Passkeys: A Shattered Dream?

Description: The choice for this week's main topic received some serious competition from some surprising legislation that came into effect yesterday in the United Kingdom. So we're going to start by taking a close look at what happened in the UK that promises to completely change the face of consumer IoT device security. As we'll see, that's not an overstatement; the world as we've known it just changed. While that exploration is going to consume most of the first half of today's podcast, I also want to look at what happened last week with Chrome's change of plan regarding third-party cookies, I have a bit of listener feedback to share, and news of the next installment in a long-running science fiction book series. I also have the welcome news that I am finally working on bringing up GRC's email communications system. Then we'll finish by taking a look at a blog posting by an industry insider that many of our listeners forwarded to me, asking "What do you think about this?"

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-972.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-972-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Last night he was sitting in his armchair, having a nice glass of Cabernet Sauvignon, reading the news, when he saw the announcement that the United Kingdom is going to make a major change to how IoT devices work. He is so excited, he's come here, he leapt to his feet and said "We've got to talk about this on Security Now!." So we will. What happened with Chrome dumping its plans to dump third-party cookies? And let's talk a lot about Passkeys. There was an interesting post Steve read, and a lot of you read, saying it's not working. Are Passkeys over? All that and more coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 972, recorded April 30th, 2024: Passkeys: A Shattered Dream?

It's time for Security Now!, the show where we - I know you wait for this all week long; don't you. We cover the security field, the computer field, every other field, the sci-fi field, with this guy right here, Mr. Steven "Tiberius" Gibson. May the Fourth be with you.

Steve Gibson: And Leo?

Leo: Yes.

Steve: You are - oh, the Fourth.

Leo: Yeah, it's coming up.

Steve: You are correct about the breadth of our coverage. I do have a little sci-fi mention, a real quickie.

Leo: Oh, good. Oh, good.

Steve: For those listeners who have been following along, what has turned out to be my favorite saga, which promises - well, actually I was going to say it promises to be never-ending, but it actually does have an end; and this author is, you know, continuing to deliver on the promise.

Leo: Nice.

Steve: Okay. But the choice for this week's main topic received some serious competition from some surprising legislation that came into effect just yesterday in the UK, in our dear beloved United Kingdom. So we're going to start by taking a close look at what happened in the UK that, I kid you not, promises to completely change the face of consumer IoT device security, like, immediately. I know that's, you know, I don't think that's an overstatement. I think that the world as we have known it has just changed. So you could see why that was competing for today's topic, which we'll get to after that. We're going to explore a little bit of, like, what happened with the news that just came in that you announced during last week's podcast, Leo, which was Chrome's sudden change of plan regarding its third-party cookie handling.

Leo: Yeah.

Steve: I'm going to get a little bit into that. But I'll explain why I'm probably going to punt most of it till next week because it turns out there's a lot. I also have a little bit of listener feedback to share. And as I said, news of the next installment in a long-running sci-fi book series. I've also got some welcome news that I'm finally working on GRC's email system, which will come as great news for our listeners. And then we're going to finish by taking a look at a blog posting by an industry insider developer that a surprising number of our listeners, this thing has gotten a lot of traction out in the industry, and our listeners kept forwarding me links to it, asking whether Passkeys is a shattered dream.

Leo: Oh.

Steve: They've all been saying, "What do you think about this?" And so, you know, some neat stuff to talk about, and of course a great Picture of the Week that's apropos of the topic that we've been covering lately of Voyager.

Leo: Yes. Passkeys: A Shattered Dream, the topic on today's - it sounds like, you know, [sound], today.

Steve: We go up, and we come down.

Leo: And we go down, yeah. I like Passkeys. It is a shattered dream. Although one could say you might have a little dog in this hunt because honestly the SQRL protocol that you created is a far better way of doing it.

Steve: It solved the problem that is dogging this. But, you know, I don't even mention it in my coverage because all of our listeners already know.

Leo: They know.

Steve: That, you know, I did solve it the right way, and that's not what we got.

Leo: And I suppose that ship has sailed.

Steve: Yeah.

Leo: And you just have to have the big guys behind it before it has any chance of being adopted.

Steve: Well, and in fact this author, this guy is the author of a very strong WebAuthn library, the one for Rust that SUSE is using, and many others have forked from. He notes that Chrome has succeeded in killing some features just by not adopting it.

Leo: Right, exactly. Yeah. That's too much power in Google's hands, if you ask me.

Steve: Well, they've got it.

Leo: We will get the - yeah, I know, it's too late now. We will - by the way, you know, I used OpenAI's chat for a long time and made some GPTs and really like it, especially v4. But I just started using on Sunday, Kevin Rose said try Gemini. Have you not tried - and so I tried Gemini Advanced, and it is mind-blowingly good. So we've got us a race, which is exciting. Very interesting. And I'm sure there'll be lots of security implications there. Gemini writes excellent code, unfortunately.

Steve: Well, and did you see the blurb? I didn't get it in the show notes where ChatGPT4, just given the list of CVEs...

Leo: Of CVEs, we talked about it, yes.

Steve: Is able to generate exploits. Just like, oh, here you go.

Leo: They took 87% of the CVEs they were given, just the description, and created the proof of concept.

Steve: Working exploits for them. So script kiddies have been elevated. Oh.

Leo: What a world. We are - this is just getting wild. That's all I can say.

Steve: Yeah.

Leo: And not in a good way, to be honest with you. Ready for the Picture of the Week.

Steve: Okay. So this resonated with me from something in my youth, and I figured that you would, being the king of pop that you are, you know, you'd go, oh, yeah, that's about this. But it drew a blank for you.

Leo: Kind of old pop, yeah. Well, it sounds like, I don't know what, a joke, maybe, an old joke.

Steve: Okay. So we have a cartoon which is apropos of the topic we've been following about the fate of Voyager 1 and how by some miracle it is back on the air. Anyway, so the cartoon shows a couple of cute little green aliens in their classic circular UFO saucer with the glass bubble. And the saucer, however, is labeled "Salvage." And it's got the tow truck hook off the back with a hook. So they're, you know, they're flying around out in space looking for stuff. And so the alien - and then so what they have done is they just - we see them looking at the Voyager 1 probe labeled Voyager 1977, you know, with a NASA on it, and its various probes and sensors and so forth. And the one alien is saying, "15 billion miles on it, but the radio still works."

Leo: It's true.

Steve: And I'm reminded of something about used - it is true, absolutely true.

Leo: Yeah, yeah.

Steve: I'm reminded of something about used cars back from my high school days.

Leo: Yeah, I think it's an old joke.

Steve: Like there was some meme about, well, it's broken down and only goes downhill or something, but the radio still works.

Leo: Right, right.

Steve: Anyway, 15 billion miles, the radio still works. I thought that was a cute little observation.

Okay. So - okay. Yesterday - mark this day in your calendars, yesterday, April 29th, 2024 - a new law went into effect in the UK. Not just like legislation got submitted somewhere; or, well, we're going to give this some consideration; and not just like, oh, recommendations like we're always getting. This is a law that went into effect yesterday. And I'm astonished by this - not only by the fact that it happened and by the content, but the depth of the quality of the baseline requirements that this proposes, which is why we're going to spend some time on this because this is huge.

Now, the Guardian, what was also odd was I guess I must have seen this, like, immediately after it happened because I looked around for, like, other more fleshed-out coverage, and nobody was picking up on it. Now there's like, overnight it's like beginning to happen in the security info sphere because it's like, whoa, what?

So yesterday, only the Guardian in the UK seemed to have anything to say. And they didn't say much, but their headline was "No more 12345: Devices with weak passwords will be banned in the UK." And the subhead was "Makers of phones, TVs, and smart doorbells legally required to protect devices against access by cybercriminals." And, you know, not just passwords, baby. That's just the tip of the iceberg. I mean, this is - it's comprehensive legislation.

In order to get more detail, I went to the source. So this is the GCHQ's National Cyber Security Centre blog posted yesterday which was titled "Smart devices: New law helps citizens to choose secure products." And actually even that's an understatement because this impacts not only the manufacturers of devices that may not be in the UK, but also anyone importing them and anyone retailing them. And it's got some teeth behind it. So it's not that the consumers in the UK are going to have a choice. There aren't going to be any non-compliant options to purchase. And of course this ends up being global because we're in a global economy.

Okay. So what GCHQ said in sort of their top-level blog announcement, they said: "From April 29th, 2024, manufacturers of consumer 'smart' devices must comply with new UK law." That is yesterday. "The law, known as the Product Security and Telecommunications Infrastructure act, or PSTI act, will help consumers to choose smart devices that have been designed to provide ongoing protection against cyberattacks. The law means manufacturers must ensure that all their smart devices meet basic cyber security requirements."

