The Bluetooth Backdoor

Description: Utah passes age verification requirement for app stores. The inside story on fake North Korean employees. Is that a Texas accent? An update on the ongoing Bybit crypto heist saga. The industry may be making some changes in the wake of the Bybit attack. Apple pushes back legally against the UK's secret order. Did someone crack Passkeys? The UK launches a legal salvo at an innocent security researcher. The old data breach we witnessed that just keeps on giving. A bit more Bybit post-mortem forensic news. A lesson to learn from a clever and effective ransomware attack. And what about that Bluetooth Backdoor discovery everyone is talking about?

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-1016.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-1016-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. He has a remarkably good solution to the age verification conundrum, a fantastic story about a fake employee coming from North Korea, and then we'll talk about the Bluetooth Backdoor. It got a lot of press, but is it really a problem? All of that coming up and a lot more on Security Now!, next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 1016, recorded Tuesday, March 11th, 2025: The Bluetooth Backdoor.

It's time for Security Now!. I know you've been waiting all week. Here we are, Tuesday, and the latest security news is here with Mr. Steve Gibson, the king of the hill when it comes to this stuff. Hi, Steve.

Steve Gibson: Hey, Leo. It's great to be with you again, March 11th and Episode 1016. And I was a little jealous of hearing you talk about the 20th anniversary...

Leo: Yeah.

Steve: ...upcoming for TWiT.

Leo: April 13th will be our 20th year of TWiTs.

Steve: Yup. And so you did that for a few months before you said, "Hey, Gibson."

Leo: Yeah, your 20th's coming up.

Steve: "I think we're ready to add a second podcast to our network." Actually, I guess that would create a network; right? It really wouldn't be a network with one podcast.

Leo: Yeah. Until then it was just a podcast, yeah. Yeah, that's right. So your 20th should be in the fall, I guess, yeah? Soon.

Steve: Yeah, it is.

Leo: Well, we could do something special for that. Think about what you want to do.

Steve: Let's just ignore it. No. We're going to let my birthday go by. We're going to let the 20th podcast, 20th anniversary podcast...

Leo: I'm the same, exactly. But I did, you know, on the thousandth episode we had all of the original hosts from Episode 1 back.

Steve: Right.

Leo: And I said, well, I can't do that again. But I thought, really, what's the most important part of all of the things we do, it's our community. It's the people who listen, the people who email, and they chat with us, all the people who are part of the family. So I said, let's celebrate them on April 13th. And I'm asking people to send us videos of when they first started watching, how they watch, you know, just memories, that kind of thing.

Steve: Oh, neat.

Leo: So that'll be a lot of fun. That show will be jam-packed with - we'll have the regular show, as well. But every few minutes we'll drop in a video from a listener or viewer. So if you want to be part of that, just post it on your favorite social with @twit in the posting so we'll see it. Or you can email Leo at Leo.fm and send that to me that way, and that'll work, too. I don't have your fancy mail system. I should just say, "Everybody mail it to Steve." No. Steve has a very clever system, which I should steal, of validating emails before you can email him on a regular basis at GRC.com.

Steve: Well, but you like dipping in on all of those social media places. I mean...

Leo: I do. I do. I do dip in.

Steve: ...you're streaming on 27 of them right now.

Leo: It's unbelievable how many there are. Yeah. They're growing like Topsy.

Steve: So I think that makes more sense for you. For me, it's like - oh, my god, I just did it. I forgot to post on Twitter again. Shoot.

Leo: Don't post on Twitter. Skeet. You've got to skeet, man. Be a skeeter.

Steve: What's that?

Leo: That's Bluesky.

Steve: Like I said, old-school for me, yes. Okay. So...

Leo: All right. Well, you could post on Twitter when I do the first ad, which is coming up. But first I'd like to know what we're going to be talking about today.

Steve: We're going to talk about, well, okay. I just gave this the title of the week, which was the most emailed thing that I saw, which is all of this huffing and puffing about a big, bad, Bluetooth Backdoor that had been discovered and was revealed by a pair of Spaniards, Spanish security researchers last week at the big annual global Spanish security conference. It's an interesting story, and we're going to cover it. But we've got to talk about Utah passing the first age verification requirement for app stores. And I'm going to spend a little time talking about age verification again. We have before, but boy is it a hot topic among our listeners. I get so much feedback from people who are mostly upset at the idea that they need to verify their age on the Internet.

My take is this is as significant as cryptography, as privacy, inasmuch as it's one of those things that - it's a problem created by the fact that cyberspace is different than physical space. So we're going to spend a little time on that. Also we've got a really interesting piece, the inside story on fake North Korean employees written with the details provided by an individual who keeps having these North Koreans trying to get hired by his firm.

Leo: Oh, wow.

Steve: And he says, you know, they really don't sound like they're from Texas. Anyway, we've got an update on the ongoing Bybit crypto heist saga, several more pieces, I mean, for something that is this big, right, there's a lot of tendrils sort of oozing from it, like where did the crypto go, what has happened. The industry looks like it's going to actually respond in some interesting ways, more like in a larger, bigger way. Also, how did this happen? We know more now about the Safe{Wallet} guys and exactly what the exploit was that caught them, that then caused them to get infiltrated and allowed them to pass the attack forward. Also Apple is pushing back against the order that never was in the UK, so we have a little bit of news about that. Also, did somebody crack Passkeys?

Leo: What?

Steve: Something happened.

Leo: I don't know.

Steve: And we'll look at that. Also the UK has launched a legal salvo at an innocent security researcher just because they can.

Leo: Oy.

Steve: Also in addition we have the old data breach, which we all witnessed, which just keeps on giving.

Leo: Oh, no.

Steve: And many people will be glad they're no longer using that particular password manager.

Leo: Mm-hmm.

Steve: We also have some additional Bybit forensic news; a lesson to learn from a clever and effective ransomware attack; and then, finally, what about that Bluetooth Backdoor discovery that everyone is talking about? So I think a lot of interesting stuff for this week's podcast. And a Picture of the Week that is difficult to believe. But it was not - it's not one of those that was blindly posted to the Internet, where people have sent it to me. This was from a listener in the State of Minnesota who said he took this screenshot himself. He said: "I took this screenshot and thought of you." So...

Leo: Oh, I can't wait. I haven't looked at it yet. We'll do as we always do. I will scroll up, absorb it, and then you'll all have a chance to see our Picture of the Week. All that coming up on Security Now!. It's going to be a great show.

I know which password manager you're talking about. And in some ways I feel like we should apologize because for so many years we told everybody to use it. You used it. I used it. We loved it. You had interviewed the guy who created it. But as often happens, private equity got involved.

Steve: Yup.

Leo: And the bottom line became more important than actual security.

Steve: I vetted the technology, and Joe had done everything right. The design was immaculate.

Leo: So sad. Especially, and this is the most vital of all, the Picture of the Week. I like it that you start with the comedy. You always end with the big one.

Steve: Sometimes it is somber, yes, we end on a somber note, like, well, good luck to you.

Leo: What could possibly go wrong? Actually, sometimes these pictures have what's good. So tell me about this picture.

Steve: Okay. This was actually what a listener of ours found when he went to the Minnesota, the State of Minnesota...

Leo: Oh, my god. Okay.

Steve: Like he found it today. Like this is not...

Leo: This is appalling.

Steve: It's unbelievable. The caption I gave it was "What year is this?" And I said, "It seems we still have a ways to go." So this is the login page for the State of Minnesota Unemployment Insurance agency there. And he's tried to put in what looks like a reasonable length password, if you count - we know that the dots that it shows when you're blanking a password don't always correspond to the length of the password. That's for additional security; right? But we're seeing [counting] maybe about 16 to 20 dots. He gets, and it shows an X on the right side of that attempt. And then the page is updated saying "Validation Error(s)." And we have then an enumeration of what's wrong with this password. "Password must not be more than 6 characters."

Leo: Okay.

Steve: And, as if that wasn't bad enough, "Password must not contain any special characters."

Leo: What? So it's six alphabetic characters.

Steve: Yeah, alphanumeric, presumably.

Leo: Oh, maybe alphanumeric, okay.

Steve: Yeah. So you have an alphabet of, what?

Leo: Twenty-six letters and then 10 more digits, yeah.

Steve: Sixty-two, yeah, 62. Or, no...

Leo: Oh, yeah, because lower and upper case.

Steve: Sixty-six, yeah.

Leo: Although I bet they don't care about case if they're doing - who knows what they...

Steve: Oh, my god. And then it's a little confusing because the standard guidance here, underneath the validation error screen, is "Password must be at least 6 characters." And then...

Leo: But cannot be more than six characters.

Steve: And then, "Password must not be more than 6 characters."

Leo: So it's exactly six characters.

Steve: Must be exactly six characters. I mean, Leo, I mean, if this didn't come from a listener who said, "Steve, I had to share this with you," and took a screenshot for me, I wouldn't believe it.

Leo: That's amazing.

Steve: And that's today, 2025. You must...

Leo: Well, it also tells you that they aren't hashing the passwords; right? Because the length wouldn't matter if they were hashing them.

Steve: One would hope. I mean, again, if they're telling you, first of all, if it must be at least six characters and must not be more than six characters, it actually says that on two successive lines.

Leo: It's kind of kooky, yeah.

Steve: You could simplify that by saying, obviously, password must be exactly six characters.

Leo: Exactly six characters, yeah.

Steve: But that would seem a little too extreme, so they're going to make it a little more mysterious, apparently, by saying must be at least six characters, must not be more than six characters. Do the math.

Leo: Did you ever play the password, that guy's password game? Do you know what I'm talking about?

Steve: You mean Mastermind?

Leo: No, no, no, no. There's a fun little game, making fun of...

Steve: Oh, right, right, right. I do know what you mean. Boy.

Leo: This whole rules thing where it tells you the rules, and you have to adjust the password as you go. So let's say monkey123. And then it tells you, nope, you've got to have an uppercase character. Okay. So let's put an uppercase character. And now it says it has to include a special character. So let me add a special character. The digits must add up to 25. Three, six, that means I need to put nine, and then another nine, and a one. Oh, there we go. Your password must include a month of the year. Oh, well, let's fix that. Here we make key. It must include one of our sponsors. Okay. And it goes from there. There's actually 36 rules. It gets harder and harder. It's hysterical. This is at Neal.fun. It's a great kind of take on what we just saw here, which is absurd password...

Steve: Yeah, so, you know, we wondered how it is, Leo, that states keep getting themselves infected with malware and being hit with ransomware. But when you see a page like this, which says, you know, six character passwords, and we don't know about special characters, it's like there is really - like, what explains this? And this is the unemployment insurance site, which, you know, it'd be nice to have some security there. Anyway, wow.

Okay. Listeners of this podcast know how I feel about age verification. In the same way that we need to make peace with the thorny issues surrounding the abuse of the absolute privacy offered by modern encryption, I believe we must also squarely address the problem of verifying someone's biological age in cyberspace, even if that means deciding not to. That is, I'm not saying we have to know. I'm saying this is an issue that we just need to stop punting because we have so far. Unfortunately, having given this issue a great deal of thought, this feels to me like another of those thorny and intractable problems. But okay. Let's explore this a bit. And I want to do that because, boy, is this of interest to our listeners.

So I'm old enough - and I think you are, too, Leo - to be able to collect Social Security.

Leo: I am, yes.

Steve: So I have the legal right, as do you, to sit at my desktop PC and do anything and go anywhere someone my age can legally do, which is pretty much anywhere and anything.

Leo: Yeah.

Steve: But, I also want my privacy preserved while I'm wandering around. Now, I understand, Leo, you've pretty much given up that battle.

Leo: Yeah, I don't care anymore.

Steve: A lot of our listeners, you know our listeners are like no, no, no.

Leo: I don't even recommend it. Everybody should care about privacy. I just - I don't get to because I spend so many hours in the day on the air, and I have no filter, and everybody knows everything about me.

Steve: And Leo, your email address, come on.

Leo: And I just gave out my email address. That just tells you right there I gave up a long time ago. But I don't recommend it. It's just, you know.

