Security Now! #874 - 06-07-22

Passkeys, Take 2

This week on Security Now!

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.

"I don't get it... we keep changing the password and we still have a leak!"

Security News

ServiceNSW Responds

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:

> *This issue is known and does not pose a risk to customer information.* [Okay.]
>
> *The blogger has manipulated their own Digital Driver Licence (DDL) information on their local device.* [Indeed.]
>
> *No other customer data or data source has been compromised.* [Okay.]
>
> *It also does not pose any risk in regard to unauthorized access or changes to backend systems such as DRIVES.* [No one ever said it did. But it's good that it doesn't.]
>
> *Importantly, if the tampered license was scanned by police, the real time check used by NSW Police (scanning mobipol) would show the correct personal information as it calls on DRIVES.* ["DRIVES" is apparently the name of their backend database system.]
>
> *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.]
>
> *Altering the DDL is against the law.* [Oh, well then. So not unlike using a fake ID which, I would imagine is also against the law.]
>
> *The DDL has been independently assessed by cyber specialists and is more secure than the plastic card.* [Huh. Were these by any chance the same cyber specialists who came up with the system's design in the first place? And what does "more secure than the plastic card" mean? How can the security of a physical card be compared against a computer solution? Everyone always says that all software has bugs. A physical card doesn't have software bugs.]

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."*

ExpressVPN pulls the plug in India

Last Thursday, ExpressVPN announced that it would be removing all of its India-based VPN servers in response to a new cybersecurity directive issued by the Indian Computer Emergency Response Team (CERT-In).

However, that doesn't necessarily mean 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.

These 'virtual' India servers will instead be physically located in Singapore and 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 27, 2022. 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. CERT-India stated that the user data being logged will only be requested for the purposes of "cyber incident response; protective and preventive actions related to cyber incidents."

CERT-India has since clarified that this rule does not apply to corporate and enterprise VPN solutions and are 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 help 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."*

In addition, the new rules which are called Cyber Security Directions, also require firms to report incidents of security lapses such as data breaches and ransomware attacks within 6 hours of noticing them. 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 the 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."*

Unfortunately, this feels more like the future than the past. Governments are increasingly uncomfortable with the idea of having no means to surveil their citizens and others who reside within their borders. The great encryption 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 while tests are being taken.

The most recent instance of this occurred last week and this week in Syria which scheduled a series of four planned outages, lasting three and a half hours each. The first two occurred last week and the next two are set for today and for the 12th. The outages are performed via BGP by removing Syria's routing from the global Internet, thus cutting it off from the rest of the world. Wow.

Prior to Syria implementing the exam blackouts in 2016, test questions would begin appearing on social media 30-60 minutes before each exam. This allowed cheating students to circulate

correct answers and compromise the integrity of the test. 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.

The pressure on Syrian students is great since their performance on these standardizes tests largely determines what higher education options they will be able to access, which, in turn, defines 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.

The exam blackouts operate in Syria by blocking all hardwired and mobile internet access in the hours before the exams, as paper tests are printed and physically distributed across the country.

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. And academic-related internet shutdowns have been reported in India. Last year, more than 25 million people faced a mobile internet shutdown in the Indian state of Rajasthan during a local teacher eligibility exam.

"Follina" under active exploitation

Under the heading of "Well, that didn't take long" we have last week's Microsoft mess which Kevin Beaumont named "Follina". Recall that this was the ms-msdt:// protocol vulnerability that was being abused through Office — ALL versions of Office. Now we have reports of widespread and quite aggressive attempts to abuse this weakness.

So now, a most likely Chinese state-aligned threat actor has been observed attempting to exploit the Microsoft Office "Follina" vulnerability to target government entities in Europe and the U.S. The enterprise security firm Proofpoint said it blocked attempts at exploiting this remote code execution flaw, which is being tracked CVE-2022-30190 (CVSS score: 7.8). No fewer than 1,000 phishing messages containing a lure document were sent to the targets. ProofPoint said *"This campaign masqueraded as a salary increase and utilized an RTF with the exploit payload downloaded from 45.76.53.253."*

The attacking 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." This second expanded script checks for the presence of virtualization, steals information from local browsers, mail clients and file services, conducts machine recon and then zips it all for exfiltration 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's wide-ranging reconnaissance capabilities it was believed to be mounted by a nation-state level actor.

The vulnerability remains unpatched with Microsoft urging their customers to disable the protocol to prevent the attack vector. And in the absence of a security update, the great guys over at 0patch.com have released one of their unofficial micropatches to block the ongoing attacks against Windows systems.

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 government 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 first 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 under active exploitation at the time 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 was doing this. But because the exploit didn't immediately reproduce and fall at the feet of the Microsoft MSRC guy, he just blew it off saying that it was not a problem.

