right here, Mr. Steve Gibson of the Gibson Research Corporation. Hey, Steve.

Steve: Leo, we are at 997. Yes. It hadn't occurred to me until, you know, we got close enough for it to be obvious that 999 is on Election Day.

Leo: Oh. That's appropriate. Holy cow.

Steve: It might have been the end of the podcast.

Leo: We'll drive off the cliff together.

Steve: Had it not been clear that, no, we're going to sail on through it and keep dealing with...

Leo: Well, it still might be if I move to New Zealand suddenly. But I don't think that's going to happen.

Steve: Well, Kevin Rose, I heard him talking to you a long time ago, like, I mean, a surprising long time ago, saying that he was, like, pushing his visa along.

Leo: If you've got millions, as Kevin does, it's good to have a hidey-hole. And the billionaires, apparently, they did think New Zealand was a place to go. Although their government has changed, and they may be not quite as organized as in the past.

Steve: As welcoming with open arms? Yeah.

Leo: Yeah.

Steve: Okay. So here we are on the 22nd of October. And as planned, we are going to talk about the Credential Exchange Protocol which was announced eight days ago during the FIDO Alliance's conference held a little bit south of me, actually, in Carlsbad, Southern California. At first when I saw the spec I thought, oh, well, there's not enough here to talk about because it's kind of more of an outline. Actually we'll have some fun with what the spec doesn't say in a minute. But when I got into it more, I realized that the meat of it was present, and there's enough to, like, make it the podcast.

So I at first renamed the podcast Credential Exchange Protocol Preview, thinking that that's all we were going to be able to do. But, no, we're going to be able to cover it. But, oh, we've got - and I should also say that I was hoping that this week I would be able to share more feedback because I'm getting just so much great listener feedback to securitynow@grc.com from those who have registered their incoming email with GRC's email system that I was wanting to share it. But, oh, is there some amazing news that we have, it just took up too much room.

Leo: Do you remember for a while we would do - we would alternate news episodes with Q&A episodes.

Steve: Yeah.

Leo: But there is just too much going on in security these days.

Steve: Yeah, there is. So we're going to answer the question. We touched on it last week, but I wanted to give it a little more attention, whether Chinese researchers did successfully break RSA encryption, as all of the tech press headlines covered.

Leo: Ah, the quantum [crosstalk], yeah.

Steve: You know, what did they do? Also, what next-level terror extortion is being powered by the NPD breach data. I actually had a buddy of mine send me something that he received, and it was - it's worth talking about. Also the EU is apparently going to be holding software companies liable for their lack of security.

Leo: Interesting.

Steve: In other words, liable for damages arising from software in a no-fault fashion, meaning even if they weren't aware of the problem. So that's, I mean, that's a sea change for the software industry. Also, Microsoft lost weeks of security logs. How hard did they try to fix the problem? The Chinese drone company DJI has sued the DoJ over its ban on DJI drones, which is interesting. Also, it turns out that the DoJ wishes to acquire deepfake technology to create fake people, complete with identities. And this is where it's like, what could possibly go wrong? Also Microsoft has bots pretending to fall for phishing campaigns and then leading bad guys to their honeypots.

Leo: Oh, good.

Steve: Which is diabolical and brilliant.

Leo: Yes.

Steve: So I just love this. And we've got a little bit of BIMI logo follow-up from the two pieces of listener feedback that I did manage to squeeze in before we take a look at this operation of the FIDO Alliance's forthcoming Credential Exchange Protocol which, as we know, the whole goal of it is to create passkey collection portability among passkey providers. So another jam-packed, I think really interesting episode for our listeners.

Leo: Yeah. I don't know about you, but the new pig-butchering scam is not just to say hello, although I still get those, and I got one with a picture of a Chinese girl saying "You remember meeting me?" So I get those, and of course immediately - but the latest ones, and I think these are probably very effective, Lisa's started getting them, too, are job offers. They're head-hunters. And, now, I obviously am not looking for work. But I think if you're a young person looking for work you might very well fall for these.

Steve: Yup.

Leo: So I am thinking that no head-hunter is going to text message you cold and say we have a job we think you'd like. If you do get that text message, I would really think twice before responding to it because it's probably just pig-butchering. It is for me, I know, because nobody's trying to hire me.

Steve: And anybody who hired me would regret it pretty quickly.

Leo: Yes.

Steve: When is that going to be ready, Steve? Ah, well...

Leo: We're not well-suited to being employees, either one of us, are we. Steve, I am prepared to demonstrate, to show to the world the Picture of the Week. I'm scrolling up now.

Steve: A useful analogy about the whole Zero Trust change is just the evolution in thinking about how a firewall should work. The first firewalls were open, and they blocked known problems.

Leo: Right.

Steve: And it became pretty quickly clear that that was the wrong strategy.

Leo: Right.

Steve: We need it to be closed by default for everything, and then selectively open ports...

Leo: Whitelist instead of blacklist.

Steve: ...for [crosstalk] we knew we wanted.

Leo: It's really cool. And it's such a simple concept, and yet it's so effective. All right. You have a title for the Picture of the Week.

Steve: So, yes. I gave this picture the caption "Generic Accessibility Requirements May Not Always Produce an Appropriate Outcome."

Leo: Okay. Oh, my god.

Steve: So what we have is a warning sign that says, "Hot Surface, Do Not Touch." And due to the need for unsighted people to be able to read the signage also, below it is braille. Which of course poses a problem...

Leo: Don't touch that, yeah.

Steve: ...for a hot surface warning sign. Now, this sign actually has an interesting history because, once again, the email for today's podcast, I was able to - I got everything wrapped up at the end of the evening and sent the email out last night. So at this point I think it was 11,314 recipients of the show notes, the summary of the podcast, and a thumbnail of this picture that you could click on to get full size. They all received that last evening, so I'll just remind our listeners that that's available to anybody who wants to subscribe to the Security Now! list. A couple people said, I know what that's supposed to be, but that's not actually braille. And so I know why. This actually came from a listener who submitted a photo of this sign, well, a sign which had this. It said "Hot Surface, Do Not Touch," and then it actually had a line of true braille along the bottom.

Leo: Oh.

Steve: The problem was it was a photo taken off-axis in bright sunlight, so it was washed out, and it had like a big shining reflection of the sun on it.

Leo: It was probably pretty hot, actually.

Steve: And so it did make, well, it did make a great standalone photo. So I had this really cool perspective correction software, so I fixed the perspective, and it looked fine, but it still didn't look great. So I thought, I wonder if AI can come to my aid.

Leo: This is AI generated?

Steve: Yes.

Leo: Wow.

Steve: And ChatGPT now has an image facility.

Leo: It didn't used to be able to do text at all. This is remarkable. You know?

Steve: So I said, could you take a sign and, like, improve the contrast and make it more legible, or something like that. And it said, yeah, happily. And so it thought about it for a minute, and it came up with a completely different sign. And, I mean, it was like as if I'd said, here's an idea, run with it, which is not what I said at all. And so but I've learned from my friend that it really helps it to be polite. So I said, wow, that's really great, but could you make it look...

Leo: Not what I was looking for.

Steve: ...a lot more like the original one that I uploaded? Oh, sure, I'd be happy to do that.

Leo: And it did.

Steve: Well, no. And I got this, which is still not what I started with, and it's also not braille.

Leo: Right.

Steve: But it's the concept. So although, Leo, I have to say I'm becoming astonished by what I'm seeing this AI stuff can do.

Leo: Well, that's a pretty good example, yeah.

Steve: I'm not a super experienced SQL coder, nor do I really program in PHP. But over the weekend there was a chunk of code that I got from - that is in the email system that I'm using. And, you know, I mean, as we know, once you understand procedural languages, they all pretty much look the same. You need to, you know, know how you do not equals, which varies from language to language; and, you know, do I put semi-colons at the end of each line or not, that kind of stuff. But so I can see what it is doing.

But it was, as always, you know, because I prefer to code in assembler, I'm wanting not just a solution, but like the absolute optimal solution. And it was doing something with SQL statements where it was doing late binding to prevent against injection attacks. And I wanted to know how much of what it was doing I could reuse for a subsequent query without having to do all of the early stage setup. So, and again, pre-AI I would have, you know, googled, right, for like I don't know, half an hour, poking around, getting an understanding of each of these statements and exactly what context they require and how much they leave behind and blah blah blah blah.

I thought, okay, I'll just ask ChatGPT because it also now has a coding, an explicit coding assistance. They call it Canvas. And so I went there, and I copied the statements that I wanted to understand in detail. I removed some of the superfluous stuff. And I pasted it in. I said, could you explain to me if I want to make another query of the same sort, you know, what this all does and how much of it does not need to be repeated. I'm just astonished.

Leo: It's really remarkable now.

