Passkeys, Take 2

Description: This week we have a response from ServiceNSW to the news of their insecure Digital Driver's License. ExpressVPN is the first VPN to pull the plug on India. Turning off the Internet is becoming a common practice by repressive regimes. The Windows Follina exploit explodes in the wild. Another Windows/Word URL scheme can be exploited. A critical cellular modem chip defect has surfaced. Named ransomware is being impacted by U.S. sanctions and ransomware is taking aim at our system boot firmware. We have a bit of errata and closing-the-loop feedback. Then, in the wake of Apple's big WWDC 2022 keynote, which mentioned Apple's forthcoming adoption of the FIDO2 Passkeys, I want to highlight one glaring concern that everyone seems to have missed.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-874.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-874-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. As always, well, actually, you know what, Steve said it was kind of a slow week. There's still, it seems to me, plenty to talk about. We'll start with a look at ServiceNSW's new driver's license. Steve talked about how insecure it was last week. This week, their response. You won't believe it. ExpressVPN is the first VPN to pull out of India. We'll tell you why. There's a critical cellular modem chip defect. Does anybody care? And then Steve is going to look at Passkeys once again in light of Apple's announcement that they're going to support it. What is the key critical deficiency of Passkeys? Steve identifies it, coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 874, recorded Tuesday, June 7th, 2022: Passkeys, Take 2.

It's time for Security Now!, the show where we protect you, your loved ones, your privacy, your wallet online with this guy right here, Steve Gibson of GRC.com. Hi, Steve.

Steve Gibson: Hey, Leo, great to be with you again for, what is this, this is the first, no, yeah.

Leo: Yeah, June.

Steve: The first episode of June.

Leo: Yeah, yeah. Hard to believe. How did we end up here?

Steve: First episode of June. I know. I know. So I, along with you and Mikah, watched and enjoyed yesterday's Worldwide Developer Conference Keynote. And I saw in passing their mention of Passkeys. And something occurred to me that I hadn't thought about before. And I initially put it in "Miscellany" because it just - it didn't seem that big a deal. But the more I thought about it and wrote about it, the more I realized that there might be something significant that we're just all kind of glossing over while we're rejoicing about Apple, Google, and Microsoft adopting this technology. So, and it was kind of a slow week, which doesn't happen that often in security.

Leo: Woohoo.

Steve: Yeah.

Leo: This is the only show where we celebrate slow weeks.

Steve: And so I was sort of thinking, well, I don't want to talk about that. I don't want to talk about that. Don't want to talk about that. So we have things to talk about. We're going to talk about the response that Service New South Wales provided about - their response to all of the news which we covered last week of their very insecure, apparently hard-to-spoof, so they said, Digital Driver's License.

ExpressVPN is the first VPN to pull the plug on India. We're going to talk about that.

Leo: Our sponsor, yeah.

Steve: Yeah. Also turning off the Internet has turned out to become a common practice among some repressive regimes.

Leo: Oh, yeah.

Steve: For an interesting reason. The Windows Follina exploit - Follina, Italy, we talked about that. It has exploded into the wild, not surprisingly.

Leo: Oh.

Steve: Yeah. And there's another Windows/Word URL scheme which has been uncovered, which can be exploited. We've got a critical cellular modem chip defect which has surfaced. And maybe I'm the only one who's not worried about it. We'll see. Also we've got an interesting consequence of U.S. sanctions which is that named ransomware is being impacted as a consequence. And also in a separate story, ransomware is beginning - we have, as a consequence of these Conti leaks, we learned that some new development in the ransomware sphere is taking aim at our systems' boot firmware, the UEFI.

Leo: Uh-oh. Uh-oh.

Steve: Which is not where we want them going next. But they're going to. We've got a bit of errata and a bit of closing-the-loop feedback. Then we're going to talk about - I titled today's podcast "Passkeys, Take 2" because four weeks ago it was the title of the podcast, and I need a little bit of follow-up.

Leo: Oh, good.

Steve: That I think all of our listeners will find very interesting.

Leo: Yeah, I'm looking forward to that. One of our listeners, Redacted in the chatroom, has noted that on your show notes, at least on the Mac, the Mac interprets this number 874 for 06-07-22 as a phone number. And I don't know what'll happen if I dial it, but I don't think I will. I just thought that was kind of interesting.

Steve: That is interesting.

Leo: Yes. I'd never noticed that before. Thank you, Redacted. I don't know what country code that is.

Steve: I have noticed, and I've never asked you about this, I've noticed that sometimes, on my Pad at least, Safari doesn't show the symbols that other browsers show. They've got the little missing icon squares. Like, why? Is this hard?

Leo: It's a font thing, Steve.

Steve: Oh, okay.

Leo: Yeah, I think that whatever font you're using there, it doesn't have those Unicode characters. I think honestly by now every font should have - but you know there's potentially, I think, 65,000 Unicode characters.

Steve: That's true.

Leo: I mean, it's just such a large number, you don't want to put that all in a font set. So occasionally some Wingdings get turned into squares.

Steve: You'd need to have the add-on ROM if you put that...

Leo: Yeah, no kidding. Every font's 100MB.

Steve: Yeah. So this one, this is not high-brow, but it's just a fun little ditty. So the caption on this picture reads: "I don't get it. We keep changing the password, and we still have a leak." And the picture is two people standing in front of a corporate bulletin board where in 48-point type it says: "Today's security access code: 55523." And again, yes, yeah, wonder where that leak could possibly be occurring. That's a mystery.

Anyway, okay. So in a follow-up to the quite unflattering coverage the tech press, including this podcast, gave to the New South Wales ridiculously insecure Digital Driver's License, someone over at ServiceNSW was told that they needed to offer a rebuttal. So here's the official reply from ServiceNSW. Starts out: "This issue" - and, like, which issue? Like take your pick. "This issue is known and does not pose a risk to customer information." Okay. "The blogger has manipulated their own Digital Driver's License information on their local device."

Leo: Yeah.

Steve: Uh-huh. Right?

Leo: Yeah.

Steve: "No other customer data or data source has been compromised."

Leo: Oh, they thought it was a breach or something.

Steve: Well, okay.

Leo: Completely misunderstood it, yeah.

Steve: Yeah. Maybe. "It also does not pose any risk in regard to unauthorized access or changes to backend systems such as DRIVES." And of course no one ever said it did. But it's good that it doesn't. Then they said: "Importantly, if the tampered license was scanned by police, the real-time check used by NSW Police" - and they said "scanning mobipol" - "would show the correct personal information as it calls on DRIVES." So DRIVES is apparently the name of their backend database system.

Leo: Oh, you mean like a physical driver's license.

Steve: Yeah.

Leo: Well, I'm glad we did this.

Steve: Oh. Then they said: "Upon scanning the license it would be clear to law enforcement that it has been tampered with." Right, since nothing that's being displayed can be trusted. So the offline aspect of this is apparently not very useful.

Leo: So they're basically saying, well, just don't trust it. Okay.

Steve: Oh, but Leo, here's the kicker.

Leo: Oh.

Steve: "Altering the DDL is against the law."

Leo: Oh, no. No one would do that.

Steve: No one.

Leo: Lot of law-abiding people there.

