SECURITY NOW!

Transcript of Episode #879

The Rolling Pwn

Description: This week we look at a recently made and corrected mistake in the super-important OpenSSL crypto library. The NIST has settled upon the first four of eight post-quantum crypto algorithms. Yubico stepped-up to help Ukraine. Apple has added an extreme "Lockdown Mode" to their devices. Microsoft unbelievably re-enables Office VBA macros received from the Internet. The FBI creates a successful encrypted message app for a major sting operation. We close the loop with some of our listeners. Then we examine an even more egregious case of remote automotive wireless unlocking and engine starting.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-879.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-879-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We're going to talk about NIST. They have finally settled on some algorithms, the first four of their post-quantum crypto algorithms. Steve will talk about them, including one called CRYSTALS-Dilithium. And Apple's extreme lockdown mode. Will Steve start using it? Plus, why you may not want to own a Honda automobile. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 879, recorded Tuesday, July 12th, 2022, "The Rolling Pwn."

It's time for Security Now!. Here he is, ladies and gentlemen, the star of our show, Bob Barker. No, no, that's the wrong show. Steve Gibson. Hello, Steve.

Steve Gibson: And I've got my lighting proper this week.

Leo: Oh, yeah. Yesterday. Because you had the skylight open.

Steve: I have, like, one, two, three, four, five, six, seven, eight, nine. I have many high-wattage LEDs, and they point at the ceiling, which is white.

Leo: Oh, nice, soft, diffuse.

Steve: Which is why I get this really nice diffused - but it lit the room so much that my face was dark in the camera.

Leo: Right.

Steve: So I thought - man, look at this. It's just amazing. It looks perfect now.

Leo: Yeah. Good lighting makes a big difference. Even with a bad camera it makes a difference; right?

Steve: That's right. Well, and of course we all have that Logitech HD 720 or 722 or whatever it is.

Leo: Yup, there you go, yup.

Steve: So we're at Security Now! Episode 879 for the 12th of July. And I was a little ambivalent about the title "The Rolling Pwn," P-W-N.

Leo: It's funny. It's funny.

Steve: Yeah. That's the official name of the hack/attack. But I wanted to do - I was toying with Rolling Your Pwn. And I thought, well, no, okay. Or The Rolling PIN. But Pwn is what it's supposed to be. So anyway, I just left it alone. But first we're going to look at a recently made and corrected mistake in the super-important OpenSSL crypto library. What you missed last week, oh, no, week before last, I forgot to mention when I was on with Jason, somebody - it was really pretty funny - somebody took the OpenSSL command line - and remember, OpenSSL is like the Swiss army knife times a thousand - took the OpenSSL command line and said, what if it was a GUI? And so our Picture of the Week was that, and it took up four pages of our show notes.

Leo: Wow.

Steve: And that was for one tab of one subset of the OpenSSL commands. Those were the options for I think it was like making a cert. Anyway, so it was just a...

Leo: That's a GUI for you, yeah.

Steve: It was a fun Picture of the Week. But anyway, we're going to talk about a problem that was found and corrected in OpenSSL. The NIST, and many of our listeners tweeted me to make sure I knew about this and wanted to hear about it, has settled upon the first four of the total of eight post-quantum crypto algorithms which will become the next standard, much as Rijndael was decided as the AES standard, and like the SHA-256 hashes. I mean, we need standards. Otherwise it's bad enough that our USB plugs won't go in the right way. Fortunately, they're all the proper shape. And at least we have that, you know. Some are not triangular, and some are not hexagonal. So anyway. Also Yubico stepped up to help Ukraine in a little blurb that passed by. And I thought, oh, I've just got to give them a shout-out and a thank you.

Leo: Good.

Steve: Apple of course has added the extreme lockdown mode that we'll talk about, or it's forthcoming. Microsoft unbelievably, and the whole security industry has just gone in meltdown over their announcement that they are re-enabling Office's VBA macros, which are received over the Internet, after telling us in February, to everyone's great relief and many sighs, that they were going to finally disable them by default. Now they're saying, oh.

Leo: You're out of date. They're decided not to enable them.

Steve: Oh, you're kidding.

Leo: No. This just in.

Steve: Wow.

Leo: Yeah. So they put out a press release yesterday because there was a lot of upset over this.

Steve: Cool.

Leo: And macros from the Internet will be blocked by default in Office, according to Microsoft, as of yesterday.

Steve: Still be blocked by default.

Leo: Still, yes.

Steve: Okay.

Leo: And I think, yeah, there's some nuance into the whole thing, but...

Steve: Well, we'll have some fun at their expense anyway because, you know, they're Microsoft.

Leo: I will show the flowchart that decides whether macros are to run. You've showed it before.

Steve: Oh, you mean there's still a way for it to happen.

Leo: Oh, yeah. Oh, yeah.

Steve: Oh, boy. Wonderful. The FBI has created a successful encrypted messaging app which participated in a major sting operation that we're going to talk about. We're also going to close the loop with some of our listeners. Then we're going to examine an even more egregious case of remote automotive wireless unlocking and engine starting. Thus the Rolling Pwn.

Leo: Ah.

Steve: Cars roll.

Leo: I get it.

Steve: After they've been pwned. And we have a really clever wonderful Picture of the Week. So I think another good podcast for our listeners.

Leo: An OR gate. Yes, a little bit of a logic lesson. It's exciting, yes. Good. Good show, as always, coming up. I look forward to Tuesday all week long to hear the latest. In fact, and I'm sure I'm not alone in this, I see articles, and I go, I can't wait to hear what Steve has to say about that. Can't wait to hear what Steve says. Without you, I wouldn't know what to think about any of this stuff. Picture of the Week time, Steverino.

Steve: So this is a visual feast. Unfortunately, it would be difficult for me to explain it in words.

Leo: Well, that's why there's a picture.

Steve: Yeah, it's a very good point. For those who do have the show notes in front of them, there's a, as I described at the beginning, and I gave it a title, a wonderful mechanical OR gate. If you - oh, Leo, it's just so good. If you - imagine that six different people had six different keys to six different padlocks. Now, if you wanted to allow any one of them to unlock, for example, a gate that was secured with a chain, well, you could interlace, you could interlock six padlocks, the hasps of six padlocks through each other, right, to sort of create one long padlock-y thing such that any one of the keys could unlock its one padlock. And because it was a chain of padlocks, that would unlock the whole chain. The problem we have here, though, is that this thing that's being secured, there's a big steel arm coming in from the left.

Leo: Looks like a gate or something, yeah.

Steve: Yeah. And it wants to come, it wants to slide outwards so there isn't really a way you could do that with interlinking hasps of padlocks. So some mechanical engineer, I mean, this is just so cool. The more I looked at it for a while, the better I liked it.

Leo: This is brilliant. And unlocking any one of these padlocks opens the gate.

Steve: Yes. Well, I guess it would be an AND, wouldn't it. Because they all have to be closed for it to work. Any one of them being opened...

Leo: It just depends whether the true value is unlocked or locked; right?

Steve: Yes, exactly. Exactly.

Leo: So it's OR if it's unlocked. It's an AND if it's locked.

Steve: Right. But what's so cool is that, again, there's no way to describe it except to say that it's this wonderful mechanism where any one of these locks, like one at the bottom, you would unlock one at the bottom.

Leo: 18 or 15.

Steve: Yes, exactly. That would allow the shim to be removed from a pin which then allows the pin to slide out, the little pin at the bottom, which then allows the big pin, the big vertical pin to be lifted out of the way of the lever that wants to come out. So it's this multistage lock trick.

Leo: It's very clever.

Steve: It's really good.

Leo: I'm not sure I understand how 15/18, so if I take 15 or 18 out, then that means, oh, I can slide this bigger bar out and then slide this little pin out.

Steve: Right.

Leo: Yeah, yeah, you just kind of have to be there. And you certainly can't describe it.

Steve: And then the big ones comes - as I said, there's really no way to describe it except to, you know, it's analogous to a chain of six locks that are interlinked, but in a way that works with this particular mechanism. Anyway, it's just wonderful, wonderful, wonderful. So whoever that was, a listener of ours who said, what do you think about this for a Picture of the Week? It's like, right on.

Leo: Wow, wow, wow.

Steve: That is dead on the money. That's our kind of picture. Okay. So OpenSSL v3.0.4 introduced a serious vulnerability which v3.0.5 just repaired. It is a potential remote code execution flaw which was recently discovered in an update, that is, this 3.0.4, to the v3 branch of OpenSSL. The issue was found, as I said, in 3.0.4. The good news is it was just released late last month, on June 21st. It impacts x64, so 64-bit Intel architecture systems having the AVX-512 instruction set extension. The good news is OpenSSL v1.1.1, as well as the two forks, BoringSSL and LibreSSL, they're not affected. So only the 3 branch. The trouble stems from a remote memory corruption vulnerability. This AVX are the Advanced Vector Extensions which add instructions to the x86 instruction set architecture from processors both from Intel and AMD.

There was some interesting back and forth about this in the GitHub issue thread where I think that the OpenSSL Foundation's guy Tomas Mraz said: "I do not think this is a security vulnerability." Of course he didn't want it to be. He said: "It is just a serious bug making the v3.0.4 release unusable on AVX-512 capable machines." Okay. So I guess he's saying that it will crash. So like the certificates, it involved RSA certificates. So they won't work. So that's a problem. A security researcher, Guido Vranken, he said it can be triggered trivially by an attacker.