Steve: I mean, every single, I mean, it was a course that it just dumped out where every single statement it explained what it was doing and then answered my question, which was, of all of that, how much was setup that I did not need to repeat when I wanted to reissue the query with a couple different parameters. I mean, and I just thought, okay, this actually, I mean, now, I'm not asking it to help me with my assembler code because I'm, you know, I'm a fish in water there.

Leo: It might be able to, though, Steve. I'd give it a little test just to see.

Steve: Nah, nah. It's not even interesting. But here, I mean, it really, you know, this was where I wanted a quick answer without in-depth studying.

Leo: Right.

Steve: So like without going and spending the time to dig through all of the individual definitions.

Leo: I've used it for regular expressions, and it's very good at interpreting regular expressions.

Steve: Oh, that would be good, too, yeah.

Leo: Very much like SQL queries. But I think the key with AI is you have to know what its limits are. And it really doesn't work by itself. But when in conjunction with a human, it can be very useful. You just have to know what you're doing and know what its limits are and not say that it's, you know, going to take over the world.

Steve: Yeah. I used it for some VB script a couple months ago. Again, not a language that I spend all my time in. And it gave me something that looked good and did not work.

Leo: Yes.

Steve: So, but I saw where the error was.

Leo: Yeah, it's a starting point.

Steve: And I thought, okay. Then I was able to fix it. So...

Leo: Yeah, it's a starting point.

Steve: Anyway, it is, I have to say it's - for some things it just - it's a time saver. So...

Leo: Yeah, yeah.

Steve: And I'm normally not looking to save time. But in this case it's like, okay, I just want to get this out of my way, so.

Leo: Yeah.

Steve: Okay. So I know from having created and written InfoWorld Magazine's TechTalk column for eight years, that the way things work in publishing, the authors of columns and news articles have absolutely no control over the title given to their work. Why that is true is something I've never understood. I complained about it back then when I was writing the column, and I was told, no, you don't do that, we do that. And it's like, okay. So it's just the way it is. And as I said, I can't begin to tell you how many times I was distressed to see the headline one of my carefully thought out and crafted columns was given after it left my control and headed to the printer turned out to have. You know, it often, I kid you not, the headline bore no relationship to what I had written. It was just so annoying.

And with that understanding, I can forgive the well-meaning author of a piece that appeared last Monday the 14th in CSOnline. The headline of that piece could not possibly have been any more misleading than it was, so I can only imagine what its author thought when they saw it in print. The incredibly provocative headline in question read "Chinese researchers break RSA encryption with a quantum computer." Did that happen? No. It didn't even remotely happen. It wasn't and still isn't even remotely close to happening, and there's no way to characterize what did happen as having "broken" RSA encryption. You know, "breakage" in cryptography has a very specific bone-chilling meaning, and this isn't it.

Okay. So fortunately, to regain some sense of order to the universe, one only needs to read past that deliberately fictitious headline to the first sentence of the actual article, which says: "The research team led by Shanghai University's Wang Chao, found that D-Wave's quantum computers can optimize problem-solving in a way that makes it possible to attack encryption methods such as RSA." Now, not nearly as catchy as "Quantum computers have broken RSA encryption." You know, basically phrased another way: "A team" - which is what happened - "of very clever Chinese researchers discovered a better way to employ some characteristics of D-Wave's quantum computers against the prime factoring problem that lies at the heart of RSA's encryption protection." Unfortunately, you know, as I said, the truth of the discovery makes for a much less exciting headline.

Through the years of this podcast we've talked a lot about the strength of RSA encryption which lies entirely in the still surprisingly, and thankfully intractable challenge of factoring extremely large - and when I say "extremely large" I mean "humongous" - numbers into their two prime factors. The basis of RSA's extremely clever system is that we first choose a very large, as in 4,096-bit large, you know, huge prime number at random, which turns out to be easier than you might expect. There's lots of them out there. That's our private key.

Then we hide that private key by choosing another similarly large 4,096-bit prime number and then multiply those two primes to obtain an 8,192-bit (two times 4,096-bit) product. The product of those two primes is the public key, inside of which is hidden the private key. So if it were possible for some computer system to factor that even more massive 8,192-bit public key, then that original private key that was hidden inside the public key could be revealed, and RSA's protection would then actually be in trouble. And we use this encryption everywhere. So yes, it would kind of be the end of the world.

The Chinese researchers explained in their paper, they said: "Using the D-Wave Advantage, we successfully factored a 22-bit RSA integer, demonstrating the potential for quantum machines to tackle cryptographic problems." That's all they said. "We successfully factored a 22-bit integer." So, news flash, quantum computers can be used to factor integers. Very small integers, at this point. And if memory serves, the last time

we looked at this a few years ago, other researchers were announcing their breakthrough by factoring a much smaller number. I think it was like they factored 13 or 11 or something. I mean, like the number 13 or 11. So 22 bits, that's a much bigger number. I have no doubt that this represents a significant discovery and, yes, another breakthrough in the application of quantum computer technology for breaking cryptography.

But at today's strength where the public key that's the thing that needs to be factored in order to retrieve the private key hidden inside it, 8,192 bits is what you would need to factor. So practical RSA factorization protection still appears to be entirely safe. At the same time, these sorts of breakthroughs are what make cryptographic researchers nervous, which is why it's a good thing that our industry has already designed and is already deploying so-called post-quantum algorithms that no longer rely upon the protection offered by the factorization problem. And in fact what they do is believed to be completely intractable by quantum computing technology.

You know, and we talked about this before in the case of, for example, the Signal messaging application. They're already quantum-safe. But because these new quantum-safe algorithms are still new and unproven, Signal took the belt-and-suspenders approach of using both the old and time-proven, as well as the new and hopefully safe, but still not yet time-proven, algorithms at the same time. In that way Signal's users are already protected because the possibility of some true breakthrough in the use of quantum computers is there. But even if that happened, we would still have the fallback of traditional crypto. Even if quantum computers were able to crack one family of crypto, they're using both new and old.

So anyway, I got swamped with email, not surprisingly, from our listeners who saw this headline. And of course it got picked up and echoed around the industry. Oh, my god, you know, the Chinese quantum computer researchers have broken RSA crypto. No. Didn't happen. This still, I mean, this is the way it's going to go; right? It's going to get chipped away at. Next generations of quantum computers will be able to increase the strength of this. Hopefully we will have moved, we will have migrated to post-quantum technology by that time. And so when it eventually does happen, nobody will be using this technology any longer. So it's certainly foreseeable that that's the case.

Okay, now, this happened over the weekend. A buddy of mine forwarded a scam PDF that had arrived in his email. But the opening line of this particular scam is what caught my attention and thus made it into today's podcast. Although his email name, you know, his email account does not have any aspect of his name in it, the PDF was correctly addressed to him with his full, correct first and last name. And I'm going to read, like, the first third of it to give you a sense for it. So it was addressed to him, you know, first name, last name, comma.

And it read: "I know that calling," and then it had his accurate phone number, area code, phone number. "So know that calling," and there's his phone number, "or visiting you at," and then it had his full current residential street address, "would be an effective way to contact you in case you don't act. Don't try to hide from this. You have no idea what all I can do in," and then his city of residence.

Leo: I get this exact email daily.

Steve: Okay. I had not seen it before.

Leo: No, and it's a PDF that's attached to the email.

Steve: Yes.

Leo: I'm not sure why that is, either.

Steve: Exactly. It's a PDF that is attached to this. He was terrified. You know, and then it goes on with the standard, you know, how horrible you are, all of your videos that you've been watching...

Leo: Oh, it's BS. Yeah.

Steve: ...and being recorded and blah blah blah. But for me what stood out - oh, and it finally ends up telling him that the only way to prevent this from being sent to all of his friends and family and contacts and social media accounts, all of which this cretin alleges to have, is to pay $2,000 to a bitcoin address.

Leo: We actually - I was actually making fun of it because our local newspaper, the Santa Rosa Press Democrat, had a "Three Santa Rosa residents have been fooled by this scam." I thought, doesn't everybody get these emails all the time? I mean, you don't get these? I get them all the time.

Steve: I've never seen this. And...

Leo: Here's one. I'll show you one. I could show you because it's a wrong - the address is an old address. I suspect this whole thing now is prompted by maybe the NPD leak. I don't know.

Steve: Well, that's where - that's exactly where I'm headed with this.

Leo: Yeah. Here, let me show you mine. I mean, this is - and you can see because look at the email address. Shawna Nelly XDFT, it's a completely fabricated email address; right?

Steve: Yup.

Leo: This is - I can show this because it's not my current address.

Steve: Right.

Leo: And this is exactly what you're talking about.

Steve: That is the email.

Leo: Yup. I get this daily, Steve.

Steve: Okay, I had never seen it. He had never seen it. So for me, this being new, what was very clear was that this was being driven, exactly as you said, by the fact that all of this data is now public. And I guess, you know, for me what really yanked my heartstrings is the idea of how many people are truly going to be terrorized by this. And again, obviously you're not, Leo. But I am absolutely sure that when people get this for the first time...

Leo: Oh, yeah.

