

Security Now! #870 - 05-10-22

That “Passkeys” Thing

This week on Security Now!

This week we look at a patch to Android to thwart an actively exploited vulnerability. We briefly revisit Connecticut's new privacy law and we take a quick look at the raft of recent ransomware victims. The U.S. State Department has added another ransomware group to its big bounty list and we look at what's being called the biggest cybersecurity threat facing the U.S. Meanwhile, the White House issues a memorandum about the threat from quantum computing and we have the discovery of a new and pernicious DNS vulnerability that's unlikely to be fixed in our IoT devices. And after looking at F5 Networks new and quite serious troubles, we close the loop with some listener feedback, briefly discuss the past week of Sci-Fi news, then finish by looking at the past week's most Tweeted-to-me question: “What's that passkeys thing that Apple, Google and Microsoft are adopting?”

When you don't know what that code does, but you assume it must be important... so you just leave it alone...



Security News

Google updates Android to patch an actively exploited vulnerability

Google released monthly security patches for Android with fixes for 37 flaws across various components, and one of them is a fix for an actively exploited Linux kernel vulnerability that came to light earlier this year. That vulnerability, CVE-2021-22600, with a CVSS of 7.8 was ranked as “High” severity because it could be exploited by a local user to escalate privileges or deny service.

The flaw was a double-free vulnerability residing in the Packet network protocol implementation in the Linux kernel that could cause memory corruption, potentially leading to denial-of-service or execution of arbitrary code. And it wasn’t just Android that was vulnerable. Patches were released by various Linux distros, including Debian, Red Hat, SUSE, and Ubuntu back in December 2021 and January 2022. It’s unclear why Google didn’t patch this one sooner.

However, now Google says: “There are indications that CVE-2021-22600 may be under limited, targeted exploitation.” And last month the vulnerability has been added to CISA’s Known Exploited Vulnerabilities Catalog due to evidence of its active exploitation. Google also patched three other bugs in the kernel as well as 18 high-severity and one critical-severity flaw in MediaTek and Qualcomm components.

Connecticut’s recently passed data privacy bill became law last Wednesday.

I was incorrect in stating last week that Connecticut's governor, Ned Lamont, would need to sign the recently passed legislation for it to become law. It turns out that the state has a rule that bills which have passed in the state assembly become law automatically five days after they are passed during a legislative session. So, consequently, Connecticut now joins California, Virginia, Colorado and Utah to become the fifth state to create its own privacy law in lieu of federal action. And there has been specific reaffirmation that once the law has ramped up to full strength, the Global Privacy Control signals being sent by browsers **must** be honored, without exception and without any further “are you sure”-style prompting by anyone with whom Connecticut residents interact online.

Ransomware victim snapshot

- Trinidad’s largest supermarket chain was crippled by cyberattack.
- The German library service is struggling to recover from a ransomware attack.
- A major German wind farm operator confirms cybersecurity incident.
- Austin Peay State University in the US was hit with ransomware.
- The ADA, the American Dental Association confirmed cyberattack after ransomware group claimed credit.
- Coca-Cola is investigating claims of a hack after a ransomware group was offering their stolen data for sale.
- Conti ransomware has deeply crippled the systems of the electricity manager in a Costa Rican town and newly elected president of Costa Rica has declared a state of emergency.
- The Agricultural equipment maker AGCO has reported a ransomware attack.
- A cyberattack has taken down the network of the State Bar of Georgia.
- And classes have resumed at Michigan community college after a ransomware attack, and

classes at Kellogg Community College will be resuming Wednesday after two days of outages caused by a ransomware attack. In Battle Creek, Michigan, nearly 7,000 students were told last Monday, May 2, that ransomware had crippled its systems the previous Friday, April 29th. The school was forced to shut down its main campus in Battle Creek as well as branches in Coldwater, Albion and Hastings.

As I've said, I don't want to make this the weekly ransomware news podcast, but neither do I want to avoid noting the severity of these ongoing attacks. So I'll just give a shortened snapshot every so often.

US State Department offering \$10 million reward for information about Conti members

US State Department offering \$10 million reward for information about Conti members
The U.S. State Department has begun offering \$10 million rewards for any information leading to the identification or location of people connected to the Conti ransomware gang. And an additional \$5 million reward is also being offered for any information that leads to the arrest or conviction of a Conti member. So \$15 million to anyone who can turn-in a member of the Conti gang. You've got to think that this would make anyone associated with Conti quite uncomfortable. Their MUST be others outside of the gang to whom Conti members have bragged.