Now, here they just highlight three, and they happen to be the first three. But they're the first three of, like, 14, and they're all really significant. But the first three are: "The manufacturer must" - get this. I mean, it sounds like I wrote this from the podcast. "The manufacturer must not supply devices that use default passwords, which can be easily discovered online and shared. If the default password is used, a criminal could log into a smart device and use it to access a local network or conduct cyberattacks." So again, this is just sort of the blog discussing the legislation. We'll get to the actual legalese here in a second. But so they're sort of like explaining the why of these requirements at this point.

Second one, the manufacturer must provide a point of contact for the reporting of security issues which, if ignored, could make devices exploitable by cyber criminals. Right. And three, the manufacturer must state the minimum length of time for which the device will receive important security updates. And then they flesh that out by explaining: When updates are no longer provided, devices are easier to hack, or may stop working as designed. So that's just the top three, which this law makes a requirement for the sale of anything; and, I mean, and again, there just - there aren't

any loopholes in this. It's like, if it's going to connect to the Internet or a network, it has to have this.

So they said: "Although most smart devices are manufactured outside the UK, the PSTI act also applies to all organizations importing or retailing products for the UK market." I mean, like, Amazon. So they said: "Failure to comply with the act is a criminal offense, with fines up to 10 million, or 4% of qualifying worldwide revenue, whichever is greater."

They said: "The law applies to any 'consumer smart device' that connects either to the Internet or to a home network, for example, by WiFi. This may include" - and here's the first of several enumerations that we're going to be encountering. But here they said: "Smart speakers, smart TVs, and streaming devices; smart doorbells, baby monitors, and security cameras; cellular tablets, smartphones, and games consoles; wearable fitness trackers including smart watches; smart domestic appliances such as light bulbs, plugs, kettles, thermostats, ovens, fridges, cleaners, and washing machines."

They finish: "The NCSC has produced a 'point of sale' leaflet for retailers to distribute in-store to their customers. It explains how the PSTI regulation affects consumers, and why it's important to choose smart products that protect against the most common cyberattacks."

So, you know, it's the end of April; right? It's not the beginning. This is not April Fool's Day. This happened on the 29th. And the first thing I need to say is "Holy crap. Where did this come from?" Turns out it's been in the works for five years and just not much is, you know, it hasn't been drawing much attention to itself. So fines in the amount of the greater of 10 million, I think that was 12.5 million USD at this point, at the current exchange rate, or 4% of a manufacturer's qualifying worldwide revenue, whichever is the greater. This is the sort of legislation that can really make a profound overnight difference in consumer security. And since a great many manufacturers have shown through their actions or, you know, deliberate inaction, that they need to be made to change, this is the change that's required to make them.

Since this is huge and potentially affects all products worldwide which might find their way to the UK because it impacts not only manufacturers, as I said, but anyone who imports or retails such products, I needed to get a bit more back story. So I did some digging. On the gov.uk website I found the actual legislation. It turns out, as I said, it's been quietly in the works for several years, and it really is a law, not just some watered-down milquetoasty "recommendations" that we see too often here in the U.S.

The Verge picked up on this, and in their coverage they noted that, here in the United States, our FCC is trying something similar in its forthcoming "Cyber Trust Mark" program. You know, they liken it to the federal ENERGY STAR program, explaining that the Cyber Trust Mark logo indicates which products comply with the program's recommendations, which include strong default passwords. But like ENERGY STAR, nobody's forcing companies to go along with it; and consumer product packaging has become so encrusted with certifications and compliance logos that it's unclear whether anyone even notices.

As we know, consumers are focused on three things: Does it do what I need? What does it cost? And what additional nice-to-have features does it offer? Whether or not a connected light switch has a default password is the last thing on anyone's, like, shopping list. In other words, while the United States continues to be completely lame on this, the United Kingdom has taken the only action that has any chance of actually producing results for the consumer. And thanks to the fact that we still live in a blessedly globalized economy, everyone everywhere will obtain the security benefits that the kingdom is now requiring as a matter of law.

Since this matters to everyone everywhere, and since it's going to change the face of Internet-connected consumer technology, let's take a closer look at what the legislation actually has to say. First of all, this comes from a non-profit organization, that is, the actual think-tank behind the legislation, which is what got enacted, known as the European Telecommunications Standards Institute, or ETSI, E-T-S-I. We've spoken of them in the past. This work has been underway, as I mentioned, for the past five years, having started back in 2019, in February, with the publication of its v1.1.1, and it's been quietly making its way forward year by year.

The Baseline Requirements document is the one that's most relevant. It's a 34-page PDF that I've given a GRC shortcut to. The shortcut is ETSI, E-T-S-I. So if you're curious, you can just put grc.sc/etsi into your browser, and you'll be bounced over to a document titled "Cyber Security for Consumer Internet of Things: Baseline Requirements."

The document's introduction explains. They said: "As more devices in the home connect to the Internet, the cyber security of the Internet of Things (IoT) becomes a growing concern. People entrust their personal data to an increasing number of online devices and services. Products and appliances that have traditionally been offline are now connected and need to be designed to withstand cyber threats. The present document brings together widely considered good practice in security for Internet-connected consumer devices in a set of high-level outcome-focused provisions. The objective of the present document is to support all parties involved in the development and manufacturing of consumer IoT with guidance on securing their products.

"The provisions are primarily outcome-focused, rather than prescriptive, giving organizations the flexibility to innovate and implement security solutions appropriate for their products. The present document is not intended to solve all security challenges associated with consumer IoT. It also does not focus on protecting against attacks that are prolonged or sophisticated or that require sustained physical access to the device. Rather, the focus is on the technical controls and organizational policies that matter most in addressing the most significant and widespread security shortcomings. Overall, a baseline level of security is considered. This is intended to protect against elementary attacks on fundamental design weaknesses," they say, "such as the use of easily guessable passwords.

"The present document provides a set of baseline provisions applicable to all consumer IoT devices. It's intended to be complemented by other standards defining more specific provisions and fully testable and/or verifiable requirements for specific devices which, together with the present document, will facilitate the development of assurance schemes. Many consumer IoT devices and their associated services process and store personal data. The present document can help in ensuring that these are compliant with the General Data Protection Regulation (GDPR). Security by design is an important principle that is endorsed by the present document."

Okay. So my goal for today's discussion of this is to accurately convey how comprehensive this document, and the legislation that backs it up, actually is, which I was astonished by. So here's the bullet point. I assembled this from the document, just to give you a sense for how comprehensive it is. So here's the list of its main topics: No default universal passwords. Implement a means to manage reports of vulnerabilities. Keep software updated. Securely store sensitive security parameters. Communicate securely. Minimize exposed attack surfaces. Ensure software integrity. Ensure that personal data is secure. Make systems resilient to outages. Examine system telemetry data. Make it easy for users to delete their data. Make installation and maintenance of devices easy. Validate input data. And data protection provisions for consumer IoT they go into at length.

And again, these are not, like, gee, we wish we had these. The legislation is about the policy. As I've always said, there's a complete difference between policy and, you know, mistakes. Anybody can make a mistake; but policy is, you know, your stated goal. And so this legislation enshrines these goals as the policy that consumer IoT devices sold in the UK must incorporate. They must have these as policies, as operating goals which are implemented in the device. So each of these major topics is broken down into multiple pieces, and each of those pieces is tagged with one of four possible requirement levels.

Now, here's where for a moment I was concerned that my enthusiasm for this was going to be dashed because we've got mandatory, recommended, conditionally mandatory, or conditionally recommended. And I thought, oh, great. Well, if everything is just recommended, then we've gotten nothing. Turns out almost everything is mandatory. So there are a few things where they've backed off of mandatory. But for the most part it's mandatory across the board.

The scope of what the document covers is also very clearly laid out, that is, you know, to not allow people to say, oh, that doesn't apply to us, so we don't have to do that. No, I don't think anyone's going to get a free pass here. It says, and this is in the baseline requirements document: "The present document specifies high-level security and data protection provisions for consumer IoT devices that are connected to network infrastructure." So they even broadened it to network infrastructure, meaning if you're connected to something, this is for you.

They say, parens: "...(such as the Internet or home network), and their interactions with associated services. The associated services are out of scope. A non-exhaustive list of examples of consumer IoT devices includes" - and it's pretty much what I just said. But there's additional ones here: "Connected children's toys and baby monitors; connected smoke detectors, door locks, and window sensors; IoT gateways, base stations, and hubs to which multiple devices may connect; smart cameras, TVs, and speakers; wearable health trackers; connected home automation and alarm systems, especially their gateways and hubs; connected appliances, such as washing machines and fridges; and smart home assistants."

They said: "The present document provides basic guidance through examples and explanatory text for organizations involved in development and manufacturing of consumer IoT on how to implement those provisions. Table B.1 provides a schema for the reader to give information about the implementation of the provisions." Now, okay, Table B.1, that's the table which breaks down all the topics and subtopics with the mandatory or recommended categories. But then the idea is that manufacturers will print this out and be required to fill in for every one of those items the compliance level of their device, attesting then to the fact that they have met the recommendations.