Steve: And so it's kind of like a fake ID, which I would imagine is also against the law. But no one's ever used one of those. Finally, the DDL has been, fortunately, "independently assessed by cyber specialists and is more secure than the plastic card."

Leo: [Buzzer sound]

Steve: Exactly. Like, okay, were these by chance the same cyber specialists who came up with the system's design in the first place?

Leo: Yeah, yeah. Our geeks tell us it's safe.

Steve: And really, what does "more secure than the plastic card" mean? How can the security of a physical card be compared against a software solution? After all, everyone always says that all software has bugs. A physical card doesn't have software bugs. Seems like that might be more secure. And of course we talked last week about the difficulty of acquiring the paper and the printer, and apparently it's got some multilayer laminated sandwich with a different color so you look at the side of the card, and you see a little orange stripe or something.

Anyway, at least now we know why nothing happened three years ago, back in 2019, when the egregious and completely avoidable problems with this system were first publicly displayed. ServiceNSW apparently employed the now-famous "These are not the droids you're looking for" diversion by claiming that "The DDL system is working exactly the way we intended, and you all just don't understand why this is what we want."

Leo: That is such a cover-your-ass response. Unbelievable. Unbelievable.

Steve: Really. It really is. Yeah. Well, you know, government here to help you.

Okay. ExpressVPN, as you reminded our listeners, a sponsor of the TWiT network, last Thursday announced that it would be removing all of its India-based VPN servers in response to a new cybersecurity directive issued by the Indian CERT, the Indian Computer Emergency Response Team. However, that doesn't necessarily mean, they said, that users of ExpressVPN will be out of luck.

ExpressVPN wrote that: "Rest assured our users will still be able to connect to VPN servers that will give them Indian IP addresses and allow them to access the Internet as if they were located in India."

Leo: Oh. How do they - I guess the pool of its - that's interesting. Yeah.

Steve: That's interesting, yeah. As long as you are able to route those IPs...

Leo: You could put a router there, not a server.

Steve: ...to somewhere else; right.

Leo: Yeah, there you go, yeah.

Steve: Right. So they said: "These 'virtual' India servers will instead be physically located in Singapore and in the U.K."

Okay, so what happened? CERT-India will be enforcing new controversial data retention requirements that are set to come into effect three weeks from today, on June 27th. These new rules require VPN service providers to store subscribers' real names, contact details, and IP addresses assigned to them for at least five years. India's CERT stated that the user data being logged will only be requested for the purposes of "cyber incident response" - which of course could be anything - "protective and preventive actions related to cyber incidents."

So India's CERT, after I'm sure a lot of pushback, they clarified that this rule does not apply to corporate and enterprise VPN solutions, and they're only aimed at those operators who provide proxy-like services to "general Internet subscribers/users." In other words, to any and all VPN service providers.

In their statement, ExpressVPN said: "The new data law, intended to fight cybercrime, is incompatible with the purpose of VPNs, which are designed to keep users' online activity private. The law is also overreaching and so broad as to open up the window for potential abuse."

And in addition to the new rules, which are called Cyber Security Directions, they also require firms to report incidents of security lapses such as data breaches and ransomware attacks within six hours of noticing them. I think in the U.S., what was it, 72? So six hours in India. India's move has not only sparked privacy concerns, but has been criticized as ambiguous and overly broad, with many pointing out a lack of clarity on the scope of incidents that come under purview of this upcoming directive.

In a statement, the Internet Freedom Foundation said: "Such excessive requirements for collecting and handing over data will not just impact VPN service providers but VPN users as well, harming their individual liberty and privacy. In the absence of sufficient oversight

and a data protection framework to protect against misuse, such requirements have the potential to enable mass surveillance." Which has got to be what's actually going on here.

And unfortunately, this feels more like the future which we're heading toward than the past we are receding from, or that is receding from us. Governments are increasingly becoming uncomfortable with the idea of having no means to surveil their citizens and others who reside within their borders. So the great decryption debate is far from over.

And speaking of pulling the plug, what do Algeria, Iraq, Jordan, Sudan, India, and Syria all have in common? Their governments have reacted to out-of-control cheating on tests by high school students by completely shutting down their national Internet service during the period of time that tests are being taken.

Leo: What? Oh, go try that here. Holy moly. Oh, my god.

Steve: I know. I was thinking the same thing, Leo. Can you imagine? It must be, although I don't think it is really, but you'd have to imagine that those countries' use of the Internet is sort of circumstantial, or not core. As you said, you could not do that in the United States.

Leo: No.

Steve: But anyway, so the most recent instance of this occurred last week - two, actually - and this week in Syria, which scheduled a series of four planned outages, each lasting 3.5 hours. The first two occurred, as I said, last week; and the next two, actually one is set for today and one on the 12th. The outages are performed via BGP by removing Syria's routing from the global Internet.

Leo: Wow, that's one way to stop cheating. Holy cow.

Steve: Thus cutting it off completely from the rest of the world.

Leo: That's mind-boggling.

Steve: It is. In this day and age, Leo, I mean, and that's why you think that the only way it could be possible would be that it's just, like, Syria doesn't depend on the Internet to the degree that we do. But how can that be? Anyway, prior to Syria implementing these exam blackouts, which began in 2016, it turns out, I mean, this actually was a problem. Test questions would begin appearing on social media sites 30 to 60 minutes before each exam, thus allowing cheating students to circulate correct answers and compromise the integrity of the tests. So now, as hundreds of thousands of Syrian high school students sit to take their national exams, Syria is taking the extreme proctoring measure of shutting down national Internet access. Wow.

The pressure on Syrian students is great since their performance on these standardized tests largely determines what higher education options they will have access to, which in turn largely controls their economic futures. Doug Madory, the Director of Internet Analysis at Kentik, said: "The stakes for the exams are so high that there's an

assumption that everyone is cheating." So the exam blackouts operate in Syria both by blocking, well, actually if you pull out BGP, you know, hardwired and mobile Internet access, in the hours before the exams, as paper tests are printed and physically distributed across the country. So in order to minimize leakage, they already, like, print, I mean, the ink is still wet on these things when the students get them, in order to keep them from escaping. And they went one step further and just pulled their BGP routing from the Internet.

And as I said, this strategy is not only being used in Syria. Iraq previously drew criticism from digital human rights groups for ordering local Internet providers to shut down during school exams in the summer of 2015. The academic-related Internet shutdowns have been reported in India, as well. And last year more than 25 million people faced a mobile Internet shutdown in the Indian state of Rajasthan during a local teacher eligibility exam. So I guess even the teachers were cheating.

Leo: Well, that's what happens when you have these high-stakes exams.

Steve: Yes.

Leo: I mean, France does this. A lot of countries do this. They're all at the same time on the same day countrywide, which is why they can shut down the whole country.

Steve: Right.

Leo: But, I mean, I guess there's an agreement among the citizenry that this is important. We're going to do this right.

Steve: We're just all going to, you know, bite the bullet.

Leo: That's pretty wild. Wow.

Steve: Wow. So "Follina," I just love that name, is under active exploitation. So under the heading of "Well, that didn't take long," we have last week's Microsoft mess which Kevin Beaumont named "Follina," as we know, after the area code of Follina, Italy, which appeared in one of the exploit documents. And recall that this is the ms-msdt:// protocol vulnerability that was being abused through Office, all versions of Office. And by using an RTF extension on the file, you were bypassing the protected viewing mode. So it's now under widespread and quite aggressive attempts of abuse.