Steve: Okay. Okay, good. In the interest of preserving as much privacy as possible, and only disclosing the bare minimum necessary and when necessary, I would argue there is never any need to share an exact date of birth. After all, none of the proposed legislation anywhere says we need to know your birthday. They just want to know how many years you've been around, round up to an integer number the number of - or round down the number of years completed. That should be sufficient. So okay.

But I also don't like the idea of having my age sprayed indiscriminately everywhere I go. So, you know, it should be on an as-needed basis. I'm just sort of talking about a theoretical framework here, like if we were going to try to solve this problem, what would that solution look like? You know, if I go to a website that has a reasonable need to verify my age, and if I agree with its need and elect to provide that, then I should have the option of in some way releasing my integer age to that site one time, and that one time only.

Now, another consideration is that age restrictions vary by region. Right? So in the United States we do not yet have any uniformity across our individual and independent state legislations. They all just kind of make crap up as they go. And internationally, restrictions often vary by country. So it would likely be necessary to be able to assert our country and state of residence as part of this voluntary age and jurisdiction disclosure. Right? Because it matters where we are. This state says you have to be this old. That state says, oh, no, you can drink when you're 12. Whatever.

Alternatively, perhaps I'm a Gen Z'er who just doesn't care at all about having their age sprayed across the Internet. In that case, this theoretical age verifier could be left unlocked with any querying website being informed of such a user's age and jurisdiction on the fly. I'm Gen Z. I don't care. If I'm in a household with younger kids, I could both lock and password- or PIN-protect this feature so that "something I know" would need to be provided any time I wished to assert my age in cyberspace.

So if something like this were to happen, this would be another Internet specification for the W3C, the World Wide Web Consortium, to design and standardize, and it would be

implemented in and dispensed by our web browsers, the way they do all this other stuff for us already. Once this was standardized, any website that was legally obligated to verify its visitors' age, or actually any website that wanted to know because after all they could ask, we don't have to tell them, rather than presenting, you know, that ridiculous "Yes, I'm at least 16 years or older," or 18 years or whatever it is or older button, that site would have returned an HTTP reply header when displaying the site's initial home page. In the Gen Z'er case, where their browser was set to "permanently disclose" or "permanently unlocked," their browser would return a query making the proper assertion, and the site's content would automatically be available if they qualified.

But in the typical case, where a web user wants to exercise some control over the disclosure of this information, the receipt of this reply header would cause the user's browser to display its own uniform pop-up prompt, saying that the site being visited requires the user to verify their age and location, either/or, or maybe it is requesting that information as opposed to requiring it. That pop-up would contain a button labeled "Please verify my age and send my location to this website." If the user agreed, the browser would generate a query containing this information, and the website would open its doors.

Now, in that regard, the model would be very much like the cookie pop-ups that we're all now plagued with, but it would be implemented by the browser, not by the website. So that's where the uniformity in its display would come from, and it would be displayed in the center of the screen, and only when sites required verification.

Now, of course, by this time everyone is thinking "Yeah, okay, fine. But how can any user's web browser possibly know their date of birth and location in any way that cannot be spoofed at will?" And of course everyone's thinking is 100% correct. That's the big problem. And it's not a problem that can be sidestepped, since it's the essential problem. But I wanted to first lay out the rest of this required framework to show that, if that essential problem could be solved, it could be the basis for a workable solution.

Okay. So now let's switch to the news from last week which triggered this re-exploration. Though it received wide coverage, The Verge's headline was "Utah becomes the first state to pass an app store age verification bill." And they followed that with the note that "Meta, Snap, and X are applauding this." So The Verge wrote: "Utah became the first state in the country to pass legislation requiring app store operators" - and, you know, that's Apple and Google; right? - "to verify users' ages and require parental consent for minors to download apps.

"The App Store Accountability Act is the latest kids' online safety bill to head to the governor's desk, as states across the country and the federal legislature have tried to impose a variety of design regulations and age gating requirements to protect minors from online harms. Much of the legislation that has advanced through the states has been blocked in the courts, and the leading bill in Congress failed to pass last year amid concerns that it could limit free expression on the Internet." Right? Our First Amendment is like what always gets marched out in order to say no, no, no, you can't do any filtering.

"Putting the onus on mobile app store operators to verify ages, rather than individual website providers, is something that Meta and other social media sites have pushed in recent months, as legislatures consider a variety of bills that could impose more liability for kids' safety across the tech industry. Apple reportedly lobbied against a Louisiana bill that would have required it to help enforce age restrictions" - you know, Apple doesn't want any involvement in this if they can possibly avoid it - "but recently voluntarily opted to let parents share their kids' age ranges with apps." And we talked about the first phase of that. We're about to talk about the update to that. "Meta spokesperson Jamie

Radice called that 'a positive first step' at the time, but noted that 'developers can only apply these age-appropriate protections with a teen's approval.'

"After Utah passed its age verification bill, Meta, Snap, and X applauded the move in a joint statement and urged Congress to follow suit," meaning let's make this go national, saying, "'Parents want a one-stop shop to verify their child's age and grant permission for them to download apps in a privacy-preserving way.'" They said: "'The App Store is the best place for it.'" And I disagree with that; but okay, we'll get there.

Leo: Parents are going to be very surprised when they ask the parents' age, as well, by the way, but okay. Go ahead, continue.

Steve: Precisely. Because, right, you have to...

Leo: You have to. Everybody.

Steve: "'The App Store is the best place for it, and more than a quarter of states have introduced'" - a quarter of states - "'have introduced bills recognizing the central role app stores play.' Apple spokesperson Peter Ajemian pointed to a white paper the company released last month which emphasizes the importance of minimizing the amount of sensitive data collected on users. Google, which runs the Play Store on Android, did not immediately provide comment on the bill. But others, including the Chamber of Progress,'" whatever that may...

Leo: Well, okay.

Steve: That's the Chamber of Progress.

Leo: We don't want to [crosstalk].

Steve: That's right, "'which counts Meta's European arm as well as Apple and Google among its corporate backers, warn that the bill could put all users' privacy and rights at risk.'" Again, this is why I think this is like a big deal. This is one of these things, one of these sticky wickets that, you know, that cyberspace brings with it that, up to now, we just kind of wanted to, like, let's not...

Leo: But you know what you're seeing, this is all something that Meta wanted. This is all something the social - they didn't want to do the age verification, so they lobbied hard, and of course probably brought big black bags of cash to members of Congress and the State Assembly saying, oh, really, the App Store should be responsible. They're the central authority of all this nonsense.

Steve: Yeah, and I actually do agree with the notion, I think it should go deeper than that. I think it should be on the platform. But because then everybody gets it.

Leo: You could keep it locally if it were just the phone; right? The phone could just say yes or no.

Steve: Yes. The phone, well, in fact we will be talking about Apple has finally capitulated with an API that is what I've been talking about us needing for quite a while. So: "The Supreme Court has long recognized that age verification requirements like those in SB 142 chill access to protected speech for everyone and are therefore inconsistent with the First Amendment." And again, yes, this is a problem; right? I mean, this is not a small thing. This doesn't have an easy answer. It's clear that this runs up against - that legislation to restrict access to the Internet runs up against this notion of unrestricted free speech because we're talking about restricting, based on age, some people's access. But we actually do that now; right? Just not in cyberspace.

So this Chamber of Progress, this legal advocacy counsel, they have a person, Kerry Maeve Sheehan, wrote in a blog post: "SCOTUS is set to weigh in on age verification this year, but in a case that deals specifically with its application to accessing porn sites." Okay. Better that than nothing, I'd say. Maybe we're going to have to chip away at this in order to get where we need to go. "As privacy experts have explained, 'strict age verification confirming a user's age without requiring additional personally identifiable information is not technically feasible in a manner that respects users' rights, privacy, and security.'" And that of course gets back to the point I was making earlier, that it's like, yes, we can invent a framework and a system for doing this. But that last piece is the problem. How do we do it in a way that cannot be easily bypassed and spoofed?

So once again we have political legislators imagining that they're able to dictate the way reality should operate. You know, much as they've been wanting to with encryption. Well, we want everything encrypted, except we need to be able to see things. What? What? But Apple, apparently cognizant of the direction things are going, last month in February published a short, eight-page document titled "Helping Protect Kids Online." I have a link to it in the show notes for anyone who wants to see it, but I'm going to cover it here.

It appears that Apple is grudgingly moving in the direction they need to go, which is to allow their platform to be used as an age verifier, much as they would clearly rather not. Their "Helping Protect Kids Online" document addressed this. Under the topic "Making it easier to set up and manage accounts for kids," Apple wrote: "For years, Apple has supported specialized Apple accounts for kids, called Child Accounts, that enable parents to manage the many parental controls we offer, and help provide an age-appropriate experience for children under the age of 13. These accounts are the bedrock of all the child safety tools we offer today. To help more parents take advantage of Child Accounts and parental controls, we're making two important changes.

"First, we're introducing a new set-up process that will streamline the steps parents need to take to set up a Child Account for a kid in their family." And they keep using the word "kid." I guess that's okay, but it just strikes me as odd, kids. "And if parents prefer..."

Leo: That's right. My folks always said, "You say children, not kids."

Steve: Yeah, exactly.

Leo: Kids are baby goats.

Steve: It seems too informal to me.

Leo: Yeah.

Steve: But okay.

Leo: It's marketing material, that's why.

Steve: Yeah. "And if parents prefer to wait until later to finish setting up a Child Account" - and this was very interesting to me - "child-appropriate default settings will still be enabled on the device." So even if you don't - if a parent just sort of flips a switch to say, yeah, we want this for - we want a child account, it defaults to safe. So they said: "This way, a child can immediately begin to use their iPhone or iPad safely, and parents can be assured that child safety features will be active in the meantime."

Leo: That's because they know parents will do the least possible.

Steve: Exactly. Okay, here you go. Get out of my hair.

Leo: Yeah. So it fails into a safe state, which it should. That's fair.

Steve: Right, it absolutely should. "This means even more kids," they wrote, "will end up using devices configured to maximize child safety with parental controls. Second, starting later this year" - and this is annoying to me. This thing is full of "coming soon" and "later this year." It's like, what? What's the problem here? Just do it.

Leo: How hard could it be?

Steve: Yeah. I know you've had endless handwringing meetings in your ivory tower up there in your golden doughnut. So get it done. Anyway, starting later this year: "Parents will be able to easily correct the age that is associated with their kid's account if they previously did not set it up correctly." What? Okay, now, this is one of my hobbyhorses. Why set the age? Just set the date of birth. Do it once. And, Leo...

Leo: Oh, that's a good point. It automatically updates.

Steve: It's a miracle. It's amazing.

Leo: It's amazing.

Steve: It's as if you had a computer that was able to do division.

Leo: To add one. Oh, well, never mind.

Steve: I don't understand. And then Apple, and this is the big revelation, once they do, parents of kids under 13 will be prompted to connect their kid's account to their family group, if they're not already connected. The account will be converted to a child account, and parents will be able to utilize Apple's parental control options with Apple's default age-appropriate settings applied as a backstop.

Okay. So under the topic - okay. So, for example, it could default to underage; right? And then what the parent does is then insert their child's date of birth instead of their child's age because that's not, as they would say, that's not going to age well. But date of birth? It's automatic. It's a miracle.

Anyway, under the topic, then, "A new privacy-protective way for parents to share their kids' age range," Apple said, again, because, you know, Leo, this is going to have to be thoroughly vetted. We have to make sure that the slide switches are the right size.

Leo: [Crosstalk] button is just...

Steve: "Later this year." Later this year, wait for it, it's coming.

Leo: Yes.

Steve: "Apple will be giving parents a new way to provide developers with information about the age range of their kids." Age range. We're not giving them - we're not going to - really you're just going to pry this information from us.

Leo: Good. It could be "under 13/over 13." That's sufficient; right?

Steve: Yeah. "Enabling parents to help developers deliver an age-appropriate experience in their apps while protecting kids' privacy." They said: "Through this new feature [coming soon] parents can allow their kids [it says 'kids'] to share the age range associated with their Child Accounts with app developers." It's a miracle. "If they do, developers will be able to utilize a Declared Age Range API to request this information [from a platform] which can serve as an additional resource to provide age-appropriate content for their users." How long do you think it took them to come up with this, Leo?