So, in a statement on Friday, State Department spokesman Ned Price told us something we already know, that Conti has been behind hundreds of ransomware attacks over the last several years. He said: "The FBI estimates that as of January 2022, there had been over 1,000 victims of attacks associated with Conti ransomware with victim payouts exceeding \$150,000,000, making the Conti Ransomware variant the costliest strain of ransomware ever documented."

The memo also notes that the group has recently claimed credit for that wide-ranging ransomware attack that targeted the government of Costa Rica as it was transitioning to a new president. The attack crippled the country's customs and taxes platforms alongside several other government agencies. And, as I noted before, the attack also brought down one Costa Rican town's energy supplier.

Conti also attacked Ireland's Health Service Executive a year ago in May 2021, which resulted in weeks of disruption at the country's hospitals. Ireland refused to pay the \$20 million ransom and now estimates it may end up spending \$100 million recovering from the attack.

The group similarly crippled dozens of hospitals in New Zealand and the group has made a point of targeting U.S. healthcare and first responder networks, including law enforcement agencies, emergency medical services, 9-1-1 dispatch centers, and municipalities within the last year, according to the FBI.

The group has suffered a number of internal breaches over the years, the most notable of which occurred a few months ago in February after it expressed public support for Russia's invasion of Ukraine. Within a few days of the message, the gang's internal Jabber/XMPP server – which carried their private messaging channel – was hacked, and two years of the group's chat logs appeared on a new Twitter handle called @ContiLeaks. The leaks revealed the group's inner

workings and illustrated the way they chose their targets. However, those leaks did nothing to slow the group down. Last Wednesday, they added New York-based architecture firm EYP to their list of victims.

So, Conti joins the ranks of those carrying a serious bounty on their heads. Last November, the U.S. State Department offered a \$10 million reward for any information that would lead to the identification and/or arrest of members of the Darkside ransomware group with a similar bounty on the operators behind REvil (Sodinokibi).

The worst threat the US faces...

Is from the Winnti Group, also known as APT 41 — Advanced Persistent Threat. Just how advanced and persistent are these threat actors?

Researchers with Cybereason recently briefed the FBI and the DoJ about Operation “CuckooBees” — a funny name but not a funny operation. This is an ongoing espionage effort by Chinese state-sponsored hackers with the charter to steal proprietary information from dozens of global defense, energy, biotech, aerospace and pharmaceutical companies. The specific individual organizations affected were not named in Cybereason’s report but they allegedly include some of the largest companies in North America, Europe and Asia. And the threat actor behind it all is the prolific Winnti Group, also known as APT 41.

Cybereason CEO Lior Div said that the most alarming aspect of the investigation into Operation CuckooBees was the evasive and sophisticated measures used to hide inside the networks of dozens of the largest global manufacturing companies in North America, Europe and Asia, dating as far back as 2019. Lior said: “The group operates like a guided missile and once it locks onto its target, it attacks and doesn’t stop until it steals a company’s crown jewels. Winnti pilfered thousands of gigabytes of data and to add insult to injury also made off with proprietary info on business units, customer and partner data, employee emails and other personal information for use in blackmail or extortions schemes at a time of their choosing.”

Cybereason said that throughout its 12-month investigation, it found the intruders took troves of intellectual property and sensitive proprietary data, including formulas, source code, R&D documents and blueprints, as well as diagrams of fighter jets, helicopters, missiles and more. The attackers also gained information that could be leveraged for use in future related cyberattacks, like details about a company’s business units, network architecture, user accounts and credentials, employee emails, and customer data.

And of greatest concern, according to Cybereason's CEO, was that the companies had no clue they were breached. In a pair of detailed reports, Cybereason attributes the attacks to Winnti based on an analysis of the digital artifacts the group left behind after its intrusions. Several other cybersecurity companies have also been tracking Winnti since it first emerged twelve years ago in 2010, and researchers have observed that the hackers are clearly operating on behalf of Chinese state interests while specializing in cyber-espionage and intellectual property theft.