A most likely Chinese state-aligned threat actor has been observed attempting to exploit this Follina vulnerability, targeting target government entities in Europe and in the U.S. The enterprise security firm Proofpoint, I think they're Israeli-based, said it blocked attempts at exploiting this remote code execution flaw being tracked as CVE-2022-30190, with a CVSS of 7.8, no fewer than 1,000 phishing messages containing a lure document which were sent to targets. Proofpoint said: "This campaign masqueraded as a salary increase and utilized an RTF with the exploit payload downloaded from the IP 45.76.53.253."

The attack payload is a Base64-encoded PowerShell script which functions as a downloader to retrieve a second PowerShell script from a remote server named "seller-notification.live." The second expanded script checks for the presence of virtualization because, you know, that's one of the ways that these scripts are analyzed is they're stuck in a VM in order not to get, you know, in order to create containment. So now increasingly scripts and malware are checking to see whether they're being run under virtualization, in which case they don't do anything bad. So it checks for virtualization, steals information from local browsers - and in the write-up of this there was a list of like all of the passwords that this thing attempts to exfiltrate from every browser that we know about - and also mail clients and file services. It conducts local machine reconnaissance and then zips it all for exfiltration back to IP address 45.77.156.179.

The phishing campaign has not been linked to a previously known group, but Proofpoint said that given the specificity of the targeting and the PowerShell payload, it has wide-ranging reconnaissance capabilities. They didn't think it was an amateur. They believed it to be mounted by a nation-state-level actor. The vulnerability, as we know from Microsoft, remains unpatched, with Microsoft urging customers, I guess who ask, they're not being proactive, to disable the protocol to prevent the attack vector.

And in the absence of a security update, the great guys over at 0patch, you know, numeral 0patch.com, have released one of their unofficial micropatches to block the ongoing attacks against Windows systems. Let's see. We're the first Tuesday of the month. So next Tuesday the 14th hopefully we're going to see that this thing gets updated and fixed. 0patch's founder Mitja Kolsek said: "It doesn't matter which version of Office you have installed, or if you have Office installed at all. The vulnerability could also be exploited through other attack vectors."

Proofpoint said that the extensive reconnaissance conducted by the second PowerShell script demonstrates an actor interested in a large variety of software on a target's computer. So this, coupled with the tight targeting of European and local U.S. governments, led them to suspect a campaign that has a state-aligned source.

And as we sign off from this follow-up, let's all remember that, as I noted last week when this nightmare began, it was the middle of April, about a month and a half ago, when this was responsibly reported by a credible security research group as being at the time, mid-April, under active exploitation. The report was made to Microsoft's Security Response Center. The group providing the report provided a copy of the in-the-wild, real-world Microsoft Office document which demonstrated the nature of the exploit. But because the exploit didn't immediately reproduce itself and fall at the feet of the Microsoft MSRC guy, he just blew it off, saying it wasn't a security problem. Well, it certainly is now. And a different Windows URL schema can be abused. Leo, I'm going to take a sip of water.

Leo: Okay.

Steve: Let's tell our listeners how we're here listening to this.

Leo: Okay.

Steve: Last week, I opened our coverage of the latest Microsoft mess by stating: "We have a new, head-buried-in-the-sand, quite pervasive Microsoft Office zero-day remote code execution vulnerability which is now being used in attacks." And of course I was referring to the issue we were just talking about, which has now since then exploded in

the wild. And of course the head buried in the sand I was referring to was Microsoft's decision to ignore the early warning that they were given a month beforehand.

And last week I also observed that this msdt:// protocol scheme wasn't a bug, it was a feature; and that this would make its remediation all the more difficult because it could not simply be turned off globally since there might well be some users who were dependent upon that feature because, again, it's not a bug. Unfortunately, it's a feature which has now been revealed to be insecure. And it's not alone.

We're back here today because, sure enough, another similar feature of Windows has just surfaced. This time it uses a different scheme. This one is search-ms:// protocol. BleepingComputer reported that a security researcher by the name of Matthew Hickey, a co-founder of Hacker House, found a way to combine a newly discovered Microsoft Office OLEObject flaw - because yes, as we've commented, Leo, OLE is still with us. You could combine this OLEObject flaw with the search-ms: protocol handler to open a Windows search window from a Word document.

And by "search window" we mean that a Windows Explorer search results window will open, showing a list of files to run; but that list of search results, that's search result files, can be sourced from a hacker-controlled remote server anywhere. Whoopsie. The result can be an extremely convincing "Your software must be upgraded to proceed" style attack. It's convincing because the dialog runs from Windows. I mean, it is a Windows dialog, and so it's not going to be, you know, like in the browser or stuck within some borders anywhere. It is coming from Windows. And although it's triggered by an untrusted Word document which the user can have received through any channel, such as spoofing mail, it also can be zero-click.

Bad guys can use this hack by sending phishing emails claiming to be security updates or patches that need to be installed. The OLEObject flaw that Matthew Hickey found bypasses what would normally be a confirmation dialog which would pop up, which Word would show, if you were trying to use the search-ms:// protocol scheme. Now that gets suppressed, making these very convincing-looking, and fasten your seatbelt.

And this might not trick and probably would not trick listeners of this podcast. But we've made our computers so complex that most users have no idea what's going on. And legitimate software does open pop-ups telling us that we need to upgrade this or that so often that it's become commonplace. Microsoft, as I mentioned, did incorporate a warning confirmation dialog in an attempt to prevent the abuse of this confusion, but this most recent hack arranged to bypass those warnings.

And it seems to me that, again, there are a whole bunch of protocol handlers in Word that can be abused. This creeping featuritis is insidious. And it appears to be unavoidable as products mature. In the case of Windows, what has now become an incredibly complex system has been created that no one fully understands. And I mean that seriously. I mean, the Office group are the size of a company themselves. And they're producing stuff, having some interaction over on the Windows side because they need this or that feature added. And you're just going to get mistakes in this process. Security, as we know, is unforgiving; and only one mistake is all that's required for a bad guy to get in.

In this particular instance of over-engineered complexity, there are many similar protocol handlers which can be triggered by Microsoft Word documents, often without requiring any user interaction. And once again, Microsoft apparently just doesn't get it, or at least they don't want to. When BleepingComputer asked Microsoft how they planned to resolve this most recent foible, Microsoft replied: "This social engineering technique requires a user to run a malicious document and interact with a list of executables from an attacker-

specified network share. We recommend users practice safe computing habits and to only open files that come from trusted sources."

Uh, duh. The whole point is that the abuse of these protocol handlers allows for the creation of zero-click exploits against users who merely open documents or for the creation of extremely convincing spoofs which hide what's really going on. So when Microsoft says: "We recommend users practice safe computing habits," well, the users think they are. I mean, they don't want to do anything bad on purpose. But it happens anyway. Grumble.