"As with everything we do, the feature will be designed around privacy, and users will be in control of their data. The age range will be shared with developers if and only if parents decide to allow this information to be shared, and they can also disable sharing if they change their mind." That's got to be another slide switch. And probably it's bigger, you know, so you...

Leo: I changed my mind.

Steve: That's right. Flashing red. And it won't provide kids' actual birth dates.

Leo: Ah.

Steve: Wow, what a concept. As I've noted before, a Declared Age Range API is exactly the right solution. Kids use specific iPhones and iPads, and Apple will even have it default in the direction of enforcing safe content. So it makes sense for the device's platform to know the age of its user and for that platform to be able to disclose that information with proper controls. I still think it makes the most sense, as I've said, for parents to set their child's date of birth internally. And as I've noted, they would be free to fudge it either way, depending upon their individual child's emotional maturity and the level of protection they feel most comfortable enforcing.

Leo: That's right. Exactly right. This is such a good solution. You know what, this is what should happen. I think it's only happening now because it's either this or the App Store, and they don't want to do that.

Steve: Right. And of course the App Store can get this from the platform.

Leo: Right.

Steve: So it's a win/win. So then, if Apple insists upon calculating the users' age within large privacy-protecting ranges, while that seems unnecessarily restrictive to me, fine. Apple has already needed to amend those dumb ranges because, well, they're dumb. Okay. They had to add another one. But, okay. If that's what they want to do, then they could do that. And it does serve to give some additional impression of increased privacy.

So in this document, Apple explained their thoughts about all this under the heading "Age Assurance: Striking the Right Balance Between Platforms and Developers to Best Serve the Needs of Our Users." They said: "At Apple, we believe in data minimization, collecting and using" we know that, god do we know that - "collecting and using only the minimum amount of data required to deliver what you need. This is especially important for the issue of 'age assurance,' which covers a variety of methods that establish a user's age with some level of confidence. Some apps may find it appropriate or even legally required to use age verification, which confirms user's age with a high level of certainty often through collecting a user's sensitive personal information, like a government-issued ID - to keep kids away from inappropriate content. But most apps don't.

"That's why the right place to address the dangers of age-restricted content online is the limited set of websites and apps that host that kind of content. After all, we ask merchants who sell alcohol in a mall to verify a buyer's age by checking IDs. We don't ask everyone to turn their date of birth over to the mall if they just want to go to the food court."

Leo: There you go. There you go. Good analogy, yeah.

Steve: Yeah. "Requiring age verification at the app marketplace level" - and here's their point - "is not data minimization. While only a fraction of apps on the App Store may require age verification, all users would have to hand over their sensitive personally identifying information to us, regardless of whether they actually want to use one of these limited set of apps. That means giving us data like a driver's license, passport, or national identification number (such as a Social Security number), even if we don't need it. And because many kids in the U.S. don't have government-issued IDs, parents in the U.S. will have to provide even more sensitive documentation just to allow their child to access apps meant for children. That's not in the interest of user safety or privacy.

"Requiring users to overshare their sensitive personal data would also undermine the vibrant online ecosystem that benefits developers and users. Many users might resort to less safe alternatives like the unrestricted web, or simply opt out of the ecosystem entirely because they can't or won't provide app marketplaces like the App Store with sensitive information just to access apps that are appropriate for all ages.

"By contrast, the Declared Age Range API is a narrowly tailored, data-minimizing, privacy-protecting tool to assist app developers who can benefit from it, allowing everyone to play their appropriate part in this ecosystem. It gives kids the ability to share their confirmed age range with developers, but only with the approval of their parents. This protects privacy by keeping parents in control of their kids' sensitive personal information, while minimizing the amount of information that's shared with third parties. And the limited subset of developers who actually need to collect a government-issued ID or other additionally sensitive personal information from users in order to meet their age-verification obligations can still do so, too.

"All in all, it gives developers a helpful addition to the set of resources that they can choose from, including other third-party tools, to fulfill their responsibility to deliver age-appropriate experiences in their apps. With this new feature, parents will even more firmly be in the driver's seat, and developers will have another way to help identify and keep kids safe in their apps."

So anyway, Apple is going to need to get, you know, over themselves to some degree, I think, and accept that, you know, what they've built in is an Internet portal. That is, you know, these iPads and iPhones are Internet portals. And they're going to have to have some way of filtering the content that children are able to see. I think this does that. You know, they do still have this notion of dividing ages up into segments. I think they have five of them now. I have it here somewhere in my notes. I'm not seeing it right now.

Leo: And maybe that's why they don't do birthdates, because they don't want to even know that much. They just...

Steve: I think you're right. I think you're right. In the same way that they don't want to have the decryption keys for their advanced data protection, they don't want their phone to know the person. I think you're exactly right. That explains this. It was a mystery to me. It's like it seemed so obvious. But you're right. They don't want...

Leo: They don't even want that. They want this vaguest thing that they can get away with, and that would be age range. That makes sense.

Steve: Yup.

Leo: Yup.

Steve: It really does.

Leo: And this is a great solution, frankly. I don't know why they didn't do this right away. This is perfect.

Steve: Yeah. And they can simply not show the apps which require an age that the viewer doesn't qualify for in the App Store. They're just not there for those viewers. You know, they shouldn't see them. They can't get them anyway. So just don't show them. It just doesn't show up in the phone.

Leo: Yeah. And it gives - and the other thing I like about it, it gives parents the ultimate authority. Because only the, I mean, the parent knows what a kid is mature enough to do or not. And if the kid is a 12 year old, but has the maturity of a 16 year old, the parents can say that. He's 16.

Steve: Yes.

Leo: And Apple doesn't get involved. Nobody gets involved. The parent is the right person to decide and give - and if there's a parent who, you know, doesn't care, let's hope they care enough to set a button that says it's a kid's phone, and that would do the closest thing to the right thing.

Steve: Right.

Leo: I like this. You know what I suspect, this sounds like something Apple came up with but hadn't implemented at all. And they know it's going to take some time.

Steve: They just wanted to push back as long and hard as they could. And now it's like, okay, fine. If we're going to start having legislation, then guess what, here's the solution we propose.

Leo: It's probably an iOS 19 feature, and that's why they're saying in the coming - it's like September we'll have it for you. I hope so, anyway, because this would completely short-circuit the whole thing. It would make it doable.

Steve: Now, what this does, though, then, is solve the problem here. It does not solve the problem for Pornhub, where if Congress weighs in, or if the Supreme Court weighs in and says, you know, you must absolutely protect minors from having access to this content, then the only way to do that is for people who do want access to lose their anonymity.

Leo: Well, you know, the State of Texas did in fact create a law which a federal judge put on hold and the Supreme Court heard arguments last month on and will decide upon. And one hopes that the Supreme Court decides in favor of the First Amendment. That's what that federal judge, the district judge in Texas said, this violates the First Amendment. So there are I think at least 20 states that have these porn laws. And what happens, what Pornhub has done is just withdraw from the state.

Steve: Right.

Leo: But that just means there's a lot of other, plenty of other porn sites, or use a VPN, or, I mean, there's all sorts of ways around it. Your phone solution is actually much more bulletproof. And you could say that the phone has to say you're over 21. I mean, you could say that.

Steve: Yup.

Leo: I mean, that's - I mean, an 18 year old probably has their own phone. Maybe even at 16 you're without parental supervision. So, but at that point they should be able to do whatever. If the parents aren't going to get involved, then they could be able to do what they want; right?

Steve: Yeah. Yeah, I mean, so again, we're...

Leo: It's a great solution.

Steve: We're looking at this because we're in cyberspace, and this is something that we've been just sort of like not wanting to deal with so far, and I think that we're finally facing the fact that we've got to answer some of these hard questions.

Leo: Yeah, yeah, maybe.

Steve: We have an easy question, Leo, which is...

Leo: Like who's the next sponsor?

Steve: That's the one I was thinking of.

Leo: I know you so well. All right. Let's take a little break. We have lots more to talk about. Steve's coming back in just a second.

Steve: Oh, we've got a North Korean job interview.

Leo: What are you talking about, North Korea? I'm from Lubbock. We'll talk about that in just a little bit. Steve Gibson. You're watching Security Now!. So glad you're here. All right, Steve. Your rest is over. Back to work.

Steve: Okay. So thanks to a listener of ours, I was made aware of one employer's experience with North Koreans faking their identities for the purpose of attaining employment in the U.S. As we'll see, at one point toward the end of his description, Roger Grimes, whose security industry work we've covered before, says: "I have now spoken with many dozens of other employers who have either almost hired a North Korean fake employee or hired them. It is not rare." So here's what Roger himself experienced.

He said: "You would think with all the global press we've received because of our public announcement of how we mistakenly hired a North Korean fake employee in July of 2024, followed by our multiple public presentations and a whitepaper on the subject, that the North Korean fake employees would avoid applying for jobs at KnowBe4. You would be wrong. It is apparently not in their workflow to look up the company they are trying to fool..."

Leo: Oh, how funny.

Steve: "...along with the words 'North Korea fake employees,' before they apply for jobs. We get North Korean fake employees applying for our remote programmer/developer jobs all the time."

Leo: Wow.

Steve: "Sometimes they're the bulk of the applications we receive. This is not unusual these days. This is the same with many companies and recruiter agencies I talk with. If you are hiring remote-only programmers, pay attention a little bit more than you usually would. North Korea has thousands of North Korean employees deployed in a nation-state-level industrial scheme to get North Koreans hired in foreign countries to collect paychecks until they're discovered and fired. Note that due to UN sanctions, it is illegal to knowingly hire a North Korean employee throughout much of the world.

"To accomplish this scheme, North Korean citizens apply for remote-only programming jobs offered by companies around the world. The North Koreans apply using all the normal job-seeking sites and tools that a regular applicant would avail, such as the company's own job hiring website and dedicated job sites like Indeed.com. The North Koreans work as part of larger teams, often consisting of dozens to over a hundred fake applicants. They're usually located in countries outside of North Korea that are friendly to North Koreans, such as China, Russia, and Malaysia. This is because North Korea does not have a good enough infrastructure - in other words, Internet and electricity - to best sustain the program, and it is easy for adversarial countries to detect and block North Korean Internet traffic.

"The North Korean fake employees work in teams with a controlling manager. They often live in dormitory-style housing, eat together, and work in very controlled conditions. They do not have much individual freedom. Their families back home are used as hostages to keep the North Korean applicants in line and working." Basically, they're slaves.

Leo: That's so awful.

Steve: "They get jobs and earn paychecks, but the bulk of the earnings is sent back to North Korea's government, often to fund sanctioned weapons of mass destruction work. The scheme is much like an assembly line workflow. The North Korean fake employee and their helpers apply for the job, interview, supply identity documents, get the job, get the related company equipment, and collect a paycheck. The North Korean applicant may do all the steps in this process or farm it off to other participants, depending upon the language skills of the applicant and the requirements of the job application process.

"They will often use made-up 'synthetic' identities, use stolen identity credentials of real people in the targeted country, or actually pay real people of Asian ancestry who live in the target country to participate. It turns out there is a burgeoning sub-industry of college-aged males of Asian ancestry who cannot wait to get paid for participating in these schemes. There are Discord channels all around the world just for this. They make a few hundred to a few thousand dollars for allowing their identity to be misused or participating in the scheme. That way, they can interview in person or take drug tests if the job requires that." Wow. So they're like subcontractors of this North Korean scheme.

"Sometimes the North Korean instigator does all the steps of the application process. Sometimes, they just get the job interview and hand it off to others with better language skills for the interview, and sometimes they hand off the job to someone who can actually do the job, and collect a kickback percentage. How the North Korean fake employee accomplishes the hiring and job process runs the spectrum of possibilities. We have seen it all.