Steve: ...and they see their name, their phone number, their physical address, which they see, I mean, they don't know about the National Public Data breach. They still imagine, they have this illusion of privacy that, like, they have any privacy now...

Leo: It's a total illusion.

Steve: ...in the online world. And so they don't get it that this is some cretin, you know, in Russia or North Korea who knows absolutely nothing about them, that has no ability to physically intimidate them at their residential stress address, which they do have as a consequence of these data breaches. Anyway, I just think that - I'm glad that the newspaper is talking about this.

Leo: We should all be, yeah.

Steve: Yes. I really think that, you know, it would be a public service announcement to make sure that everyone understands that this is where we're headed, that our data, I mean, you know, I suspect that the NPD breach was an example of this.

Leo: This is probably from Street View, Google Street View, I would guess, this picture.

Steve: Oh, so it actually even had a picture of your property.

Leo: Oh, yeah, yeah. Those online tips about covering your camera aren't as useless as they seem.

Steve: Wow.

Leo: So here's the giveaway to me. This is the same verbiage that's used when Chinese scammers say, "I have your iPhone, and you'd better take it off Find My iPhone." They also use this line, "You have no idea what I'm capable of in your town, Petaluma."

Steve: Right, right.

Leo: And that to me is a little bit of a giveaway. I'm going to say these are Chinese scammers, and this is the same bunch of people who do a bunch of this kind of pig-butchering stuff. It's really too bad. And yeah, I really fear for people like my mom, older people, yeah.

Steve: Exactly, exactly. Somebody who has never seen it before, again, who has this illusion of a private life.

Leo: They just don't know what the modern world is; you know?

Steve: Yeah.

Leo: They don't, yeah, you know, it's very sad. Well, I guess we should - it's too bad I don't do the radio show anymore. I made a habit of talking about these on the radio show, hoping to reach the general audience.

Steve: Good. Well, and so, and obviously you were in touch then with that kind of audience. And I'm sure you understood, I mean, they called up and said, oh, my god. And, I mean, so this is what people are going to do when they see this. And, like, there's their phone number and their street address.

Leo: If you read it, it's terrifying.

Steve: Yes, it is. It is. And there is no need to read it because a lot of people have seen these before. But it is, it is absolutely, you go through this. And again, it is terrifying. So I was - I had never seen that, all of that information.

Leo: "Been keeping tabs on your pathetic existence for a while now. It's just your bad luck I discovered your bad deeds."

Steve: Yes. "And I've got footage of you doing filthy things in your house. Nice setup, by the way."

Leo: Yeah.

Steve: And, I mean, if somebody read this, they would - and again, and they didn't know better.

Leo: Yeah. That's the problem. Unfortunately, this is the world we live in now. That's what's really sad about this. This is just one of many. Somebody's saying they send it as a PDF to evade email scan detecting.

Steve: That's what I was sort of thinking, except that I thought, of all [crosstalk] is now opening PDFs now and looking inside.

Leo: Yeah. I don't know.

Steve: We're half an hour in, Leo. Let's take a break. And then I'm going to - we're going to talk about what the European Union just did, and it's big news for software product liability.

Leo: I feel like we've heard this, like we heard it was coming. I feel like we've talked about this before.

Steve: Well, this landed, and wait till you hear...

Leo: Oh, boy.

Steve: ...what they're going to try to do.

Leo: Oh, baby.

Steve: It's such a big deal, I can't believe it's going to happen.

Leo: Good.

Steve: I mean, it's too big a change.

Leo: Good.

Steve: Okay. So as our long-time listeners know, one of this podcast's longest standing observations has been over the distortion in the software industry created by software license agreements that universally disclaim any and all responsibility for any consequences of the use and operation of the software. The wheels don't fall off of cars which we drive only because it would be the end of any automaker whose cars' wheels did fall off, because the rigid enforcement of product liability would end that company's existence overnight.

But that has never, bizarrely, been the situation in the software business where software users have no choice other than to contractually sign away all of their rights in a software license agreement in return for the privilege of using the software, regardless of its quality. It's like, you know, hey, if you don't want to use it, fine, don't sign this. But if you agree, then we're not making any representations about the product's quality or its fitness for any particular purpose. That language is in all of those license agreements.

So our listeners also know that I 100% understand that mistakes happen, and that the perfect operation of a complex software system can be impossible to achieve. But at the

same time, through the years of this podcast we've examined instance after instance of the consequences of deliberate policies - not mistakes - that can only be characterized as enabling continuing egregious conduct on the part of some software producers. This conduct and the policies that enable it are explicitly protected by the license agreements under which software is used. And I've also often wondered here when and how this will change because it feels like it's wrong the way things are today.

Well, change may be coming. I don't know what to make of this next piece of major earthshaking news because the changes that the European Union proposes to make in its product liability laws to explicitly include software liability, while at the same time eliminating software licensing exemptions, seems too radical to actually occur. But it has actually happened. So anyway, time will tell. And the fact that this is moving into law certainly means something, even if it doesn't happen immediately or at full strength. And I should note that it doesn't come into effect for 24 months, so that gives some time for something to happen. I'm not sure why they installed this two-year time delay. But we're going to find out.

Okay. So let's back up a bit and explain what's in the works. The first clue that I had about this was from the first news item in the Risky Business most recent newsletter. Here's what it describes, and listen to this carefully because this is it. They wrote: "The European Union has updated its product liability law to cover software and associated risks, like security flaws and planned obsolescence. The new EU Directive on Liability for Defective Products replaces one of the EU's oldest directives and will provide consumers with the legal tools to hold companies liable in court if they sell defective products.

"The biggest change to the old directive is the addition of software products to the list of covered goods. Companies that sell or want to sell in the EU will have to make significant changes to how they are currently doing business if they have failed to invest in proper software development and cybersecurity practices. The new directive extends liability to vendors for software that contains security flaws [wow] where those flaws lead to any damage to consumers. This includes both physical damage caused by defective or insecure software, but also material damage, such as loss of functionality and features, loss of financial assets, and others.

"The directive also classifies the lack of a software update mechanism to be a product defect and makes the vendor liable. Software vendors are also forbidden to withhold information about a software update's negative impact. The only exemption in liability coverage is when the software update requires the consumer to manually install an update. But generally the directive sees vendors liable as long as they have control over their product after a sale. The directive also extends liability to vendors who use any type of planned obsolescence system to artificially reduce the lifespan of their products." And I have to say some of this read like, you know, touching on the fringe of some of the things that we've seen Apple doing over time.

Leo: Yup, yup.

Steve: They said: "This includes software designed to slow down a device, hardware components engineered to fail after a certain period, or an update that degrades a software's performance..."

Leo: It's totally aimed at Apple. That's hysterical.

Steve: Yes, "in order to entice users to move to a new service, tier, or product. Companies can also be held liable for misleading consumers about a product's durability, repairability, or expected lifespan. The directive requires victims to prove a product's defectiveness, but it also adds a new legal mechanism to force vendors to make required evidence available. The new rules exclude free and open-source software..."

Leo: Ah, good.

Steve: Uh-huh, "...from its requirements. The new directive was approved earlier this year by the EU Parliament and earlier this month by the EU Council. It is set to go into effect in 24 months, in the fall of 2026."

Okay, now, I trust Catalin's reporting, but I needed to see this for myself, and our listeners need to hear this. So I found the 63-page document from the EU, and I've got the link to it there in the show notes at the bottom of page 5, Leo. And as far as I can see, he did not get anything wrong. Okay. So I'm just going to pick and choose a couple of paragraphs from the whole document to give everyone a taste of this.

After a bit of explanation about how and why the very old previous Directive is no longer useful, this new Directive explains that, rather than attempting to edit and amend the old one, it is being replaced in its entirety by this new Directive. And that brings us to paragraph 6, which says: "In order to ensure that the Union's" - European Union - "the Union's product liability regime is comprehensive, no-fault liability for defective products should apply to all movables, including software, including when they are integrated into other movables or installed in immovables."

Leo: What is a movable? Like a phone? A car?

Steve: They actually describe it, I think it was earlier, but they were saying including software, which is what I keyed on. And just so everyone is clear about the legal definition of "no-fault liability," an example I found online says: "No-fault liability is the legal responsibility to compensate someone for an injury, even if you were not negligent or at fault. For example, if you own a dangerous animal, and it hurts someone, you're responsible for their injuries, even if you didn't mean for it to happen."

Okay. So it's clear that from the standpoint of a software publisher, unintentional damage will not waive their liability under this new Directive for any damage it may cause. Paragraph 13 explains: "Products in the digital age can be tangible or intangible. Software, such as operating systems, firmware, computer programs, applications, or AI systems" - and by the way, AI also figures heavily here - "is increasingly common on the market and plays an increasingly important role for product safety. Software is capable of being placed on the market as a standalone product or can subsequently be integrated into other products as a component, and it is capable of causing damage through its execution.