Okay. What is billed as a critical UNISOC (U-N-I-S-O-C), I'm sure SOC is System on a Chip, vulnerability affecting millions of Android smartphones probably isn't anything to worry about. If you were worried by headlines that you saw like that, I don't think there's anything to be too concerned about. Checkpoint went to the trouble of reverse engineering the firmware of a widely used cellular modem chipset. And they uncovered a buffer overflow which could be exploited to hang the chip. But there's no suggestion that attacker-provided code could be injected. And even if it could be, it's unlikely that much damage could be done since the chipset in question here is only running on a tiny subsystem of the entire device. It's the cellular modem. And while in theory there could be some damage done, in this case it just creates a denial of service, meaning that the service crashes.

But the headlines were followed by breathless statements such as: "A critical security flaw has been uncovered in UNISOC's smartphone chipset that could be potentially weaponized to disrupt a smartphone's radio communications through a malformed packet." Well, yeah. That's true. The company in question, UNISOC, is based in Shanghai and is the world's fourth-largest mobile processor manufacturer following MediaTek, Qualcomm, and Apple. So, I mean, they're significant. They currently account for around 11% of all System on a Chip components.

The problem has been patched after having received a somewhat surprisingly high CVSS of 9.4. So that seems high to me for a denial of service on a cellular radio. But I suppose the fact that an adversary could simply send a malformed packet to crash a handset's radio seemed like a big worry to whomever assigned the CVS. In any event, Google actually will be pushing an update in their June 2022 release for Android, and I do congratulate Checkpoint for their reverse engineering work.

As I've noted before, it seems wrong to me that widely used proprietary products need to be reverse engineered to have their security verified by a third party, but that's the closed-source world we live in today. And increasingly it seems clear that having the source in the open, the source for very important significant components of the ecosystem should be a requirement, really, of somebody agreeing to license it and put it in a large number of devices.

So, ransomware sanctions are causing trouble for the ransomware cretins. Yay. In an interesting bit of ransomware news, sanctions are turning out to have quite an impact on the ransomware business. The problem is that even though the attackers are not law abiding, they're as we know anything but, their victims are. So enterprises which have been hit by ransomware and are now being held at ransom are legally prohibited from making any ransom payments, through any mechanism, even if they wanted to.

Mandiant's research paper published last Thursday was titled: "To Hades and Back: UNC2165 Shifts to LockBit to Evade Sanctions." And that headline requires a bit of explanation, which Mandiant then provides. They explain: "The U.S. Treasury Department's Office of Foreign Assets Control (OFAC) sanctioned the entity known as Evil Corp back in December of 2019, citing the group's extensive development and use and

control of the Dridex malware ecosystem. Since the sanctions were announced, Evil Corp-affiliated actors appear to have continuously changed the ransomware they use."

And I have a chart from the Mandiant report showing a block of time when they were using BitPaymer. Then they shifted to DoppelPaymer. And then for a while they were using WastedLocker. Then they used Hades for a while, then Hades PhoenixLocker, and then finally Hades PayloadBin. So again, continually making these changes over time. They said: "Specifically, following an October 2020 OFAC advisory, there was a cessation of WastedLocker activity and the emergence of multiple closely related ransomware variants in relatively quick succession." In other words, the names were changing, but not much else was changing. "These developments," they wrote, "suggested that the actors faced challenges in receiving ransom payments following their ransomware's public association with Evil Corp.

"Mandiant," they wrote, "has investigated multiple LockBit ransomware intrusions attributed to UNC2165, a financially motivated threat cluster that shares numerous overlaps with the threat group publicly reported as Evil Corp. UNC2165 has been active since at least 2019" - and notice when that occurred, so yup, somebody new popped up, perhaps changing their name - "and almost exclusively obtains access to victim networks via the FakeUpdates infection chain, tracked by Mandiant as UNC1543. Previously," they said, "we've observed UNC2165 deploy Hades ransomware. Based on the overlaps between UNC2165 and Evil Corp, we assess with high confidence that these actors have shifted away from using exclusive ransomware variants to LockBit, a well-known Ransomware as a Service (RaaS), in their operations, likely to hinder attribution efforts in order to evade sanctions."

So Mandiant's paper then delves into all the details of the intelligence that they collected to track these activities and to draw these conclusions. But I mostly wanted to make the observation, which I thought was interesting, that just as the Conti gang apparently shut down and disbanded due to the trouble they caused for themselves by siding so clearly with Russia, and then falling under the Russian sanction umbrella, thus being unable to receive ransom payments from the West, similarly, U.S. Treasury Department sanctions on many other ransomware operators prevent them, too, from receiving payment from U.S. resident victims. So the prior practice of building up a big public reputation no longer serves the financial interests of these ransomware gangs.

The next thing I wanted to talk about is what this is in the middle of, or picks up on, which is the fact that ransomware groups have now been spotted successfully compromising motherboard firmware. One of the consequences of Conti's boasting about how they side with Russia, as we talked about a couple times, is that somebody within the Conti membership who had good access to what was going on leaked a bunch of the internal chats that was always meant to stay private.

So the following leaked message posted exactly one year ago on June 7th, coincidentally, 2021. It reads, and this is good English, considering it was probably a native Russian speaker, from someone named Stern. He wrote: "Hi. Things are good. I apologize for not immediately responding. I haven't communicated through a toad" - that's what it says, maybe that's, I don't know, a typo or something. "I haven't communicated through a toad for a long time." And Leo, for what it's worth, I doubt that you or I have ever communicated through a toad. I'm just saying.

Leo: It probably means something in the hacker world. I don't know. Yeah.

Steve: So Stern said: "I haven't seen what you wrote. Now I'm finishing a full report on the mechanism of operation of the Intel ME (Intel Management Engine) controller and the

AMT technology based on it. Recovered a bunch of undocumented commands using reverse" - meaning reverse engineering - "interface dump, and fuzzing. Unfortunately, the starting theory based on the presentation of Embedi/PositiveTechnologies reporters was not confirmed in the form in which they presented it."

Okay. So that was probably referring to back at the time a report of vulnerabilities in the Intel Management Engine. So this guy jumped on those, rubbing his hands together, thinking, oh, cool, I'm going to go pursue this. So he's saying: "Unfortunately, the starting theory based on the presentation of Embedi/PositiveTechnologies reporters was not confirmed in the form in which they presented it." Meaning it wasn't what they said, but he stayed with it.

He said: "But there is another legal mechanism to activate AMT, but so far it has not reached the working software. At the moment I make a sniffer buffer that provides the HECI interface because it is all configured in UEFI. Then the sniffer took a little longer. After I fully restore the command set, the POC will be prepared. There are ideas. If we talk about the topic of UEFI, then this is not just a load dropper, but also perhaps some daemon of the level of SMM processors" (System Management Mode processors).

He said: "Plus since now I have tightly studied the Management Engine controller, the idea is to test such functionality as rewriting the SPI flash drive through it. Usually this controller is allowed to write to the flash drive, which cannot be said about the processor, and some commands were found that are responsible for this functionality."

So bottom line is, in this posting, which was leaked from Conti, and this was made, the original posting was from a year ago, this guy is saying, "I have achieved what I needed to as a consequence of this reverse engineering of the Intel Management Engine firmware on the motherboard of pretty much everything."

So the conversations among the Conti members have shed light on the syndicate's attempts to search for vulnerabilities related to the Management Engine firmware and BIOS write protection. They reverse engineered the system to locate undocumented commands and vulnerabilities in the Management Engine interface, achieved code execution in the Management Engine to access and rewrite the SPI flash memory, and dropped System Management Mode-level implants which could be leveraged to modify the OS kernel. That is, the operating system running, that is booted on this motherboard.