So they said: "Devices that are not consumer IoT devices, for example those that are primarily intended to be used in manufacturing, healthcare, or other industrial applications, are not in scope of the present document." So this is squarely aimed at consumer IoT residential-style devices. Of course, once that's all that's available for the consumer, other enterprises and anyone who purchases the devices get the benefit of all these features which have been required at this level. And they said: "The present document has been developed primarily to help protect consumers; however, other users of consumer IoT equally benefit from the implementation of the provisions set out here."

Okay, now, the document does differentiate something it considers to be a "constrained device," basically. And again, nobody gets a free pass on this one. But they did recognize that there are some things that, while they're connected and consumer devices, they just - they have too many constraints to meet what is otherwise a set of very high bars. So they said:

"The present document addresses security considerations specific to constrained devices. For example, window contact sensors, flood sensors, and energy switches are typically constrained devices."

So to give everyone a sense for how well thought-out and specific this is, here's the document's definition of what it means by a "constrained device." They said: "A constrained device is a device which has physical limitations in either the ability to process data, the ability to communicate data, the ability to store data, or the ability to interact with the user, due to restrictions that arise from its intended use." They said: "Note: Physical limitations can be due to power supply, battery life, processing power, physical access, limited functionality, limited memory, or limited network bandwidth. These limitations can require a constrained device to be supported by another device, such as a base station or companion device."

And they give some examples: "A window sensor's battery cannot be charged or changed by the user; this is a constrained device." Another example: "The device cannot have its software updated due to storage limitations, resulting in hardware replacement or network isolation being the only options to manage a security vulnerability." And that ends up being important because this astonishing baseline requirement also deals with firmware updating. And yes, it has to be enabled by default. This is just like, where did this come from?

Third example: "A low-powered device uses a battery to enable it to be deployed in a range of locations. Performing high-power cryptographic operations would quickly reduce the battery life, so it relies on a base station or hub to perform its validations of updates. Another example, device has no display screen to validate binding codes for Bluetooth pairing. Or five, the device has no ability to input, such as via a keyboard, any sorts of authentication information." So obviously, where things are either impossible or impractical for these requirements to be met, well, they're excused under the grounds that it is a constrained device. But otherwise no free pass.

Okay. So the document's too long and detailed for me to go through in detail here; but I do want to, again, give everyone a sense for how well thought out, thorough, and serious this is. So I want to look deeply at this "No universal default passwords." And Leo, let's take our second break, and then I will do that.

Leo: I'm sorry, I'm talking to Mom. You want to say hi? You want to say hi to Mom? Say hi to Steve.

Steve: Hey, Mom.

Leo: I'm going to run. Go to dinner; okay? I love you, Mama. Bye.

Steve: That's right, it's 5:00 o'clock there.

Leo: It's 5:00 o'clock there. And you know what happens is she's got an alarm on her Amazon Echo to wake her up. So she called me, and then it starts going off, and I had to explain to her how to go over to it and turn it off. So I guess what she normally will do is leave, go to dinner, and by the time she gets back the alarm has stopped because I don't think she knew how to stop it.

Steve: Ah, these fancy consumer devices.

Leo: She loves it. You know, she wanted to thank me because I gave her - and I hope that this has a good password on it. I gave her a - there's a company called Nixplay makes a 15-inch frame, it looks like a painting frame, but it's a digital photo frame. And I can email pictures to it. I control it from here. So, and she just sits and looks at it all day.

Steve: Nice. Aww.

Leo: So it just makes her really happy. She says it's like having my family here. So anyway. We will continue in a moment. Sorry for that personal interlude.

Steve: Yeah, no problem. And, boy, May 7th is going to be a busy day.

Leo: It is. The Apple event is May 7th. RSA conference is going on. You should come up sometime for that.

Steve: I have a large inventory of very aging and slowing down iPads.

Leo: Oh, iPads. You're an iPad user, I know.

Steve: So I've not - I love my iPads, and I've not purchased any for many years because there's, you know, no new iPads.

Leo: There's no reason to.

Steve: Right.

Leo: And I think that this time they're going to - they know this, by the way. I mean, I've been using - I have an iPad 6 which I've had for three years. I very rarely use my iPad Pro. They've got to get a way to get...

Steve: I didn't even know they had numbers.

Leo: Oh, well, they don't. You have to know. You have to know.

Steve: Oh. I think I have a - I know I don't have the iPad 1. Remember how that was like - it was like a whale?

Leo: Oh, yeah.

Steve: It had a weird bowed back.

Leo: Oh, yeah. You don't want that, yeah. No, we've come a long way. But you will want these because they have OLED screens. And what's intriguing is this rumor you're going to have the M4 and a lot of AI built in. So this might be something you really do want, I think.

Steve: Well, and I read on a black screen with amber type, so that would be...

Leo: Perfect for you.

Steve: ...great for low power consumption.

Leo: I just ordered - there's a new Kobo reader, the Libra 2 Color, that uses color eInk. And I'm very intrigued by that. And because it will have some color. It won't have amber on black, probably, like you just said.

Steve: All the color I've seen has been on...

Leo: So it's washed out, yeah, yeah.

Steve: Yes, exactly, very low contrast.

Leo: Well, we'll see. It also has a stylus, and you can take notes and stuff. So, you know...

Steve: I do love my ReMarkable. You turned me onto that.

Leo: Isn't that great? Yeah.

Steve: That is the best thing. It really is...

Leo: Yeah, yeah, yeah. It's great for coding because I use it all the time to sketch out, like, problems and what I'm [crosstalk].

Steve: Oh, that's exactly what I do. I'm a big diagram drawer.

Leo: Exactly, yeah.

Steve: When I'm, like, trying to parse something. It's like, wait a minute, because that's the only way you can deal with the off-by-one problems is do little examples of it.

Leo: Yup, see it. Yup, I'm the same way.

Steve: Okay. So as I said, I'm not going to go through the entire document. It is too long and wonderful. I mean, I already sound like I'm over-caffeinated at this point because I'm so excited by what is in here. It's just astonishing. I mean, again, it's like you took 20 years of this podcast and distilled all of its recommendations that we've come up with on the fly, when things have been obvious and when we've seen something go wrong over and over and over, it's like, okay, when are we finally going to do this? And here it is. And they just didn't miss anything.

Leo: That's great. That's fantastic. Wow.

Steve: It's just, you know. And again, it's not like - again, it's not recommendations, it's law.

Leo: Yeah.

Steve: In the UK. Okay. So I do want to, like, do a little bit of a deeper look into the first one they address because it is crucially important, under the banner "No Universal Default Passwords." They say: "Where passwords are used, all consumer IoT device passwords shall be unique per device or defined by the user. There are many mechanisms used for performing authentication, and passwords are not the only mechanism for authenticating a user to a device. However, if they are used, following best practice on passwords is encouraged." By which they mean, again, they have not softened this as a recommendation. They're saying, you know, encourage people when they use a password to use a good one.

They said: "Many consumer IoT devices are sold with universal default usernames and passwords such as admin/admin for user interfaces through to network protocols. Continued usage of universal default values has been the source of many security issues in IoT, and the practice needs to be discontinued. The above provision can be achieved by the use of pre-installed passwords that are unique per device and/or by requiring the user to choose a password that follows best practice as part of initialization, or by some other method that does not use passwords. For example, during initialization a device generates certificates that are used to authenticate a user to the device via an associated service like a mobile application.

"To increase security, multifactor authentication, such as use of a password plus OTP procedure, can be used to better protect the device or an associated service. Device security can further be strengthened by having unique and immutable identities. Where pre-installed unique-per-device passwords are used, these shall be generated with a mechanism that reduces the risk of automated attacks against a class or type of device. Pre-installed passwords must be sufficiently randomized. Passwords with incremental counters such as 'password1,' 'password2' and so on are easily guessable. Further, using a password that is related in an obvious way to public information sent over the air or within a network, such as MAC address or WiFi SSID, can allow for password retrieval using automated means.

"Authentication mechanisms used to authenticate users against a device shall use best practice cryptography, appropriate to the properties of the technology, risk, and usage." And I should mention that also in here is a complete description of their use of all the

terminology. So here they just said "best practice cryptography." But in an addendum they specifically outline what that means that is required to be used where they use this term. So again, there's nothing that they missed. They said: "Where a user can authenticate against a device, the device shall provide to the user or an administrator a simple mechanism to change the authentication value used.

"When the device is not a constrained device, it shall have a mechanism available which makes brute-force attacks on authentication mechanisms via network interfaces impractical. For example, a device has a limitation on the number of authentication attempts within a certain time interval. It also uses increasing time intervals between attempts. Or the client application is able to lock an account or to delay additional authentication attempts after a limited number of failed authentication attempts. This provision addresses attacks that perform 'credential stuffing'" - it actually says that in this document for consumer IoT devices - "or exhaust an entire key space. It is important that these types of attacks are detected by the consumer IoT device and defended against, whilst guarding against a related threat of 'resource exhaustion' and denial of service attacks."