"In the interest of legal certainty, it should be clarified in this Directive that software is a product for the purposes of applying no-fault liability, irrespective of the mode of its supply or usage, and therefore irrespective of whether the software is stored on a device, accessed through a communication network or cloud technologies, or supplied through a software-as-a-service model. Information is not, however, to be considered a product, and product liability rules should therefore not apply to the content of digital files, such as media files or eBooks or mere source code of software. A developer or producer of software, including AI system providers, should be treated as a manufacturer."

And this is followed by paragraph #14, which fully exempts open source software. It reads: "Free and open-source software, whereby the source code is openly shared, and users can freely access, use, modify, and redistribute the software or modified versions thereof, can contribute to research and innovation on the market. Such software is subject to licenses that allow anyone the freedom to run, copy, distribute, study, change, and improve the software. In order not to hamper innovation or research, this Directive should not apply to free and open-source software developed or supplied outside the course of a commercial activity, since products so developed or supplied are by definition not placed on the market.

"Developing or contributing to such software should not be understood as making it available on the market. Providing such software on open repositories should not be considered as making it available on the market, unless that occurs in the course of a commercial activity. In principle, the supply of free and open-source software by non-profit organizations should not be considered as taking place in a business-related context, unless such supply occurs in the course of a commercial activity. However, where software is supplied in exchange for a price, or for personal data used other than exclusively for improving the security, compatibility, or interoperability of the software, and is therefore supplied in the course of a commercial activity, this Directive should apply."

Then we have the question of products that are enhanced by or dependent upon external services. Where does liability lie then? Paragraph 17 says: "It is becoming increasingly common for digital services to be integrated into, or interconnected with, a product in such a way that the absence of the service would prevent the product from performing one of its functions. While this Directive should not apply to services as such, it is necessary to extend no-fault liability to such integrated or interconnected digital services as they determine the safety of the product just as much as physical or digital components. Those related services should be considered components of the product into which they are integrated or with which they are interconnected, where they are within the control of the manufacturer of the product.

"Examples of related services include the continuous supply of traffic data in a navigation system, a health monitoring service that relies on a physical product's sensors to track the user's physical activity or health metrics, a temperature control service that monitors and regulates the temperature of a smart fridge, or a voice-assistant service that allows one or more products to be controlled by using voice commands. Internet access services should not be treated as related services, since they cannot be considered as part of a product within a manufacturer's control, and it would be unreasonable to make manufacturers liable for damage caused by shortcomings in Internet access services. Nevertheless, a product that relies on Internet access services and fails to maintain safety in the event of a loss of connectivity could be found to be defective under this Directive."

And, finally, I was thinking about the exclusion that is always present in license agreements, which as we know has been a hobbyhorse of mine. This addresses that directly. Paragraph 56 of the legislation says: "The objective of protecting natural persons would be undermined if it were possible to limit or exclude an economic operator's liability through contractual provisions. Therefore no contractual derogations should be permitted. For the same reason, it should not be possible for provisions of national law to limit or exclude liability, such as by setting financial ceilings on an economic operator's liability."

Okay, now, not being trained in the law, I cannot render any opinion about the eventual impact of what the European Union has just done. But I can read. And what should be abundantly clear is that a sea change of some sort is coming to the product liability side of the software industry, at least as it applies in the European Union.

Even if this is met with a great deal of industry pushback, and it's difficult to imagine that it won't be, it appears that the past half-century of software publishing operating with impunity in a world without accountability or consequences may be approaching its expiration date. Over the past 50 years, software and the Internet have gradually grown to become truly mission-critical. But many older aspects of the way things have always been done have remained in place due to, you know, inertia, and no immediate forcing of change. Newer tools have been created that could enable software to be, and we've talked about this, significantly more robust than it is today. But programmers still choose to recklessly code in crazy, unsafe and unmanaged languages like C and Assembly. Imagine that.

You know, we've seen reports of major projects being deliberately recoded in fast and safe languages which will at least be able to deal with ridiculously persistent errors, such as use-after-free, that keep causing problems and continue to plague today's code. But these deliberate and expensive recoding efforts remain, you know, they are far and few between exceptions. It needs to become the norm. So it may be that legislation such as the EU has just put into place, having a 24-month grace period before it goes into effect, will up the ante and finally induce serious consideration of how future coding should be accomplished to reduce the incidents that might subject its publisher to warranted product liability claims.

And, you know, I just dissed two of my favorite languages. Let me be clear, it is entirely possible to write safe and secure code in C or Assembly.

Leo: Of course.

Steve: It's just far more expensive to really do so.

Leo: Yeah. Yeah.

Steve: You know, the flight computers controlling both the American Shuttle program and the two Voyager space probes, they were hand-coded in assembly language, and they both proved to be extremely reliable accomplishments. It all boils down to economics.

Leo: It's expensive, yeah.

Steve: We know that I write everything in assembly language, and that none of what I produce has ever had a problem with bugs. I rarely revise my final product other than to add new features. But I also have the unusual freedom of not having a boss and, more importantly, not writing under any sort of delivery deadline. That's not a luxury most of the world's coders enjoy. So for nearly everyone else, the thing that makes the most economic sense is using next-generation memory-safe languages. That's the only strategy that makes sense for keeping uncaught errors from turning into exploitable security vulnerabilities.

So I'm going to be keenly interested to see what comes of the EU's new software liability legislation. I mean, it is a big deal.

Leo: It's coming here, too. But this is part of the Biden administration's national cyber strategy they announced last year with software liability. And that's for security reasons as much as for, you know, liability reasons.

Steve: Well, look at the problems we've had, like Microsoft doesn't update that one old tenant, and China gets in and is in the U.S. government agency's email.

Leo: Yes. I mean, I'll never forget the first shrink wrap license I saw, which was probably for an Atari 800 in the '80s, and reading the lines "We make no warranty that this software is usable for anything, will do anything, is going to do what we say it's going to do. It's not our fault."

Steve: Complete disclaimer of all responsibility.

Leo: And I was kind of blown, it was like, wow, really?

Steve: It's astonishing.

Leo: Yeah. But I understand people don't have the confidence in software, and they never have. Didn't the DoD adopt Ada as an attempt to have a secure programming language that would be reliable? And what happened to that initiative?

Steve: I don't know. People were still programming in COBOL at the time.

Leo: At the time, yeah. Ada was supposed to be memory safe, memory hard. It was very strongly typed. I think programmers didn't like it because I was so strongly typed, it required a lot of boilerplate code, and they didn't really like doing that. That was my sense of it. But for whatever reason, I don't think it's widely used anymore.

Steve: No. I think there's no question that we have so much computing power now that we can afford to sacrifice some strict level of efficiency in trade for security, and in trade for using a language that protects the programmer. And, you know, if I were counseling people, and I know we have listeners in college and at high school level who are wondering what they should do. I would not - everyone argues that learning assembly language, for example, you know, which is basically machine language using mnemonics to make it more intelligible, is useful to really understand what's going on down at the hardware level, you know, in the computer. And I can't argue with that.

But if you want to get a job, and you want to be in demand, I'll bet you that the future is in being really up to speed on secure, safe computer programming. I think that's where we're going to head is, I mean, you know, initiatives like this are going to change - again, it's about economics. That's the driver. And a lot of inertia, too. And we know that, you know, the only way I'm going to quit programming in assembly is when I'm buried.

Leo: Yeah. And, you know, people are mentioning in the chatroom Rust, which is memory-safe, strongly typed.

Steve: Rust is what immediately comes to mind.

Leo: And a lot of people are choosing that, yeah, yeah, yeah.

Steve: Yup.

Leo: But there are other choices out there. I mean, this is definitely a movement among coding. People who write languages are definitely working on this.

Steve: And, you know, one of the things that we see, Leo, is these changes occur slowly. And so the industry's been dabbling around these things. It all began in academia where all kinds of wacky languages exist to explore the idea. It takes a long time for them to actually move from there into production. And you have to have people who know them. So I would seriously look at Rust or another language that's entire purpose is security because programming secure applications is coming.

Leo: Yeah. Good. It's about time.

Steve: BleepingComputer's headline was: "Microsoft warns it lost some customers' security logs for a month."

Leo: Whoops. Whoops.

Steve: Uh-huh. And TechCrunch reported under the headline "Microsoft said it lost weeks of security logs for its customers' cloud products." And since going to the source is usually best, I tracked down Microsoft's own report of this. Under the section of that titled "What happened?" they wrote - this is Microsoft who wrote: "Starting around 23:00 UTC on September 2nd, a bug in one of Microsoft's internal monitoring agents resulted in a malfunction in some of the agents when uploading log data to our internal logging platform." You know, okay, no one knows what any of that means, but it sounds good.