The leaked chats show that the work ultimately resulted in proof-of-concept code last summer that can obtain System Management Mode code execution by gaining control over the Management Engine after obtaining initial access to the host through traditional vectors like phishing, malware, or supply chain compromise. Basically, we're always talking about how you get into a machine. And then often you need to elevate your privilege. And what you're probably doing is looking to see where you are and then moving laterally. Now this group, some members somewhere, have developed the technology to go down rather than out and across, that is, they are now able to infiltrate the firmware of motherboards.

Security researchers who have been privy to these chat logs have observed that: "The shift to ME firmware gives attackers a far larger pool of potential victims to attack, and a new avenue to reaching the most privileged code and execution modes available on modern systems." And of course as we know, and as I mentioned in my notes that I deleted or somehow didn't make it into the show notes, it's enough trouble getting people to update their operating systems and their application software. The bad guys know that there is a serious update log, or lag, rather, in getting that to happen. So imagine how many more systems' firmware is never touched.

And to that add the fact that I'm sure many of us here listening to this podcast, because we tend to have propellers spinning on our heads, have done firmware updates. I've encountered, and I'm sure everybody has encountered, warnings saying basically leave the firmware that you have alone unless something, like there's some hardware problem that you're coming here to update your firmware for. That is, if it's not broke, don't fix it. And so this notion in firmware land is still sort of pre-viral. It's the only reason to update firmware would be to fix some hardware-level compatibility problem, not because there might be something evil crawling around in there that you want to update.

So as a consequence, I mean, I'm sure the majority of systems that are out and running are using firmware that has never been updated since the system was first installed, despite the fact that this podcast has covered multiple firmware problems, and it was one of those apparently a year ago that got a member of the Conti group, or at least somebody posting in the Conti secure chat channel, that hey, you know, I've figured this out. We now, if we can run code in the user's OS, we can now infiltrate their UEFI BIOS and get persistence and complete invisibility...

Leo: Terrible.

Steve: ...from traditional antimalware. Yeah, Leo, it really is.

Leo: If you have, I mean, don't most PCs now have in the firmware a signing certificate for the firmware? I know they do for the boot, right, the secure boot; right?

Steve: Right.

Leo: Is the firmware not checked?

Steve: There's no way for it to check itself.

Leo: Oh, of course. Duh.

Steve: Because the root is the ultimate authority.

Leo: Right.

Steve: And so there's no one, there's no third party to say...

Leo: No higher authority, yeah.

Steve: ...yeah, you're fine, go ahead, yeah.

Leo: Boy, that seems like something you might want to put in BIOS or something, somewhere.

Steve: Well, what you really want is to rigorously write-protect this.

Leo: Right.

Steve: And, you know, in the old days you had motherboards with jumpers, and some server motherboards will still have a physical write-protect jumper.

Leo: Right.

Steve: That will just disable the ability for the firmware to be modified.

Leo: Well, that protects BIOS, but it doesn't protect the UEFI because that's on the hard drive; right?

Steve: Well, no.

Leo: And you write to that all the time, I think. Yes?

Steve: Yeah, well, the UEFI is on the motherboard. There is some information stored on the hard drive. But of course you're able to put new hard drives in, and the UEFI survives. So the UEFI is actually in, well, is in NVRAM.

Leo: All I know is from installing Linux and stuff, if you put a new hard drive in, it wouldn't boot because you don't have an operating system on it.

Steve: Right.

Leo: If you install a UEFI-aware operating system, it is going to write some stuff on the hard drive.

Steve: Ah, right. But so that's like the boot sector on the partition. But UEFI is that stuff that you get to by hitting F2 or...

Leo: It's confusing because there's - some of it's in firmware, but some of it's in software; isn't it. So if you...

Steve: Well, what's in software is what the UEFI BIOS...

Leo: Boots to.

Steve: ...boots to and executes, yes.

Leo: But if you deleted your UEFI partition on a hard drive, you wouldn't boot.

Steve: Right.

Leo: Right. So there is firmware UEFI, but there's also software UEFI.

Steve: Right.

Leo: Are they talking about something that modifies firmware?

Steve: Yes. Yes. That's where the Intel Management Engine and System Management Mode and all of that is.

Leo: Oh, dear.

Steve: Yeah. So they're talking about going down and physically modifying the firmware on the motherboard so it doesn't matter if you reformat your drive or if you install a new operating system or you do anything.

Leo: Wow. Wow.

Steve: Yeah. And I imagine before long, since the UEFI firmware has known bugs, we've talked about them on the podcast, it's got known bugs in the motherboard firmware, and that is not being updated. And as I said, it's hard enough to get operating systems updated, and applications.

Leo: I get BIOS updates all the time, though; right? I mean, don't I? I mean...

Steve: Well, and that's the problem is that the fact that you get BIOS updates means that the BIOS is not protected against being modified.

Leo: Right, it can be modified.

Steve: By software.

Leo: That's right, yeah, that's right.

Steve: Yup. And once something bad gets in there, because the BIOS is updating itself, it can disable the self-update so that the BIOS will no longer accept an update after it becomes malicious. I mean, it's really bad.

Leo: Wow.

Steve: And, you know, and this is the situation we've created for ourselves.

Leo: Yup.

Steve: I had a fun piece of errata to share from @anocelot, I think that's how I pronounce his name. He said: "@SGgrc: Security Now! Errata - Grover the Muppet was blue."

Leo: Okay.

Steve: "Not green."

Leo: Don't mess with that.

Steve: He said: "He's not green. You're probably thinking of Oscar the Grouch." And he said: "Just trying to save you from the Muppet Mafia who will not tolerate inaccuracies." And I hope I don't disappoint you if I have no idea what color any of them are. I know the Cookie Monster was blue because I had a Cookie Monster with big goo-goo eyes. It was an early GRC mascot.

Leo: Oh, how funny.

Steve: As for Grover and...

Leo: No idea.

Steve: ...Grouch or Oscar...

Leo: Oscar the Grouch, yeah.

Steve: Yeah.

Leo: Lives in a garbage can, yeah.

Steve: Oh, and several ServiceNSW Digital Driver's License owners, that is, we've got listeners, not surprisingly, who have these DDLs; right? And they assured me they have not tampered with them, so that's good.

Leo: Good boys, yes.

Steve: They did note that a simple screen capture would fool no one because the screen incorporates a number of animated effects. Which that's where all the time got spent; right?

Leo: Right.

Steve: Doing animation instead of the crypto. There's a flower that animates in the upper left-hand corner, and the phone's inertial positioning is used to animate a large multi-colored flower in the background wallpaper. So you look at it, and then you turn it back and forth. And, you know...

Leo: Oh, can't fake that. Ooh.

Steve: Ah, no, Leo. That's high tech.

Leo: Yeah.

Steve: Yeah. So if only its developers had given as much thought to the security of the device as they gave to its flash.

Leo: It's better than plastic.

Steve: That's right. And Larry Wilson tweeted as closing the loop, he said: "I think you've undersold the benefit of triply encrypting and going from 256 bits to 768 bits. Because those are logarithms, and adding represents multiplying, the growth in security is immense. Compare the difference between 64-bit keys and 128-bit keys."