Incredible. What I just summarized is broken into five individual provisions in the document, but each and every one of them is tagged as "Mandatory." So, for example, if a device offers password-based authentication, it can no longer be shipped from the factory with a default password, and the device must also incorporate proactive defenses against brute force and credential stuffing attacks. It must incorporate some form of lock-out mechanism. This legislation changes everything. It takes the well-understood, but still not often implemented, best practice, you know, even not yet implemented from, you know, high-end enterprise level devices, and mandates its use today for a residential doorbell. This is huge.

Section 5.3 is titled "Keep software updated." And picking some bits from it, for example, it says: "Developing and deploying security updates in a timely manner is one of the most important actions a manufacturer can take to protect its customers and the wider technical ecosystem. It is good practice that all software is kept updated and well maintained. All software components in consumer IoT devices should be securely updateable. When the device is not a constrained device, it shall have an update mechanism for the secure installation of updates." They have in quotes: "'Securely updateable' and also in quotes 'secure installation' means that there are adequate measures to prevent an attacker misusing the update mechanism.

"Measures can include the use of authentic software update servers, integrity protected communications channels, verifying the authenticity and integrity of software updates. It is recognized that there are great varieties in software update mechanisms and what constitutes 'installation.' An anti-rollback policy based on version checking can be used to prevent downgrade attacks. Update mechanisms can range from the device downloading the update directly from a remote server, transmitted from a mobile application, or transferred over a USB or other physical interface. If an attacker compromises this mechanism, it allows for a malicious version of the software to be installed on the device." Meaning that there are provisions for preventing that happening. Thus they're explaining what the danger is.

"An update shall be simple for the user to apply. The degree of simplicity depends on the design and intended usage of the device. An update that is simple to apply will be automatically applied, initiated using an associated service such as a mobile application, or via a web interface on the device. If an update is difficult to apply, then that increases the chance that a user will repeatedly defer updating the device, thus leaving it in a vulnerable state. Automatic mechanisms should be used for software updates. If an automatic update fails, then a user can, in some circumstances, no longer be able to use a device. Detection mechanisms such as watchdogs and the use of dual-bank flash or

recovery partitions can ensure that the device returns to either a known good version or the factory state.

"Security updates can be provided for devices in a preventative manner, as part of automatic updates, which can remove security vulnerabilities before they are exploited. Managing this can be complex, especially if there are parallel associated service updates, device updates, and other service updates to deal with. Therefore, a clear management and deployment plan is beneficial to the manufacturer, as is transparency to consumers about the current state of update support. In many cases, publishing software updates involves multiple dependencies on other organizations such as manufacturers that produce sub-components; however, this is not a reason to withhold updates. It can be useful for the manufacturer to consider the entire software supply chain in the development and deployment of security updates.

"It is often advisable not to bundle security updates with more complex software updates, such as feature updates. A feature update that introduces new functionality can trigger additional requirements and delay delivery of the update to devices. The device should check after initialization, and then periodically, whether security updates are available." Again, the device should check. They said: "If the device supports automatic updates and/or update notifications, these should be enabled in the initialized state and configurable so that the user can enable, disable, or postpone installation of security updates and/or update notifications."

They said: "It is important from a consumer rights and ownership perspective that the user is in control of whether or not they receive updates. There are good reasons why a user may choose not to update, including security. In addition, if an update is deployed and subsequently found to cause issues, manufacturers can ask users to not upgrade their software in order that those devices are not affected." But they're saying, you know, again, secure by default. Secure by policy. If it can update itself, it should. But recognize that there are reasons that it might not.

They said: "The device shall use best practice cryptography to facilitate secure update mechanisms. Security updates shall be timely. 'Timely' in the context of security updates can vary, depending on the particular issue and fix, as well as other factors such as the ability to reach a device or constrained device considerations. It is important that a security update that fixes a critical vulnerability. i.e., one with potentially adverse effects of a large scale, is handled with appropriate priority by the manufacturer. Due to the complex structure of modern software and the ubiquity of communication platforms, multiple stakeholders can be involved in a security update."

Okay. So what we're talking here - and I just shared the tip of the iceberg. But it is all like that.

Leo: I have this vision of you last night or the night before, sitting by the fire, your feet up, you've got a little something to drink, you're reading through this. Your beautiful wife is across the way sitting reading her magazine. And every five minutes you go, "Wow." Or, "Yes." I imagine every step of the way because this is all the stuff you've been saying all this time.

Steve: I know. It's astonishing, Leo. I mean, it is. They even have in there protocols which are used on the WAN should not be exposed - or on the LAN should never be exposed to the WAN.

Leo: They came this close to saying "three dumb routers"; right? I mean, it's impressive. I am very impressed.

Steve: It is astonishing that this thing exists.

Leo: Mm-hmm.

Steve: So again, grc.sc/etsi, E-T-S-I. That will bring this 34 or whatever number of page things it was to you, and it is all like that. What we're talking about here amounts to nothing less than the forced and immediate maturation of more than a decade of lazy consumer product security design.

Leo: Yup, yup, yup.

Steve: End-user security is finally being prioritized over device development and support costs and, yes, even convenience. It had to happen sometime, and all indication is that sooner or later it was going to need to be forced. That happened yesterday.

Leo: Wow. And it's law; right?

Steve: It's law.

Leo: I mean, in the UK this is the law. Wow.

Steve: It is the law.

Leo: That's amazing.

Steve: Yes. And a law with teeth in it. It's not like, you know, oh, you pay a $2,500 penalty. No. It's 10 million, or 4% of your revenue, whichever is greater.

Leo: It's significant, yeah.

Steve: Yeah. So, I mean, basically, like, everything on the shelf that isn't already in compliance. And we'll note that many routers, many consumer routers now are; right? Although I don't think they've gotten - I think they're still, you know, like the default username and password still starts off with admin and password or something. So, but they are updating themselves, and they do have that on by default. So, you know, that's good.

But, you know, one of the reasons, in fact there was a reference somewhere to this crazy, huge, like the Mirai botnet that owe their existence to the fact that these policies have never been required, enforced, or present before. And as a consequence of web authentication being on the WAN, and default usernames and passwords, these things

could just be taken over. And so the UK is finally saying, enough of this. What? You know, come on. You guys haven't gotten your act together in the last decade? Well, we're going to require it now. If you want to sell stuff in the UK, you have to do this.

And it will probably increase cost a little. But as we know, once this is done, it's in a chip. You just stamp it out. And so, yeah, users are going to have to learn that they're going to have to, you know, like read the directions instead of assuming it's admin/admin or just has no password. And like, oh, well, I'll give it one later, and then never get around to it. Instead, every device will have a unique password which they'll have to write down or change to make it one that they want. But it won't be, you know, off the shelf, all of them sitting there on the shelf with the same password to start. And again, that's just, like, one of 40 different topics that they've dealt with.

Leo: Amazing. Amazing. It's amazing.

Steve: It's astonishing.

Leo: Yeah, yeah.

Steve: Okay. So while we were recording last week's podcast, the news dropped, thanks to you, Leo, covering it, that Google's plans for Chrome's full phaseout of third-party cookies would not be occurring this year as had been planned and, needless to say, much anticipated. For this week's podcast I had hoped to follow up on that news to learn and then report on what was going on. The beginning of my research into Google's interactions with the UK's CMA, their Competition and Markets Authority, suggested that everything is actually going quite well overall, only somewhat slower than was expected. My preliminary examination suggested that the UK's concerns are actually more along the lines of whether Google's new system goes far enough.

And actually, having just covered what we covered, I'm not that surprised at this point because they really seem to be getting busy. Where concerns had previously been raised, for example, that smaller advertisers may be disadvantaged, the UK now appears to be discounting those concerns and complaints. But I ran out of time for this research because, Leo, in front of the fireplace as I was last evening, my glass of Cabernet got empty, and I thought, okay, well, we'll tackle this next week. We ran out of space. So I will have the news of this next week because there was lots of material, and it's possible to get a much better understanding of what's going on.

Leo: You need an overflow podcast. That's what you need.

Steve: That's right.

Leo: No one would object if you decided to do one, that's for sure.

Steve: Okay. So a bit of Closing the Loop, just two pieces of Closing the Loop feedback. Guillermo Garca said: "Hi, Steve. Listening to the feedback on the counter race condition," which our listeners had a lot of fun with, he said: "I wonder what would happen to a process that wants to increase the counter if the previous owner of the counter was switched out of context and did not return the ownership before being

switched off. Would this active process get stuck and have to wait for the previous one to regain context and return it? Again, many thanks."

Okay. Many of our listeners reported that they found the discussion of object ownership within a multithreaded environment very interesting. But at no point did I talk about what happens when something goes wrong. Right? I just talked about if everybody does the right thing and behaves themselves, this is how cool it is when everything is perfect. But, for example, notice that nothing actually prevents the shared counter from being incremented by a thread that does not first acquire ownership of the object. In this instance, in the example that I painted, there is no enforcement. It's all and only by agreement among the process's threads. And since they're all part of the same process, it's in their best interest to abide by the rules mutually.