The group used a previously unknown and undocumented malware strain called DEPLOYLOG, as well as new versions of malware like Spyder Loader, PRIVATELOG, and WINNKIT. The malware

included digitally signed, kernel-level rootkits as well as an elaborate multi-stage infection chain that enabled the operation to remain undetected. The group also managed to abuse the Windows Common Log File System (CLFS) mechanism, which allowed the intruders to “conceal their payloads and evade detection by traditional security products.” CLFS is a logging framework that was first introduced by Microsoft in Windows Server 2003 R2 and has been included in all later Windows operating systems.

Cybereason explained that “The attackers implemented a delicate ‘house of cards’ approach, meaning that each component depends on the others to execute properly, making it very difficult to analyze each component separately.” And unsurprisingly, “Operation CuckooBees” generally took advantage of existing weaknesses including unpatched systems, insufficient network segmentation, unmanaged assets, forgotten accounts and lack of multi-factor authentications.

Cybereason said that the attackers generally obtained their initial foothold in the organizations through vulnerabilities in Enterprise Resource Planning (ERP) platforms. Last month, FBI’s director Chris Wray told 60 Minutes that the “biggest” threat American law enforcement officials face is from Chinese hackers stealing proprietary information. He said that the FBI opens a new China counterintelligence investigation about every 12 hours. Think about that. Wray said that “They are targeting our innovation, our trade secrets, our intellectual property on a scale that’s unprecedented in history. They have a bigger hacking program than that of every other major nation combined. They have stolen more of Americans’ personal and corporate data than every nation combined. It affects everything from agriculture to aviation to high tech to healthcare, pretty much every sector of our economy. Anything that makes an industry tick, they target.”

The White House and Quantum Computers

This one made me shake my head. The headline was “White House wants nation to prepare for cryptography-breaking quantum computers”. To give everyone a sense for this, the reporting on this in “The Record” started out:

A memorandum issued Wednesday by President Joe Biden orders federal agencies to ramp up preparations for a day when quantum computers are capable of breaking the public-key cryptography currently used to secure digital systems around the world.

The document, National Security Memorandum 10 (NSM-10), calls for “a whole-of-government and whole-of-society strategy” for quantum information science (QIS), including “the security enhancements provided by quantum-resistant cryptography.”

Uh huh. Why don't we just cure cancer? Oh Wait! That was what Biden was going to do while he was Barack's VP. How'd that work out?

But seriously... “order the federal government to ramp up preparations for a day when quantum computers are capable of breaking public-key cryptography, which, by the way, doesn't yet exist?” The federal government is apparently unable to update its own software when being handed patches to do so. Someone... somewhere... says... uuhhhh, not today. We haven't yet secured our computers for technology we already have against hackers we already have.

So... I don't know... how about having the White House issue a memorandum ordering the various agencies of the federal government to please just reboot their computers? How would that be? We could actually get more security, right now, today.

And, sure. Quantum computing technology shows promise. But let's remember that it's been showing promise for quite some time. We've had nascent quantum computing technology since around the late 1970's, so for more than four decades. It's intriguing and interesting and it's been moving forward gradually, like most really big problems do -- think fusion power. And the federal government should **absolutely** be funding ongoing research in universities to allow our nation's brightest young minds to continue pushing this frontier forward. There's clearly something tantalizingly possible there. And I agree that we should not forget that we have adversaries. China is also hard at work on this problem. So if we patch and reboot our computers, we might be able to keep them from stealing it from us once we figure it out.

The ongoing threat from predictable DNS queries

As I have often said, I'm stunned by the elegance and fundamental simplicity of the Internet's design. It was so beautifully conceived. Yet it is not without some blemishes. One of the original sins of the Internet's early design was —and still is— a lack of entropy in some fields which are critically important. This entropy is crucially necessary for robust attack resistance. In defense of the Internet's early designers, the last thing they were thinking, while they were trying to get this whole thing to work, was about active and aggressive adversaries. That just wasn't on the map at all. So they designed and built beautiful technology which has withstood decades of explosive growth being put to use in applications they could never have, and did never, imagine.

But there are a few problems. For example, the endpoints of a TCP connection are identified by an IP address and a port number. And the progress of the connection's data flow is tracked by a 32-bit byte sequence number. In the early days of this podcast we examined how the predictability of the sequence numbers being issued by TCP/IP software stacks could be weaponized and used by attackers to splice into an existing TCP connection. Since nothing identified the other endpoint other than its source IP address and source port, TCP packets carrying spoofed source IP and port, and guessing a sequence number that would be accepted by the receiving endpoint could — and did back then — succeed in injecting malicious traffic into established TCP connections.