Another person participating in the thread, Alex Gaynor, wrote: "I'm not sure I understand how it's not a security vulnerability. It's a heap buffer overflow that's triggerable by things like RSA signatures, which can easily happen in remote contexts like," he says, "a TLS handshake." And the post-grad Ph.D. student who originally discovered and reported the bug chimed in to the thread, stating, he said, "Although I think we shouldn't [as in should not] mark a bug as 'security vulnerability' unless we have some evidence showing it can, or at least may, be exploited," he says, "it's necessary to release version 3.0.5 as soon as possible given the severity of the issue."

Which is what did in fact soon happen. The issue has been assigned CVE-2022-2274, described in that CVE as a case of a heap memory corruption within RSA private key operations. The advisory notes that "SSL/TLS servers or other servers using 2048-bit RSA private keys running on machines supporting AVX-512-IFMA instructions of the x86_64 architecture are affected by the issue." And calling it a "serious bug in the RSA implementation," but still apparently not wishing to call it a vulnerability, the maintainers of OpenSSL said that the flaw could lead to memory corruption during computation that could be weaponized - sounds like a vulnerability to me - by an attacker to trigger remote code execution on the machine performing the computation. So anyway, as I said, smacks of a security vulnerability.

Well, anyway, the flaw has been patched, and all users of OpenSSL v3 should move to 3.0.5, especially if you had diligently moved to 3.0.4, which is the buggy release. On the other hand, the window was a couple weeks. So the chances are nobody even had a chance to get the buggy one before you got the good one. So anyway, just FYI.

Okay. Last Tuesday, the U.S. NIST, right, our National Institute of Standards and Technology, announced that the results of a six-year - these things do take a while. And I'm glad. This is not something you want to rush because we're going to be living with this for a long time. The six-year competition among a set of post-quantum algorithms had resulted in the selection of four initial algorithms. That is, because there are going to be a total of eight. So the first half have now been chosen. After editing out the various self-congratulatory statements from various bureaucrats who have no clue what this is all about and who certainly didn't even write what they are quoted as saying in this official announcement - I read through this. It's like, oh, come on. You have no idea what you're talking about. Anyway...

Leo: What are you talking about? I've always been a fan of elliptic curve cryptography.

Steve: I use it to clean my sheets every week. So here's what the people who actually wrote something about this and were involved in the choosing had to say. They said: "The U.S. Department of Commerce's National Institute of Standards and Technology (NIST) has chosen the first group of encryption tools that are designed to withstand the assault of a future quantum computer, which could potentially crack the security used to protect privacy in the digital systems we rely on every day" as in today - "such as online banking and email software. The four selected encryption algorithms will become part of NIST's post-quantum cryptographic standard, expected to be finalized in about two years.

"The announcement follows a six-year effort managed by NIST" - but not in any way poisoned by them, this has all been done open on GitHub in full public view - "managed by NIST which in 2016 called upon the world's cryptographers to devise and then vet encryption methods that could resist an attack from a future quantum computer" - that is, you know, one with more than four Qubits - "that is more powerful than the comparatively limited machines available today. Today's selection constitutes the beginning of the finale of the agency's post-quantum cryptography standardization project." And this is clearly a good thing. I don't remember, Leo, whether they were able to do the factorization of, what was it, 33 or something?

Leo: It was some ridiculous number, yeah.

Steve: Yeah, it's like, oh, okay.

Leo: How hard is that, huh?

Steve: We don't have to worry about it right now.

Leo: I'd say we are making progress, though. This was a story a couple of days ago that scientists in Germany have showed spooky action at a distance of 20 miles. Two atoms. So that's when the...

Steve: Quantum entanglement.

Leo: Yeah, quantum entanglement. And the two atoms somehow are communicating instantaneously across a distance of 20 miles.

Steve: And Leo, not at lightspeed. Not at lightspeed.

Leo: Much faster.

Steve: Instantaneously.

Leo: Yeah, yeah.

Steve: Which tells you, definitely a simulation. Okay.

Leo: Or there's some dimension we don't know about in which those two atoms are right next to each other.

Steve: Or are the same thing.

Leo: Or the same thing.

Steve: See, the whole idea of space could be just an illusion; right?

Leo: Right.

Steve: There isn't actually any. It's just...

Leo: It's mindboggling that they're doing it.

Steve: It is. And it does, it Hertz.

Leo: It Hertz, yes.

Steve: It Hertz, yes. Okay. Four additional algorithms are under consideration still for inclusion in the standard, and NIST plans to announce the finalists from that round at a future date. NIST is announcing its choices, they wrote, in two stages because of the need for a robust variety of defense tools. As cryptographers have recognized from the beginning of NIST's effort, there are different systems and tasks that use encryption, and a useful standard would offer solutions designed for different situations.

And that, yes, how many times have we - we talk about the toolbox that we have today and how cool it is that you can just put these things, these little components together in all different ways. So use varied approaches for encryption, and offer more than one algorithm for each use case in the event one proves vulnerable. And that's what they've done here. This is like we're - it feels like we're beginning to understand collectively as a planet how to do these sorts of things correctly.

So explaining this for the masses, NIST added: "Encryption uses math to protect sensitive electronic information, including the secure websites we surf and the emails we send. Widely used public-key encryption systems, which rely on math problems that even the fastest conventional computers find intractable, ensure these websites and messages are inaccessible to unwelcome third parties. However, a sufficiently capable quantum computer, which would be based on different technology than the conventional computers we have today, could solve these math problems quickly to defeat today's encryption systems. To counter this threat, the four quantum-resistant algorithms rely on math problems that both conventional and quantum computers should have difficulty solving, thereby defending privacy both now and down the road."

Leo: Okay. I've got a tough question for you. Do we now trust NIST? Because remember they intentionally recommended a weak algorithm at the behest of the National Security Agencies.

Steve: Yeah. And I would say those were days gone by.

Leo: Yeah.

Steve: There's no cryptographer who doesn't know that this random bit generator that RSA Corporation was sort of defaulting to had some sketchy background. There was no reason for the NSA not to describe where the magic numbers came from that that digital random bit generator used. And that would have made everyone feel good. If somebody had said this is how we chose these numbers, then everyone would have gone, okay, that makes sense. Instead, it was "Thou shalt use these numbers." And it was like, uh, that's not the way we do things here. And the point is it wasn't the way - it sort of was the way we did things then because nobody was that focused on those things. Now we really are.

So I don't think that could ever happen again. I mean, and so this is a - this was done very much like the way Rijndael was chosen, where a number of like really good candidates were examined, and sample implementation code was created. And things like how fast can we make this work on a 64-bit x64 architecture, and can we design algorithms which will not be subject to side channel attacks. I mean, just think about everything we've learned in the last 20 years. All of that is now rolled into this. And lots of debate. That's why it took six years, you know, to decide this.

So in this case these first four of the eight total algorithms, the first four are designed for two main tasks, for which as we know encryption, or crypto, is typically used: general encryption, which is used to protect information exchanged across a public network, and digital signatures used for identity authentication. All four of the algorithms were created by academic experts collaborating openly from multiple countries and institutions.

To provide for general encryption, NIST - and it's not NIST as much as it's the collective. And that's just it. NIST is just sort of saying, yeah, we're going to do the press release. But it was this - it wasn't NIST that chose it I guess is the point. It was everybody coming to a final agreement that, okay, to do encryption we're going to use the CRYSTALS-Kyber algorithm, which is what was chosen for encryption. And it was chosen because it uses comparatively small encryption keys which two parties will be able to exchange easily, as well as very good speed of operation. On the digital signatures side, NIST and the collective selected three algorithms.

Leo: I like his first one.

Steve: I do, too, Leo. It's the Dilithium CRYSTALS algorithm.

Leo: Yes.

Steve: And then we also have FALCON, and we have SPHINCS. It's actually SPHINCS+ because there was some tweaking that was done later, so it's S-P-H-I-N-C-S-+, which we're supposed to read as SPHINCS+. Reviewers noted the high efficiency of the first two, and NIST recommends CRYSTALS-Dilithium to be used as the primary algorithm, with FALCON for applications that need smaller signatures than Dilithium is able to provide. The third, and this is, again, why the thinking was so good on this, SPHINCS+ is somewhat larger and slower than the other two, but it's valuable to have as a backup for the reason that it is based on entirely different math than all of the other three NIST selections. The other ones are based on lattice math, and SPHINCS isn't. So again, we've learned that where crypto is concerned, there's nothing wrong with using a belt and some suspenders.

As I said, the first three of the four selected algorithms are based on a family of math problems known as "structured lattices," which is why the word "CRYSTAL" appears as part of the names of the first two; while SPHINCS+ uses hash functions. Now, the next four algorithms to be chosen, which are still under consideration, are designed for general encryption and do not use structured lattices or hash functions in their approaches. So again, we're going to do, like we're looking at a variety of different solutions like in advance. And once we deploy these, all of them will be selectable in the various algorithms so that, if something is found to be wrong, it'll be like, whoops. And it'll be easy to just switch over. Or remember how the early versions of TrueCrypt allowed you to use like multiple different algorithms like at once, under the theory that, well, if one of them was broken, then the other ones would still be good. So anyway, we sort of have a little bit of that, too.