And a Windows Search URL schema can be abused, too

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 0-day remote code execution vulnerability which is now being used in attacks." I was referring to Microsoft's decision to ignore the early warning of this impending doom until after it had occurred.

Last week I also observed that this msdt:// protocol scheme problem 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 we're back here, today, because, sure enough, another similar feature of Windows has just surfaced. This time it uses the search-ms:// protocol URI scheme. 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 with the search-ms: protocol handler to open a Windows search window from a Word document. By "search window" we mean that an Windows Explorer search results window will open, showing a list of files to run, but that list of search results can be files sourced from a hacker-controlled remote server. Whoopsie. The result can be an extremely convincing "your software must be upgraded to proceed" attack. It's convincing because the dialog runs from Windows and appears to be coming from Windows — because it is — though it's triggered by an untrusted Word document which the user can have received through any channel, such as a spoofing eMail.

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 confirmation dialogs to allow Word to open these convincing looking search windows directly.

This might not trick listeners of this podcast. But we have made our computers so complex that most users have no idea what's going on. And legitimate software DOES often pop up, telling us that we need to upgrade or update this or that. Microsoft has incorporated warning confirmation

dialogs in an attempt to prevent the abuse of this confusion, but this most recent hack arranges to bypass those warnings.

"Creeping Featuritis" is insidious. And it appears to be unavoidable in maturing products. In the case of Windows, an incredibly complex system has been created that no one fully understands. And security is unforgiving. A single mistake is all that's required. 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 doesn'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."

Uhhhhhh. Duh... The whole point is that the abuse of these protocol handlers allows for the creation of either 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 and to only open files that come from trusted sources." ... that's exactly what users think they are doing.

"Critical UNISOC Chip Vulnerability Affects Millions of Android Smartphones"

If you were worried by headlines saying things like: "A critical chip vulnerability has affected millions of android smartphones" you probably need not worry about it.

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 is only running a tiny subsystem of the entire device.

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, we wouldn't want that.

The company in question, UNISOC, is based in Shanghai and is the world's fourth-largest mobile processor manufacturer after Mediatek, Qualcomm, and Apple. So they currently account for around 11% of all System on a Chip shipments.

This problem has been patched after having received the somewhat surprisingly high CVSS of 9.4. 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 cellular radio seemed like a big worry.

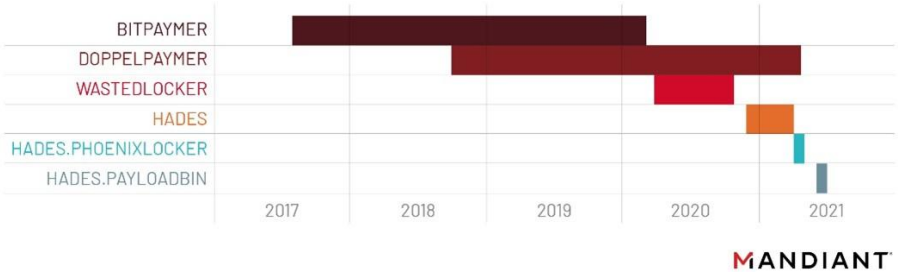
In any event, Google will be pushing an update in their June 2022 release, and I do congratulate

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 3rd parties, but that's the closed-source world we live in today.

Ransomware sanctions are causing trouble

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 through the attackers are not law abiding, their victims are. So enterprises which have been hit by ransomware are legally prohibited from making any ransom payments — through any mechanism — even if they wanted to.