Another quite famous lack of entropy may, and often does, exist in DNS queries. Being UDP, the spoofing task is much easier. If a DNS client emits a query to a DNS server having a knowable IP address, and if the 16-bit source port of its query and the 16-bit transaction ID is predictable, it is not difficult for an adversary to jam a bogus DNS reply into that client which looks identical to the reply it's expecting to receive from the authentic DNS server. And as we know, this form of DNS cache poisoning spoofing attack can have devastating consequences. The IP being looked up will be altered and traffic will be silently redirected.

It was the realization of that, which hit Dan Kaminsky back in 2008, when nearly all DNS servers in the world were vulnerable to this form of attack — because their queries had very low effective entropy — that caused the world to secretly prepare and then synchronize a simultaneous global update of all affected DNS servers.

But we missed something. Something that today afflicts many IoT devices, such as routers by Linksys, Netgear and OpenWRT, as well as Linux distributions like Embedded Gentoo. This exposes many millions of IoT devices, once again, to this once-solved security threat.

What we missed, or at least weren't worrying about 14 years ago was that it's not only DNS servers and our desktop operating systems which emit DNS queries — they were all fixed. But many other low-end IoT-ish devices do too. Maybe it wasn't a big problem or concern back then. But the crucial fact is, the lesson of the need to deeply randomize source port and transaction query IDs was not learned well enough.

Nozomi Networks Labs discovered a vulnerability, now being tracked as CVE-2022-30295, which affects the DNS implementation of all versions of uClibc and uClibc-ng which is a very popular C standard library used in IoT products. The flaw, which was found and fixed in all major DNS servers back in 2008 is the predictability of transaction IDs in the DNS requests generated by the library:

```
1300
1307         /* first time? pick starting server etc */
1308         if (local_ns_num < 0) {
1309             local_id = last_id;
1310         /*TODO: implement /etc/resolv.conf's "options rotate"
1311         (a.k.a. RES_ROTATE bit in _res.options)
1312         local_ns_num = 0;
1313         if (_res.options & RES_ROTATE) */
1314             local_ns_num = last_ns_num;
1315             retries_left = __nameservers * __resolv_attempts;
1316         }
1317         retries_left--;
1318         if (local_ns_num >= __nameservers)
1319             local_ns_num = 0;
1320         local_id++;
1321         local_id &= 0xffff;
1322         /* write new values back while still under lock */
1323         last_id = local_id;
1324         last_ns_num = local_ns_num;
1325         /* struct copy */
1326         /* can't just take a pointer, __nameserver[x]
1327         * is not safe to use outside of locks */
1328         sa = __nameserver[local_ns_num];
1329         __UCLIBC_MUTEX_UNLOCK(__resolv_lock);
1330
1331         memset(packet, 0, PACKETSZ);
1332         memset(&h, 0, sizeof(h));
1333
1334         /* encode header */
1335         h.id = local_id;
1336         h.qdcount = 1;
1337         h.rd = 1;
1338         DPRINTF("encoding header\n", h.rd);
1339         i = __encode_header(&h, packet, PACKETSZ);
1340         if (i < 0)
1341             goto fail;
1342
```

The relevant code used to form a DNS query is shown above where we see "local_id++;" which increments the transaction ID, then "local_id &= 0xffff;" which masks and retains only its lower 16 bits, effectively causing the incremented value to wrap around from 65,535 back to 0. The problem here, of course, is that the use of this widely used library produces entirely predictable sequentially incrementing DNS transaction IDs, the presence of which in our DNS servers panicked the entire networking world 14 years ago.

While doing a bit of background research into this uClibc, I found that the original pre-forked uClibc's last update was May15th of 2012. So exactly 10 years ago this coming Sunday. Because this library had become unsupported, it was forked to create uClibc-ng (presumably "ng" stance

for next generation): <https://uclibc-ng.org/> The good news is, it's being actively maintained. But under its home page's History section it explains:

uClibc-ng is a spin-off of uClibc (from Erik Andersen) from <http://www.uclibc.org>. Our main goal is to provide regularly [misspelled] a stable and tested release to make embedded system developers happy.

The first release 1.0.0 with the code name Leffe Blonde was made while visiting Fosdem 2015. It was prepared in a hotel room in Brussels on 1 February 2015. All releases are prepared while drinking a pair of Belgian beer, since then.