So NIST wrote: "While the standard is in development, NIST encourages security experts to explore the new algorithms" - oh, all the source is public, by the way, and posted online.

Leo: And that's why we shouldn't worry about NIST being involved in this, obviously.

Steve: Yeah. I mean, they really weren't. They were just, again, based on, Leo, that text that I excluded from the announcement, you would know that whoever it was who wrote that nonsense...

Leo: Senator Foghorn Leghorn believes in CRYSTAL lattices as the finest way to protect yourself.

Steve: Yeah. That's exactly it, yeah. Boy. So they said: "While the standard is in development, NIST encourages security experts to explore the new algorithms to consider how their applications will use them, but not to bake them into their systems quite yet, as the algorithms could change slightly before the standard is finalized."

Leo: Can we use these if they're not baked? I mean, can we use them now?

Steve: Yeah. Yeah, yeah, yeah. Oh, I mean, they've been pounded, had the crap pounded out of them already.

Leo: They've pretty well baked, yeah.

Steve: I think it's likely that they're pretty much. But they're just, you know, again, they're hedging their bets. They don't want to be, like have fingers pointed at them saying, hey, you said these were final, and we burned it into our firmware and sent it up into outer space. So, like, no, okay, don't do that quite yet.

Leo: Don't do that. So but what tools...

Steve: Elon is welcome to use them right now.

Leo: Use them all you want, Elon. PGP or, I mean, what kind of tools - TLS, I guess, yes?

Steve: Yeah, yeah. Basically our crypto uses signatures all over the place and uses encryption all over the place.

Leo: Yeah. I would use PGP for that, which of course is an ancient and kind of crapulous bundle of algorithms, none of which are these.

Steve: Yeah. We could hope that PGP does not incorporate these so that once quantum computing comes along, sorry, PGP, your time has come.

Leo: SSH, though, would probably implement it, I would imagine.

Steve: Oh, oh, well, yeah, you mean, yeah, TLS definitely would. And it'll be used for hashing and, I mean, like a next-generation set of functions, the idea being that assuming that big quantum machines actually do happen, and again, it's like these are fast enough that there's no reason not to switch over to them. And that's the point; right? At some point because remember the other danger that we've talked about is that things that are encrypted today cannot be decrypted today, but they could be decrypted tomorrow. So this is why the NSA has that massive facility in Nevada which is why Vegas's lights are dimmer now than they used to be, is that the NSA's just storing everything. They're like, well, we can't decrypt it yet, but we think that once quantum comes along we'll be able to retroactively decrypt all this crap that we've been storing. So let's just keep it because storage doesn't cost anything.

So the point is we want to switch over to post-quantum crypto as soon as we know that we can trust it, assuming that it's not going to be a lot slower, and these algorithms are not slower than the ones we have, they're just bigger and different. We want to switch over so that we start giving the NSA stuff that, oh, sorry about that, but this is the Dilithium CRYSTAL quantum crypto.

Leo: I love it.

Steve: And you're SOL. So, yeah. So...

Leo: And by the way, you can't crack these Dilithium CRYSTALS.

Steve: So I obviously have no problem with the idea that adopting advanced post-quantum cryptography under the name Dilithium CRYSTALS, it's like what we ended up with. But I have to say, Leo, that in scanning through all of the candidate entries from which these four winners were selected, I did breathe a sigh of relief when I saw that the quantum algorithm named Frodo had not won.

Leo: Yeah.

Steve: I would have a hard time choosing Frodo as my post-quantum solution.

Leo: I refuse to use an algorithm with hairy toes, I'm sorry, it's just not going to happen.

Steve: Thank you. On that note, I'm going to take a sip of water. And then we're going to talk about Yubico.

Leo: Adrian in the chatroom says, "You mean Dilithium's a real thing? I thought it was just some fictional Star Trek thing."

Steve: It's real now, baby.

Leo: It is now.

Steve: It's real now.

Leo: Congratulations. You have entered the Dilithium Zone. Our show today brought to you by - let me find out who it's brought to you by. That's what I need to do. Oh. I know what I did wrong. I should never use Bing for a browser.

Steve: Oh. Lorrie keeps saying to me, how can I get rid of this thing? It keeps coming back. I hate it. I hate it. I hate it.

Leo: I hate Bing.

Steve: To help Ukraine hold off Russia's cyberattacks, Yubico donated 30,000 FIDO YubiKeys.

Leo: Bravo.

Steve: I know. I thought that was so neat. I just happened to see it in passing. So far, more than half of those 30,000, around 16,000 YubiKeys have been deployed to Ukrainian government executives, workers, and employees of private companies in

Ukraine's critical sectors. The initiative is being coordinated by a company named Hideez, H-I-D-E-E-Z, which I guess ID as in identity, Hideez, they're a Ukrainian security firm specializing in identity services and FIDO consultancy, so they know their way around FIDO. Earlier this spring Hideez secured a donation of 30,000 YubiKeys from Yubico, and way to go, Stina.

Since then Hideez's staff has been working with Ukrainian government agencies like the Ministry of Digital Transformation, the National Security and Defense Council, and the State Service of Special Communications and Information Protection, that's the SSSCIP, of Ukraine to ensure the devices can be - and it's one thing to have YubiKeys; right? But you've got to know what to do with them. So they're ensuring the devices can be imported into the country, that government infrastructure is prepared for the YubiKeys rollout, and that the recipients receive the necessary training to know how to use them.

So the idea is that once government and critical sector workers have a security key as an extra layer of protection, their accounts would finally be safe from what amounts to an onslaught of nonstop spear-phishing attacks which have been constantly hitting their inboxes every day.

Yuriy Ackermann, VP of War Efforts at Hideez, told the publication Risky Business, he said: "We got YubiKey-certified, so they are allowed to be deployed into Ukraine instances." He said: "We have quite a few ministries that have moved a lot of their stuff to G Suite and Azure. With them it's quite easy. We just get them a key. We made instructions in Ukrainian, video instructions and so on. So it's really fast. We had a department that pretty much moved to using FIDO, like 500 people in less than a week because they just needed to understand their policies, read our documentation, and that's it. They just give the keys and roll them, and voila."

So meanwhile, efforts are underway to roll out the keys to individuals in other departments - you know, they still have, what, 17,000 of them available, or 14,000 rather - including those without the proper server-side infrastructure. In these cases, Ackermann says Hideez has been providing the government with the company's solutions at minimal costs.

Anyway, as I said, I just happened upon this nice bit of news and wanted to acknowledge what Yubico had done to help Ukraine in their war effort.

Leo: Is FIDO related to Passkeys? FIDO2?

Steve: Yes. FIDO was the original, and that was the one which didn't get off the ground because it absolutely is tied to a hardware token. Whereas FIDO2 you're allowed to use devices that have some sort of biometrics in order to do the unlocking.

Leo: Same concept, really.

Steve: Yes, yes. Same concept. So Apple's new - oh, I should mention though also that FIDO2 uses WebAuthn also as its protocol to the web server, whereas FIDO is not a WebAuthn user.

Leo: Oh, okay.

Steve: So you have to have specific support for FIDO on the server side. Which is why Hideez is having to bring in some of its own technology where that's not available.

Leo: Got it, yeah.

Steve: So Apple's new extreme "Lockdown Mode." And "extreme" is their word, which I thought was kind of fun. In a blog post last Wednesday, Apple took the wraps off of Lockdown Mode, which will be rolled out later this year, first seen in macOS Ventura, iOS 16, and iPadOS 16. This is an optional mode which will, again, in their words, severely restrict some features. I mean, they've gone to a great degree here. I guess on one hand it's selling the idea that it is so restrictive. But they're also, like, making it clear that, yeah, we're not sure that this is for everybody. So if you were to turn this on, be prepared for a bunch of stuff not to work.

The aim is to protect specifically highly targeted individuals such as human rights workers and researchers, by reducing their devices' available attack surface. And they provided in their announcement a screenshot where this is the screen where you would go to turn this on. And it says: "Lockdown Mode is an extreme, optional protection that should only be used if you believe you may be personally targeted by a highly sophisticated cyberattack. Most people are never targeted by attacks of this nature."

Then they said, second paragraph: "When iPhone is in Lockdown Mode, it will not function as it typically does. Apps, websites, and features will be strictly limited for security, and some experiences will be completely unavailable." And then they've got a button to learn more, or this big scary one at the bottom, "Turn On Lockdown Mode."

So the way Apple put this in their announcement, and they used a term I hadn't seen before I thought was interesting. They said: "Apple expands industry-leading commitment to protect users from highly targeted mercenary spyware."

Leo: Mmm.

Steve: And they said: "Apple is previewing a groundbreaking security capability that offers specialized additional protection to users who may be at risk of highly targeted cyberattacks from private companies developing state-sponsored mercenary spyware."

Leo: Wow.

Steve: Yeah. They said: "Apple today detailed two initiatives to help protect users who may be personally targeted by some of the most sophisticated digital threats, such as those from private companies developing state-sponsored mercenary spyware." Okay, we get the message, Apple. "Lockdown Mode the first major capability of its kind, coming this fall with iOS 16, iPadOS 16, and macOS Ventura is an extreme, optional protection for the very small number of users who face grave, targeted threats to their digital security. Apple also shared details about the $10 million cybersecurity grant it announced last November to support civil society organizations that conduct mercenary spyware threat research and advocacy." In other words, researchers who were like going to dig into what this is all about.