And, okay. So Larry was referring to Bryant McDiarmid's question I replied to last week about whether triply-encrypting with 256-bit keys each time would be equivalent to taking the sum or the product of those bits. Now, I answered correctly. But in re-reading Bryant's question, I could see what Larry meant. Bryant asked: "Hey Steve, quick question. If I encrypt a file with a 256-bit encryption three times with three different passwords, what is the resulting bit strength? Is it 256 plus 256 plus 256? Or 256 times 256 times 256?"

I answered Bryant's question correctly in that the resulting bit strength is the sum of the individual separate encryption key lengths. But as we know, each single additional bit of key strength that we add doubles the number of all possible keys since you have all of the original number of keys when that new bit is off, and all of the original number of keys again when that new bit is on. So obviously, each bit you add doubles the total number. So encrypting three times with 256 bits each time would result in 2^768

possible keys, which is a ridiculously large number which I, for the heck of it, put in the show notes just because it's kind of glorious. I mean, I didn't even count the groupings by three, or the digits. It is a huge number.

Leo: There is an Emacs command that will turn that into, you know, whatever it is, 180 gazillion. I will do that.

Steve: Oh, like it'll speak it for you.

Leo: It doesn't speak it. It actually spells it out in English.

Steve: Oh, my lord. And has man ever run out of names for those groups of three?

Leo: Oh, god, yeah; right? I mean, isn't that why googol, there's a googol? Because that's a one with a hundred zeroes.

Steve: Yeah, this has more than that.

Leo: Yeah.

Steve: So, I mean, at some point someone must have given up naming these decades, these like divisions.

Leo: Well, it's based on a kind of Latin base, so you can probably deduce it.

Steve: Oh. Great. Please, nobody tweet me the answer. Actually, it wouldn't fit in a tweet. So we're safe. And Leo, let's take our last break, and we're going to talk about Passkeys.

Leo: I am going to use the Calculator Soup...

Steve: Uh-oh.

Leo: ...word to number converter.

Steve: Good.

Leo: Since I don't have Emacs on this machine. Convert this number. Let's see if it can do it. It broke it.

Steve: Yeah. Oh, I mean, it's big.

Leo: It never came up with an answer. That's ridiculous.

Steve: 15525180930...

Leo: No, no, no, no, no. No need to - no need. That's funny. This is too big. All right. Oh, it has to be less than 200 characters. So, yeah, too big.

Steve: [Buzzer sounds]

Leo: I bet Emacs will do it. It's a bignum, but we'll handle it. I will check into it. All right. Now, Steve has been thinking about Passkeys. So is this going to eliminate passwords forever?

Steve: I originally had these thoughts filed under "Miscellany" because I didn't want to make a big deal about it. But the more I thought about them, the more they grew. And as I said, no other news of the week rose to any greater significance. So I decided to sort of more formally take a second look at Passkeys in the wake of yesterday's Apple WWDC presentation. So a bit of an @SGgrc tweet storm arose from our listeners yesterday following the keynote, where Apple...

Leo: Yes, because Apple announced it, yeah.

Steve: Yeah. Where Apple made a point of highlighting their forthcoming adoption of essentially the revised and much more practical FIDO2 public key authentication system under the unofficial designation, which has taken hold of Passkeys. And as we know, that was the title of our podcast on May 10th, four weeks ago. And so I wanted to thank and acknowledge all those who took the time to tweet. What I think set most people off was that this was the first time outside of SQRL that we've seen the very SQRL-like use of a QR code to allow logging onto someone else's machine using an identity stored in the user's smartphone. We haven't seen that from Google yet; but if Apple does it, then Android will have to follow. So I imagine it's just a matter of time.

One thing I did not highlight when I first talked about this four weeks ago was the uncomfortable unanswered questions surrounding manufacturer lock-in. Apple seems quite uninterested in allowing me to send and receive iMessages from my Windows desktop. Being an avid iPhone and iPad user, this imposes a constant and very real inconvenience for me. If I was using a Mac? No problem. From a Mac I have access to my iMessages. But not from Windows. And I've sort of come up with a workaround using iCloud for Windows in order to send things over into the Apple world and then stick them in a posting or an iMessage. But it's more work than it should be.

So I worry that Apple's use of this Passkeys technology will be similarly and characteristically an Apple-only solution, synchronizing only among Apple's authentication devices and not to Windows and Android. And that's a huge practical problem for Passkeys adoption which SQRL never had. So I want to make sure that everyone understands what the difference is because there is a takeaway from this, and why we're almost certainly heading for trouble which, among all the celebration, no one seems to have talked about this yet.

Okay, so why is that important? Both systems, FIDO and SQRL, share the common property that the authenticating client a smartphone, a fob, or a PC creates a public key pair for a website. The process of registering the user's identity to that website just involves providing the remote website with only the public key of the pair. And that's the essence of both approaches. That's FIDO and SQRL.

Subsequently, verifying any user's identity amounts to verifying that the user is holding the matching private key. So to perform that verification, the website sends a random unique nonce, it's a challenge, to the user's authenticator, which signs it using its private key, which it never lets go of. The signed blob is returned to the site, which verifies the signature using the public key that it originally received from the client during its registration. And that's it.

Leo: This is not so very different from public key crypto in general, how you would verify a signed email, for instance, when I send you an email.

Steve: Yes, that's all it is. I mean, that's exactly what it is. Everybody has your public key, and you sign your email using your private key, and they're able to verify the signature on the email which proves that it came from you, or at least someone holding the private key.

Leo: And because Apple has widespread biometric identification, they have a really good way of verifying you're you. I don't know if you caught this. I thought this was very SQRL-like. They said, well, what if you log into a site at a library, a computer you don't own? And did you catch this? They showed a QR code.

Steve: Well, that is SQRL-like.

Leo: Yeah.

Steve: In fact, that's what SQRL allows, and it was part of my original demos of SQRL. And that's why so many people sent tweets is they said, hey, Apple's using QR codes.

Leo: Right.

Steve: With Passkeys. And it's like, well, good, because that's a use case which is important. I mean, I'm not - don't get me wrong. I'm not imagining that SQRL is going to get adopted, at least not yet. We've clearly - we're going FIDO2. But there are a couple differences which are significant.

Okay. So we have the private key. Websites have the public key. They send us something that they've never sent before, which we sign. They verify the signature. Thus they know we have the private key, exactly as you said, Leo, like email. Where the two systems, FIDO and SQRL, that is to say Passkeys and SQRL, crucially and importantly differ is how those original key pairs are created. The FIDO2 Passkeys system creates key pairs randomly, whereas SQRL calculates them deterministically.

Okay, so remember that SQRL's primary design feature, the core concept behind SQRL from the start, was the idea of using a single grand master key and hashing each

website's logon domain to automatically create per-domain public key pairs. That simple innovation eliminates all need for dynamic synchronization of randomly generated Passkeys among devices because each separated device will derive the same per-site private key from one grand master key that they share. So you load that one master key into each of your various authenticating devices under SQRL, and that master key generates all of the per-domain subsidiary key pairs forever.