But it doesn't take a deliberate act to mess something up. Bugs happen, as we know. A typical bug in a complex multithreaded environment is that, for example, a thread will acquire ownership, then follow some code path that causes it to fail to release its ownership. At that point, that counter can never be incremented again, and any and all threads that need to may stall, waiting for an object's ownership to be released. And who among us, especially back in the early days of Windows, has not experienced an application lock up and freeze? Or its menuing UI mysteriously stops responding. Or the app apparently dies and refuses to do anything, although it's still there on the screen and looks like it should be going, but it's not. Some things might still be functioning, whereas other things suddenly become non-responsive.

One of the most common causes of such things happening is that somehow the ownership of a shared object was not freed by its owner. Threads can sometimes get into trouble. If a thread, for example, attempts to divide by zero, that thread will be terminated by the operating system. Or if a thread mistakenly attempts to execute some data, an illegal instruction can be encountered and, again, the thread will be immediately killed. In any event, if that thread happened to own some shared objects at the time of its termination, they would likely not be freed. And other threads, or even a re-spawn of the terminated thread, might then be unable to succeed ever again after that. Shutting down and restarting the application might be the only way to clear out such stuck ownership.

Not surprisingly, many solutions for these sorts of problems have been created over time. One of the best things about software is if you've got a problem, there's probably a way to fix it, which is what makes it so fun. Not surprisingly, people have been very clever. For example, there's a system known as "Structured Exception Handling," or SEH for short, which actually allows a thread to protect the system and itself from its own possible misdeeds. I've implemented Structured Exception Handling in assembler, and I've used it when my code had no choice, for some reason, other than to try to do something that might fail catastrophically. And what doing this allowed me was then to try something and not lose control, but to have that failure recoverable, and then I could deal with the consequences. And there are also entirely different ways to manage shared object ownership than that simple exchange instruction which I chose specifically to demonstrate the simplest of all possible solutions, which it is.

But before I close out this conversation, I would be remiss if I did not mention one of the classic problems with multithreaded environments which is known as the "deadlock." A deadlock can be created when two threads each separately own an object, but also need ownership of another object that the other owns. In other words, say that there are two objects, both of which need to be simultaneously owned by a thread in order to complete some work. One thread currently owns the first object, and the second thread currently owns the second object. And each of them needs to obtain ownership of the object that the other one already has.

Both threads will patiently wait for something that will never occur, since neither will relinquish its ownership of the object it owns until it, however briefly, is able to obtain ownership of the object it needs. Which the other thread owns, and it's also waiting for the object the first thread has. So consequently, neither will ever succeed, and we have a classic deadlock, as it's known in computer science.

But Guillermo's question highlights something else that I did not talk about. He talked about multiple processes sharing objects, in other words, inter-process object sharing; whereas all of my discussion has been about multiple threads within a single process, intra-process object sharing. It is possible to share objects between processes. For example, Windows allows this by using unique names to identify objects. Then separate processes that know the common name for an object can open the object to obtain its handle, very much like opening a file by name, after which operating system calls can be used to check the shared object's status, to obtain and release ownership of it, and so on.

And it's also possible to set timeouts while waiting for an object's ownership to be granted. If that amount of time passes, the object wait will be ended, and the waiting process will be notified that the object never became available during the amount of time that the thread said it was willing to wait for it.

And even returning to our original example with the exchange instruction, remember that a thread that wants to obtain ownership attempts to obtain it, and the result of the exchange instruction informs it whether or not it was successful. If it was not successful, it's fully able to decide what to do next. Right? I mean, it's not - it doesn't have to wait forever. It can go do other things and try again later. Or it might ask the operating system to put it to sleep for some length of time. That's an extremely friendly thing to do since the thread is voluntarily giving up the rest of its running time slice, which allows the OS to schedule other threads. Then when the thread is reawakened by the operating system, it can again attempt to obtain ownership, and then decide what to do.

Anyway, I know I'm weird. All this fascinates me. And I have never encountered, as I said before, anything as pure and clean and gratifyingly complex as coding. I get it. It's not for everyone. But if it is, it can be terrifically rewarding. And I know, Leo, that you also love to code.

Leo: Oh, I love it. I live for it.

Steve: One last tiny bit of news. Apparently The New York Times last week picked up on the story of, you know, we were talking about the LexisNexis selling drivers' driving habits. Turns out that The New York Times had a story on Wednesday that General Motors had "accidentally," says GM, enrolled millions of people into its "OnStar Smart Driver+" program. Consequently, if consumers chose not to enroll through the phone app, it would do it anyway. Unenrolling requires consumers to contact OnStar customer support line. However, turns out some people do not trust them and have started stripping the electronic devices out of their cars. So reports The New York Times.

Anyway, just a little bit of follow-up on that. Mistakes had happened, and people, you know, we showed that detailed report last week, and I've seen several others since then, all looking identical because it's coming from the same company. So, Leo, let's take our last break, or no, our second-to-last break. I want to again update our listeners on a bit of sci-fi and where I am in my work. And then we're going to get into our main topic.

Leo: You bet. By the way, the book club loved the Bobiverse. So much so, a lot of them are now on to Book 2. Anthony Nielson, who read Book 1 for the book club - it was Stacey's book club. Stacey was a little reluctant. She wasn't crazy about it because she didn't like Bob. Which is, you know, if you don't like Bob, there's a lot of Bob in the Bobiverse. But Anthony Nielson said, oh, yeah, Book 2 kind of eased his concerns about Book 1. I'm going to have to reread the whole thing because September Book 5 is coming out. Very exciting. The fifth book of the Bobiverse.

Steve: Wait, there's going to be a fourth Bobiverse book?

Leo: Yes. There are four. There's going to be another one.

Steve: You mean a fifth one; right.

Leo: Yeah, it's amazing.

Steve: Cool.

Leo: Anyway, so that was your recommendation. Thank you, and I look forward to hearing more.

Steve: Well, and it came through our listeners. It was from our listeners.

Leo: Yes, I remember that, yeah.

Steve: Who kept saying to me, Steve, you know, check it out. And I have to say that my taste often differs from our listeners.

Leo: It's a little lightweight. It's lightweight. It's fun. It's not...

Steve: Yeah. And actually what I'll be recommending in a minute is also.

Leo: Lightweight?

Steve: Yeah.

Leo: Okay. I'm reading "Hyperion" right now, which is the opposite of lightweight, one of the classic science fiction novels that I never got around to reading, so I'm enjoying it quite a bit.

Steve: Yeah, I think I have the paperback around here.

Leo: Yeah, I mean, it's a classic; right? I mean, that I've never read it is amazing. Now back we go to Mr. G. and sci-fi time.

Steve: Yes, on the science fiction reading front I wanted to mention to our many listeners who've been enjoying Ryk Brown's ongoing Frontiers Saga, that book 11 of 15 in his third of five 15-book story arcs, became available yesterday in the Amazon U.S. store. I received Ryk's announcement that his latest novel titled "The First Ranger" is now available for download in the U.S., although apparently international availability may lag a bit, as is apparently common.

I've received so much feedback through the years from our listeners who've enjoyed following this adventure - and it is one long adventure, now at 41 full-length novels - that I wanted to make sure everyone knew that book 11 was here now. I told some family members and friends, and they just jumped up and down because there's just - if it's right for you, then it is really right for you. It's primarily character driven. He offers us very fully formed individuals with very distinct and at times annoying personalities. And in a way it's a bit like Star Trek, where it's less about whiz-bang science fictional technologies than about how the various characters whom we've come to know over time, how they deal with what comes their way. I find it very satisfying. And in addition to many in this podcast audience, as I said, you know, I've turned friends and family members onto it, and they're completely hooked.

For those who have never looked at the series, as I mentioned, Amazon Kindle is where it is. It's part of the Kindle Unlimited plan. And the novels, if you just buy - if you're not a member of Kindle Unlimited, they're not very expensive if you just purchase them outright. So here's what I know. Anybody starting the first book will know within an hour whether they have just started into a journey that already has 41 additional books waiting for them. And they are just as compelling as the first one. So it's, you know, sci-fi is a passion of mine. We've talked about it, Leo, you and I, through the, you know, we're in our 20th year of the podcast now.

Leo: Oh, my god. And your skin is still beautiful, by the way.

Steve: Many authors, many adventures, a lot of fun.

Leo: Yes, yes. I'm referring to something that happened before the show. Don't worry, folks. You didn't miss anything.

Steve: Okay. And finally, on my own work front, last week I finished updating various GRC pages with the news that 6.1 was now available. This is not 6.1's documentation, which is still quite sorely needed. This is just enough to hold us over until I have email communication up and running, after which my plan is to plow into SpinRite's extreme need for documentation. I cringe whenever someone asks a question that really should be there on the website, documented. But I'm getting there as soon as I can.