Apple's head of Security Engineering and Architecture was quoted: "Apple makes the most secure" - yeah, blah blah blah - "mobile devices on the market."

Leo: You can always tell when you're reading from an Apple press release. Yeah, blah blah blah.

Steve: Okay. "Lockdown Mode," he said, "is a groundbreaking capability that reflects our unwavering commitment to protecting users from even the rarest, most sophisticated attacks." (Which we're unable to block.) Okay, he didn't really say that. "While the vast majority of users will never be the victims of such highly targeted cyberattacks, we will work tirelessly to protect the small number of users who are. That includes continuing to design defenses specifically for these users, as well as supporting researchers and organizations around the world doing critically important work in exposing mercenary companies that create these digital attacks.

"Lockdown Mode," he said, "offers an extreme, optional level of security for the very few users" - and they really don't want you to turn this on, but they want you to feel very special if you do - "who because of who they are or what they do may be personally targeted by some of the most sophisticated digital threats" - because after all, otherwise it wouldn't get through the regular iOS security - "such as those from" - oh, and we're naming names - "the NSO Group and other private companies developing state-sponsored mercenary spyware. Turning on Lockdown Mode in iOS 16, iPadOS 16, and macOS Ventura," you might as well just turn off your device. No, it doesn't say that - "further hardens device defenses and strictly limits certain functionalities, sharply reducing" - and I actually believe in this a lot - "the attack surface that potentially could be exploited by highly targeted mercenary spyware."

Okay. So at launch, Lockdown Mode includes the following protections. We have five. Messages: Most message attachment types other than images are blocked. Some features, like link previews, are disabled. Because yes, those could be abused.

Web browsing: Certain complex web technologies - bravo, Apple - like Just-in-Time JavaScript compilation. Remember, we saw Microsoft experimenting with disabling that in Edge because it just seems to be where all the problems are. Just-in-Time JavaScript compilations are disabled unless the user excludes a trusted site from Lockdown Mode.

Third, Apple services: Incoming invitations and service requests, including FaceTime calls, are blocked if the user has not previously sent the initiator a call or a request. Wired connections with a computer or accessory are blocked when iPhone is locked. And maybe you guys talked about this over on MacBreak. I'll ask you about that in a second, Leo.

Leo: Sure.

Steve: And finally, configuration profiles cannot be installed, and the device cannot enroll into mobile device management while Lockdown Mode is turned on. So to my eye, those all sound like very useful and sane restrictions.

Leo: You bet.

Steve: They would not hugely impact even most users, I think, while they would very clearly and significantly restrict the device's attack surface. So I say "Bravo, Apple, nice going." I'm sure that they've closely looked at the history of the way their devices have been compromised and then took steps to address future threats in a way that will keep their devices useful and useable, while being far less easily compromised. So again, bravo.

So Leo, that fourth thing, wired connections with a computer or accessory are blocked when iPhone is locked.

Leo: This is how I interpreted it, and the panel seemed to agree. That's so that you can charge but not have a data connection with a USB port. Right? So it's to prevent you from plugging your iPhone into some strange port. It actually allows you to do so. It's like your USB condom, I think. So what happens normally with an iPhone when you plug in a USB cable to a device, it says do you trust this device, you say yes, and now you can exchange data.

Steve: Right.

Leo: Which is obviously a bit risky.

Steve: That's very, very possible.

Leo: I think it's good.

Steve: A built-in condom for your...

Leo: Yeah, a built-in condom. That's how Apple should sell it, I think, yeah? A rubber for your phone.

Steve: Most of you will not need a built-in condom. But...

Leo: I had a question about the Just-in-Time stuff. Didn't we talk about that at one point, that Google's research showed that Just-in-Time JavaScript was problematic?

Steve: Yeah, it was Microsoft that were doing this because now they have Bing. And as we know, Bing is now based on the Chromium engine. And so it was their analysis that showed I think it was 80%, eight zero percent, of the problems that they were seeing in JavaScript resulted from tiny flaws in this squeezing the last, every last bit of performance out of JavaScript. And what they were saying was, you know, maybe five years ago, 10 years ago, computers were still slow. I mean, remember back then we didn't want to use encryption because it slowed down things. It was too slow. Now it's like, bring it on. We're going to go quantum, baby. We're going to do Dilithium encryption. So anyway, Microsoft said, hey, just turn this off, and you're going to be automatically protected from 80% of the problems that we're having. So Apple is saying the same thing.

Leo: And it's not like turning off JavaScript. It's just turning off the JIT compiler or JIT...

Steve: Right.

Leo: So you frankly maybe would run a little slower, probably not on a modern machine, and it eliminates a lot of those security flaws. So I think that's - I agree with you, this didn't seem too onerous. Google has their superior security where you have to have two Titan keys and all of that. I tried that for a while, and you lose so much functionality from Google that it wasn't worth doing it. But I could see turning this on for a normal person.

Steve: I'm going to turn it on and see how it feels.

Leo: Yeah, I will, too.

Steve: Because, like, why not? And, you know, I'll bet that would be - you know they're going to have some telemetry. I'll be it'll be some interesting metrics that they're going to get back about how many people go, yeah, you know, I don't need all that crap that's...

Leo: I forgot I turned it on, yeah.

Steve: Yeah, exactly.

Leo: It's all the things that people do that are dangerous, like links in messages. And the message rendering engines, I also use my Security Now! knowledge on MacBreak Weekly about that because we talked about this is where you see a lot of flaws, on Windows as well as on Apple products with this interpreter that has to somehow render this content in messages. And it's so often a security issue.

Steve: Yup. Okay. So as you said, it's already the case that Microsoft is not going to do what they said they were going to do last week, which...

Leo: It really confused the hell out of people.

Steve: Caused such a hubbub. Although Paul Ducklin, who writes for Sophos, I loved what he said about this. And I'll share it just because it's a great blast from the past. So that everyone understands, this macro abuse that we've suffered for so long, I remember when they announced it at the beginning of the year, Leo. You and I, back in February, were looking at the notice bars that used to be, and that would be, where it was so easy for you to click on "Allow Macros." And so, like, you know, you'd get some piece of email, and it would say, oh, this is not going to display properly unless you allow macros. Click here. It's like, who would not click that? Of course you would because it's like telling you to click it.

And so Microsoft said, oh, no, okay, we realize this has been causing lots of problems. We're going to turn this off. So what shocked everybody was when they said, uh, we're not going to tell you why, and they actually refused to tell people why exactly they changed their mind, but they announced last week they were going to do that.

Anyway, so in Sophos Paul said: "Remember 1999? Well," he said, "the Melissa virus called, and it's finding life tough in 2022." He said: "It's demanding a return to the freewheeling days of the last millennium, when Office macro viruses didn't face the trials and tribulations they do today." He said: "In the 1990s you could insert VBA (Visual Basic for Applications) macro code into documents at will, email them to people, or ask them to download them from a website somewhere. And then you could just totally take over their computer.

"In fact, it was even better or worse than that. If you created a macro subroutine with a name that mirrored one of the common menu items, such as FileSave or FilePrint, then your code would magically and invisibly be invoked whenever the user activated that option. Worse still, if you gave your macro a name like AutoOpen, then it would run every time the document was opened."

Leo: Yuck.

Steve: I know. How did we survive, Leo? And he says: "Even if the user only wanted to look at it. And if you installed your macros into a central repository known as the global template, your macros would automatically apply all the time. Worst of all, perhaps, an infected document would implant macros into the global template, thus infecting the computer; and the same macros, when they detected they were running from the global template but the document you just opened was uninfected, could copy themselves back out again to that document." He said: "That led to regular 'perfect storms' of fast-spreading and long-running macro virus outbreaks. Simply put, once you'd opened one infected document on your computer, every document you opened or created thereafter would, or at least could, get infected as well, until you had nothing but infected Office files everywhere."

Leo: Everywhere. Nice.

Steve: "As you can imagine," he said, "at that point in the game, any file you sent or shared with a colleague, customer, prospector, investor, supplier, friend, enemy, journalist, random member of the public, would contain a fully functional copy of the virus, ready to do its best to infect them when they opened it, assuming they weren't infected already. And if that wasn't enough on its own, Office macro malware could deliberately distribute itself, instead of waiting for you to send a copy to someone, by reading your email address book and sending itself to some, many, or all of the names it found there.

"The first macro malware, which spread by means of infected Word files, appeared in late 1995 and was dubbed Concept" - remember that? - "because at that time it was little more than a proof-of-concept." And, you know, the Concept virus was a thing. That's what it was. "But it quickly became obvious that malicious macros were going to be more than just a passing headache. Microsoft was slow to come to the cybersecurity party, carefully avoiding terms such as virus, worm, Trojan Horse, and malware, resolutely referring to the Concept virus as nothing more than a 'prank macro.'

"Over the years, however, Microsoft gradually implemented a series of functional changes in Office by incrementally, for example, variously, first, making it easier and quicker to detect whether a file was a pure document, thus swiftly differentiating pure document files and template files with macro code inside. In the early days of macro viruses, back when computers were much slower than today, significant and time-consuming malware-like scanning was needed on every document file just to figure out if it needed scanning for malware."