"If they actually win the job, they will have another participant in the targeted country pick up the computing equipment sent by the employer and set it up. They're known as 'laptop farmers.' These laptop farmers have rooms full of computing equipment sitting on tables, marked with an identifier of what computer belongs to what company, to keep them straight. They power on the laptops and give the fake North Korean employees remote access to the laptop. Using this scheme, North Korea has illegally 'earned'" - he has in air quotes - "hundreds of millions of dollars to fund its illegal weapons programs over the last few years.

"There have been North Korean fake employee part-time contractors for over a decade, but the fake full-time remote employees took off when COVID-19 created a ton more of fully remote 'work-from-home' jobs. There is far more money to be made. If your company offers high-paying, remote-only programmer/developer jobs, you are likely receiving fake job applications from North Koreans. It is rampant. Hundreds to thousands of companies around the world likely have North Korean fake employees working for them right now. It is common. We regularly get applications from North Korean fake employees. We routinely reject most of them. Occasionally, we accept a few and interview the fake employees to learn more about them."

Leo: Wow.

Steve: Like, deliberately; right?

Leo: That's wild.

Steve: "And to keep up on any possible developing trends. Luckily, so far North Korea does not seem to be changing their tactics that much from our original postings. The signs and symptoms of a North Korean fake employee we described last year still apply today. They're apparently still having great success using them. If you and your hiring team are educated about these schemes, it's fairly easy to recognize and mitigate them. You just have to know and look for the signs and symptoms.

"We recently interviewed 'Mario,' [and he has that in quotes], 'Mario,' supposedly from Dallas, Texas. Here's part of his resume." I have it in the show notes on page nine. So it shows: Dallas, Texas and then a 754 phone number, and they blocked out the rest. Mario something @gmail.com, blacked out.

Leo: I'm Mario.

Steve: That's right. At the very top line, GCP, Python, C#, Rust, Microservices, Cloud, and he has in parens AWS and Azure. Then all of the counseling about how to prepare it like a one-page rsum. So it says: "Experienced Senior Software Engineer with 8+ years of experience in Python, C#, Rust, microservices, REST/GraphQL API development, cloud infrastructure, (AWS and Azure), and containerized application deployment. Specialized in cloud-native architectures, high-availability systems, and secure coding practices. Passionate about building scalable, reliable, and high-performance applications for cybersecurity and enterprise solutions."

Leo: I'd hire him. This guy looks good.

Steve: Like, where do you sign?

Leo: Yeah.

Steve: Then under Experience, from 07/22 through 12/2024, a Senior Software Engineer with Cloud-Native Microservices & Security, Amazon Web Services (AWS), and Remote. And during this time he, Mario, designed and developed cloud-native microservices in Python, C#, and Rust, ensuring high availability and fault tolerance. Well, what's what you want. Built secure and scalable REST and GraphQL APIs, enabling seamless interoperability between cloud services and enterprise applications. Checked that box.

Leo: A buzzword festival.

Steve: Led cloud infrastructure development on AWS (using Lambda, EC2, S3, RDS, and DynamoDB); and Azure (AKS, Cosmos DB, Key Vault, Event Grid). Whoo. Implemented zero-trust security models, incorporating OAuth 2.0, JWT authentication, and end-to-end encryption. Developed containerized application. And this goes on and on and on. But, you know, one full page of this is what I can do for you.

Leo: Patrick said he wouldn't hire him because of the use of what looks like Comic Sans in the header. That right there, that's - he's out. He's out.

Steve: That is a bad choice of font.

Leo: I think it was Tekton or one of those architectural fonts. But yeah, probably not very professional.

Steve: So Roger wrote: "We have hidden Mario's last name and contact information because it is the name of a real American."

Leo: Oh, interesting.

Steve: "Who is likely unaware that his identity has been hijacked and used..."

Leo: Interesting.

Steve: So, like, when people go and check him out and google him and look him up, oh, look, there he is. He's a real guy.

Leo: Just like the Jackal did. You go to the cemetery, you get a child that's died young and then go to get the birth certificate. And then you get the passport in their name. Oh, no, that's just a TV show. But anyway, same idea.

Steve: And he said: "So who's likely unaware that his identity has been hijacked and used in this scheme, and we don't want hiring companies to accidentally be given the rogue contact information and think they have a real employee candidate." He said: "'Mario' [in quotes] claimed that he was an American citizen who was born and raised in Dallas. Despite this, he had a fairly strong Asian accent."

Leo: Yee haw.

Steve: "Likely North Korean. The Mario who showed up for our Zoom interview had the same voice as the Mario we interviewed over the phone during the first stage of the application process."

Leo: Now, we should say they're in on this; right? I mean, they know, they're just playing with this guy because they want to learn about this, yeah.

Steve: Yes.

Leo: So they're ready.

Steve: As he said, occasionally they go ahead and do an interview, even though they are highly suspicious from the get-go, because they want to, like, stay up to date on what North Korea is doing. So he said, in this case he said, I loved this, "The Mario who showed up for our Zoom interview had the same voice as the Mario we interviewed over the phone during the first stage of the application process. But sometimes they're different."

Leo: I have a cold today.

Steve: Wow.

Leo: Sometimes the American who they're using as a patsy who's doing the interview probably; right?

Steve: Right. So he said: "We had three KnowBe4 people on the Zoom call, including myself." Which as we'll see comes in here in a minute. He said: "Over the next 45 minutes, we asked all sorts of questions that would be asked of any real developer candidate. Whenever we asked a question, Mario would hesitate, spend 5 to 15 seconds repeating our question, and then come back with the perfect answer - most of the time. It was clear that Mario or someone participating with him was typing the question subject into a Google search or AI engine and repeating the results. Mario started off by saying how he had a special interest in social engineering." Roger here writes "no kidding" because of course this whole thing is social engineering.

Leo: That's what he's doing. Yeah.

Steve: Yeah. And security culture. He mentioned "security culture" over and over. He said: "I soon realized that if you go to our main website, we say 'security culture' all over the place. He was repeating phrases he found on our website. But he was very friendly and smiling; and his English was heavily accented, but not super hard to understand most of the time." Although born and bred in Dallas, eh. He said: "I would say that based solely on this first part of the interview, if we were unaware of what was going on, we would all have liked what he said and how he responded. He was friendly and smiley, and we liked him."

Leo: Aww.

Steve: "Mario claimed on his rsum and in person to have programmed for Amazon, Salesforce, and IBM. He supposedly has the exact advanced programming skills we had advertised."

Leo: Of course.

Steve: "I wish all job applicants knew as well how to best match what we advertised in a job ad with what they responded." Of course it was all fake, but still. "During his initial statements, he said he had a personal interest in cryptography and security. When it came time for me to ask technical questions, I used his mentioned interests as the basis for my questions. I started off by asking if he'd ever done post-quantum cryptography, and if he had implemented it in his past projects. He hesitated, repeated the question, and then gave me an excellent dissertation on post-quantum cryptography, including mentioning NIST, which is probably the top search result you get when researching post-quantum cryptography, and a list of the various post-quantum cryptography standards."

Leo: Maybe he listens to Security Now!. You know? Maybe he just...

Steve: Maybe.

Leo: Yeah.

Steve: "I asked him if his previous projects were all using post-quantum cryptography. He said yes."

Leo: No.

Steve: "Which is absolutely untrue."

Leo: Right.

Steve: "Almost no American company is currently implementing post-quantum cryptography. Strike one. I asked what post-quantum encryption standard he liked the most. He said Crystals-Dilithium. It is a digital signature algorithm, not encryption. He frequently mixed up encryption algorithms, like AES, with hashes like SHA-2 and digital signatures like Diffie-Hellman. Strike two for someone who is really into cryptography and regularly does post-quantum crypto."

Leo: He should have listened to the show better. He obviously was drifting off at some point.

Steve: Yeah, he was. He was like, you know.

Leo: Wasn't paying close attention.

Steve: He was fascinated by the sponsors.

Leo: Yeah.

Steve: "I asked what size an AES cipher key would need to be to be considered post-quantum strength. This seemed to throw him for a loop, and he wasted more time than usual. Finally he replied '128-bits.' That's wrong. AES keys have to be 256-bits or longer to be considered resilient against quantum cryptography. Strike three on the technical questions. He wrongly answered every technical question I asked. At this point, I decided to throw out a random bad fact that any normal U.S. candidate should be able to spot and correct. I said, 'Bill Gates, CEO of Microsoft, says that all future programming will be done by AI agents. What do you think?'

"Okay, now, Bill Gates has not been the CEO of Microsoft since '08, but most people outside the industry would likely think Bill Gates was still the CEO because that's how the media often references him, as the 'former CEO of Microsoft.' He's still a cultural icon associated with Microsoft. This is the type of mistake that a North Korean employee who does not have great access to the Internet would make."

Leo: Aha. Gotcha.

Steve: "And sure enough, Mario repeated the fact that Bill Gates was the CEO of Microsoft instead of the current CEO, Satya Nadella. Mario did give a great answer on agentic AI and programming using AI agents. If he were a real employee, I would give his answer top points, well, except for not noticing my CEO switch-a-roo.

"Finally, with the technical part of the interview over, we switched to the 'personal' questions. If you are concerned that you may have a North Korean fake employee candidate on your hands, it cannot hurt to think of and ask for cultural references that anyone in your country or region should readily know, but that would be harder for a foreigner with limited knowledge of the culture to understand. One of my co-interviewers asked him what he did in his free time. This seemed to surprise him. My co-worker asked if he likes any sports. He said he loved badminton."

Leo: Okay.

Steve: Okay. "Which he probably did not realize that, although super popular in Asian cultures, is not among the top sports if you grew up in Dallas, Texas."

Leo: No.

Steve: "Or nearly anywhere in America. Sure, there are plenty of people who play badminton, especially Americans of Asian-American ancestry; but it is an unlikely response out of all the possible responses you could offer. I asked how excited he was that the Cowboys won the AFC. I figured he would not know that the Dallas Cowboys got creamed and did not win the AFC. For one, they're in the NFC and not the AFC conference division."

Leo: See, I would have missed that one. So I don't know.

Steve: "He again hesitated, but then seemed to get that I was mentioning the Dallas Cowboys and that they had been eliminated from contention. I was surprised this one did not trip him up as much as I thought it would."

Leo: See, the right answer is I don't follow sport ball.

Steve: Right.

Leo: If you really were a geek.

Steve: Right. Ask me a question about badminton, and I got you.

Leo: Yeah. I don't think badminton's so disqualifying, to be honest.

Steve: No, no. "My co-worker said he was going to visit Dallas soon and did the candidate have any favorite food spots. Mario said his mother's cooking."

Leo: Oh, good answer.

Steve: He said: "I thought that was a great response so he did not have to look up any restaurants in Dallas. So my co-worker persisted, asking the candidate if they had any restaurants to recommend. Mario did not. I offered up the 'book repository.'"

Leo: Oh.

Steve: "One of the most famous tourist sites in Dallas, where people are dying to eat their 'Nashville hot chicken.'"

Leo: No.

Steve: Mario wholeheartedly agreed with my recommendation.

Leo: Oh, god. Whoopsies.

Steve: "My co-worker asked the candidate if there was anywhere in the world he would want to travel. In our hidden Slack channel, my co-worker said that when he asked this question of North Korean candidates, their eyes always lit up."

Leo: Oh.

Steve: "And they got excited." Yeah. "Sure enough, Mario began to excitedly describe his dreams of visiting Paris and South Africa."

Leo: That's sad.

Steve: And Roger said: "I think it was at this point that we all began to have some empathy. Yes, we were dealing with a fake job candidate who was trying to steal our money, or worse; but in reality, this was a young man likely forced to do what he was doing, destined never to receive any big salary or visit those dreamed-of vacation destinations. It's strange, but I think we started to feel a little ashamed at conducting a fake interview. So we stopped and asked if he had any questions. The normal job candidate would likely ask more about the job, the tools used, the benefits and things like that. Mario had no questions other than how many other people we were interviewing and how he was doing in the job interview.

"We ended the job interview. We had not picked up any new tactics or information, other than noticing that a lot of the North Korean fake employee candidates lately had been claiming to have been born and raised in Dallas, Texas, and all with very heavy accents. However, the last fake employee interview switched from a heavy Asian accent from the initial phone interview to a savvy Pakistani person whom we interviewed on Zoom." And then they said: "He must have been hired to handoff the interview."