But I have a method here. And getting email up and having our podcast audience help me develop an email presence and a reputation is, you know, part of what it takes these days because, you know, as we know, spam is such a problem that the large receivers of email have gotten very picky about the bounce rates and spam flaggings and so forth. So anyway, I want to get that working. And then while that's happening I will be able to start working on the documentation. So anyway, I'm working on email, and I know how

many of our listeners are excited that Twitter will not be the only way to get a hold of me.

So our main topic. Actually, after that first topic, you can see why it was competition for this one. So we have two big ones. Today's podcast title, "Passkeys: A Shattered Dream?" It gets its title from a blog posting from last Friday. It was a thoughtful posting by a guy named William Brown, who is the author of a popular WebAuthn package for Rust. In fact, it's pretty much the WebAuthn package for Rust. It generated significant attention within the security community, and a little bit later within our own listener community because, after I had already chosen it as our topic, everybody started tweeting me links saying, oh, my goodness, what do you think about this? So his WebAuthn package is "webauthn-rs," which describes itself as "WebAuthn Framework for Rust Web Servers."

To remind everyone how and where WebAuthn fits within the overall Passkeys solution, it's the protocol and specification that a Passkeys client on the user's side uses to communicate with a web server that supports WebAuthn. So, for example, just as a web server will offer some form of username and password login, possibly with additional factors such as time-based one-time passwords or something else, such a server might also offer support for the WebAuthn protocol as a means for allowing remote clients to identify and authenticate their identity over a network. In the case of this author's Rust implementation of WebAuthn, he described WebAuthn by writing:

"WebAuthn is a modern approach to hardware-based authentication" - notice he says hardware-based authentication - "consisting of a user with an authenticator device, a browser or client that interacts with the device, and a server that is able to generate challenges and verify the authenticator's validity. Users are able to enroll their own tokens through a registration process to be associated to their accounts, and then are able to log in using the token which performs as a cryptographic authentication. This library" - meaning his that he wrote - "aims to provide useful functions and frameworks allowing you to integrate WebAuthn into Rust web servers. This means the library implements the Relying Party component of the WebAuthn/FIDO2 workflow. We provide template and example Javascript and web asm bindings to demonstrate the browser interactions required."

Okay, now, the only thing I'll note about what this author wrote, as I pointed out right at the start, and it might be significant, is that this appears to have first been written back in the earlier FIDO1 era when hardware dongles were the only way the FIDO group was willing to roll. As we know, the requirement for purchasing a piece of hardware, while potentially ensuring greater security, was finally accepted to be a bar too high. So the FIDO group basically capitulated to allow software-only solutions the privilege of authenticating with what evolved into FIDO2.

So my point is that the author of this WebAuthn crypto library for Rust appears to have started this back at the hardware-only dongle stage of FIDO1, and he simply changed "FIDO1" to "FIDO2" in his introduction. This may be significant for what he subsequently wrote and published last Friday, since the introduction of FIDO2, with its accompanying Passkeys, promised to make his work far more relevant.

Before I share what he wrote Friday, I wanted to note that the section following that brief introduction, I really loved it, it was titled "Blockchain Support Policy." Okay, now, this is for, right, a WebAuthn package that has nothing to do with blockchain.

Leo: I'm thinking it's going to say "none," but okay, yeah.

Steve: Uh-huh, exactly. So he said: "Blockchain Support Policy." And he wrote: "This project does not and will not support any blockchain-related use cases. We will not accept issues from organizations, or employees thereof, whose primary business is blockchain, cryptocurrency, NFTs, or so-called 'Web 3.0 technology.'"

Leo: Right on, right on, right on.

Steve: Period.

Leo: Period. End of statement. Yeah.

Steve: And of course, you know, we know why he said that; right? There's been so much nonsense surrounding, you know, the "Blockchain will solve all of society's ills" nonsense, and especially within the identity authentication space, that it's easy to imagine how much of that this guy may have been fending off through the years. Elsewhere he notes that his library has passed a security audit performed by SUSE Linux's product security, and that other security reviews are welcome.

And as a total aside, I thought it was also interesting that on the topic of compatibility he wrote, under "Known Supported Keys/Hardware," he said: "We have extensively tested a variety of keys and devices, not limited to Yubico 5c, 5ci, FIPS, and Bio; Touch ID, Face ID, meaning iPhone, iPad, MacBook Pro; Android; Windows Hello with TPM; and soft tokens." And then he said, and under "Known BROKEN Keys/Hardware," he notes: "Pixel 3a, Pixel 4 + Chrome does not send correct attestation certificates, and ignores requested algorithms." And he said: "Not resolved." And "Windows Hello with Older TPMs," he said, "Often use RSA-SHA1 signatures over attestation, which may allow credential compromise or falsification."

Okay. So Friday he gave his blog posting the title, as I said, that I reused for today's podcast, although, well, his was "Passkeys: A Shattered Dream," although I added the question mark. His posting was not a rhetorical question. His was meant as a statement. So before I share what William has written, I wanted to take a moment to note that in order to do justice to his choice of words, I'm going to again need to use a term on this podcast that makes me uncomfortable.

Leo: Uh-oh.

Steve: Although the fact that the term was the American Dialect Society's Word of the Year for 2023 suggests that it's a term we're all destined to be encountering more and more often. That term is "enshittification."

Leo: Oh, yeah. Oh, yeah.

Steve: The American Dialect Society's Word of the Year last year, Leo.

Leo: Yeah.

Steve: So I am somewhat eased about its usage due to its lineage and the fact that Wikipedia does not shy away from devoting a rather extensive page to its definition, its description, and its discussion with extensive examples of this happening. Wikipedia describes "enshittification" as "the pattern of intentional decreasing quality observed in online services and products such as Amazon, Facebook, Google Search, Twitter, Bandcamp, Reddit, Uber, and Unity. The term," they write, "was used by writer Cory Doctorow in November 2022, and the American Dialect Society selected it as its 2023 Word of the Year. Doctorow has also used the term 'platform decay' to describe the same concept."

And, for what it's worth, you know, allow me to commend this Wikipedia page to our listeners. I found it to be somewhat gratifying and affirming because, while I was reading and agreeing with everything it said, I felt a little bit less like the crotchety old-timer yelling at the kids to get off the lawn. You know? In other words, objectively and, sadly, deliberately, some things actually are getting worse. It's not just that you and I, Leo, are getting older, and everything seems worse.

And also, not far into this, he refers to - "he" meaning, sorry, done with Wikipedia, "he" meaning William Brown, the author of this - refers to a system whose name is spelled "Kanidm," a term I had never encountered before. Its home page explicitly explains it's pronounced "kar-nee-dee-em." Even though there's no "R" anywhere to be seen. I mean, I guess it's, I would say, "ka-nee-dee-um," maybe. But, you know, "kar-nee-dee-em," which of course reminded me of "carpe diem." And I don't know, maybe it's - I don't know where the name came from.

Leo: It's intentional, I'm sure, yeah.

Steve: Yeah.

Leo: Sade is pronounced "shar-day." I think it's a British-ism. The "R" is inserted, yeah.

Steve: Oh, that's interesting, you're right, it is.

Leo: Yeah.

Steve: Yeah. Okay. Kanidm. So it is a sprawling open source identity management platform developed in Rust by the SUSE Linux project.

Leo: I've been looking for something like this.

Steve: Well, there it is.

Leo: Okay.

Steve: It appears that William's WebAuthn library was adopted into that multi-faceted identify project to provide its WebAuthn functionality. So when we hear him refer to, in

his blog posting, "kar-nee-dee-um," he's refer to his library's significant participation in that project. And that explains also why SUSE's security people have reviewed and approved of his library, because it's the one they chose for their big identity management platform.

Okay. So he said: "At around" - you know, Leo, let's take our last break before I get into this.

Leo: Yeah, yeah. Keep it a secret what he says.

Steve: Then I won't have to break in the middle of it.

Leo: Right, right. This is good. I'm enjoying it. And, yeah, we've all decided here that you can say "enshittification" because it's just a word with a bad word in the middle, but not in the beginning or the end.

Steve: Well, and encrappification...

Leo: Doesn't have the same...

Steve: ...doesn't have quite the same ring to it.

Leo: Yeah. We've been saying "enshirtification," just like Kanidm. You're adding a phantom "R." I don't know. That's confusing.

Steve: But does that mean you have to wear more shirts?

Leo: Need to wear more shirts.

Steve: Yeah.

Leo: Okay, now, back to the enshittification.

Steve: Of Passkeys.

Leo: Of Passkeys.

Steve: Okay. So the author of this WebAuthn library, well regarded, built into SUSE's Linux identity platform, written in Rust, lots of experience from back in the dongle days, the early FIDO1 days, he wrote last Friday: "At around 11:00 p.m. last night, my partner went to change our lounge room lights with our home light control system. When she tried to log in, her account could not be accessed. Her Apple Keychain had deleted the Passkey she was using on that site. This is just the icing on a long trail of enshittification

that has undermined WebAuthn. I'm over it at this point, and I think it's time to pour one out for Passkeys. The irony is not lost on me that I'm about to release a new major version of webauthn-rs today as I write this.