Okay, but SQRL is not the system we're going to get. And that almost might be deliberate since the system it appears we're going to get really will create lock-in. I've read the glowing celebratory announcements about how Apple, Google, and Microsoft are going to be implementing Passkeys. But the tech press really needs to start asking, what about cross-platform Passkey sharing and interoperability? You know, awkward though a username and password are, the one thing they have going for them is that they are platform agnostic.

Okay. In the near future, when you use an iPhone, an iPad, or a Mac to register your identity as a Passkey on a website, that iPhone, iPad, or Mac creates a random public key pair, and the private key is held close and never released. Which of course is important because releasing it would defeat the system's security. Apple will back it up securely to iCloud. And I'm sure they will freely synchronize all of a user's Passkeys among the devices Apple controls and that are the user's. But is Apple going to dynamically synchronize those private keys among Android and Windows authentication devices? When pigs fly.

If they keep their keys to themselves, and if Google and Microsoft each keep the private keys that they create to themselves, then we have a fragmentation disaster on our hands. You register a Passkey on a Windows device, but none of your Apple devices will know that secret private key. So you can only log on with the device family which originally registered at that website. Well, that's a mess. In today's highly heterogeneous computing environment, the single most compelling benefit of password managers is that they are cross-platform. Would anyone be comfortable using a password manager that was not?

Now, I suppose if you are 100% all-in on Apple, which of course is what they want, then an Apple-only solution would be okay. But it's not clear how you could ever change your mind. I'm all-in on iPhone and iPad, but the things I need to do can only be accomplished with Windows, and I dare say DOS. So as nice as Apple's Passkey system may be, and as much as I'm a devoted iPhone and iPad user, I can't use Apple's Passkey system until I'm sure there will be a means for also using its Passkey registrations which I created on my iPhone or an iPad, under Windows because we should be able to.

Leo: Would it be sufficient, yeah, would it be sufficient to say here's your private key? In other words, if I could take my private key, it's on my iPhone obviously, and just send it to my Android phone, and register it with Passkeys on my Android phone, isn't that all I need is that private key?

Steve: Yes. And, for example, SQRL does that because that's the solution.

Leo: You have to be portable. So it wouldn't be a hard thing for Apple to make it visible somehow; right?

Steve: All they have to do is put it up on a QR code and allow...

Leo: What they're going to say is, oh, no, because people don't understand the difference between public and private keys.

Steve: Exactly.

Leo: In fact, we know that because that's how NFTs keep getting stolen.

Steve: Exactly.

Leo: And so they're going to give out their private key inadvertently. And then of course the jig is up.

Steve: Apple is going to hold this close, they are going to synchronize their devices, and they are never going to allow those keys to be exported.

Leo: Do you have but one private key? You wouldn't have - you'd have a private key per person; right?

Steve: Per site. No, that's what - that's the difference.

Leo: Oh. So it is a large amount of data then.

Steve: Yes.

Leo: You don't - oh. Why don't they just - is that what SQRL does? I have my key that ultimately...

Steve: SQRL was one key.

Leo: Yeah. That's what we should do.

Steve: I know.

Leo: In the long run it could be tattooed on your inner thigh or something. But everybody should have that one private key. That's what I do with PGP.

Steve: Wouldn't want it tattooed. That is the SQRL gift...

Leo: Yeah, one key.

Steve: ...was one key, and it allowed you to have anonymous, secure, unhackable, anonymous logon everywhere.

Leo: Because I'm me. Only I have access to that.

Steve: That's right. That's not what FIDO did. FIDO generates them at random.

Leo: Oh, that's a mistake.

Steve: I know. It's bad.

Leo: That's a mistake. Because now you have a - this is just another password system.

Steve: That's all it is.

Leo: Because now you have a password, which is just...

Steve: It is different. The thing that's different is that the dynamic authentication, that the website sends you a challenge which you sign and return. So what that means is all they have is your public key.

Leo: Right.

Steve: It doesn't matter if it escapes from them. That's the only thing we've achieved with FIDO is that websites can now - remember how many times I said SQRL gives websites no secrets to keep.

Leo: Right, right.

Steve: That's this.

Leo: It eliminates breach problems. But this still is a terrible...

Steve: It is, it's awful.

Leo: Why would they solve it that way? It seems so obvious that everybody should have - and then you solve it by if I have multiple accounts, for instance, for tax preparation software, I have multiple accounts because I do my mother's thing; right? My mother's taxes. So I could generate a private key for her that I would keep and a private key for me that I would keep.

Steve: Yup.

Leo: And I would have then multiple logins to a single site.

Steve: Yup.

Leo: But I still...

Steve: You're describing SQRL.

Leo: Yes.

Steve: You're describing SQRL.

Leo: One key per person. I don't understand why that's not obvious.

Steve: That's the right solution, and that's not what we got.

Leo: I didn't realize that - I should have listened more carefully when you were describing this. I didn't realize Passkeys generated a unique private key for every site and app. That's...

Steve: Well, and if you think about it, they have to. Otherwise they'd be giving every site the same public key.

Leo: Right.

Steve: And that can't happen because then you would have no - you would have no security.

Leo: Something unique, yeah. So how do you solve that with FIDO?

Steve: Well, the only way to do it is cloud synch. You need cloud synchronization. And so my takeaway from this whole thing is, until Apple, Google, and Microsoft figure out like...

Leo: Interoperability.

Steve: ...interoperability, do not do this. Wait for a password manager to support this. Because what we need is the same cross-platform capability that we have now with passwords. We need that with...

Leo: Yes. Portability. You need portability.

Steve: We need portability, yes.

Leo: And by the way, don't blame Apple. This is the FIDO2 spec.

Steve: Right, right. I'm not blaming Apple.

Leo: No, no, no, some people in the chatroom are. This is not Apple saying - the one thing you could blame Apple is if they decided not to make it portable. Because they could make it portable; right?

Steve: Well, we could also blame Apple for not adopting SQRL.

Leo: Well, obviously.

Steve: Which solves this problem.

Leo: They should have done that. Wow. Wow. So I wonder, though, if you're going to use the Passkeys name, which presumably FIDO trademarked...

Steve: No, it's not official. It's not an actual name.

Leo: It's not official? Oh, it's just...

Steve: No. There's nothing called Passkeys in FIDO.

Leo: Okay.

Steve: The marketing people said, oh, instead of passwords it's Passkeys.

Leo: Passkeys. But there's no requirement.

Steve: There's no proprietary ownership of that term.

Leo: There's nothing you could say that, oh, if you want - see, they could have trademarked it and said, well, if you want to use the word Passkeys trademark, you've got to have portability. They didn't do it.

Steve: Well, remember FIDO began with a dongle; right?

Leo: Yeah, yeah.

Steve: I mean, it was never meant to be like mass, the way it has become. And it's because they backed away from that and said, okay, well, we're not happy about it, but we'll let you use your phone. We don't think it's as good as a token; but we tried to do tokens, and no one bought them.

Leo: Nobody's going to buy tokens. We all have phones with strong, I mean, here's the good news, we all have phones with strong biometrics.

Steve: Yes, yes, yes.

Leo: And Secure Enclave. So they're pretty good. In fact, I would argue, if you just have a YubiKey without biometrics, they make them with biometrics, but if you have one without biometrics, this isn't tied to me. If I lost it, anybody could use it.