"I've now spoken with many dozens of other employers who have either almost hired a North Korean fake employee, or actually hired them. It is not rare. And sometimes the fake employees, when discovered, switch to a ransomware encryption scheme or steal

your company's confidential data and ask for a ransom, so it is not always just about getting a paycheck. Employers beware."

Leo: I think, though, it's really interesting to say he felt some sympathy for the guy because I feel the same way, you know, when you kind of punk people who are trying to scam you on the phone. Often they're as much the victim as you would be. And it's good to remember that.

Steve: Right, they're in some big farm, and some robo dialer is connecting them to you, and unfortunately they're being rated on their success percentage.

Leo: Right, right. Wow. What a story. Wow, that's just - that's fascinating.

Steve: So I wanted to be sure that the employers and interviewers among our listeners were fully aware and appreciated the degree to which these fake North Korean employee farm scams are real. I have a link on page 12 of the show notes to Roger's far more detailed 21-page report on this, which also has - it is heavily linked to other resources. It's KnowBe4.com, and then the URL has the title North-Korean-Fake-Employees-Are-Everywhere. So anyway, I just - I wanted to put this on our listeners' radar because it's really not something you want to do. And of course it is the case that the moment they start to feel that they might be found out, that the jig might be up for them, there is a serious danger of them switching their use of your network to ransomware and exfiltration and extortion. So, you know, it also needs to be taken seriously.

Leo: And next time say "Billy Bob's Texas," if they ask you what restaurant you like.

Steve: And the other thing that needs to be taken seriously, Leo.

Leo: Yes. Our fine sponsors?

Steve: That's right.

Leo: You are getting really good at this, Steve. It's scaring me a little bit.

Steve: Your job is secure, my friend. Don't worry.

Leo: Back to you, Steve.

Steve: Okay. So before I share the latest news on the movement of 1.5 billion USD worth of stolen Ethereum tokens, I should note that the 10% bounty on that $1.5 billion is not $150,000, as I apparently mistakenly said...

Leo: It's a little more than that.

Steve: ...last week, yeah. Several of our listeners politely wrote to say, "Uh, Steve, that would be $150 million in bounty."

Leo: A little more.

Steve: "Not $150,000." So indeed, I am happy to share that correction. And thank you, you know, listeners who are paying attention.

Okay. So what do we know today? Crypto.news reports under their headline "Nearly 20% of Bybit's $1.46 billion in stolen funds 'gone dark,' said Bybit's CEO. CEO Ben Zhou now says nearly 20% of the funds are now untraceable, less than two weeks after the exchange lost over $1.4 billion in a highly sophisticated attack by North Korea-backed hackers. In a March 4th post on X, Zhou shared an update on the ongoing investigation into the cyberattack, revealing that around 77% of the stolen funds remain traceable, but that nearly 20% has 'gone dark' through mixing services.

"The hacker primarily used THORChain, a cross-chain liquidity protocol which came under scrutiny for unwillingness to prevent DPRK hackers from laundering the funds, to convert stolen Ethereum into Bitcoin. Approximately 83% of the funds, or around 1 billion, were swapped into bitcoin across nearly 7,000 [that's actually 6,954] individual wallets." So as I said, this was that dispersion that I talked about, where just they scattered it to the four corners, you know, in order to make it, you know, much more difficult to track and to chop this huge amount into smaller, less suspicious-size chunks.

"As Crypto.news reported earlier," they wrote, "while other protocols took steps to prevent the movement of stolen funds, THORChain validators failed to take meaningful action. Pluto, a core contributor, resigned in protest after nodes rejected a governance proposal to halt ETH transactions. Of the stolen funds, 72% (900 million) passed through THORChain, which remains traceable, says Zhou. However, around 16% of the funds, totaling just shy of 80K Ethereum, valued at around 160 million, have now gone dark through ExCH, a centralized crypto mixing service.

"Zhou mentioned that the exchange is still waiting for an update on these transactions. Another portion of the funds, around 65 million, also remains untraceable as Zhou says more information is needed from OKX's Web3 wallet. In addition, the Bybit CEO revealed that 11 parties, including Mantle, ParaSwap, and blockchain sleuth ZachXBT, have helped freeze some of the funds, resulting in over 2.1 million in bounty payouts so far."

Leo: So that's 2.1 billion in saved money; right?

Steve: Yeah.

Leo: That's pretty good. That's a good start, yeah.

Steve: Yeah. So Bybit is recovering some of their stolen money in return for those 10% bounty payouts which, you know, allows them to keep those moneys legally. Which, you know, is certainly the way to do it. And Leo...

Leo: I would check Mario in Dallas, if I were them. I just, I don't know, I think that's one possible place to look.

Steve: Well, you know, maybe one of his cousins is part of the Lazarus Group. Wow. And just listen as I'm sharing what Crypto.news wrote. It's like, this is clearly just a world unto itself.

Leo: Yes, that's right.

Steve: When you talk about all this stuff moving back and forth and sloshing around, and it's just...

Leo: It's the wild west, absolutely.

Steve: Yeah, really.

Leo: And while there were for a while some attempts to regulate it with the FCC, I think that horse has left the barn.

Steve: Doesn't seem to be much interest at the top.

Leo: Not anymore.

Steve: On doing that. So, yeah. Okay. Also, meanwhile, what of the Safe{Wallet} service whose malicious infiltration was the proximate cause of this very expensive breach in the first place? Crypto.news also reports under their headline "Safe Wallet responds to Bybit hack with major security improvements," which is what you call, you know, closing the door after the horses have all left the barn.

They wrote: "Ethereum-based crypto wallet protocol Safe implemented 'immediate security improvements' to its multisig solution following a cyberattack on Dubai-based exchange Bybit on February 21st. North Korea's Lazarus stole [as we know] over 1.4 billion in Ether from Bybit's Ethereum wallet by exploiting vulnerabilities in Safe Wallet's UI. The infamous hacking group injected hostile JavaScript code specifically targeting Bybit, siphoning more than 400,000 ETH. To prevent further attacks" - again, whoops - "Safe placed its Wallet in lockdown mode before announcing a phased rollout and a reconfigured infrastructure." Right.

"Martin Koeppelmann, co-founder of Safe, said in a March 3rd X.com post that their team had developed and shipped 10 changes to the UI. The protocol's GitHub repositories showed updates to 'show full raw transaction data now on the UI' and 'remove specific direct hardware wallet support that raised security concerns,' among other upgrades.

"Bybit CEO Ben Zhou discussed the incident on the When Shift Happens podcast with host Kevin Follonier, explaining that the attack occurred shortly after he signed a transaction to transfer 13,000 ETH. Zhou mentioned using a Ledger hardware wallet, but noted that he couldn't fully verify the transaction details. The issue is known as 'blind signing,' a common vulnerability in multisig crypto transactions. Safe's latest updates aim to provide signers with more detailed transaction data, according to Koeppelmann.

"In response to a post from Kyber Network CEO Victor Tran regarding industry-wide security efforts, Koeppelmann emphasized the importance of collaboration, but noted that immediate damage control remains the priority, writing: 'We're still in the putting-out-fire mode, but once we have that behind us we need to come together and improve overall frontend and transaction verification security,' Koeppelmann stated, adding that 'This will take involvement of many parties to solve it for good.'" Okay. So it does sound as though in the longer term broader sense some good will eventually come from all this, though it certainly was expensive, an expensive lesson. There is so much liquidity sloshing around in this crypto world, it still boggles my mind. You know, I mean, we're just, like, oh, yeah, we lost 1.2 billion, well, maybe $1.5 billion. But, you know, we've got that covered.

Leo: Almost as if they built a technology designed to easily anonymously transfer funds from one party to another.

Steve: You think? Wow.

Leo: It's almost as if it was designed to do that.

Steve: Wow. And that there's a lot of interest in having that done. You know, like, oh, hey, I've got some application for anonymous big dollar transactions.

Leo: Used to be you had to bank with a big suitcase to hold all the cash. Now it's this little tiny wallet, and it can hold billions.

Steve: And you have to have Mario, who is a big guy, able to, you know, because those luggages are heavy when they [crosstalk].

Leo: Yeah, I just watched - I was just watching an old heist show called "Heat," where it was back in the day so you had to rob...

Steve: Oh, classic movie.

Leo: Yeah, Al Pacino, Robert De Niro.

Steve: Yup.

Leo: And you had to rob, you know, armored trucks to get cash, or rob banks and make [crosstalk].

Steve: That's where the money is, yup.

Leo: And they brought these big bags in to carry the cash out. And it's like, no, no one brought - you know what? No one with any brains robs banks or armored trucks

anymore. That's not the way to get it. You just need a little thumb drive and a computer.

Steve: And hire some geeks.

Leo: A few geeks named Mario.

Steve: That's right. Okay. So meanwhile, back on the encryption front, last week the BBC reported under the headline "Apple takes legal action in UK data privacy row." This of course would be in response to a legal demand whose very existence Apple is prohibited from divulging. But it seems that particular cat is well out of the bag. So the BBC wrote: "Apple is taking legal action to try to overturn a demand made by the UK government to view its customers' private data if required. The BBC understands that the U.S. technology giant has appealed to the Investigatory Powers Tribunal, an independent court with the power to investigate claims against the Security Service. It is the latest development in an unprecedented row between one of the world's biggest tech firms and the UK government over data privacy.

"In January, Apple was issued a secret order by the Home Office to share encrypted data belonging to Apple users around the world with UK law enforcement in the event of a potential national security threat. Data protected by Apple's standard level of encryption is still accessible by the company if a warrant is issued, but the firm cannot view or share data encrypted using its toughest privacy tool, Advanced Data Protection.

"Last week, Apple chose to remove ADP from the UK market rather than comply with the notice, which would involve creating a 'backdoor' in the tool to create access. Apple said at the time it would never compromise its security features and said it was disappointed at having to take the action in the UK. The UK's order also angered the U.S. administration, with President Donald Trump describing it to The Spectator as 'something that you hear about with China.' Tulsi Gabbard, U.S. head of intelligence, said she had not been informed in advance about the UK's demand. She wrote in a letter that it was an 'egregious violation' of U.S. citizens' rights to privacy, and added that she intended to determine whether it breached the terms of a legal data agreement between the U.S. and the UK.

"The Financial Times, which first revealed Apple's legal action, reports that the tribunal case could be heard in the next few weeks, but may not be made public. The Home Office refused to confirm or deny that the notice issued in January even exists. Legally, this order cannot be made public. But a spokesperson said: 'More broadly, the UK has a longstanding position of protecting our citizens from the very worst crimes, such as child sex abuse and terrorism, at the same time'" - wait - "'at the same time as protecting people's privacy.'" Because we want both. "The UK has robust safeguards and independent oversight to protect privacy, and privacy is only impacted on an exceptional basis, in relation to the most serious crimes, and only when it is necessary and proportionate to do so."

Leo: Yeah. I believe that, for now.

Steve: The intent, the intent...

Leo: The intent is good.

Steve: Yes, the intent is good.

Leo: I don't deny that.

Steve: Now, myself being a glass half full sort, I'm still holding out hope that Apple's initial move will have shaken up the UK's legislators sufficiently for them to allow Apple's appeal to succeed, and for Apple's very public shot across the bow threat to pull their strongest encryption entirely from the UK will be sufficient to put this troublesome issue back to bed for a while. We'll see.

The unresolved question is, given that we now have the technology to create and enforce absolute privacy of communications and data storage, in a modern democracy which is designed to be by the people and for the people with elected representation in government, do the benefits of this absolute privacy obtained by the overwhelming law-abiding majority outweigh the costs and risks to society created by its abuse by a small criminal minority?