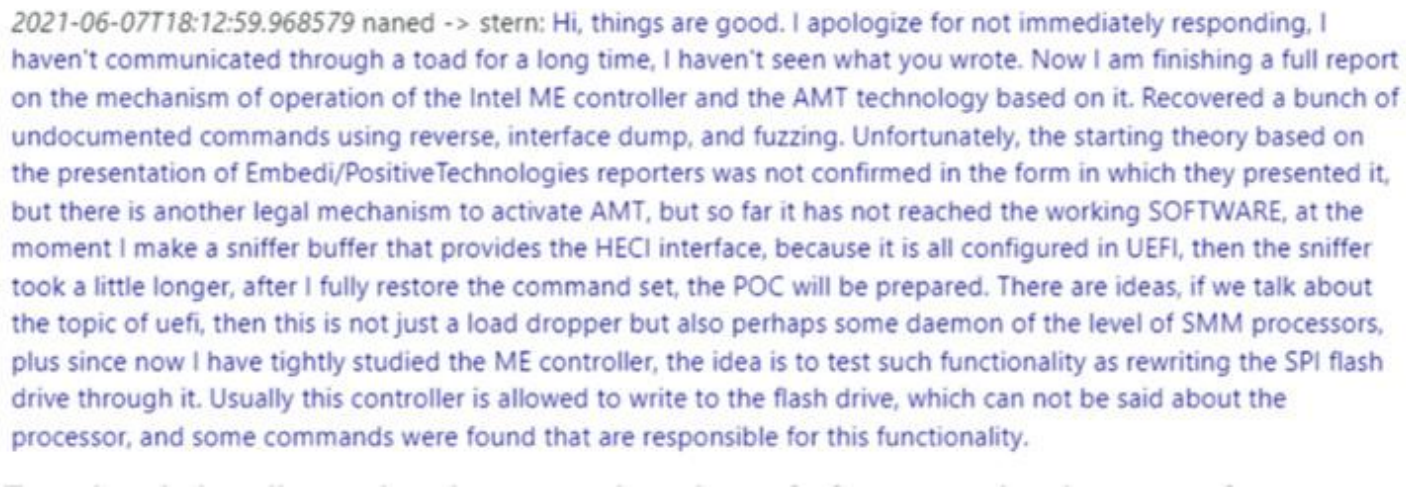
Mandiant's research paper published last Thursday was titled: *"To HADES and Back: UNC2165 Shifts to LOCKBIT to Evade Sanctions"* That headline requires a bit of explanation, which Mandiant then provides, writing:

> *The U.S. Treasury Department's Office of Foreign Assets Control (OFAC) sanctioned the entity known as Evil Corp in December 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:*
>
> 
>
>
> *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. These developments suggested that the actors faced challenges in receiving ransom payments following their ransomware's public association with Evil Corp.*
>
> *Mandiant 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 almost exclusively obtains access into victim networks via the FAKEUPDATES infection chain, tracked by Mandiant as UNC1543. Previously, we have 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.*

Mandiant's paper then delves into all of the details of the intelligence that they collected to track these activities and draw these conclusions. But I mostly just wanted to make the observation, which I thought was interesting, that just as the Conti gang apparently shutdown 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, US 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 following leaked message was posted exactly one year ago on June 7th, 2021:



The conversations among the Conti members have shed light on the syndicate's attempts to search for vulnerabilities related to ME firmware and BIOS write protection. They reverse engineered the system to locate undocumented commands and vulnerabilities in the ME interface, achieved code execution in the ME to access and rewrite the SPI flash memory, and dropped System Management Mode (SMM)-level implants, which could be leveraged to modify the OS kernel.

The leaked chats show that the work ultimately resulted in proof-of-concept (PoC) code last summer that can obtain SMM code execution by gaining control over the ME after obtaining initial access to the host through traditional vectors like phishing, malware, or a supply chain compromise.

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."

Errata

anocelot / @anocelot

> *@SGgrc: Quick #SecurityNow Errata - Grover the Muppet was Blue. Not green. You're probably thinking of Oscar the Grouch. See https://muppet.fandom.com/wiki/Grover Just trying to save you from the #MuppetMafia who will not tolerate inaccuracies. ;)*

The ServiceNSW Digital Driver's License

Several of our listeners have a ServiceNSW digital driver's license (which they assure me they haven't tampered with). They noted that a simple screen capture would fool no one because the screen incorporates a number of animated effects. There's a flower that animates, and the phone's inertial positioning is used to animate a large multi-colored flower in the background wallpaper. So that's nice. If only its developers had given as much thought to the security of the device.

Closing The Loop

Larry Wilson / @dunster96

> *I think you've undersold the benefit of triply encrypting and going from 256 bits to 768. 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.*

[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. 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 possible keys since you have all of the original number of keys when that new bit off and all of the original number of keys again when that new bit is on. So, encrypting three times with 256 bits each time would result in 2^768 possible keys, which is a ridiculously large number:

1,552,518,092,300,708,935,148,979,488,462,502,555,256,886,017,116,696,611,139,052,038,026,050,952,686,376,886,330,878,408,828,646,477,950,487,730,697,131,073,206,171,580,044,114,814,391,444,287,275,041,181,139,204,454,976,020,849,905,550,265,285,631,598,444,825,262,999,193,716,468,750,892,846,853,816,057,856.]

Passkeys, Take 2

I originally had these thoughts filed under "Miscellany" because I didn't want to make a big deal about it. But as I worked on them, they grew... even finally to the point of perhaps being significant. And no other news of the week rose to any greater significance. So I finally decided to take a second look at Passkeys in the wake of Apple's WWDC 2022 presentation, yesterday.