"This resulted in partially incomplete log data for the affected Microsoft services. This issue did not impact the uptime of any customer-facing services or resources. It only affected the collection of log events," which, you know, we call "putting a good face on it." They said: "Additionally, this issue is not related to any security compromise." Except as it would have been nice to have logs so you could detect security compromises, which you can't detect if you don't have logs. But the next sentence is the one that got me. "The issue was detected on September 5th. Following detection, our engineering teams began investigating and implemented a temporary workaround to reduce the impact of these failures beginning on September 19th."

Now, okay, those dates caught my eye. They say that the issue was detected on the 5th of September, and that their engineering teams began investigating and implemented a temporary workaround to reduce the impact of these failures beginning on September 19th. In other words, two weeks lapsed between their initial detection of this issue and their beginning to investigate and implement a temporary workaround. It sounds as though logging is not an urgent priority for them, though after all the problems they've had surrounding a lack of logging for their customers, one would really imagine that it might receive more attention. I guess not.

Okay. DJI sues the DoJ.

Leo: The DoD.

Steve: The what?

Leo: The DoD, not the DoJ. They're suing the Defense Department.

Steve: Oh, you're right, you're right, the Department of Defense, sorry.

Leo: Yes.

Steve: Yes. So DJI, the Chinese manufacturer of what are arguably the best small consumer drones in the world...

Leo: [Crosstalk] love them. Yeah, I have several, yeah.

Steve: Everybody does, has sued the United States Department of Defense over the DoD's listing of them as agents of the Chinese military. Reuters News Service carried the news which contained some interesting details. They wrote: "WASHINGTON, October 18th, (Reuters). China-based DJI sued the U.S. Defense Department on Friday for adding the drone maker to a list of companies allegedly working with Beijing's military, saying the designation is wrong" - that is, DJI is saying the designation is wrong - "and has caused the company significant financial harm." Yeah, no kidding.

"DJI," writes Reuters, "the world's largest drone manufacturer that sells more than half of all U.S. commercial drones, asked a U.S. District Judge in Washington to order its removal from the Pentagon list designating it as a 'Chinese military company,' saying it 'is neither owned nor controlled by the Chinese military.' Being placed on the list," they write, "represents a warning to U.S. entities and companies about the national security risks of conducting business with them. DJI's lawsuit says because of the Defense Department's 'unlawful and misguided decision,' it has 'lost business deals, been stigmatized as a national security threat, and been banned from contracting with multiple federal government agencies.'" Yeah, that would happen. "The company added: 'U.S. and international customers have terminated existing contracts with DJI and refuse to enter into new ones.'

"DJI said on Friday it filed the lawsuit after the Defense Department did not engage with the company over the designation for more than 16 months, saying it 'had no alternative other than to seek relief in federal court.' Amid strained ties between the world's two biggest economies, the updated list is one of numerous actions Washington has taken in recent years to highlight and restrict Chinese companies that it says may strengthen Beijing's military.

"Many major Chinese firms are on the list, including aviation company AVIC, memory chip maker YMTC, China Mobile, and energy company CNOOC. DJI is facing growing pressure in the United States. Earlier last week DJI told Reuters that Customs and Border Protection is stopping imports of some DJI drones from entering the United States, citing the Uyghur Forced Labor Prevention Act. DJI said no forced labor is involved at any stage

of its manufacturing. U.S. lawmakers have repeatedly raised concerns that DJI drones pose data transmission, surveillance, and national security risks, something the company rejects." And finally: "Last month, the U.S. House voted to bar new drones from DJI from operating in the U.S. The bill awaits U.S. Senate action. The Commerce Department said last month it is seeking comments on whether to impose restrictions on Chinese drones that would effectively ban them in the U.S., similar to proposed Chinese vehicle restrictions."

Okay. So we've talked about this previously, so this is not surprising. And this is one of those situations, I think, where it's entirely possible to see the logic being applied by each side of this argument. It cannot be argued that nothing could ever make a more perfect spying device than a camera-equipped flying drone. You know, they are by definition flying cameras, and DJI's are among the best. We previously talked about how DJI drones are being actively used within military bases in the U.S., and even on secret military bases. And DJI drones receive software updates. So it's theoretically possible for - again, theoretically - for the Chinese government to order DJI, a Chinese manufacturer, to alter their firmware so as to turn their drones into active spying cameras. And whether or not it's fair, "theoreticals" are what keep our military planners and our generals up at night.

The only way I can see for this to work would be for DJI to essentially create a wholly separate U.S. version of DJI as an independent U.S.-based division. DJI China could produce the drone chassis and all the hardware, which is where the majority of the cost and value lies. But the sole exception would be the drone's circuit board, which would be manufactured using U.S.-known components in the U.S. which have been sourced for that purpose. And that U.S. DJI drone control board would then be flashed with firmware that had been audited and inspected by technical representatives of the United States. DJI would need to establish camera footage uploading cloud servers in the U.S. without any ties to China, and the only connection would be the receipt of brainless drone chassis from China.

This would all obviously represent a huge burden and a cost for DJI. But I can't see reaching any other compromise. It's not strictly fair; but the danger, even if only theoretical, is so great that I think DJI will need to consider some sort of a solution along these lines, you know, if they want to keep the U.S. market. Unfortunately, you know, we're a big market for them, and tensions are on the rise between the U.S. and China. And not without cause. I mean, you know, how many times, Leo, have we talked about Chinese-sponsored cyberattacks on the U.S., you know, inside the U.S.

Leo: Right.

Steve: And so of course tensions are going to be high. And presumably we're giving as well as we get.

Leo: Right. I'm sure the Chinese would have no hesitation banning U.S. drones in the Chinese market, if such things existed. That's one of the reasons we don't make them in the U.S.

Steve: Well, and remember, historically China has been very unfair to U.S. importers.

Leo: Well, you know, one of these reasons, the drones took off shortly after the iPhone came out. I remember going to CES and seeing my first drones in the parking

lot of CES in the late 2000s, 2008 or '9. And the reason was we taught Chinese manufacturers how to make all these components, they started making them in quantity, like accelerometers, and then started putting them in their own products. And, I mean, that's ideally how things should work, frankly. It's really, it is, it's a real shame. Those DJI drones are amazing. They just released a brand new one that's 200 bucks and impossible to crash. I mean, it's just, it's very - but I also completely understand the concern because you're right, these would be perfect spy deals.

Steve: Absolutely, yeah.

Leo: And we don't normally upload to the cloud. I mean, you don't - you could disable that feature. That's not critical to their functionality. I don't know if that would make it better.

Steve: Well, and of course the problem would be...

Leo: They could do it anyway.

Steve: Yeah, well...

Leo: Because they are Internet-connected, yeah.

Steve: Exactly. In some way arranging to make that verifiably the case.

Leo: Right, yeah. It's challenging.

Steve: Let's take a break.

Leo: Yes, sir.

Steve: And then we're going to talk about the Department of Defense's operations command wanting to acquire sophisticated deepfake - actually, this is the DoJ this time - deepfake capability.

Leo: No. No.

Steve: I'm sorry, DoD, it is.

Leo: It is DoD, yeah.

Steve: I was stuck on the DoJ for some reason.

Leo: Well, all the DO's are, you know, they all overlap a little bit. So I understand that. Back to you, Steve.

Steve: Okay. The Intercept reports that our U.S. Department of Defense is in the market for sophisticated deepfake technology. The Intercept's headline was "The Pentagon Wants to Use AI to Create Deepfake Internet Users" with the subhead "The Department of Defense wants technology so it can fabricate online personas that are indistinguishable from real people." And once again I find the details of this quite interesting. Here's the start of The Intercept's coverage of this.

They wrote: "The United States' secretive Special Operations Command is looking for companies to help create deepfake Internet users so convincing that neither humans nor computers will be able to detect they are fake, according to a procurement document reviewed by The Intercept.

"The plan, mentioned in a new 76-page wish list by the Department of Defense's Joint Special Operations Command, or JSOC, outlines advanced technologies desired for the country's most elite clandestine military efforts. The entry reads: 'Special Operations Forces (SOF) are interested in technologies that can generate convincing online personas for use on social media platforms, social networking sites, and other online content.' The document specifies that JSOC wants the ability to create online user profiles that 'appear to be a unique individual that is recognizable as human, but does not exist in the real world,' with each featuring 'multiple expressions' and 'Government Identification-quality photos.'

"In addition to still images of faked people, the document notes that 'the solution should include facial and background imagery, facial and background video, and audio layers,' and JSOC hopes to be able to generate 'selfie video' from these fabricated humans. These videos will feature more than fake people. Each deepfake selfie will come with a matching faked background, 'to create a virtual environment undetectable by social media algorithms.'

"The Pentagon has already been caught using phony social media users to further its interests in recent years. In 2022, Meta and Twitter removed a propaganda network using faked accounts operated by U.S. Central Command, including some with profile pictures generated with methods similar to those outlined by JSOC. A 2024 Reuters investigation revealed a Special Operations Command campaign using fake social media users aimed at undermining foreign confidence in China's Covid vaccine.