He says: "Microsoft also made it harder for template macros to copy themselves out into uninfected files. Unfortunately, although this helped to kill off self-spreading macro viruses, it didn't prevent macro malware in general. Criminals could still create their own booby-trapped files upfront and send them individually to each potential victim, just as they do today, without relying on self-replication to spread further."

He also noted that Microsoft "popped up a 'dangerous content' warning so that macros couldn't easily run by mistake. As useful as this feature is, he wrote, because macros don't run until you choose to allow them, crooks have learned how to defeat it. They typically add content to the document that helpfully explains which button to press, often providing a handy graphical arrow pointing at it."

Leo: Click Allow Here.

Steve: Click here, yes, with a little eh eh eh eh, and giving a believable reason that disguises the security risk involved. And finally, he said: "Adding Group Policy settings for stricter macro controls on company networks. For example, administrators can block macros altogether in Office files that came from outside the network, so that users cannot click to allow macros to run in files received via email or downloaded from the web, even if they want to."

So he says: "At last, in February 2022, Microsoft announced to sighs of collective relief from the cybersecurity community that it was planning to turn on the 'inhibit macros in documents that arrived from the Internet' by default, for everyone, all the time. The security option that once required Group Policy intervention was finally being adopted as a default setting. In other words, as a business you were still free to use the power of VBA to automate your internal handling of Office documents; but you would not, unless you went out of your way to permit it, be exposed to potentially unknown, untrusted and unwanted macros that weren't from an approved, internal source."

And of course yay. As this podcast celebrated at the time, Microsoft described that change then by saying: "VBA macros obtained from the Internet will now be blocked by default. For macros in files obtained from the Internet, users will no longer be able to enable content with a click of a button. A message bar will appear for users notifying them with a button to learn more. The default is more secure and is expected to keep more users safe including home users and information workers in managed organizations."

So everybody was excited about that. Sophos was enthusiastic, too, although a little bit less so than I was at the time. Back then they said: "We're delighted to see this change coming, but it's nevertheless only a small security step for Office users because VBA will still be fully supported, and you will still be able to save documents from email or your browser and then open them locally. The changes won't reach older versions of Office for months, or perhaps years, given that change dates for Office 2021 and earlier haven't even been announced yet," they wrote. "Mobile and Mac users won't be getting this change, and not all Office components are included. Apparently only Access, Excel, PowerPoint, Visio, and Word will be getting this new setting." On the other hand, that's by far the majority of Office things.

So anyway, Leo, you have the - so the news I was reporting yesterday was that they decided, well, actually what Microsoft said last week was: "Following user feedback, we have rolled back this change temporarily" - more temporarily than I thought - "while we make some additional changes to enhance usability." They said: "This is a temporary change, and we are fully committed to making the default change for all users. Regardless of the default setting, customers can block Internet macros through the Group Policy settings described in the article 'Block macros from running in Office files from the Internet.' We will provide additional details on timeline in upcoming weeks." And oh, boy, look at that...

Leo: So you know what happened, which was - so until recently it would be a pop-up, it'd say there's a macro in here, and then a button that said, yeah, go ahead, run it. And that was just too easy. So what they were going to do was take that button, move it into the properties of the document, so you have to know to get the info on the document, go into the properties, check a box, run the macro. And I think pretty clearly what happened is a lot of businesses said, but no, but it's too hard. And we have to train people how to do that. So initially Microsoft said, okay, we're not going to do that. Now, of course, everybody else has said, no, no, it's too easy. And so they just kind of backed off on that. This is, according to the Microsoft blog post, this is the new way to do it, and you can see that new.

Steve: Do you have to click your heels three times?

Leo: Well, so basically, if there's a macro, a VBA macro in there, this is the decision tree. And if it's from a trusted location, if it's digitally signed and trusted publisher blah blah blah, it used to be that you could use Group Policy or Cloud Policy to block or unblock. But now, if none of that's true, you get this final flow-through where in fact Office default macros blocked so show trust bar security risk with learn more. This is what we were talking about.

Steve: Okay.

Leo: And then there will be a process. So it looks like they're going to kind of bring that back. But to make businesses happy there are a lot of situations...

Steve: Yeah, allow them to be signed. I mean, that's going to...

Leo: Yeah, signed, because if you open - it was a previously trusted document. And you'll still have Group Policy that can default to block or unblock. So I think this actually is Microsoft, they have to compromise all the time because business users, right?

Steve: Yeah. It is so sad, though, that it is so difficult to turn up the security.

Leo: Well, and now you know why. Now you know why. Every business says, well, yeah, but I don't want to retrain employees. We need those macros. We use them in our weekly spreadsheet flow. And we don't want to have to have to do all that.

Steve: Wow.

Leo: So I think that this is the process, and it's a good process ultimately, where all the stakeholders get to weigh in.

Steve: And then you just push it in the direction you want to incrementally.

Leo: Yes. Just like Google does; right? And so you can see Microsoft's heart is in the right place. They want to do this. And they just kind of do it in a way that it doesn't upset people as much as it did, I guess.

Steve: Yeah.

Leo: It should be, you know, for all Microsoft Home users it should be off. But they're not going to do that, either.

Steve: Yeah, you're right. That would absolutely.

Leo: Yeah, yeah. Anyway, so yeah, they rolled back the rollback.

Steve: Wow.

Leo: That's what TechCrunch's headline is, is "Microsoft Reverses Its Reversal."

Steve: Wow. Okay. So Motherboard published an interesting story under the headline "This Is the Code the FBI Used to Wiretap the World."

Leo: Oh, yeah, wasn't that interesting, yes.

Steve: Yeah. And they followed that opening up with the subheading: "Motherboard is publishing parts of the code for the ANOM encrypted messaging app, which was secretly managed by the FBI in order to monitor organized crime at global scale."

Leo: And you nailed it, by the way.

Steve: I know.

Leo: You figured it out. You figured out how they did it. That's pretty smart.

Steve: Yeah, well, and I actually talk about that here in a second. But you're right, they did it the way I keep saying this is the way you would do it. So what I thought was interesting is that the approach that Motherboard says the FBI took to pull this off was precisely the solution I have often hypothesized as being the obvious way in which an end-to-end multi-party messaging system would be compromised. So here's how Motherboard's story begins.

They said: "The FBI operation in which the agency intercepted messages from thousands of encrypted phones around the world was powered by" - what Motherboard described as, actually they had people describing as cobbled-together code. Okay. I disagree with the characterization. They used open source code for an XMPP encrypted messaging system. So it's like okay. I guess you could describe open source as cobbled together. It all kind of is. But okay. Anyway, they said: "Motherboard has obtained that code and is now publishing sections of it that show how the FBI was able to create its honeypot. The code shows that the messages were secretly duplicated and sent to a 'ghost' contact that was hidden from the users' contact lists. This ghost user, in a way, was the FBI and its law enforcement partners, reading over the shoulder of organized criminals as they talked to each other."

Now, our listeners will recall that this has been my greatest criticism of any supposedly private and secure system where a user's keys are being in any way managed for them. The reason that Threema's approach has always appealed to me is that the user is 100% responsible for their own key management. And as we've often observed, if you're not managing your own keys, someone or something is managing them for you because the one thing any secure and private instant messaging system needs is keys. The point being, key management must be occurring somewhere. So if it's not something you're doing for yourself, then you don't have any direct control over what's going on.

Okay, now, that's not to say that I think that people should be doing their own key management. When today's podcast is finished, I'll shoot an iMessage to my wife, Lorrie, and let her know that I'm heading home. My point is, top-level state secrets are not being exchanged in my iMessages. The fact is, when you get right down to it, no consumer smartphone can really be trusted absolutely. But again, most people don't need that level of secrecy.

Anyway, Motherboard continues. They wrote: "Last year the FBI and its international partners announced Operation Trojan Shield" - I love the name - "in which the FBI secretly ran an encrypted phone company called ANOM for years and used it to hoover up tens of millions of messages from ANOM users. ANOM was marketed to criminals and ended up in the hands of over 300 criminal syndicates" - you've got to love this - "worldwide. The landmark operation has led to more than 1,000 arrests, including alleged top-tier drug traffickers and massive seizures of weapons, cash, narcotics, and luxury cars."

So, wow. The FBI mounted a good old-fashioned high-tech sting operation. Good going. But Motherboard doesn't sound very impressed with the FBI's coders. They wrote: "Motherboard has obtained the underlying code of the ANOM app and is now publishing sections of it due to the public interest in understanding how law enforcement agencies are tackling the so-called "Going Dark" problem, where criminals use encryption to keep their communications out of the hands of the authorities."

Now, okay. I'm unconvinced that there's any true public interest here, but okay. Mostly Motherboard seems to want to embarrass the FBI over what they think is the low quality of the code. They wrote: "The code provides greater insight into the hurried nature of its development, the freely available online tools that ANOM's developers copied for their own purposes" - also known as open source - "and how the relevant section of code copied the messages as part of one of the largest law enforcement operations ever."

They said: "The app uses XMPP to communicate, a long-established protocol for sending instant messages." You know, Jabber uses XMPP. "On top of that, ANOM wrapped messages in a layer of encryption. XMPP works by having each contact use a handle," they wrote, "that in some way looks like an email address. For ANOM, these included an XMPP account for the customer support channel that ANOM users could contact. Another of these was 'bot.'" Now, I do think it was a little inartful for them to name the secret account "bot," but okay.