"In 2019 I flew to my mate's place in Sydney and spent a week starting to write what is now the WebAuthn library for Rust. In that time I found a number of issues in the standard and contributed improvements to the WebAuthn working group, even though it took a few years for those issues to be resolved. I started to review spec changes and participate more in discussions. At the time there was a lot of optimism that this technology could be the end of passwords. You had three major use cases: second factor, passwordless, and usernameless. Second factor was a stepping stone toward the latter two. Passwordless was where you would still type in an account name, then authenticate with PIN and touch your security key. And usernameless was where the identity of your account was resident and thus discoverable on the key. This was, from my view, seen as a niche concept by developers since, really, how hard is it for a site to have a checkbox that says 'remember me'?

"This library ended up with Kanidm being, to my knowledge, the very first open source identity management platform to implement 'passwordless,' which is now Passkeys. The user experience was wonderful. You went to Kanidm, typed in your username, and then were prompted to type your PIN and touch your key. Simple, fast, easy. For devices like your iPhone or Android, you would do similar, just touch your Touch ID and you're in. It was so easy, so accessible. I remember how it almost felt impossible that authentication could be cryptographic in nature, but so usable and trivial for consumers. There really was the idea and goal within FIDO and WebAuthn that this could be 'the end of passwords.'

"This is what motivated me to continue to improve webauthn-rs. Its reach has gone beyond what I expected, with parts of it being used in Firefox's authenticator-rs, a whole microcosm of Rust Identity Providers being created from this library and my work, and even other languages' WebAuthn implementations and password managers using our library as the reference implementation to test against. I cannot understate how humbled I am by the influence webauthn-rs has had.

"However, warnings started to appear that the standard, the WebAuthn standard, was not as open as people envisioned. The issue we have is well known: Chrome controls a huge portion of the browser market, and development is tightly controlled by Google. An example of the effect was the 'Authenticator Selection Extension' of the WebAuthn specification. This specification extension is important for sites that have strict security requirements" - like, you know, the government - "because the extension supports the attestation of the make and model of the authenticator in use. If you know that the website's attestation will only accept certain devices, then the browser should filter out and only allow those acceptable devices to participate."

So, like, just to pause here for a second, that would be so cool; right? If your bank, for example, required more than just a browser-based Passkey because it is pure software, but needed a hardware dongle, or needed a biometric, you know, reaffirmation of your identity when you tell it that you want to transfer some amount of money somewhere, then you absolutely want this protocol to be able to specify the type of authentication device that would be used and for the browser to then prompt for that level of authentication.

Anyway, he says: "However, Chrome never implemented it. That alone led to the entire feature being removed from the spec. It was removed because Chrome never implemented it. This demonstrates that if Chrome doesn't like something in the specification, they can just veto it without consequence. Later, the justification for this not being implemented was: We never implemented it because we don't feel that

authenticator discrimination is broadly a good thing. They, users, should have the expectation that a given security key will broadly work where they want to use it." He says: "I want you to remember this quote and its implications: Users should be able to use any device they choose without penalty."

He says: "Now, I certainly agree with this notion for general sites on the Internet; but within a business where we have a policy around what devices may be acceptable, the ability to filter devices does matter." So he says: "This makes it possible to go to a corporate site and apparently successfully enroll a security key, only to then have it fail to register. Even better if this burns up" - you know, consumes - "one of your limited resident key slots which cannot be deleted without a full reset of your device. This might happen since the identity provider rejected the device's attestation." And he says: "That's right. Even without this, identity providers can still discriminate against devices without this extension; but the user experience is much worse, and the consequences far more severe in some cases."

He says: "The kicker is that Chrome has internal feature flags that they can use for Google's needs. They can simply enable their own magic features that control authenticator models for their policy, while everyone else has to have a lesser experience. The greater warning here is that many of these decisions are made at F2F," as he puts it, "Face to Face meetings held in the U.S. This excludes the majority of international participants, leading some voices to be stronger than others. It's hard to convince someone when you aren't in the room, even more so when the room is in a country that has a list of travel advisories for foreign travelers, including 'Violent crime is more common in the U.S. than in Australia,' 'There is a persistent threat of mass casualty violence and terrorist attacks in the U.S.,' and 'Medical costs in the U.S. are extremely high. You may need to pay upfront for medical assistance.'"

Okay, now, the point he's making here is that Google has outrageously outsized power to decide what does and does not succeed in the world, due to their unilateral control of their Chrome browser. That which Chrome does not support, dies. And he's also observing something that might not ever occur to those of us who are happily camped out here in the U.S., which is that, unfortunately, the U.S. can apparently be somewhat frightening and expensive for volunteer open source developers wishing to have their voices heard from other countries. His point is those voices are too easy for Google to ignore.

And Leo, when I was thinking about this, this brought to mind something that Stina Ehrensvard often mentioned to me through the years. After founding Yubico in Sweden, she understood the critical importance of geographic location. So she deliberately uprooted her young family and relocated to Silicon Valley. She knew that, if she was going to succeed, she needed to be where the action was, and specifically to be able to attend face-to-face meetings with Google executives and others. In the list of authenticators on William's webauthn-rs site, Yubico's products are all mentioned first because, when it mattered, she was there in person. And I know, as you know, Leo, truth be told, it's often quite difficult to say no to Stina.

Leo: Yeah, and that's how you met her at RSA coming down the escalator.

Steve: That's right.

Leo: So you're right. In person makes a big difference.

Steve: Yup. Yup.

Leo: But don't ask Marcus about what it means to be an open source developer in the United States, Marcus Hutchins, because of course he was arrested on the tarmac...

Steve: Trying to leave.

Leo: Trying to leave the United States.

Steve: Yup.

Leo: So I understand why there's a little chilling effect on open source developers here.

Steve: Well, and I guess you really do, I mean, I'm sure those travel advisories exist. I don't know how much you have to heed them, but still.

Then, under the topic of, or the subtopic of "The Descent," as he put it, he said: "In 2022 Apple announced Passkeys. At the time, this was really just nice marketing, a nice marketing term for passwordless, and Apple's Passkeys had the ability to opportunistically be usernameless, as well. It was, all in all, very polished and well done. But of course thought leaders exist, and Apple hadn't defined what a Passkey was, exactly. One of those thought leaders took to the FIDO conference stage and announced 'Passkeys are resident keys,' while at the same time they unleashed a Passkeys dev website.

"The issue is described in detail in another of my blog posts. But to summarize," he writes, "this push to resident keys means that physical hardware security keys are excluded because they often have extremely low limits on storage, the largest being 25 for YubiKeys. That simply won't cut it for most people, who have more than 25 accounts." And that, I'll just mention, that's one of the biggest annoyances with the whole Passkeys technology is the requirement for significant storage per Passkey. That is, you know, the big thing that I don't have. The approach that I took with SQRL explicitly avoided that by, you know, being able to create per-domain, similar security per-domain keys, meaning that you only had to have one, instead of this problem, in the case of the YubiKey, they're able to store 25. But once you hit that limit, you need another one.

Leo: I know. That's a frustration for me. I wish they'd add more memory.

Steve: Yeah, yeah. Anyway, William then coins a term that Cory Doctorow might appreciate. He terms the period following the announcement of Passkeys as "The Enshittocene Period."

Leo: I like it.

Steve: Yeah.

Leo: All right.

Steve: The Enshittocene Period. He says: "Since then, Passkeys are now seen as a way to capture users and audiences into a platform. What better way to encourage long-term entrapment of users than by locking all their credentials into your platform; and, even better, credentials that cannot be extracted or exported in any way. Both Chrome and Safari will force you into using either hybrid where you scan a QR code with your phone to authenticate. To use a hardware security key requires clicking through multiple menus. And even their default is not a good experience, taking more than 60 seconds' work in most cases. The UI is beyond obnoxious at this point. Sometimes I think the password game has a better user experience.

"The more egregious offender is Android, which won't even activate your security key if the website sends the set of options that are needed for Passkeys. This means the identity provider gets to choose what device you enroll without your input. And of course all the developer examples only show you the options to activate 'Google Passkeys stored in Google Password Manager.' After all, why would you want to use anything else?

"A sobering pair of reads are the GitHub Passkey Beta and GitHub Passkey threads. There are instances of users whose security keys are not able to be enrolled as the resident key slots are filled. Multiple users describe that Android cannot create Passkeys due to platform bugs. Some devices need firmware resets to create Passkeys. Keys can be saved on the client, but not on the server, leading to duplicate account presence and credentials that don't work; or, worse, lead users to delete the real credentials. The helplessness of users on these threads is obvious, and these are technical early adopters, the very users we need to be advocates for changing from passwords to Passkeys. If these users cannot make it work, how will normal people from other disciplines fare?