"Last year, Special Operations Command, or SOCOM, expressed interest in using video 'deepfakes,' a general term for synthesized audiovisual data meant to be indistinguishable from a genuine recording, for 'influence campaigns, digital deception, communication disruption, and disinformation campaigns.' Such imagery is generated using a variety of machine learning techniques, generally using software that has been 'trained' to recognize and recreate human features by analyzing a massive database of faces and bodies.

"This year's SOCOM wish list specifies an interest in software similar to StyleGAN, a tool released by Nvidia in 2019 that powered the globally popular website 'This Person Does Not Exist.' Within a year of StyleGAN's launch, Facebook said it had taken down a network of accounts that used the technology to create false profile pictures. Since then, academic and private sector researchers have been engaging in a race between new ways to create undetectable deepfakes and new ways to detect them. Many government services now require so-called 'liveness detection' to thwart deepfaked identity photos,

asking human applicants to upload a selfie video to demonstrate they are a real person an obstacle that SOCOM may be interested in thwarting."

And of course this struck home with me because, as I shared last week, I was asked to hold my ID up next to my head and then move my hand around behind and in front of it while talking to a DigiCert person to verify my liveness.

So Leo, we are nowhere near Kansas. And we are also a long way from Mayberry. Wow.

Leo: Honestly, I mean, this is in response to the Russians doing the same thing; right?

Steve: Well, exactly. And as we know, North Korea has been signing up their own operatives and pretending to be domestic job seekers in order to infiltrate U.S. enterprises.

Leo: Yeah. I think a lot of this is for social networks. I mean, Twitter or X is full of Russian cutouts, pretending to be Americans, with fairly plausible identities. And I am sure that we're just trying to do the same thing right back to them.

Steve: Yeah.

Leo: It's, I mean, it's inevitable, I guess.

Steve: And we live in a society where, you know, the things that our government are doing like this is not top secret. It's like, yeah, well, you know, we put out a requisition saying this is the technology that the Department of Defense needs.

Leo: Help us.

Steve: Yup.

Leo: They're going to do this. Wow.

Steve: Okay. And I just love this next piece. While we're on the subject of things being faked, Microsoft is running a massive deception campaign that is providing phishing sites with fake credentials. The credentials lead to Azure tenants for fake companies. So in other words, Microsoft has bots which are reading email to detect phishing. When such phishing is detected, these bots visit the phishing site on purpose, pretending to be actual people who have been fooled by the phishing campaign. But the phishing victim bots provide fraudulent login credentials which, in turn, lead to fake company sites which have been established in Azure cloud tenants.

So basically they're baiting the bad guys that have created phishing sites, by leading them to believe that a real person got caught up in this, and then provides their credentials. Microsoft said that threat actors then use the credentials to log into these Azure honeypots in around 5% of the cases. But that's, you know, one in 20, and that's

sufficient. Microsoft then uses the data that they collect from the honeypots to learn of, discover, and document new techniques that the bad guys are using. And they said it takes around 20 days for the threat actors to catch on to the deception and to stop logging into the accounts, but by then Microsoft has collected all the data they need.

Leo: Good, and waste their time, waste the bad guys' time.

Steve: Yup.

Leo: Yeah, that's [crosstalk].

Steve: So, you know, I suppose if this is what they were doing, instead of fixing their problem with broken logging for a couple of weeks, I ought to cut them a bit of slack because this sure seems wonderfully proactive.

Leo: It's a big company. Got lots of people.

Steve: Yeah, they can do two things at once. You're right.

Leo: Can do two things at once, yup.

Steve: Maybe three.

Leo: Yeah, maybe.

Steve: Justin Long wrote, saying: "Hey, Steve. After listening to your coverage of BIMI, the technology behind BIMI seems solid. However, I would never even tell a user about this, let alone have them rely on it. It is the kind of thing that gets simplified down to 'It's easy. Just look for the logo, and you'll know it's safe.' My fear is that the scammers will start including logo files in the body of the email, with 'Verified by BIMI' next to it. Then," he says as an example, "Gary in Accounting sees a logo and thinks it's safe to click on it. In my opinion," he writes, "BIMI doesn't do anything to help the problem. If anything, it provides a false sense of security to most risky users." He said: "Thanks for everything you do. This podcast helps me every episode. Justin."

Leo: This was my exact concern, as well, is that, you know, how do you know it's real?

Steve: Yes. And for the record, I completely agree with Justin. I like the idea of having GRC's logo appearing in those boxes where anyone's BIMI logo might appear. And while, as we saw last week, it can be quite a royal pain in the butt to get it to happen, for GRC's weekly podcast mailings and for our other much less frequent software update mailings, for the moment at least, if only as an experiment, it's worth it to me. But I think it's clear that email is already so messed up that this is all it can ever be is just, you know, an opportunistic logo, a way for those who care enough to make it happen have their

corporate identity represented in the inboxes of any recipients whose clients will do so, and nothing more.

You know, and the reason that I believed these BIMI guys created all of this almost nutty-seeming, over-the-top security and authentication is that anything we do moving forward, and this comes back to what language should you now learn, anything we do moving forward should be as secure as we can make it. As I noted toward the end of last week's exploration of BIMI, our industry has continually set the bar too low out of a fear of low adoption from setting the bar too high.

We could argue that FIDO, the first FIDO, which absolutely positively required hardware tokens, you know, separate physical dongles, it never got off the ground because that bar was set too high, and it turned out FIDO was wrong. The world did not rush to go buy tokens for this. But as soon as they loosened that up and allowed our smartphones and biometric login computers to also be FIDO clients, then suddenly we got passkeys, and it actually happened. So, I mean, there really is something about that. But in the case of email, I think this is the right thing to do.

So anyway, if it turns out that this also serves as another signal for spam filtering, then I'll be happy for GRC to get the credit for having an officially approved BIMI logo for those providers who care. But otherwise, I agree. And Leo, I agree with your point, too, is that it's just a little - it's asking too much to put too much behind it.

At the same time, our second listener, Kevin de Smidt, wrote. He said - oh, and he's currently the Head of Technology at CURE International but was earlier at Valimail, who was one of the participants in this whole BIMI effort, so he is well acquainted with BIMI. He sent a sample and wrote: "This is how Mastercard emails appear in my Google Workspace email." He said: "Notice the blue checkmark and the text when hovering over it."

And sure enough, in his case there is a little blue seal with a checkmark. And if you hover over it, you get a little popup that says: "The sender of this email has verified that they own Mastercard.com and the logo in the profile image." And then it has a highlighted link labeled "Learn more," and if you click on it you have BIMI explained to you. So Google is surfacing more than just the logo, which as we've often seen can just be a website's favicon, you know, just pulls the icon from there. But here Google is showing a little blue checkmark. So we'll see where this goes.

And that's it for feedback. As I said at the top of the show, I had initially planned to have more, but there was so much cool stuff to talk about that gets us to this point where we need to talk about Credential Exchange Protocol that I didn't really have any time for more. So Leo, let's take out last break.

Leo: Okay.

Steve: And then we are going to look at how it's being made possible for providers of passkeys, you know, collections of passkeys to move them between environments.

Leo: Excellent. We will get back to this most important topic of the Credential Exchange Protocol in just a moment. You know, Steve, the whole point of the question and answers is just to get your thoughts on things. So as long as, you know, I mean, the whole show...

Steve: Oh, but Leo, there are so many good ones that I couldn't include.

Leo: I know. I know. We love our beautiful community. They really are an amazing group. All right. Let's talk about, since we're talking about passkeys, passkey portability.

Steve: So I should caution everybody that all we have so far is an outline of the protocol. The most recent version of the specification still has a long way to go before it's ready for the world. For example, what I found in the most recent documents looks like - and I put a sample of it, a snapshot in the show notes. In Section 4 under Usage Guidelines, it says "Offer guidelines for using the CXF format to import and export credentials securely."

Leo: What programmers call a "stub."

Steve: That's right. And then 4.1. Importing Credentials, "Explain the steps and considerations for importing credentials using the CXF format." And not surprisingly, 4.2. Exporting Credentials says "Provide instructions for exporting credentials to the CXF format."

Leo: They're very organized. They're very organized. They know what they want.

Steve: In other words, we know what we're going to say, but we haven't gotten around to saying it yet. And I don't even know if they've gotten around to working out the details yet.

Leo: That's the key.

Steve: In other words, we have an almost comical lack of meat on this particular bone. I have no doubt, though, that the various participants are all rowing in the same direction and that they fully intend to turn this into an actionable specification document at some point. But at this moment, what we have is evidence mostly of good intentions. However, scant though it may be, there is enough here to piece together a coherent picture of the system's operation. We're far short of having sufficient information to create a working implementation. I don't even know if that exists yet. But we're going to be able to get a feel for how the system works.