Unlike the support channel, "bot" hid itself from ANOM users' contact lists and operated in the background, according to the code and to photos of active ANOM devices obtained by Motherboard. In practice the app scrolled through the user's list of contacts, and when it came across the bot account, the app filtered that out and removed it from view. So in that sense a little bit like having a rootkit. That finding is corroborated by law enforcement files Motherboard obtained which say that bot was a hidden or "ghost" contact that made copies of ANOM users' messages.

Authorities have previously floated the idea of using a ghost contact to penetrate encrypted communications. In a November 2018 piece published on Lawfare, Ian Levy and Crispin Robinson, two senior officials from UK intelligence agency GCHQ, wrote that "It's relatively easy for a service provider to silently add a law enforcement participant to a group..."

Leo: Mmm.

Steve: Uh-huh. I know, Leo, "...to a group chat or call." And "You end up with everything still being encrypted end-to-end, but there's an extra 'end' on this particular communication." Uh-huh.

Leo: Wow.

Steve: Yeah. The code also shows that in the section that handles sending messages, the app attached location information to any message that is sent to bot. On top of that, the AndroidManifest.xml file in the app, which shows which permissions an app accesses, includes the permission for "ACCESS_FINE_LOCATION," as in fine-grained location. This confirms what Motherboard previously reported after reviewing thousands of pages of police files in an ANOM-related investigation. Many of the intercepted ANOM messages in those documents included the precise GPS location of the devices at the time the message was sent. So, yeah, I mean, this is a golden honeypot operation.

Motherboard concluded their story by noting that Operation Trojan Shield had been widely successful, and that on top of the wave of thousands of arrests, authorities were also able to intervene using the intercepted and cloned messages to stop multiple planned murders. Using a well-established open protocol and open-source software allowed the application to be assembled without excessive cost, and it got the job done. And I just say, you know, well going. I thought that was a very nice piece of work. Wow.

Okay. We have some closing-the-loop bits and our final discussion. Let's take our last break, Leo.

Leo: Okay.

Steve: I'm a little parched. And then we'll do that.

Leo: Dealio. Yeah, I mean, I guess now that it's known how to do this, law enforcement can do it all sorts of places.

Steve: Yeah, we've got some great listeners who tweet. This is Mark Thoms. He said: "How should I vet client-side third-party EXEs I use as part of my web application? How could I verify it's not malicious?" He says: "Example: NeoDynamic.com JSPrintManager, a utility to allow silent printing from a website."

And, you know, my go-to is VirusTotal. And we've talked about it before. It is something like 70-plus different virus engines look at something. And I use it all the time. If I'm downloading something, like for example when I'm working on SpinRite, and I'm needing to do debugging on an ancient motherboard that has some strange LAN adapter, I need to get the motherboard on the 'Net. I've got to find a device driver for the LAN adapter. I go on the Internet. And I'll, like, find it. And, like, it looks like it's the right thing, but it's like, uh, kind of a sketchy site.

So I'll take it and drop the files on VirusTotal and have them give it a scan. Oftentimes VirusTotal has seen the exact file that I'm dropping already because what it does is first thing it does is it makes a hash of the file to create a signature, and then it just looks at the signature. And so it'll say, yup, already seen this one. But you can ask it to rescan it just for your own peace of mind, if you want to. Anyway, VirusTotal.com. And it's a free service. They're getting the benefit of lots of input from people who are wondering what this is.

And I know anybody who's listened to the podcast has heard me talk about it often because that's what security researchers use. And they'll, in fact, if they find something, they're able to go back and look at the first time that somebody else submitted it to VirusTotal in order to get some sense for how long a piece of malware has been roaming around the Internet. So anyway, a great solution for just checking out stuff that you're not sure about. And free. The price is right.

Isaiah tweeting from @boosted37, he said: "@SGgrc You often recommend a separate IoT network from the main one. However, devices like Chromecasts require your phone or tablet to be on the same network as the streaming stick to manage a show. How would you recommend separating those from your main network?"

That's a good question. I've had the problem myself. I do have stuff, as I said, on a separate private guest network. I'll just switch my WiFi from one to the other. I don't normally - now, the Chromecast would be more difficult because you're wanting to be using it probably all the time. And I guess I would say that it's a less sketchy device than some $5 plug that you got in a flea market or a garage sale somewhere, or freshly off of Amazon, for that matter.

So the solution I've used is that my various devices, my iPhone, my iPads, they've got - they know the passwords for both networks, and I'm able to switch back and forth between them with relative ease. So in the case of a Chromecast you may want that to - you may trust it enough for it to run on your main WiFi, but not your less trustworthy devices, which you probably also don't need to access on a daily basis. You know, you set the schedule for an outlet or a light switch, and then you let it go until daylight savings time changes.

Tom Terrific, he said: "You mentioned that you used an Asus router that you really like. Which model is it? It's time for me to buy a new router, and I always take your advice. Thanks. Tom."

Okay. So I haven't researched routers recently. It happens that the one I'm using I still like, and that's an Asus RT-AC68U, which is now at v3, so it looks like it wasn't something that lived for just a short time. I looked it up. Amazon tells me that I purchased it from them in October of 2017. So I know exactly when. That's when I was setting up my life with the woman who became my wife, back in October of 2017, almost five years ago. So it's an AC1900 WiFi gaming router, again, Model RT-AC68U. Looks like it's highly rated on Amazon. So anyway, that's what I'm using. I don't know that it's the best. There are cheaper routers. I think it's like $119 at the moment for a Prime subscriber. In fact, I think today is Prime Day. Maybe it's even cheaper if you're a Prime person. I don't know. Anyway, that's that one.

And Michael Horowitz, who knows his way around routers, he was for years the columnist, I don't know if he still is, at Computer World. He did his Defensive Computing column. He tweeted, he said: "About your Asus router." He said: "Are you sure the networks are isolated?" Now, remember I mentioned that this thing I discovered had either three or four, I don't remember which, three or four guest SSIDs. He said: "You might find all the guest SSIDs share the same subnet." He said: "I no longer have an Asus router so can't verify."

And actually if he's thinking that they might share the same subnet, then I guess I could test that. I was thinking that I'd have to actually try to send data across guest subnets to see if they're individually isolated from each other. But I should mention also when I went over to Michael's Defensive Computing site, that pointed me to his Defensive Computing Checklist. And Leo, this is the kind of thing I'll bet you would like. It's defensivecomputingchecklist.com. But I also made it this week's shortcut of the week so you can get to it more easily, or at least by typing fewer characters: grc.sc/879. Look how tiny the scroll thumb is.

Leo: It's like yours. I think I know why you like this. This is pure HTML, baby.

Steve: It is just the facts, baby. Just the facts.

Leo: Ain't no JavaScript here. No sir.

Steve: Just the facts. But if you scroll you'll notice that the thumb is not moving down on the right because this page is...

Leo: This long page.

Steve: ...so long.

Leo: Long page.

Steve: What it is, is a page of advice. And it is really good advice. It's like just all kinds of stuff. So I just think our listeners ought to know about it for their own purposes and also if they want to recommend it to people. There's just lots of information there on this massive page: grc.sc/879 or defensivecomputingchecklist.com.

Leo: It ain't a checklist. It's just a list.

Steve: It is, you're right.

Leo: A long list.

Steve: It is a long list. Just like Michael's collected ideas and thoughts about security over time.

Leo: It should be a book. This would actually - and I'll probably mention it on the radio show because there is so much good stuff in here.

Steve: It is really, really, really good stuff. Also, when I went there, at the top of that page, but also his main page, there's a note that he'll be giving a presentation on Defensive Computing at the HOPE conference in New York City about 11 days from today. It's actually called "A New HOPE," I'm sure probably...

Leo: HOPE is Hackers On Planet Earth, just in case you didn't know that, yeah.

Steve: Exactly, Hackers On Planet Earth. And wasn't "A New Hope"...

Leo: Star Wars.

Steve: Yeah, yeah.

Leo: The original Star Wars, yeah.

Steve: The original Star Wars movie.

Leo: Episode 3, "A New Hope."

Steve: "A New Hope." And I'm sure that's what they're...

Leo: Well, I think they had suspended for a while, so that's probably why. It's "The New Hope."

Steve: I think they did because they also talk about, I went there in order to see what was up. And they talk about this will be their first in-person gathering, no doubt since the COVID lockdown mess. So anyway, it's a three-day conference. It's not free, $200 for in-person all three, or streaming live for $99.

Leo: This used to be a great hacker conference.

Steve: Yeah, it is a...

Leo: Trying to remember if 2600 sponsored it or not. I feel like it was related to 2600 at the time.

Steve: And of course that's the famous frequency that was used in order to disconnect the normal communications during a long-distance call and drop you down to the billing layer.

Leo: Right.

Steve: Where you could do a so-called "phone phreaking" back in the day. Anyway, finally, Simon Dearn.

Leo: Episode 4, sorry, not 3, 4.

Steve: Was it 4?

Leo: Yeah, it's 4 because remember he started in the middle. He did 4, 5, and 6.

Steve: And that's just always bugged the crap out of me. It was like, what? What happened to the first three?

Leo: I think he was being clever because, I mean, you don't do that and say, oh, yeah, I'll be definitely making nine of these. I think he was just being clever. But then it turned out...

Steve: Oh, my god. And then we had to put up with the first three, which were...