Don't know. The trouble is that individual governments may decide these issues differently, yet the Internet is global and has always promised to be unifying. When we stand back to look at these issues surrounding privacy through encryption and the challenges presented by the biological ages of Internet users and the perceived need to filter their access to this global network, what becomes clear is that up to this point these fundamental issues and concerns, created by cyberspace having very different rules from physical space, have largely been ignored until now. It feels as if this has all happened so quickly that society has been busy catching its breath, you know, waiting for the dust to settle, waiting for services to be developed and to mature, waiting for those who govern us to catch up. It appears that our societies are finally gearing up to deal with these issues. We've had a really interesting first 50 years of this, Leo. What are the next 50 going to look like?

Leo: Yeah. Well, that's a question we're all asking in a variety of ways. You know, listening to this makes me think that Apple is probably the party that leaked, you know, they're not supposed to reveal that they've received this request. But now that I think about it, they probably leaked this off the record to a couple of news agencies who took it and ran. And that gave Apple the cover then to continue to do what they did, which is pull Advanced Data Protection and appeal. The appeal is kind of like our FISA court. The appeal is to a secret court.

Steve: Right, a tribunal in this case.

Leo: And you may never know, you'll never hear the arguments pro or con. And you may not even know the result. The only way we'll know is the canary that Apple has put out now, which is pulling ADP from England.

Steve: Yup.

Leo: It's very interesting.

Steve: Very, very interesting.

Leo: Which, you know what, we're really on the cusp. We could go either way in all of this right now.

Steve: I do, yes, it feels to me like, you know, the pressure has been mounting. And it's like, as they say, it's going to blow.

Leo: It's going to blow. Let's just hope it blows in the right direction.

Steve: Well, and, you know, whichever way it goes, I mean, it may be that we had a decade or so of privacy. Remember those ridiculous days when you couldn't export a key greater than 128 bits?

Leo: Yeah, 56. It was 56 bits.

Steve: It was 40 bits, 40 bits.

Leo: Forty, that's right, it was really low.

Steve: It was the limits.

Leo: So they could crack it, basically.

Steve: Basically yes. So because it was like, oh. And cryptography was classified as a munition. Legally it was a munition because you were unable to export munitions to foreign hostile countries. So, I mean, maybe it's going to be that crypto is outlawed.

Leo: Yeah.

Steve: Or maybe some compromise will be made. Maybe it will be necessary for anyone who wants to offer it to offer it selectively, and for there to be a master key. Or maybe governments will just say, okay, it's more important to have it than not. You know, more benefit is derived from it than harm is created from it.

Leo: Well, ultimately I think, if you care, you probably should now act to secure strong encryption. The good news is it's fairly easy to implement locally. You can do it.

Steve: That's exactly it. And that is ultimately the argument is, if it is outlawed, only the bad guys will use it.

Leo: Only outlaws will use it, yeah.

Steve: Yeah.

Leo: And people who care about their privacy. And I think this is, you know, why everybody should just learn a little bit of crypto.

Steve: Well, and of course we've been advocating TNO, Trust No One, encryption, or PIE, Pre-Internet Encryption. The idea is, if you encrypt it yourself, then it doesn't matter what happens after it leaves your control.

Leo: Right. Yeah. That's the key. Don't put it on iCloud. Encrypt it and then put it on iCloud. And you're fine; right? They don't have the key to it now. Then of course people come to your house, but that's a trouble for another day. All right. Sorry. I didn't mean to interrupt.

Steve: Okay. I think we should take a break.

Leo: Oh, okay, we can do that.

Steve: Because we've got two more small things to talk about, and then our big topic. So...

Leo: Oh, yeah. Well, this would be a good time, then. All right. Glad you're here. We're watching Security Now!. We're listening to the master. I feel like I should be sitting on the floor with my legs crossed, just listening to the master as we learn about all of this stuff. And it's great; isn't it? We're learning so much. Thank you, Steve. I don't say thank you enough, but thank you for what you do. It's really, really valuable for all of us. We appreciate it. And for Mario in Dallas, who learned everything he knows about AES from this show.

Steve: Mario, listen to those post-quantum post-show episodes again. You're missing out on a few of those questions.

Leo: Yeah. We did some good stuff on that. You can really get that down right; yeah. Practice. All right. On we go with the show, Steve.

Steve: So I wanted to let our listeners know that if they encounter reports claiming that there's a flaw that's been found in Passkeys, the truth is somewhat more nuanced.

Leo: Oh, I hope so because this is scary.

Steve: Yeah. It wasn't a flaw in Passkeys. But there was a problem found. There was a very specific and difficult-to-perpetrate account takeover flaw that was only possible due

to URL link navigation mistakes which had been made in mobile Chrome and Edge. They fixed it back in October of last year. Mobile Safari fixed it in January of this year, and Firefox patched the problem last month in February.

At one point in the Passkeys FIDO flow, mobile browsers are given a link with the scheme FIDO://, unfortunately, that they were all allowed to navigate with that URL. And that's where this really subtle, very difficult to implement but still possible sort of end-around was created. But once the three browsers all started blocking this FIDO:// scheme from being navigable, then that small loophole which a researcher had discovered, very clever guy, was closed, and Passkeys returns to being what we want, the extremely robust network authentication solution that the world needs it to be.

Okay. So I don't know what's going on in the UK. First, of course, as we know...

Leo: I think they don't know, either, as a matter of fact.

Steve: They don't. They order Apple to accomplish the impossible by decrypting data for which the UK knows Apple does not hold the keys. Then I read that a court in the UK had demanded that a U.S.-based security researcher remove their reporting of an embarrassing cyberattack and data breach which occurred at HCRG, which was formerly known as Virgin Care, one of the largest independent healthcare providers in the UK. So seeing that made me curious.

So I first found a nice summary of the situation which TechCrunch reported. They wrote: "A U.S.-based independent cybersecurity journalist has declined to comply with a UK court-ordered injunction that was sought following their reporting" - that is, this cybersecurity journalist's reporting - "on a recent cyberattack at UK private healthcare giant HCRG.

"Law firm Pinsent Masons" - which is the UK firm. So "The UK law firm Pinsent Masons, which served the February 28th court order on behalf of HCRG, demanded that DataBreaches.net 'take down' two articles that referenced the ransomware attack on HCRG. The law firm's notice to DataBreaches.net, which TechCrunch has seen, stated that the accompanying injunction was 'obtained by HCRG' at the High Court of Justice in London to 'prevent the publication or disclosure of confidential data stolen during a recent ransomware attack.'" What? You know, they wanted to report, they wanted to prevent the reporting of the attack, which is not at all the same as preventing the disclosure of confidential data. They apparently felt, well, the fact that we were attacked should be confidential.

Leo: No one should know about that. That's a secret.

Steve: No, certainly not. That would embarrass us.

Leo: Yes.

Steve: What would our shareholders think, and all of those people? You won't even believe how much data was stolen. Anyway, the firm's letter states that if DataBreaches.net disobeys the injunction, the site may be found in contempt of court, which "may result in imprisonment, a criminal fine, or having their assets seized."

"DataBreaches.net," writes TechCrunch, "run by a journalist who operates under the pseudonym Dissent Doe, declined to remove the posts, and also published the details of the injunction in a blog post Wednesday. Dissent, citing a letter from their law firm Covington & Burling, said they would not comply with the order on grounds that DataBreaches.net is not subject to the jurisdiction of the UK injunction [no kidding] and that the reporting is lawful under the First Amendment in the United States, where DataBreaches.net is based. Dissent also noted that the text of the court order does not specifically name DataBreaches.net, nor reference the specific articles in question. Just says you're bad."

So TechCrunch says: "Legal threats and demands are not uncommon in cybersecurity journalism, since the reporting often involves uncovering information that companies do not want to be made public. But injunctions and legal demands are seldom published over risks or fears of legal repercussions. The details of the injunction offer a rare insight into how UK law can be used to issue legal demands to remove published stories that are critical or embarrassing to companies. The law firm's letter also confirms that HCRG was hit by a 'ransomware cyber-attack.'" So now they've even admitted that as a consequence of this.

Okay. So that made me interested enough to go to the source, where I discovered some additional head-shaking detail which picks up where TechCrunch left off. Remember that the site is being represented by Covington & Burling, and in the UK we have the firm Pinsent Masons. So on his site, the subject of this injunction Dissent Doe wrote: "When Jason Criss of Covington & Burling [his firm] sent an email to Pinsent Masons informing them that DataBreaches.net is a U.S. entity with no connection to the UK, and that neither the UK nor the High Court of Justice has any jurisdiction over this site, that should have been the end of the matter; right? But it wasn't, and that's partly why DataBreaches is reporting on this.

"Yesterday morning, DataBreaches.net received an email from its domain registrar that it had been served with the injunction by Pinsent Masons, and that if DataBreaches did not remove the two posts in question within 24 hours, this website would be suspended. The two posts were not even particularly exciting. They mainly summarized some of SuspectFile's great reporting and linked to those posts. For those who would like to see what HCRG or the court demanded I remove, the posts can be seen at," and in his posting he provided two links, which I've duplicated here. One is UK: "More details emerge about ransomware attack on HCRG by Medusa." And the second link is "Medusa Unveils [get this] Another 50TB of Stolen Data from HCRG Care Group, Giving Greater Insight Into the Scope of the Breach."

He said: "DataBreaches informed the registrar" - that is, their domain registrar - "that the injunction was not valid and that DataBreaches.net is not under the jurisdiction of the High Court of Justice or of the United Kingdom. Jason Criss of Covington & Burling also notified the registrar that not only was DataBreaches.net a U.S. entity, but as the site's domain registrar for many years, they could see for themselves that the site was registered to a U.S. person at a U.S. postal address with a U.S. telephone number.

"Later yesterday, the registrar responded: 'Since your lawyer has already sent notice to the complainant, Pinsent Masons, we confirm that we will not be taking any action on your domain, DataBreaches.net."

Leo: Yes. Good.

Steve: Yes. "Additionally, we will be informing Pinsent Masons to contact your lawyer directly should they have any further issues. This ticket is now closed."

Leo: Woohoo!

Steve: "Pinsent Masons did not respond to Monday's email notification by Jason Criss that this site was not under UK or High Court jurisdiction. And at no time yesterday did Pinsent Masons contact the domain registrar to say that it was withdrawing the demand for the removal of the posts. That, too, was surprising. Is it over? Or will there be more? DataBreaches hopes it is over."

Leo: There's a little TWiT connection with this. It was Iain Thomson at The Register, a regular on our shows, who revealed this in The Register, and even has a screenshot of the site and the ransomware notification on it. So it's pretty hard to deny it at this point. And it's out there. And, you know, thank you, Iain.

Steve: Wow, yeah.

Leo: Doing good work, as always.

Steve: So, you know, a major firm like Pinsent Masons must be fully aware of the First Amendment free speech protections...

Leo: There's no First Amendment in the UK.

Steve: But, you know, we're here; right?

Leo: Yeah.

Steve: And they certainly knew that DataBreaches.net was a U.S.-based website, registered in the U.S. So it had to be pure baseless intimidation.

Leo: Yeah, of course.

Steve: Somewhere, some stuffed shirt at the UK healthcare provider was annoyed by the fact that this embarrassingly massive 50TB data breach of their systems was being reported on, and decided to aim their law firm at the reporter. You know, just sort of, you know, maybe we can make it go away. Wow.

Okay. Get a load of this one. Everyone's going to hear a very familiar name pop out of this little piece of news, which reads: "The FBI has recovered 23 million worth of crypto stolen from Chris Larsen, the co-founder and executive chairman of the Ripple cryptocurrency, which trades under XRP, or is named XRP. The recovered funds are just a small part of the tokens stolen from Larsen in January of last year. The funds were estimated at over 110 million last year but are now worth over 700 million." And here it comes. "Hackers stole the Larsen funds by first stealing password stores from password manager LastPass in 2022."

Leo: Oh.

Steve: "Since the attack, the hackers have been slowly cracking passwords and emptying crypto wallets. As of May 2024, over $250 million worth of crypto assets had been stolen using the data obtained from LastPass." Okay, now, remember at the time we talked about this. Bad guys largely don't care, could not care less about random people's laundry. They want one thing, which is money. So they're known to be targeting any crypto passwords suspected of being stored in LastPass vaults.