Okay. So let's begin with the news coverage WIRED offered eight days ago. This is what clued us into it last Tuesday, and it happened the day before, on October 14th, which was the day of the big FIDO Alliance Authenticate Conference held in Carlsbad, California. WIRED wrote: "The password-killing tech known as 'passkeys' has proliferated over the past two years, developed by the tech industry association known as the FIDO Alliance as an easier and more secure authentication alternative. And although superseding any technology as entrenched as passwords is difficult, new features and resources launching this week are pushing passkeys toward a tipping point.

"At the FIDO Alliance's Authenticate Conference in Carlsbad, California, researchers announced two projects that will make passkeys easier for organizations to offer, and easier for everyone to use. One is a new technical specification called Credential

Exchange Protocol (CXP) that will make passkeys portable between digital ecosystems, a feature that users have increasingly demanded. The other is a website called Passkey Central, where developers and system administrators can find resources like metrics and implementation guides that make it easier to add support for passkeys on existing digital platforms.

"Andrew Shikiar, CEO of the FIDO Alliance, told WIRED: 'To me, both announcements are part of the broader story of the industry working together to stop our dependence on passwords. And when it comes to CXP, we have all these companies who are fierce competitors willing to collaborate on credential exchange,' he said.

"CXP comprises a set of draft specifications" - very draft - "developed by the FIDO Alliance's Credential Provider Special Interest Group. Development of technical standards can often be a fraught bureaucratic process, but the creation of CXP seems to have been positive and collaborative. Researchers from the password managers 1Password, Bitwarden, Dashlane, NordPass, and Enpass all worked on CXP, as did those from the identity providers Okta, as well as Apple, Google, Microsoft, Samsung, and SK Telecom." Which is all what we want.

They said: "The specifications are significant for a few reasons. CXP was created for passkeys and is meant to address a longstanding criticism that passkeys could contribute to user lock-in by making it prohibitively difficult for people to move between operating system vendors and types of devices. In many ways, though, this problem already exists with passwords. Export features that allow you to move all your passwords from one manager to another are often dangerously exposed and essentially just dump a list of all your passwords into a plaintext file.

"It's gotten much easier to sync passkeys across your devices through a single password manager, but CXP aims to standardize the technical process for securely transferring them between platforms so users are free and safe to roam the digital landscape. Importantly, while CXP was designed with passkeys in mind, it is really a specification that can be adapted to securely exchange other secrets as well, including passwords or other types of data.

"Christiaan Brand, identity and security group product manager at Google, told WIRED: 'In the future, this could apply to mobile driver's licenses, say, or passports, any secrets that you want to export somewhere and import into another system. We've got most of the rough edges sanded down with passkeys, but one of the main pieces of negative feedback over the past year has been around portability and potential vendor lock-in. I think with this we are signaling to the world that passkeys are growing up.'

"The goal of Passkey Central, a resource repository, is similarly to help the ecosystem expand and mature. Product leads or security professionals who want to implement passkeys for their user base may need to make a business case use to executives to get budget for the project. The FIDO Alliance is basically aiming to help them with the pitch, providing data and communications materials, and then support their rollout with prefab materials like implementation and roll-out guides, user experience and design guidelines, documentation around accessibility, and troubleshooting.

"FIDO's Shikiar said: 'We've made amazing progress on passkeys. Usability and user experience are pretty much there. But we do have a punch list, and we're actively working on it. Portability is an important feature on that list. And while the biggest brands on the planet are now using passkeys at scale, there's a very long tail of companies that haven't gotten started yet. So we want to offer resources and the assets they need to be successful.' Craig Newmark Philanthropies..."

Leo: Do you want me to play the jingle? I can play the jingle if you want.

Steve: Craig Newmark, who we all know...

Leo: Philanthropies.

Steve: Philanthropies, thank you, Leo, "Cyber Civil Defense coalition provides some funding to advance passkeys. In an interview with WIRED, Newmark said he believes that passkeys can make a real difference, both for the digital security of individual people and for Internet security overall." And of course we agree with him. Craig said: "There are a lot of vulnerable systems out there. You need to make it a lot harder for bad actors to defeat password schemes. You need to make everything more secure, and passkeys is part of that."

Okay. Now, having noted that there was very little meat on this bone, there was some. The specification we have today has a useful introduction to the problem and application space that this protocol is expected to fill, and it turns out to be more than just passkey credential transport, as we said. So here's how the Credential Exchange Protocol specification (CXP) introduces the problem. It's just a few paragraphs and a bullet point or two.

So they said: "Individuals and organizations use credential providers to create and manage credentials on their behalf as a means to use stronger authentication factors. These credential providers can be used in browsers, on network servers, and on mobile and desktop platforms, and often sharing or synchronizing credentials between different instances of the same provider is an easy and common task.

"However, the transfer of credentials between two different providers has traditionally been an infrequent occurrence, such as when a user or organization is attempting to migrate credentials from one provider to another. As it becomes more common for users to have multiple credential providers that they use to create and manage credentials, it becomes important to address some of the security concerns with regard to migration." So they said "currently," and we have four bullet points.

"Credential provider applications often export credentials to be imported in an insecure format, such as CSV (comma separated values), that undermines the security of the provider and potentially opens the credential owner to vulnerability." Two: "Credential providers have no standard structure for the exported credential CSV, which can sometimes result in failure to properly migrate one or more credentials into a new provider." Third: "Some credentials might be unallowed to be imported due to device policy or lack of algorithmic capability on the importing credential provider." And finally: "Because organizations lack a secure means of migrating user credentials, often they will apply device policy that prevents the export of credentials to a new provider under any circumstances, opting to create multiple credentials for a service." In other words, you know, they're just not exportable, which is what we've seen so far. The idea being, oh, you know, no problem, create one over in the Apple world and create another one over in the Windows world.

So they finish, saying: "In order to support credential provider interoperability and provide a more secure means of credential transfer between providers, this document outlines a protocol for the import and export of one or more credentials between two credential providers on behalf of a user or organization in both an offline or online context. Using Diffie-Hellman key exchange, this protocol allows the creation of a secure channel or data payload between two providers."

Okay. So that introduction paints a picture of a more generalized secret exchange protocol. It's clearly useful; and, surprisingly, it's also completely lacking in our industry today. Somehow we've managed to come this far without a universal definition of how the owner of some secrets could move them elsewhere. The fact that this is finally being proposed demonstrates, I think, the arrival of some much-needed maturity. Up to this point, much of our industry has relied upon closed and proprietary ecosystems. That closure was first pried somewhat open by the promise and delivery of competitive open source software and open development. But the profit motive runs deep, and we've seen how shaky some of open source software foundations can be.

The CXP document noted that the name was subject to change. I'd vote for something that's explicitly more generic than "Credentials." Maybe Secrets Exchange Protocol, for example would be good. Anyway, so under "Scope" they briefly wrote. They said: "This protocol describes the secure transmission of one or more credentials between two credential providers on the same or different devices managed by the same credential owner, capable of function in both online and offline contexts. This protocol does not make any assumptions about the channels in which credential data is passed from the source provider to the destination provider. The destruction of credentials after migration by the credential provider source is out of scope, as well."

Okay, so that's good. They're explicitly keeping this extremely general, and it's significant that it can be an offline system. The spec does sketch an overview of how this protocol would work; and, frankly, it's nothing special. And that's not criticism. Quite the opposite, in fact, because we're past the point where crypto should be surprising. We now have established and well-proven ways of accomplishing pretty much anything we need.

Okay. So the sketch looks like this: The planned recipient of the credential collection is asked to create an "export request." So the recipient of the collection is asked to create an export request, which will then be provided to the credential provider; right? The side, the end which is going to be exporting the credentials. That export request includes the necessary details including a challenge, the details of the type of information that the recipient wishes to receive, the set of encryption schemes it's able to use. And unless it has access to the credential provider's public key, it will also include - and I'll get back to that a little bit later - the public side of a Diffie-Hellman key agreement.

Okay, now, remember that what Whit Diffie and Martin Hellman invented was this brilliant scheme which allows two parties to exchange public keys in full view of any attackers. And upon receipt of each other's public Diffie-Hellman keys, each is able to construct and arrive at the same shared secret key. It's bizarrely counterintuitive, but it works. And actually I used their system in several places inside SQRL.

Okay. So the credential importer uses a cryptographic-grade random number generator to create a unique Diffie-Hellman key pair. It stores the private half internally and includes the public half in this credential "export request" which it's been asked to generate. If an end-user or other authorizing party then approves and provides this export request, the exporter uses the information to create an encrypted payload.

What the exporter does is to similarly synthesize its own Diffie-Hellman pair, but it doesn't need to retain any record of it. It will combine its own private half with the importer's public half, which was in the export request, to create a secret. And that creates this automatically shared what they term a "migration key." And it uses that to encrypt the payload using the other parameters that were provided in the importer's export request. It signs the challenge provided by the importer and includes the public half of the Diffie-Hellman key pair that it just created in the exported response packet.