Leo: Yes, the prequels were not good, yeah.

Steve: No. So anyway, Simon Dearn tweeted: "There are obviously good reasons for a company to use a VPN for allowing staff to connect remotely. But what are your thoughts on domestic use? Do you use a VPN yourself?"

And the answer, Simon, is I don't. But that's only because during my daily current life I'm never needing to get online using OPNs. "OPNs" is a handy abbreviation everyone should keep in mind. It stands for "Other People's Networks." And Leo, I grinned at the beginning of the show because you were talking about being on shipboard and how you would definitely be using a VPN. If my life involved travel, so that I was wanting to be online in airports, in coffee houses, and hotels, there's no question that a VPN would be the only way I would feel comfortable getting online when I was not at home.

This is much less of a problem today, where everything is encrypted over TLS with server-authenticating certificates. Back at the start of this podcast, everything was being done over HTTP with a brief switch to TLS only when logging in, for login forms and credentials were being exchanged. Remember when we used to explain that it was important to verify that the URL of a form's submit button was HTTPS and not just HTTP? Fortunately, we survived that.

Leo: And Firesheep and all sorts of other horrible things.

Steve: Oh, my god, yes. But it's still true that without universal certificate pinning, which I really don't see ever happening, or DNSSEC being used to publish certificates, which we're still a long way from, there's a vulnerability when a malicious man-in-the-middle could have control over our traffic. It's uncommon, I'm sure, but it's probably still possible for state-level actors to mint their own certificates that our browsers and operating systems will trust without question. It's true that the bad guys could be operating at the other end of a VPN endpoint. Although Leo, your comment about ExpressVPN changing IPs all the time, that's also going to tend to make the VPN endpoint more diffuse.

But something about hotel WiFi networking in particular gives me the creeps. For a long time it was unencrypted, and there's just no way to use an unencrypted hotel WiFi safely. And I suppose if I was downloading lots of torrent content I might want to hide that fact from my Internet provider. But I've found torrents to be more trouble than they're worth. So I don't care whether Cox knows what I do on the Internet. And if anyone was watching my Internet use mostly it would just put them to sleep. So not a problem for me. But again, I don't use a VPN during my normal daily life because I'm not traveling. If I were a traveler, then yeah. VPN, no question.

Leo: Also I'm thinking you're not watching a lot of Manga and anime stuff on Netflix Japan, that kind of thing.

Steve: No, I'm not. I'm not needing to appear to be other than in Southern California. Good point.

Okay. So the Rolling Pwn. And that begs the question, do you own a Honda? And are you sure it's still parked out in front? Enterprising security researchers at Star-V Lab - maybe it's Star 5 Lab - one using the handle Kevin2600, and of course we know where he got 2600 - and the other named Wesley Li, have revealed, as Larry David might say, "a pretty, pretty serious" vulnerability in Honda's automobile keyless entry systems. It rather trivially allows attackers to remotely unlock and start potentially all Honda vehicles currently existing on the market. It's really, like, sobering.

Okay. So it works in a way that's similar to, but even easier to pull off, than that recently discovered Bluetooth replay attack which we talked about not that long ago affecting some Tesla cars. Both use readily available off-the-shelf SDR (software defined radio) equipment, which allowed the researchers to eavesdrop and to capture codes, then broadcast them back to the car in order to gain access.

Recall that we talked about this related attack, the one I mentioned, that affected Teslas. And it was very clever. The key fob produced a high-quality cryptographically unpredictable pseudorandom sequence of codes which are derived from a monotonically increasing counter which is then encrypted by an unknown shared secret key. So the vehicle knows the key, so it's able to decrypt the received code to determine whether it's the next code in sequence or at least forward of the key fob counter's last known position. But the primary security feature is that the vehicle will NEVER accept any code that it has seen previously or that represents a count behind the most recently seen count that the key fob is known to have. In other words, it will only allow its knowledge of the key fob's counter to be moved forward, never backward. Okay. So this would seem to robustly prevent any possibility of a replay attack. But not so in the case we talked about before.

What the clever hackers did before was to receive the fob's code while simultaneously arranging to jam its receipt by the car. The user would think, after pressing the button and nothing happened because the car's reception was jammed, they would think, well, okay, whatever, I guess I was too far away, or who knows. So they would press the key fob a second time, causing it to generate the next successive code in sequence. Now the hacker would capture the second code while replaying the first code which the car had never received, and the door would unlock, or the car would start, or whatever. And that's the cool part that's so clever. The attacker would have obtained and retained that second code which had never yet been seen by the car, and would still be available as the next unlocking code in that otherwise unpredictable sequence.

Okay. So that was the clever, very active, got to get involved, very much man in the middle. You're an active man in the middle, able to jam, on the fly in real-time jam the receipt of the code by the car.

Today Honda is the target, and their problem is significantly worse since the attack on Hondas is almost completely passive and much easier to conduct. The Honda-attacking researchers have been able to remotely unlock and start the engines of Hondas dating from as far back as 2012, so a decade ago, up to and including this year. So nothing's been fixed in the last 10 years. The good news is, according to "The Drive," which independently tested and verified the vulnerability on a last year's 2021 Honda Accord, the key fob attack does work, allows the car door to be unlocked, and the engine to be started, but the attacker is unable to drive off with the vehicle, though it, as I said, can start the engine. Presumably the car contains some dynamic continuous key fob presence technology that's able to sense the presence of the key inside the vehicle, which at least prevents it from being drivable.

Okay. So the more active attack against Teslas worked the way I just described. What Kevin and Wesley discovered was, as I said, significantly worse because it is so much easier. They found that the counter in Honda's cars would be resynchronized when the car receives lock and unlock commands in consecutive sequence even when that sequence is from long ago. This means that Honda's counter can be reset to an earlier state. So by being induced into moving backwards, this will cause the car to accept codes from previous sessions that should have been forever invalidated after they were used.

In their write-up, Kevin and Wesley said: "By sending the commands in a consecutive sequence to the Honda vehicles, it will be resynchronizing the counter. Once the counter is resynced, commands from previous cycles of the counter will work again. Therefore, those commands can be used later to unlock the car at will." They wrote: "We've successfully tested the 10 most popular models of Honda vehicles from the year 2012 up to the year 2022 from the attacker's perspective. Therefore, we strongly believe the vulnerability affects all Honda vehicles currently existing on the market. Tested and known-vulnerable vehicles include the Honda Civic 2012, the Honda X-RV 2018, the Honda C-RV 2020, the Honda Accord 2020, the Odyssey 2020, the Inspire 2021, the Fit 2022, the Civic 2022, the VE-1 2022, and the Honda Breeze 2022." So, bad.

Then we have the always-entertaining FAQ, from which I will excerpt a couple questions. They asked: "Why it is called the Rolling-Pwn, not a Honda-Pwn?" And they answered: "Because this bug may exist in other brands of vehicles, too." "Am I affected by the bug?

As long as a vulnerable Honda vehicle is in use, it can be abused." "Is there an assigned CVE for Rolling-Pwn?" Yup. "2021-46145 is the official reference to this bug." "Can I detect if someone has exploited this against me?" And they wrote: "Probably not. The exploitation does not leave any traces in traditional log files. But considering the ease of exploitation and attacks leaving no trace, this threat should be taken seriously."

"Is this a Honda-vehicle-only bug? No, although the main targets for the research is Honda automobiles. We have leads to show the impact of this vulnerability also applies to other car manufacturers." They said: "We will release more details in the future." "Is the risk real? We've successfully tested the latest models of Honda vehicles, and we strongly believe the vulnerability affects all Honda vehicles currently existing on the market." "What makes this bug unique, or what's the difference between CVE-2022-27254 and CVE-2019-20626?" In other words, vehicle hacking has a rich history.

They wrote: "During the research, we noted the other researchers have found similar vulnerabilities in Honda vehicles, based on the description of 'The remote keyless system on Honda HR-V 2017 vehicles sends the same RF signal to each door-open request, which might allow a replay attack.'" Uh-huh. "What they found is a simpler fixed code vulnerability where an attacker can simply record the transmission in advance and replay it later to cause the door to lock or unlock. However, most modern vehicles including Honda automobiles implemented a proprietary rolling code mechanism to prevent fixed code replay attacks. The bug we discovered," they wrote, "is in regard to the design flaw in the rolling codes mechanism implemented by Honda Motors, which needs to be taken very seriously."

Question: "Is there more technical information about Rolling-Pwn?" They said: "You can follow the author on Twitter @kevin2600. However, we will not be releasing any tools required to go out and steal the affected vehicles. At a later stage we will release technical information in order to encourage more researchers to get involved in car security research." Which, boy, really does seem to be lacking.

"How to patch the modern automobile for Rolling-Pwn bug like this? The common solution requires us to bring the vehicle back to a local dealership as a recall. But the recommended mitigation strategy is to upgrade the vulnerable BCM firmware through over-the-air updates if possible. However, some older vehicles may not support over-the-air." And I should mention Honda offers no patching for this and indicates they have no plans to.

"What does Honda think about this Rolling-Pwn Bug?" They said: "We have searched through the Honda official website, but we can find no contact info to report the vulnerability. It appears that Honda Motors does not have a department to deal with security-related issues for their products. And a person who works at Honda has told us 'The best way to report the Honda vulnerability is to contact customer service.'"

Leo: Yeah, sure. That'll work.