With LastPass's failure to increase the repetition counts of their PBKDF system, accounts which had been created in the early days of LastPass were left with very low or even in some cases zero iteration counts of their hashing algorithm. This made cracking the passwords protecting those early adopters extra easy. Our advice at the time, for anyone who had stored crypto access passwords in LastPass was to immediately create a new wallet and transfer the assets from the now unsafe wallet into the newly created wallet. We can see why that advice, when taken, could help to protect people from exactly this problem. And this was the great problem was that this massive blob of data was everybody's vaults, which were encrypted, but in some cases not strongly enough encrypted. And so over time you can do offline decryption in order to obtain people's data in the clear.

Leo: Wow.

Steve: Also, in more post-mortem news, we're still learning more about the early genesis of that attack which ultimately affected Bybit. The North Korean hackers compromised, we know, the multi-signature wallet provider Safe{Wallet}. It turns out this was conducted through a social engineering attack which targeted one of its developers. And remember, social engineering is now the way these things are happening more and more. Pretty much, you know, a lot of the other infrastructure has been shored up and tightened up. Social engineering, the human factor, is still - has now become the weakest link. According to a new post-mortem report, the point of entry appears to have been a malicious Docker file...

Leo: Uh-oh.

Steve: ...that was executed on one of the employees' computers. The Docker file deployed malware that then stole his local credentials. The attackers then used the developer's AWS account to add malicious code to the Safe{Wallet} infrastructure which targeted a specific multisig wallet which was used by the Bybit cryptocurrency exchange. And so that's the chain of events. Social engineering attack, guy downloaded and installed a malware containing Docker file, ran it on his machine. It deployed malware on his computer. That malware grabbed his AWS credentials, sent that back to the bad guys. They used that to get into Safe{Wallets} infrastructure, make the changes, and then infect the Bybit transaction. The change that Safe has made is to now display, prominently display the transaction details which they hadn't been fully bothering to display until now.

So they're just trying to make the transaction event more transparent in the hope that that will help people catch any further problems. It's a little bit like, you know, how right now everyone kind of glazes over when they look at a bitcoin wallet ID. It's just like gibberish. And so you just copy and paste it. Well, if you can make it somehow more

obvious that what you've pasted is not what you copied, then that would help you catch clipboard attacks. So that.

I wanted to share a terrific look at how a Windows-centric network, that is, okay, a network, an enterprise using secured Windows systems, nevertheless was hit by ransomware, even though they had strong and effective malware protections in place. A security research group has been tracking the Akira ransomware group that we've referred to a few times. What they found as they dug into a forensic reverse engineering of a distressingly successful attack was interesting, and it was surprising even to them. Here's what they shared.

They wrote: "Until the compromise, this incident had followed Akira's typical modus operandi. After compromising the victim's network via an externally facing remote access solution..."

Leo: Oh, I was just talking about that. Oh, yeah.

Steve: Uh-huh. "The group deployed AnyDesk, a remote management and monitoring tool, to retain access to the network, before exfiltrating data. During the latter stages of the attack, the attacker moved to a server on the victim's network via remote desktop protocol, RDP once again. Akira commonly uses RDP" - Akira being the bad guys, right, the ransomware group - "as it enables them to interact with endpoints and blend in with system administrators, who use RDP legitimately. The threat actor initially attempted to deploy the ransomware on one of the Windows servers as a password-protected zip file win.zip, that contained the ransomware binary win.exe. However, the victim's endpoint detection and response (EDR) tool immediately identified and quarantined the compressed file before it was unzipped and deployed."

Leo: Oh, see? You're fine. You're safe. Everything is good.

Steve: So it works; right?

Leo: Yeah. But...

Steve: "At this point, the threat actor likely realized they had alerted the EDR tool and would not be able to evade its defenses. They therefore pivoted their approach. Prior to the ransomware deployment attempt to this Windows server, the attacker had conducted an internal network scan to identify ports, services, and devices."

Leo: First thing you do, yup.

Steve: "This network scan identified several Internet of Things (IoT) devices on the victim's network, including webcams and a fingerprint scanner. These devices presented an opportunity to the threat actor to evade the EDR tool and deploy the ransomware successfully. The threat actor likely identified a webcam as a suitable target device for deploying ransomware for three reasons. First, the webcam had several known critical vulnerabilities, including remote shell capabilities and unauthorized remote viewing of the camera. Second, it was running a lightweight Linux operating system that supported command execution as if it were a standard Linux device."

Leo: The camera was?

Steve: Well, that's - everyone builds, I mean, Linux is...

Leo: What could possibly go wrong?

Steve: That's right. Got Linux in your camera.

Leo: Holy cow.

Steve: "Making the device a perfect candidate for Akira's Linux ransomware variant."

Leo: Wow.

Steve: "Third, the device did not have any EDR tools installed on it. Why would it? That left it unprotected. In fact, due to the limited storage capacity, it's doubtful that any EDR could be installed on it. But the ransomware could. After identifying the webcam as a suitable target, the threat actor began deploying their Linux-based ransomware with little delay."

Leo: Oh, my god.

Steve: "As the device was not being monitored, the victim organization's security team were unaware of the increase in malicious Server Message Block (SMB) traffic to and from the webcam to the impacted server..."

Leo: Oh, my god.

Steve: "...and the webcam successfully fully encrypted the servers on the victim's network."

Leo: Oh, my god.

Steve: "Akira was thus able to encrypt files across the victim's network."

Leo: Boy, I mean, this answers the question when you say, you know, protect your IoT devices. Oh, so they could get into my camera. What's the big deal? Or my light bulbs. Well, they can actually launch ransomware from these devices.

Steve: Yes.

Leo: Oh, my god.

Steve: Yes.

Leo: Wow.

Steve: I thought this was a super interesting case.

Leo: No kidding.

Steve: Here, as you said, the vulnerable IoT device was not the initial entry point. The honor belonged to some unspecified remote access solution running on a Windows machine.

Leo: As it often does, yes.

Steve: But even though the IoT device wasn't in their way...

Leo: No, wasn't their way in. That's not how they got in.

Steve: It wasn't their way in, exactly.

Leo: Yeah, yeah.

Steve: It was not their way in. The attackers needed an unprotected host for their malware. They were unable to run their ransomware on any of the Windows systems or servers on the network because all of those systems were being protected by effective real-time EDR (endpoint detection and response) security.

But their network scan had discovered some Linux-based webcams, and that's all they needed. And the security of those cams was quite lacking, which made their jobs even easier. So they loaded their malware into the cam's RAM, and it reached out over the network, using Windows file and printer sharing SMB (server message blocks) protocol to read and write back the encrypted files.

Under the prevention and remediation section of their report, the security firm wrote: "Preventing and remediating novel attacks like this one can be challenging. At a minimum, organizations should monitor network traffic from their IoT devices and detect anomalies. They should also consider adopting the following security practices." And what do you think their number one first recommendation was? They wrote: "Network restriction or segmentation."

Leo: Zero Trust.

Steve: "Place IoT devices on a segmented network that cannot be accessed from servers or user workstations or restrict the devices' communication with specific ports and IP addresses."

Leo: Oh, and all it needs is three routers. Actually, a VLAN would do it; right? To segment it?

Steve: Yup.

Leo: Yeah.

Steve: A VLAN would do it.

Leo: Yeah.

Steve: Yup. You know, it takes more work, and it can limit functionality, and it means you cannot just randomly plug anything in anywhere you'd like. So some ongoing network management discipline will be needed, too, always. But this company learned that lesson the hard way.

Leo: Put your IoT devices on a separate VLAN.

Steve: Yeah.

Leo: And don't give them access to the secure VLAN.

Steve: Yeah.

Leo: Wow. That's a great story.

Steve: Isn't that really just...

Leo: I can't believe that there's enough RAM and memory in a webcam running Linux to, I mean, obviously memory's cheap now. Right? So you're going to run Linux on this. I wonder how many IoT devices are running some little Linux kernel in the background. Why not? It's a free operating system. Why not? Wow. Great story. All right. I am proud of myself, Steve, because I saw this Bluetooth Backdoor story, I read it, and I decided not to do it on TWiT on Sunday. There was just something fishy about it. Tell us what happened. Tell us all about it.

Steve: You got it exactly right, my friend. Okay. So I deliberately titled today's podcast the Bluetooth Backdoor because that's what nearly all of the tech press has been calling it. But in this instance it does feel like the appropriate use of that loaded term, if it was

right. So okay. Last Saturday BleepingComputer's headline was "Undocumented backdoor found in Bluetooth chip used by a billion devices." And in fact, probably the reason it's made so much news was that there are so many of these things. It is the most popular chip used by radio-connected Bluetooth and WiFi connected IoT devices.

A Chinese firm Espressif, it's the ESP32, which is like it's the go-to chip. It costs nothing. It's two euros for one of these things. They're just amazing little 32-bit processors. So there's more than a billion of them. Actually it was a billion as of two years ago. 2023 the Chinese site was saying, yeah, we've made more than a billion of these things. So it's a lot more than that now.

Anyway, so last Saturday BleepingComputer said "Undocumented backdoor found in Bluetooth chip used by a billion devices." Then the next day, on Sunday, they softened that headline, saying: "Undocumented commands found in Bluetooth chip used by a billion devices." And to explain the change they wrote: "After receiving concerns about the use of the term 'backdoor' to refer to these undocumented commands, we've updated our title and story. Our original story can be found here." And in that "here" was a link, and I got a kick out of the fact that they actually link to the Internet Archive for a copy of their own previous page. So okay.

This podcast has spent some time batting this issue of "when is a backdoor not a backdoor." Right? You know, would forcing Apple to deliberately and publicly redesign their Advanced Data Protection iCloud synchronization and backup to incorporate a Master Key, be adding a backdoor? You know, in this instance I would say no because this feature of ADP, which would then be added to ADP, the master key, would be neither secret nor malicious; whereas the classic definition and use of the term "backdoor" is both. You know, it definitely needs to be secret. And if it's not secret, it cannot be a backdoor. So that leaves us with the question of malice. In Apple's case there's clearly no malice anywhere. Thus the term "backdoor" fails to qualify for what Apple has apparently been asked for by the UK on both of those counts.

So what about today's news of what nearly everyone is calling a backdoor? We know for sure that what a pair of Spanish security researchers discovered lurking in an astonishingly widely used Chinese microcontroller chip was at least undocumented and also maybe powerful and prone to abuse if it were to become known by a malicious party. But that part is not even clear. All the reporting said, oh, my god. But I'll explain to you what I did and what happened. The intent of why these instructions, these commands, 29 of them, were left undocumented, will never be known. My guess is it's just because they're not that important, not because they were meant to be super secret and allow something to be done.

Okay. So here's what we know. And this is from BleepingComputer's updated coverage after they toned down the language and backed away from the use of the term "backdoor," which was the right thing to do. They said: "The ubiquitous ESP32 microchip made by Chinese manufacturer Espressif and used by over a billion units as of 2023 contains undocumented commands that could be leveraged for attacks. The undocumented commands allow spoofing of trusted devices" - okay, and we'll get back to that later - "unauthorized data access [maybe], pivoting to other devices on the network [that's a variation of the first case], and potentially establishing long-term presence." Okay, because it has flash RAM.

"This was discovered by two Spanish researchers with the security firm Tarlogic who presented their findings at RootedCON in Madrid." This was last week. "A Tarlogic announcement shared with BleepingComputer reads: 'Tarlogic Security has detected a backdoor in the ESP32, a microcontroller that enables WiFi and Bluetooth connection and is present in millions of mass-market IoT devices.' Okay. So that's where everyone got

the idea that there was a backdoor; right? The big, you know, the firm themselves, the discoverers of this, clearly labeled it a backdoor in their presentation."

They said: "Exploitation of this backdoor would allow hostile actors to conduct impersonation attacks and permanently infect sensitive devices such as mobile phones, computers, smart locks, or medical equipment by bypassing code audit controls." Okay. Again, we'll come back to that. But eh.