"Externally, there are other issues. Apple Keychain has personally wiped out all my Passkeys on three separate occasions. There are external reports we've received of other users whose Keychain Passkeys have been wiped just like mine. Consequently, as users we have the expectation that keys won't be created correctly, or they will have disappeared when we need them most. In order to try to resolve this, the working group seems to be doubling down on more complex JavaScript APIs to try to patch over the issues that they created in the first place. All this extra complexity comes with fragility and more bad experiences, but without resolving the underlying core problems. It's a mess."

And then for the future, he says: "At this point I think that Passkeys will fail in the hands of the general consumer population. We missed our golden chance to eliminate passwords through a desire to capture markets and promote hype. Corporate interests have overruled good user experience once again. Just like ad blockers, I predict that Passkeys will only be used by a small subset of the technical population, and consumers will generally reject them."

Leo: Wow.

Steve: "To reiterate, my partner, who is extremely intelligent, an avid computer gamer, and veterinary surgeon has sworn off Passkeys because the user experience is so crappy. She wants to go back to passwords. And I'm starting to agree. A password manager gives a better experience than Passkeys. That's right. I'm here saying," he writes,

"passwords are a better experience than Passkeys. Do you know how much it pains me to write this sentence? And yes, that means multifactor authentication with time-based one-time passwords is still important for passwords that require memorization outside of a password manager.

"So do yourself a favor." This is what he writes. "Get something like Bitwarden; or, if you like self-hosting, get Vaultwarden. Let it generate your passwords and manage them. If you really want Passkeys, put them in a password manager you control."

Leo: Oh, I agree 100%. Yes.

Steve: "But don't use a platform-controlled Passkey store."

Leo: Yes, yes.

Steve: "And be very careful with physical hardware security keys. If you do want to use a security key, only use it to unlock your password manager and your email.

"Within enterprise, there still is a place for attested security keys where you can control the whole experience to avoid the vendor lock-in parts. It still has rough edges, though. Just today I found a browser that has broken attestation, which is not good. You still have to dive through obnoxious user-experience elements that attempt to force you through the default QR code path, even though your identity provider will only accept certain security models, so you're still likely to have some confused users.

"Despite all this, I will continue to maintain webauthn-rs and its related projects. They're still important to me even if I feel disappointed with the direction of the ecosystem. But at this point, in Kanidm we're looking into device certificates and smartcards instead. The UI is genuinely better. Which says a lot considering the state of the PKCS11 and PIV specifications. But at least PIV won't fall prone to attempts to enshittify it. PIV stands for 'Personal Identity Verification.' It's a standardized physical smartcard system that's heavily used by government and military. The technology to create digital identity cards has been around for a long time, and they are so fraught with their own problems that they aren't really an alternative to solve the web's authentication needs."

So I think that, for me, the thing that's so sad is that Cory Doctorow's term, and the examples of enshittification that Wikipedia documented, make very clear that these are deliberate usury outcomes. The shortest of Wikipedia's examples is what Uber did. Wikipedia writes: "App-based ridesharing company Uber gained market share by ignoring local licensing systems such as taxi medallions, while also keeping customer costs artificially low by subsidizing rides via venture capital funding. Once they achieved a duopoly with competitor Lyft, the company implemented surge pricing to increase the cost of travel to riders and dynamically adjust the payments made to drivers."

So nearly all of the problems William observed in his posting are the things we on this podcast independently noted from the start as inherent problems with the way Passkeys have been rolled out. As I've observed on several occasions, the fact that Passkeys were implemented in a non-portable way, as a vehicle for creating implicit platform lock-in, is almost a crime.

But the new thought that William proposes is something that had never occurred to me. Perhaps it's because I've been blinded by Passkeys' superior public key technology which offers so many potential authentication benefits. Even though the benefits are

theoretical, I've never questioned whether or not Passkeys would eventually become the new standard for the web. But William writes: "At this point I think that Passkeys will fail in the hands of the general consumer population. We missed our golden chance to eliminate passwords through a desire to capture markets and promote hype."

When I read that the first time I was surprised. But at the same time, I have still not adopted Passkeys. I've never registered a single Passkey. I don't have even one, anywhere. I don't encounter websites that offer Passkey authentication, so there's that. But mostly, because authentication matters crucially to me, I want to feel that I'm in control of my authentication. And that starts with thoroughly and deeply understanding it. I do thoroughly and deeply understand Passkeys' underlying cryptography. But as William explains, what's then been done with that underlying crypto has been made deliberately opaque as a means of "just trust us" individual platform lock-in. The problem is, my authentication is far too important for me to entrust to any company that might choose to, dare I say, enshittify it.

Having unique, per-site, insanely long high-entropy passwords that I can touch and feel and copy and paste and see, stored and managed by a cross-platform password manager, which is everywhere I need it to be, allows me to really understand the status of my authentication. One thing I also have is a long and growing list of TOTP one-time passcodes. And the reason I'm absolutely comfortable with that is that, again, they're tangible things that I can control, see, and understand.

Has the entire techie insider industry just been playing with itself this whole time? Have we been imagining that authentication can and should be made entirely invisible because Passkeys can theoretically make that happen? Will end users who don't know anything about the underlying technology say, "Well, I don't know how it works, but it certainly was easy?" But then what about when it doesn't work? What about when someone needs to log on from a device that's outside the provider's walled garden? These are all problems we've previously identified and questions we've asked before. I've always assumed that this was just the typical extreme adoption inertia we always see. It never occurred to me that Passkeys might ultimately fail to ever obtain critical mass and to eventually become more dominant than passwords.

While poking around to get a broader perspective, I encountered a recent piece in Wired titled "I Stopped Using Passwords. It's Great - and a Total Mess," with the intro "Passkeys are here to replace passwords. When they work, it's a seamless vision of the future. But don't ditch your old logins just yet." The author explained that, as William said, things didn't always work. He also noted that having multiple clients all popping up and asking whether you want to save Passkeys with them had become annoying. But the biggest problem he had was remembering where he had stored which Passkey. Now, as someone who spends some time pondering which of the multiple streaming providers carries the show my wife and I have been watching, that definitely resonated.

Our advice at the start of this Passkeys saga was to wait until a single provider offered Passkeys support across every platform that might conceivably be needed, since Passkeys portability was not something that anyone was even talking about back then. In fact, back then it was clearly an overt password lock-in move. So I wanted to share the news that Bitwarden, the solution that William's posting referred to and a sponsor of the TWiT network, earlier announced on the 10th of this month that Passkeys for iPhone and Android clients had just entered beta testing. So I'm very glad to see that my chosen open source password manager will soon be offering Passkey support.

Now what's going to be needed, based upon the experience of the author of the Wired article, will be the ability to assign a single Passkey handler to a platform, much the way we currently assign a handler for a platform's URL links. Having all of a platform's Passkey-aware clients popping up solicitations to store a Passkey with them seems like it

would quickly become annoying. On the other hand, you know, setting up a new Passkey doesn't happen that often.

On balance, I still don't feel much pressure to give up my use of passwords since they're working perfectly for me today. The other factor is that website login has become so persistent that I rarely need to re-authenticate to most sites. Each of the browsers I use carries a static cookie for each of the sites I frequent, so I'm already known everywhere I go. For the foreseeable future, I expect to hang back and wait. The dust is still settling on Passkeys, and Passkeys doesn't solve any problem I have today, even though they're cool.

Leo: If SQRL had only taken off. But you see the problem, which is...

Steve: Anything. Anything new.

Leo: Well, but it's not just that. The platforms aren't going to support it unless they own it.

Steve: Right.

Leo: And it gives them lock-in.

Steve: Right.

Leo: That's why Apple loves this. That's why Google and Microsoft and - they want the lock-in. And that's why I use, and I agree with him, only use Bitwarden or some sort of open source manager that you can at least take with you to do it. But, oh, this is sad because it really - it's a great idea. But the writing's on the wall. Very few websites use it.

Steve: Yes. Yes. And if they start to use it, and then their users have problems with it, they'll pull it.

Leo: Right.

Steve: I mean, it could disappear as an option because it's not worth it to them.

Leo: Not worth it. No benefit.

Steve: Everybody knows how to use a username and password.

Leo: Sigh. SQRL really solved all these problems, and that's sad because it really - it was - oh, well. What can you do? You've kind of gotten over it. I'm still...

Steve: I've gotten over it. I solved the problem. I satisfied myself. And we had a lot of fun developing it and working out all of the edge cases and so forth. And, you know, now I'm on to solving other problems.

Leo: This is where you jump up from your easy chair, grasp your Cabernet, and you shake your fist at the skies and say, "Why, I oughta...." Steve Gibson, GRC.com. That's his home on the Internet, the Gibson Research Corporation. Go there to get SpinRite, the world's best mass storage maintenance and recovery utility. 6.1's out and fantastic. If you've got an SSD, this is the kind of unexpected benefit of it. You can use 6.1 to speed up your SSD. That's fantastic. What an improvement.

Steve: Yeah. As I say, "recover lost performance."

Leo: Yeah. That's his promise, yeah.

Steve: It is data recovery and performance recovery.

Leo: Yeah. We'll have to say mass storage performance, recovery, and maintenance utility. We'll add that.