Steve: No. If the valet got a hold of it by mistake when he was parking your car.

Leo: Exactly.

Steve: Yup.

Leo: Whereas my phone, you know, you could get it, but you'd have to unlock it. And you can't.

Steve: Yup. And so the beauty is this leverages all of the work that Apple has done on security and biometrics and privacy. Unfortunately, they adopted FIDO, which is a separate key pair per site. And so you have to synchronize it. You've got to use the cloud. And you don't end up with one key per person, which is what SQRL is.

Leo: Yeah. How does SQRL solve this issue of unique public keys?

Steve: So I actually have in the notes, I said: "The entire SQRL idea hit me after I had visited one of Dan Bernstein's cryptography site pages." And I have the link: cr.yp.to/ecdh.html. That's elliptic curve Diffie-Hellman. So on that page Dan mentioned that to create - if you scroll down a little bit, you'll see some C code, four lines, or three lines of C code, right there, those three lines.

So he explains, he mentions that to create a private key using his Curve25519, you took any random 256 bits of entropy, turned two specific bits off and one bit on, and you had a valid private key. And from that you could derive its matching public key. What I realized that morning at IHOP when I was working on SpinRite 6.1, it just hit me like a flash of lightning. I realized that, rather than starting with a completely random 256 bits of entropy, we could instead start with a 256-bit hash of a domain name.

Leo: Oh, smart.

Steve: And turn that into a private key.

Leo: Smart.

Steve: And that was it.

Leo: Yeah, that's brilliant. So each domain has a unique private key.

Steve: It automatically gets calculated.

Leo: Now, I save that on my end; yes?

Steve: Yes.

Leo: So we could do the handshake.

Steve: Yes. Well, so we hash the domain name. But it's a keyed hash; right? Remember the HMAC is a cryptographic - so it's a keyed hash. So that key is your one thing. Your identity as Leo is the key for the hash. That means that when you hash Amazon.com, and I hash Amazon.com, with our different keys, we get different private keys. So your identity keys the hash of the domain name to create a private key which - and you don't have to store it because it creates it every time you need it. You say, what domain do you want to log onto? And it says, oh, in that case, this is your private key.

Leo: It's very simple and fast, yeah.

Steve: It's so, yes, I mean, it's just the right answer.

Leo: Right. Ah, sigh.

Steve: So I guess, if I was guilty of anything, it was of not evangelizing this thing and spending a lot of time pounding on people and being like Stina Ehrensvard.

Leo: It would have been a pretty uphill battle.

Steve: That was my worry. FIDO already was rolling and had steam. And I was, I mean, I spent seven years on this. And I dropped SpinRite 6.1. And I'm so happy I'm back to SpinRite 6.1 now. But anyway, this is the problem with FIDO2 and with Passkeys. And so my advice would be wait until a third-party independent solution exists. If you start

creating Passkeys with Apple, unless, I mean, maybe they're going to show the Passkey on a QR code. I don't think so. There's just no way. See, they like this lock-in.

Leo: Yeah. Of course. Yeah. They don't want to change a thing.

Steve: No. And so it means that when you create, when you register with an account on your iPad, Apple will synchronize it in the same way they do our messages right now. And then your phone will know the Passkey that the iPad generated on the fly and add it to your collection of Passkeys.

Leo: Wow.

Steve: But you end up with hundreds of them.

Leo: Okay, Steve. You're right. The world is wrong, dammit. But this is the world people like you have to live in, I'm sorry to say.

Steve: Oh, I love this world.

Leo: I'm just going to quickly see if I can convert that long number.

Steve: Oh, my goodness. Oh, my goodness.

Leo: Number to - install an Emacs package here. Oh, can't find it. Oh, well. Next time.

Steve: Next time.

Leo: Next time.

Steve: Perfect.

Leo: I bet Emacs would do it happily.

Steve: Wow. I want to hear that. I want to see this.

Leo: Lisp has an integer that's called bignum that is not tied to the register size. It can just go and go and go and go.

Steve: Wow.

Leo: It's one thing Lisp does very well. Mr. Gibson, you are the man, and they should have just damn well listened to you about SQRL, gosh darn it. But that's the world we live in.

Steve: Yup.

Leo: If you wish to hear this show at a 16Kb version...

Steve: Sounding like a SQRL.

Leo: Sounds like Thomas Edison on a cylinder, or you wish to read along, those are quite nice. Elaine Farris writes those out as we speak. Or the 64Kb audio. Steve's got all of that at his website, GRC.com. Now, while you're at GRC.com, first of all, set aside a few hours because it's a rabbit hole you're going to want to go down. There's all sorts of good stuff, free stuff, things like ShieldsUP!. But there's also his bread and butter, which is called SpinRite, the world's best mass storage maintenance and recovery utility. If you have an SSD or a hard drive, you need SpinRite. 6.0's the current version. Imminently, Steve will be 6.1. If you buy 6.0 now, of course you'll get a free upgrade to 6.1. You'll also be able to participate in the ongoing development, which I say that, it's pretty much done, I think; right? I mean, just a few little I's to dot.

Steve: All the hard stuff is behind us, and now it's a matter of sort of regluing the front end to the new back end. And that's what I'm in the process of doing.

Leo: Do not use wallpaper paste. That's all I'm saying. You should go to GRC.com, get SpinRite, support Steve and, by the way, get a great tool that will be very useful to you down the road. We have a show at our website, shows, 64Kb audio and video actually at TWiT.tv/sn. There's also a YouTube channel. So if you want to share it, for instance, if you heard something, and you said, you know, somebody's got to hear this, you can go to that YouTube channel and just get that little snippet.

There's also, of course, it's a podcast so you can just go to any podcast player and subscribe. Search for Security Now!, you should be able to find it. And that way you'll get it automatically every Tuesday, the minute we're done here. We record the show 1:30 Pacific, that's 4:30 Eastern, 20:30 UTC. You can watch us do it, live.twit.tv. There's audio and video streams there. People who watch live often like to chat live. We have a wonderful chatroom where they're all talking about what you're talking about.

Steve: And just to follow up, the one reason you can't use one key for all your sites is then you lose anonymity.

Leo: Oh, yes. Good point.

Steve: It would be trackable totally.

Leo: Excellent point.

Steve: From one website to the next.

Leo: Yes.

Steve: That's why they create them randomly.

Leo: Yeah. That makes perfect sense. But you solved it. You do have, though, the burden of keeping a vault with all of those numbers in it. And that's why you have cloud. But those things can be leaked without harm, right, because no one else has the private key.

Steve: Correct.

Leo: Yeah. Well, that's it, Steve. Are you watching any good TV?

Steve: You know, we're currently watching "The Good Doctor."

Leo: Yeah. That goes a long time because it was the original one, and then they made "The Good Doctor," so you've got many, many episodes, yeah.

Steve: And it's by the creator of "House," and of course "House" was a classic series.

Leo: Good, good.

Steve: So, yeah, we're watching that, and we've got a backlog of things. Everyone is saying that "The Lincoln Lawyer" on Netflix is really good.

Leo: Ah. I loved the movie.

Steve: I did, too.

Leo: Yeah. I'll have to look at that, okay.

Steve: And so apparently the series is worth doing.

Leo: Excellent. Excellent. I always ask Steve.