"The researchers warned," wrote BleepingComputer, "that ESP32 is one of the world's most widely used chips for WiFi and Bluetooth connectivity in Internet of Things (IoT) devices, so the risk is significant. In their RootedCON presentation, the Tarlogic researchers explained that interest in Bluetooth security research has waned, but not because the protocol or its implementation has become more secure. Instead, most attacks presented last year did not have working tools, did not work with generic hardware, and used outdated or unmaintained tools largely incompatible with modern systems."

Now, I should explain that they're taking this position because that's the thing that they created. What they actually did was create a new set of tools which are modern, which are multiplatform, and which offer the ability to explore Bluetooth connectivity. So their main thing was that they're solving the problem for researchers, and then they used it to do some research, and that's what led them to this discovery.

They said: "Tarlogic first developed a new C-based USB Bluetooth driver that is hardware-independent and cross-platform. This provided direct access to the hardware without relying on OS-specific APIs. Armed with this new tool, which enables raw access to Bluetooth traffic, Tarlogic discovered hidden vendor-specific commands (Opcode 3F) in the ESP32 Bluetooth firmware that allowed low-level control over Bluetooth functions."

Now, again, BleepingComputer got it exactly right. Armed with this new tool, which was written in C, which was a hardware-level driver that did not rely on OS-specific APIs, so it was direct to the hardware, they discovered Opcode 3F in the ESP32 Bluetooth firmware that allowed low-level control over Bluetooth functions. Oh, and Ghidra was also involved, the famous NSA-sponsored reverse engineering tool that helps to reverse-engineer firmware. So they were looking at the firmware in the ESP32 and had a tool that let them poke at the hardware. Bleeping Computer said: "In total, they found 29 undocumented hardware commands, collectively characterized as a 'backdoor.'" And now BleepingComputer has that in quotes.

Leo: Oh, good.

Steve: Yeah, "...that could be used for memory manipulation to read or write RAM and Flash, MAC address spoofing for device impersonation, and packet injection. Espressif has not publicly documented these commands, so either they are not meant to be accessible, or they were left in by mistake." I think the third, there's a third option. They didn't think it was necessary. They're not all-powerful Oz commands. They just actually don't matter. You don't need them, so they didn't bother mentioning them.

Leo: Right, right. They're there for their internal use.

Steve: Well, no. They're actually there - okay. Okay. So they're there - well, okay.

Leo: I'm sorry. I'll shut up.

Steve: So we have a CVE issued, 2025-27840. So BleepingComputer said: "The risks enabled by these commands include malicious implementations on the OEM level and supply chain attacks. Depending on how Bluetooth stacks handle HCI commands on the device, remote exploitation of the commands might be possible via malicious firmware or rogue Bluetooth connections." Well, not rogue Bluetooth connections. And if you've got malicious firmware, then you're already on the device with malicious firmware. So who cares?

They said: "This is especially the case if an attacker already has root access" - again, if you already have root access, you're already on the device - "planted malware, or pushed a malicious update on the device" - again, already on the device - "that opens up low-level access. In general, though, physical access to the device's USB or UART interface would be far riskier and a more realistic scenario." Actually, it's the only possible scenario.

The researchers explained: "In a context where you can compromise an IOT device with an ESP32, you will be able to conceal an Advanced Persistent Threat inside the ESP memory and perform Bluetooth or WiFi attacks against other devices" - yeah, rogue device - "while controlling the device over WiFi or Bluetooth." Sure, if you're using your own firmware on your rogue device. "Our findings would allow the full takeover of ESP32 chips and the gaining of persistence in the chip via commands that allow for RAM or Flash modification." Okay, sure. "Also, with persistence in the chip, it may be possible to spread to other devices because ESP32 allows for the execution of advanced Bluetooth attacks." You would need those other devices to be vulnerable, and no one says they are.

Okay. So BleepingComputer said that they had contacted Espressif for a statement on the researchers' findings, but they had not received any comment. And I think that's because the Chinese people said, what? Who cares? Yeah. Okay.

So next we need to look at what the researchers have explained about their own technology. I've edited it down somewhat to remove the market-speak and the redundancy. They said the ESP32 - you know, I'm going to skip this because it turns out it doesn't matter.

Leo: Bottom line.

Steve: The rest of their posting talks about the broader scope of their mission, which is to create a platform to support Bluetooth security audits, which is certainly a very worthwhile endeavor.

So what have we got here? Okay. The Bluetooth HCI defines the boundary between - and I should have said that what they talk about in their presentation is this, oh, we found undocumented commands in the Bluetooth HCI. Over and over and over they say that. That defines the boundary between the host processor and the Bluetooth hardware controller. You know, that's what that is. HCI is the abbreviation for Host Controller Interface. And the jargon's become standardized. Our listeners will have often heard me talking about adding AHCI support to SpinRite 6.1. AHCI is the Advanced Host Controller Interface that was created to manage SATA-connected mass storage devices. So HCI, Host Controller Interface, is a generic reference describing the hardware boundary, the register set, between a peripheral device and its processor. The processor talks to the peripheral by writing into these registers.

So this Spanish security group designed and developed a technology, created a new capability that would allow them to audit the operation of Bluetooth registers in devices. And what did they discover? They discovered that by far the most widely used microprocessor that lives at the heart of by far most IoT devices contains an array of undocumented HCI register commands that they implied could be received over the chip's Bluetooth radio, but it can't.

I deliberately chose to use the word "undocumented" because it's less freighted with intent than the word "secret." I have a picture in the show notes of these commands, from their slide which they presented in Spain. It was conducted in Spanish, and the slide set is all Spanish; except, as we often see in code, English appears, you know, in code snippets. But staring at the portions of their 46-slide deck, you know, those portions that were understandable to me in English, and also chunks of reverse-engineered and disassembled code, I began to get the sneaking suspicion that, while these commands might indeed be undocumented HCI commands, which would be executed by Bluetooth hardware, it wasn't clear to me that they were remotely accessible. They appeared to be running their own - the researchers appeared to be running their own code on the ESP32 hardware and also reverse-engineering pieces of its firmware. Nowhere did they ever talk about remotely connecting to a generic ESP32 and executing an attack.

Since in this era of helpful AI you can do translation now, I uploaded the Spanish slide deck to ChatGPT's latest 4.5 model, which was overkill, and asked for a translation into English. It did a beautiful job for me, and my suspicions were confirmed. Now I could read the entire slide deck beautifully translated. The Tarlogic posting ended by writing: "Over the coming weeks, we will publish further technical details on this matter." It may be that they have more than they're saying, but I don't think so.

The only thing I believe they've discovered is that the ESP32's Bluetooth HCI controller, the Bluetooth hardware in this Espressif 32 chip, contains some commands that are undocumented because documenting them was not important. Discovering that an HCI controller contains a command which the host CPU issues to it that allows the controller to write to main memory could hardly be considered earth-shattering. The host which issues the command is just as able to write to main memory, if it wants to, so big deal.

If an unauthorized external Bluetooth radio were able to issue such a command remotely to an ESP32-based device, while presumably providing the data to be written into the system's main memory, and if this discovery existed in more than a billion of the devices we're all using, well, then, that would indeed be the end of the world as we know it. But the world is still here, and I haven't seen any evidence of that capability in their presentation. I just really think they have made a big mountain out of a little tiny molehill. And in fact it now seems clear that this amounts to a "host-side" access to an HCI controller, and that the threat that this poses is more like a mouse hole than a backdoor.

Leo: You have a convenient illustration of what that just might look at. Did you generate this with AI? I think you did.

Steve: Yes, I did.

Leo: I think you did.

Steve: I did indeed.

Leo: It's very good. It's cute.

Steve: BleepingComputer noted that Espressif, the creator of more than a billion of these amazing little chips, had not replied to their inquiry. That's likely because they also know that this is nothing. At one point in the presentation the security researcher mentioned cloning another device's MAC address. Whoop-de-do. Sure enough, one of the 29 undocumented commands was "Change MAC Address." Well, that's got to be there somewhere because you're obviously able to set the MAC address of the device when it comes out of the assembly line. You know, that's certainly neither a backdoor nor big news.

So anyway, I'm strongly inclined to come away from all this with the conclusion, exactly as you did, Leo, it's what you sniffed from the beginning, that there's really not much here. It made some attention-grabbing headline news, but nothing I have seen has suggested that the ESP chip is not still completely secure from external attack. You know, they talk about being able to establish persistence. But if you're running code in a flash-enabled chip, persistence is not difficult to obtain.

So, you know, among the undocumented commands is write to flash. But I'm sure you can write to flash from the native instruction set of the chip. So who cares if the hardware blue chip controller can also do it? It just seems crazy to me. Maybe something more will be revealed in the future. It seems unlikely because they would have gone for it in their main security presentation. I think they just found, oh my god, some undocumented commands in the hardware of the chip. Who cares? It doesn't look like...

Leo: And most importantly that you need hardware access to the chip to get to them.

Steve: Yeah, the registers, the registers on the blue chip controller, the Bluetooth controller, that's all they found, registers on the Bluetooth controller.

Leo: There's a whole category of hair-on-fire attacks that require somebody sitting down at the device. This one is even, you know, more ridiculous because you have to actually connect something to the device so that you can write to it and so forth. But, right, I even think that hardware attacks that require somebody on your machine really don't deserve the attention they often get. They should be fixed, of course, because if somebody's on your machine, all bets are off anyway. By that time, doesn't matter what; right? They're in.

Steve: Yeah.

Leo: Yeah. This is - yeah. So I kind of got that. It's just, I mean, they should have been documented, I guess. Right?

Steve: Yeah, and this is...

Leo: I mean, who cares? It is development tools.

Steve: Yeah. I don't even see any reason to document them. They are not necessary for programming Bluetooth. They're useful for managing the chip's deployment, like setting the MAC address.

Leo: Right, right.

Steve: So that they all have different MAC addresses.

Leo: Right.

Steve: And, you know, whoop-de-do. So we discovered how they did that.

Leo: Pretty much every chip in the world will do this; right?

Steve: You can change your MAC address. Yes, yes.

Leo: If it's got flash, you're going to be able to write to it. Yeah, I think...

Steve: And oh, gee, persistence. Well, that's what flash is for, persistence.

Leo: Right. Why would it have flash otherwise? Right?

Steve: Right.

Leo: Okay.

Steve: Yeah. So mostly this is a case of the press picking up a headline from a conference and saying, oh my god, you know, ESP32, more than a billion devices, and these guys say it has a backdoor. And it's in Spanish, so we're not really sure what they said.

Leo: Well, and it's a great thing to put in a headline, "Used by a billion devices." That's always a good bit of link bait. It did get a CVE number.

Steve: Yeah.

Leo: But that's not a big deal, to get a CVE number.

Steve: No. And you're able to just request one. And the CVE, when you look it up, and I did under NIST, it says "undocumented functions."

Leo: Right.

Steve: That's what the CVE is. It's like, oh, boy, we've got - scroll down, and you'll see that it shows undocumented functions is the...

Leo: It's maybe a little higher here somewhere.

Steve: I saw it somewhere.

Leo: Yeah, so that's important, too, that just because something's in the National Vulnerability Database doesn't by itself...

Steve: Oh, there it is, hidden functionality.

Leo: Hidden functionality, that's the CVE name, yeah.

Steve: Yeah, right.

Leo: All right. Good. Well, I'm reassured.

Steve: Yeah. Again, there's just - it can't be that you're able to remotely change the programming of an unsuspecting ESP32 chip, or it would be the end of civilization as we know it.

Leo: Right. That would be a bad thing.

Steve: And we're here, we're still here talking, Leo.

Leo: If you could do it via Bluetooth, that would be a bad thing.

Steve: That, well, yeah.

Leo: I mean, we've talked - this is one of the kind of regular topics on the show about Bluetooth vulnerabilities. There are plenty; right?

Steve: Yes. It is a very complex protocol for the first half of this podcast's nearly 20 years.

Leo: Yeah.

Steve: They were happening all the time.

Leo: I remember Bluetooth snarfing.

Steve: Oh, yeah. And you notice it sort of - it's gone now.

Leo: We don't hear about it much anymore.

Steve: We got that stuff settled down; you know?

Leo: We've locked it down, yeah.

Steve: Yeah.