A bit of an "@SGGRC" tweet storm arose from our listeners following Apple's WWDC 2022 Keynote where Apple made a point of highlighting their forthcoming adoption of the revised and much more practical FIDO public key authentication system under the unofficial designation "Passkeys" — which was the title of our May 10th podcast four weeks ago. I just 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. I haven't seen that from Google yet, but if Apple does it on iOS, Android will have to follow.

One thing I didn't highlight when I first talked about this four weeks ago was the uncomfortable unanswered questions surrounding manufacturer lock-in. Apple seems quite **un**interested 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 would have access to my iMessages. But not from Windows. I've sort of worked around Apple's deliberate denial of support for cross-platform connectivity, by using iCloud for Windows, but it's still way 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 across to Windows and Android.

And that's a huge practical problem for Passkey's adoption that SQRL never had. So I want to make sure that everyone understands what the difference is, and why we're almost certainly heading for trouble which, among all of the celebration, no one seems to have picked up on 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 involves providing the remote website with only the public key of the pair. And that's the essence of both approaches.

Subsequently verifying any user's identity amounts to verifying that the user is holding the matching **private** key. To perform that verification, the website sends a unique random nonce challenge blob to the user's authenticator, which it signs using its matching private key. 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. That's the entire essence of both systems. They vary widely in the details of the way this is done, but not conceptually. On this level both systems are that simple.

Where the two systems crucially & importantly differ is how those original key pairs are created:

The FIDO2 “Passkeys” system creates the key pairs randomly whereas SQRL calculates them deterministically...

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 a per-domain public key pair. 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 the one grand master key that they share. You load that one grand master key once into each of your various authenticating devices, and that grand master key subsequently generates all of the per-domain subsidiary key pairs forever.

But, unfortunately, SQRL is not the system we're going to get, at least not yet. And that might just 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?” Awkward though a username & password are, the one thing they have going for them is that they are platform agnostic.

In the near future, when you use an iPhone, iPad or 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. And that's 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 **they** control. But is Apple going to dynamically synchronize those private keys among Android and Windows authentication devices? Yeah, when pigs fly. If they keep their keys to themselves, and if Google and Microsoft each keep the private keys their Passkeys apps generate 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 logon with the device family which originally registered at that website. What 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?

I suppose if you are 100% all-in on Apple, 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. So, as nice as Apple's Passkeys system may be, and as much as I'm a devoted iPhone and iPad user, I can't use Apple's Passkeys system until I'm sure there will be a means for also using its Passkey registrations under Windows.

And even if Apple were to provide some means for exporting a Passkey, say, as a QR code so that it can be cloned into another device, the problem then becomes keeping devices synchronized. Because the other crucial feature that our password managers provide is continuous dynamic synchronization across devices. I create a new logon at home and when I go to the office the next day that browser already knows how to login. With Apple’s Passkeys you

can have that too, but only if you use Apple products at every location.

The use of a QR code displayed on the screen can offset this inconvenience somewhat, assuming that it works the way it does with SQRL. But that means that you can never logon natively on Windows, Android or Linux. The ONLY WAY for a Windows, Android or Linux user to logon would be with their iPhone. And that might be a bit of a pinch.

Because SQRL synthesizes all key pairs from a single grand master key, there's no need for any dynamic inter-device synchronization. And the benefit of that design feature is becoming quite apparent.

So what are our takeaways?

For those of us here, in the know, we're probably better off allowing the Apple, Google and Microsoft Passkeys fanfare to drive industry-wide adoption (assuming that it does) while holding off, ourselves, until either the cross-platform Passkeys dynamic sharing question is answered — and I don't know how it can be — or waiting until the inevitable 3rd-party supplier creates a deliberately cross-platform solution. That's probably the answer.

I haven't examined FIDO2 closely enough to determine whether SQRL's domain-based deterministic keypair generating solution could be used with FIDO2's chosen encryption. The entire SQRL idea hit me after I had visited one of Dan Bernstein's cryptography site pages:

https://cr.yp.to/ecdh.html

On that page, he mentioned 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. I realized that rather than starting with a completely random 256 bits of entropy, we could instead start with a keyed 256-bit hash of a domain name and turn **that** into a private key. I was so excited by that back in 2013, that I stopped working on SpinRite v6.1 and developed that seed of an idea into a finished and working system. And along the way many other problems were solved, too.

But it's clear that the system we're going to get will be this FIDO2 / Passkeys solution. I'm sure that everyone now understands that the biggest problem with this system was the sole reason I created SQRL: No need for dynamic cross-device synchronization.

So, be careful with Passkeys adoption until the question of passkey export has been answered. And since the Passkeys solution utterly and absolutely requires dynamic cross-device and should have cross-platform synchronization, my advice would be to wait for the inevitable 3rd-party Passkeys logon manager. It seems clear that Apple, Google and Microsoft won't be providing one.