Steve: That'll work. Yeah. Is your seatbelt tight? Are the brake lights working?

Leo: Unbelievable.

Steve: "Therefore," they said, "we filed a report to Honda customer service, and we have not had any reply." No kidding.

Leo: Oh, my god. That's horrible.

Steve: Earlier in March of this year, following similar remote keyless entry attacks on Honda Civics, BleepingComputer reached out to Honda and learned that Honda had no plans to update any of their older model cars. BleepingComputer wrote: "Honda told us multiple automakers" - in other words, it's not just us, it's not just us - "use legacy technology for implementing remote lock-unlock functionality, and as such may be vulnerable to 'determined and very technologically sophisticated thieves." And of course that's until Amazon starts selling the Honda Unlocker for 19.95 from China.

They said: "At this time, it appears that the devices only appear to work with close proximity or while physically attached to the target vehicle" - not so in this case - "requiring local reception of radio signals from the vehicle owner's key fob when the vehicle is opened and started nearby" - okay, that's true - "a Honda spokesperson told BleepingComputer. In their statement, Honda explicitly mentions it has not verified the information reported by the researchers" - although others have - "and cannot confirm if Honda vehicles are actually vulnerable to this type of attack." And one must imagine that Honda doesn't want to know, since knowing might make them culpable.

And Honda did add that, should the vehicles be vulnerable, "Honda has no plan to update older vehicles at this time. And it's important to note, while Honda regularly improves security features as new models are introduced, determined and technologically sophisticated thieves are also working to overcome those features." In other words, we give up.

So all this begs the question, why does this appear to be such a difficult problem to solve? Both the Tesla-style forward-only counter advance and Honda's bi-directional resettable counter solutions are transparent to their users. In other words, the keys just work. Right? But Tesla's forward-only system is clearly superior from a security perspective. On a Honda, the ability to passively record a series of unlocking codes, which can then be replayed at any time later, seems like a significant oversight that any engineer who was designing this system would have understood. Of course they would.

One thought is that there are likely some intellectual property issues here. There's no question that the first implementers of such rolling codes would have sought patents. That thought led to me ask the Google, where I immediately found U.S. patent 6225889 titled: "Method of producing rolling code and keyless entry apparatus using the same."

The abstract of the patent starts off reading: "A rolling code producing method is provided which may be employed in a keyless entry system for automotive vehicles designed to compare a rolling code derived in a receiver installed in the vehicle and a rolling code derived in a portable transmitter to allow authorized access to the vehicle if both rolling codes match. The rolling code producing method includes producing different rolling codes" - thus rolling - "in sequence, using an initial code variable according to a given algorithm and changing the initial code variable in response to insertion of an initial code variable memory card carried by a vehicle operator into the receiver."

Okay, now, the good news is this patent was issued in 1995, and it expired in 2016. Around the same time, also in 1995, garage door openers were suffering from the same lack of security. So Brad Farris and James Fitzgibbon "invented," and I have that in quotes, a similar system for their garage door opener employer The Chamberlain Group, and obtained U.S. patent US44688695.

There's been a lot of litigation over these patents through the years, and there's a long trail of bodies. But that's what the patent system does; right? It mostly amounts to a significant source of revenue for intellectual property attorneys. But all of these various patents appear to have finally expired back in 2016. So it's unclear why Honda would still be using their broken system today. They apparently were back in 2012, maybe to avoid any litigation or having to license somebody else's patent. It was Nippon that got the auto patent back in 1995. But it appears that for at least the past six years there's been no reason not to move to a much stronger forward-only counter scheme such as what Tesla and presumably others have implemented.

Overall, this truly is, if not a difficult, at least a little more expensive problem to solve. It is difficult to robustly solve it in a one-way-only system. The ultimate way to solve the problem is for there to be a full handshake. The user presses the button on the key fob, which sends a fixed-code ping out into the world, identifying itself to any car within range. The car that key is registered to receives the "Hello it's me" ping and replies with a nonce challenge. The key fob receives the nonce challenge and either reversibly encrypts it under its shared secret key, or hashes it with an HMAC keyed with a shared secret. Either way, it returns the result of that cryptographic operation to the car, which verifies that the key fob must know the secret they share, so the car performs the requested action.

That system, while ultimately secure and Internet-proven, is significantly more expensive, since now both the key fob and the car must support bi-directional communications. The key fob must also be able to receive, and the car must also be able to transmit. Given the cost and the complexity of this full solution, and the comparatively small additional security margin provided above the forward-only counter advancement used by Tesla and presumably other forward-thinking automakers, I would say that the small added security is probably not worth the cost.

But given that forward-only counter technology has been freely available in the public domain, unencumbered by any patent licensing requirements since at least 2016, Honda's continued use of resettable counter protection since then can only be ascribed as them just not caring. Given the popularity of Hondas, and who knows what other car makers may also have been similarly lazy, the relative ease of collecting key fob codes and the ability to later replay them in an entirely passive attack likely opens Honda to some future consumer litigation, I would think. So we'll see what more we hear of this in the future. Fun stuff.

Leo: What - so does that mean if you have a Honda you should keep your fob in a bag, one of the mesh bags?

Steve: No, no. That won't solve the problem.

Leo: Oy.

Steve: It really means that...

Leo: You're screwed. You need a padlock on your Honda's door.

Steve: It does mean you're screwed. It means, I mean, essentially it means that, I mean, this completely defeats the replay. The fact is this is inexpensive hardware to do this now. It's just, you know, a little SDR that you can buy on Amazon, a Software Defined Radio. Borrow a friend's Honda fob, see how it works, capture the codes, and then you can capture the codes from anybody else.

Leo: Wow.

Steve: And now we know that basically this renders the forward-only protection completely vulnerable.

Leo: For a long time Honda Accords were the most stolen cars in America. I think they're going to reclaim their crown.

Steve: Honda anythings, apparently.

Leo: Oh, wow.

Steve: So it may well be that you can't drive off with the car, but you certainly don't want to leave anything important. You don't to trust your car door locks, essentially, is what it means.

Leo: Right, right, right. The number one car in America stolen, Honda Civic. Number two, Honda Accord.

Steve: Wow.

Leo: And I don't know if that's related to this. Probably not. Just to their desirability in the stolen car market. But still.

Steve: And their, yeah, their generic nature.

Leo: Their generic-ness, yeah, because the Camry's third, the Nissan Altima's fourth. Man, if I'm going to steal a car, I'm going to steal a Hummer. I'm going to steal a Jaguar, a Porsche.

Steve: Going to have some fun.

Leo: Get something good, yeah.

Steve: And arrested and taken away.

Leo: Yeah. Actually, maybe that's old because now I'm looking - no, that says May 13th. Now I'm seeing number one is a Ford pickup. Also popular. You know, I have on my Ford, I hope it's a good technique, but I have on my Ford, and many Fords do, have a keypad on the thing. And unfortunately it's only four digits. No, I guess it's more than that.

Steve: The ones I've seen, they use all 10 digits, but they double them up on five keys.

Leo: That's the problem. There's only five keys.

Steve: So it's like, what the hell?

Leo: And I think this is related to what Honda is doing because it's basically cheapness; right? These are the cheapest possible components.

Steve: Yeah.

Leo: Five keys is half the price of 10.

Steve: It would be nice to know, I mean, because everybody now has key fob unlock; right? I mean, like, you know. I mean, even I...

Leo: And car start.

Steve: Leo, even I have key - I do have to still - I have to stick it in a slot and twist it, yeah. I've got to do that.

Leo: Oh, that's very old-fashioned, yeah.

Steve: But at least I've got the fob on that. And, wow, interesting.

Leo: What a world. All right, Steve. Great show, as always.

Steve: We're going to miss you next week.

Leo: I won't be here. Jason Howell will take over for the day. I will be back the following Tuesday. I'm going on that TWiT cruise with 120 devoted fans who I am sure to a person will say "Tell Steve hi." No matter where I go, that's what I get. "Tell Steve hi." So hi.

Steve Gibson is at GRC.com. You can tell him hi yourself. Just go to GRC.com/feedback. While you're there, pick up a copy of SpinRite, the world's best hard drive, I'm sorry, mass storage maintenance and recovery utility, soon to be 6.1, version 6 the current version. You'll get a free upgrade to 6.1 if you buy 6 today. You also get to participate in the development of 6.1, which is imminent.

While you're there, you can also get a copy of the show. Steve has two unique versions of Security Now!, a 16Kb version for the bandwidth-impaired, and really nicely done, human-written transcripts so you can read along as you listen, but you could also use it to search and find a part of the show that you're looking for.

GRC.com, the place to go. He's on the Twitter, @SGgrc. You can DM him there, as well, @SGgrc.

We've got copies of the show at our website, TWiT.tv/sn. We also have a YouTube channel devoted to Security Now!, makes it easy to share. You can send people clips. And of course you can subscribe on your favorite podcast client and get it the minute it's available of a Tuesday afternoon.

We do the show every Tuesday, right after MacBreak Weekly. That's 1:30 Pacific, 4:30 Eastern, 20:30 UTC. Live audio and video streams are at live.twit.tv. Chatroom is at irc.twit.tv That's open to all. And after the fact Steve's got forums at GRC.com. We've got forums at TWiT.community. Those are free and open to all, TWiT.community. There's also TWiT.social, which is a Mastodon instance.