So the exported response packet is then, one way or the other, carried or through a network provided to the credential importer, where you want the credentials to go. That includes, obviously, this blob of encrypted credential data, the signed challenge response, and the public half of the credential provider's Diffie-Hellman key pair which was used to create the shared migration key. The credential importer has been holding onto the secret half of the Diffie-Hellman key pair it generated as part of the export request. So it validates the challenge, and I'll explain more about that in a second. Then it combines the secret it's been holding onto with the exporter's public key that was provided in the exported packet. This will recreate the identical "migration key," which it's able to use to securely decrypt the contents of the exported package.

So what we have is a straightforward application of Diffie-Hellman key agreement where the two parties created the shared secret and used it to exchange an encrypted package containing the user's credentials. And at every stage the entire process was state-of-the-art secure. That is, nobody getting hold of the packet would be able to decrypt it. Nobody seeing the export request would be able to use that in any way to decrypt the packet when it was coming back to the importer side. That system is absolutely secure.

What's currently missing from the specification is, well, everything else to make it actually go. As we saw, a lot of empty paragraph headings, but empty paragraphs. But the overall mechanism is clear, and it's been proven, and it will work. We have so far no idea what the user experience would be, you know, whether the Internet will be used in some way for, like, both sides to rendezvous and automatically exchange the packet, or whether that might only be an option. It would be possible to do all of this using a USB thumb drive and, you know, so-called "sneakernet," where you literally go to the side where you want to import it. You say "Please create an export request." The USB key has that. You take the USB key over to the side that currently has the credentials, and you say "Here's an export request from the importer. Please honor it and export my credentials." And that would then add a blob to the USB key.

Then you take it back to the original side where you want this to be imported and say "Here is the packet." And that side would then be able to decrypt that packet and import the credentials. So the gist is that the user asks the credential recipient to create an export request for the credential sender. That export request is then provided to the credential sender, which uses it to prepare an encrypted package. And when that encrypted package is returned to the credential recipient, the residual information which the recipient retained on its side from the original export request allows it to securely decrypt the sender's package.

Now, a well-known characteristic of Diffie-Hellman is any lack of protection from man-in-the-middle attacks. While Diffie-Hellman brilliantly creates a mechanism for secret key agreement between two parties, it has no mechanism for authentication. Nothing I've described prevents an attacker who's somehow able to interpose themselves between the parties from impersonating each end to the other. Because you'll notice there's nothing special about the ends at this point. They're just sharing keys that they assume the actual other endpoint generated. But it could be something that managed to interpose itself in between. So if that happened, doing that would allow the impersonator to decrypt the package as it moved past.

Now, all we know from the specification is that the credential importer will include a challenge for the exporter to sign. That's all it says. That's all we know today. We do know that the signer of the challenge would need to use a private key, and that the credential recipient would need to verify the signature with a matching public key. But from where and how does the credential recipient obtain the credential sender's public key? Maybe from DNS? Maybe from some sort of central FIDO registry of CXP users? We don't know.

Leo: Could it be a PGP key at the PGP key server? Or does it have to be [crosstalk] authentication?

Steve: Well, yeah, exactly. It could be something like a - it needs to be some sort of source of, you know, like authoritative source of public keys so that - and that's the one missing piece. That way one end would be able to authenticate against, you know, would be able to authenticate the other. And that would completely cut out, you know, any vulnerability from man in the middle.

Okay. So that's the big first part. The other part is the announcement of this Passkey Central website. It's at PasskeyCentral.org. And having read through the site, it's clear that, more than anything else, it's intended to be passkey adoption lubricant. It's taken a few years, but passkeys have matured to the point that if any sort of friction is holding an organization back, now might be the time, I would say, to apply some lubrication.

In the early days, anyone could be forgiven for feeling that passkeys were not there yet, or were not ready yet, or hadn't been proven, or might turn out to be another FIDO failure like the first attempt was, which never achieved critical mass. Or even that what we already had was well proven and working well enough with multifactor authentication or with password managers that make the use of super-strong passwords effortless. You know, the argument could have been made that, you know, this problem was solved well enough.

The Passkey Central site and its companion Passkeys.dev developer site make a very strong case for passkeys having arrived, and for those who do not get busy with its adoption being left behind. At some point it's going to be regarded as "doing it wrong" not to have some system for asymmetric key public key authentication. That's the big difference. As we talked about recently, Meta was recently excoriated for storing their users' passwords in the clear without any hashing. The difference between the inherent insecurity of any traditional secret-keeping authentication system such as static passwords, or even one-time passwords, which are still asking the server to keep something secret.

You know, that difference, compared to the extreme security offered by an asymmetric key authentication system like passkeys, which requires no secrets to be kept at all, that means that at some point anyone who is not employing the free-to-use, widely available, and increasingly ubiquitous passkeys asymmetric system will similarly cause some eyebrows to be raised. It's like, wait, you're still using passwords? That's, you know, they're not secure, no matter how you store their hashes.

So the point is, I'm here to say it's been a couple years. I think it's clear, once this CXP specification happens, and we actually see that Apple is willing to allow us to move our collection over between password managers, and we're able to aggregate them, passkeys will have made the grade. The benefits of the system have proven to be sufficiently strong that the question has moved from "whether" to "when." And "when" should be "as soon as possible, what are you waiting for?" because there's no longer any rationally supportable argument to be made for waiting any longer. The Passkey Central site should now provide sufficient lubrication to help overcome any residual adoption friction. The Passkeys.dev site provides sample code in Rust, TypeScript, Java, .NET, Go, Python and Ruby; and Passkeys test sites are available at WebAuthn.io, WebAuthn.me, with Yubico and Akamai also offering test facilities.

Once passkeys' Credential Exchange Protocol has been fleshed out - and make no mistake, it does still have quite a ways to go, although its overall shape is quite clear - the last piece of the passkeys solution-set will have been put in place. And given that all

of the major players have signed onto supporting CXP, the last roadblock to further passkeys adoption I think has been removed.

Leo: Yay.

Steve: Yeah. We're there.

Leo: Of course your SQRL solution, which was similar but better, I mean, obviously, unless you get Microsoft to suddenly say, hey, you know, SQRL is better than passkeys, has been replaced. But what are the things that passkeys is missing that you wish it had, that SQRL had? Recovery is one; right?

Steve: Well, it works so differently. With SQRL, you had one secret.

Leo: Right. Passkeys every site is.

Steve: Passkeys are a collection of secrets.

Leo: Right, right, right.

Steve: So it's so...

Leo: It's a very different thing, yeah.

Steve: Yeah. It's entirely different. Also there was a way of - there are still some vulnerabilities that, if your passkeys got away from you, you're pretty much screwed.

Leo: Right.

Steve: And SQRL provided a mechanism for getting that back.

Leo: For recovery, yeah.

Steve: From recovering from the loss of your secret. So, I mean...

Leo: What that means is that passkeys is always going to have passwords as a fallback, I think. I mean, I think that's probably it; right? I guess we do email.

Steve: Actually, I think all authentication is always going to have it. I mean, this is a fundamental weakness is that you will always say, you know, the dog ate my homework.

Leo: Right. A lot of people don't do passwords. They put in a random string of junk, and every time they go to the site they say "I forgot." And they rely on their email as password authentication.

Steve: Yup.

Leo: Anything wrong with that as a recovery method?

Steve: And as I said, passwords need to be regarded as a login accelerator.

Leo: Right.

Steve: Because we already have a fallback of "I forgot my password."

Leo: Right. That's the weakest link.

Steve: So as long as that's there - and in fact that was one of the other things that I built into SQRL was after you got comfortable with it and you understood how it worked, you could set a checkbox that put a beacon on your identity. And anytime you went to a website with that set, it said "Please disallow all fallback."

Leo: No fallback, yeah.

Steve: And so that if a bad guy got a hold of your email, it wouldn't help them.

Leo: Right. So this is a really good example of sometimes the perfect is the enemy of the good, or something like that. Which is you create a perfect system, but maybe good enough is all we need.

Steve: I agree. I mean, I'm - right now XenForo, the software that I use for my forums, I am a dot release behind because we're using SQRL there, and I haven't asked Rasmus to change to support the next dot. The reason I bring it up is that the next dot release supports passkeys. And I want passkeys for GRC's forums.

Leo: Right.

Steve: Because they're what the world is going to use. And, I mean, and they do work.

Leo: When they work, they work amazing. It really is a great solution.

Steve: Yeah, it's completely transparent. It's the way it should be.

Leo: Yeah, I really like it, yeah.

Steve: The way it should be.

Leo: Yeah. Steve Gibson is the way it should be, as we rapidly approach Election Day/999.

Steve: And Episode 999, baby.

Leo: But the good news, for those of you who don't know, Steve has agreed to go four digits. We don't know how he's going to do it. It's a mystery right now. He may not know how he's going to do it.

Steve: I haven't made the change yet. I get to do that pretty soon.

Leo: But we're going to keep going because you know what? This is no time to stop.

Steve: Nope.

Leo: Bye-bye.

Steve: 998. Bye.

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/