[Because, you know, that's what you want in the replacement for the Standard C library that everyone's embedded IoT devices are using, is for it to be maintained by a drunken Belgian.]

The idea to fork uClibc started in July 2014 and was discussed on the Buildroot and OpenWRT mailinglists.

So, we've identified a well-understood flaw that has been present in embedded Linux-based IoT devices for the past decade or so. Apparently, no one thought to look before now. Now the world knows. I'm pretty sure that the OpenWRT folks will get on this to fix it. The fix, after all, is trivial. The transaction ID sequence simply needs to be unpredictable. An embedded device without a good source of local entropy could come as it's starting up by using things like high resolution packet timings to obtain an unpredictable seed. Send out some pings to some known static IPs and time their return at the device's clock speed. That will generate a value that's unknowable by any external attackers. I'd use that value to key a simple symmetric cipher which encrypts a sequential counter. That will produce an unpredictable sequence.

What we don't know, is everywhere this embedded library has been used in embedded Linux systems, whether Netgear and Linksys will care to update, and most importantly, where and how this flaw will surface in the future. But the bad guys will make it their business to know because that knowledge is valuable to them. And this is the legacy we're building which most worries me: The growing number of well known problems that are accruing, mostly under the radar, and which are not being diligently fixed. These things don't go away on their own. What's happening over time — mark my words — is that one of the favorite vehicles of fiction writers, which is that anything can be hacked, offensive as I have always found that idea, is gradually becoming true. And this is another perfect example of the way it's going to happen; little by little and bit by bit.

F5 Networks Remote RCE warning and exploitation

Here comes another example of a serious vulnerability that's far more high profile and should get the attention of anyone using F5 Networks' so-called BIG-IP equipment. But both we and the bad guys already know that patching is badly broken and that there will be F5 BIG-IP equipment online which remains unpatched. Remember that list of ransomware victims from earlier? This is exactly where attacks such as those, begin.

Last Wednesday, May 4th, F5, a major cloud security and application delivery network provider, released patches to repair 43 bugs spanning its products. Of the 43 issues addressed, one is rated Critical, 17 are rated High, 24 are rated Medium, and one is rated low in severity.

But that Critical one — oh baby! — it carries a CVSS of 9.8 which arises from a lack of an authentication check which will allow attackers to take control of an affected system. As we're seeing more often, now, the flaw took only a few days to reverse engineer and a working proof of concept has been made public. So the use of terms such as "might" "may" or "could" in F5's bureaucratically worded disclosure should be replaced with "will" "did" and "have." They wrote:

"This vulnerability may allow an unauthenticated attacker with network access to the BIG-IP system through the management port and/or self IP addresses to execute arbitrary system commands, create or delete files, or disable services."

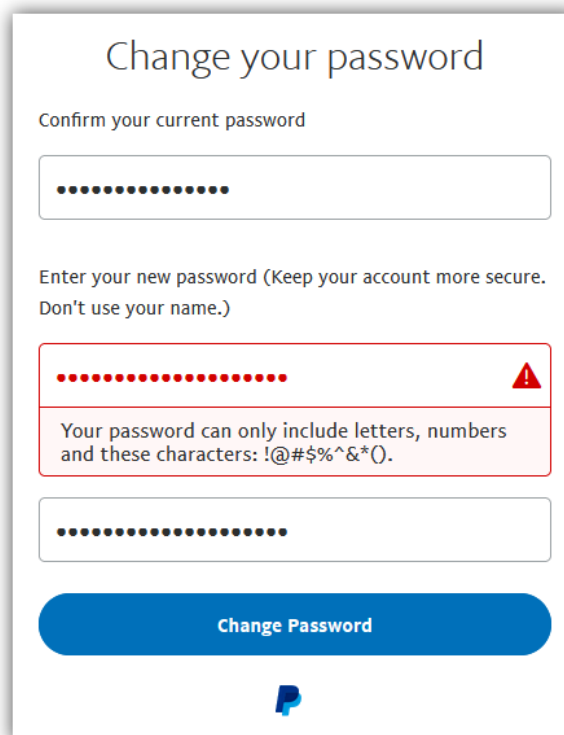
And this security vulnerability appears to be longstanding, since it affects all six most recent major version releases, v11 through v16. It doesn't appear that they'll be patching the oldest major versions 11 and 12 since patches for the iControl REST authentication bypass flaw have been released for versions 17.0.0, 16.1.2.2, 15.1.5.1, 14.1.4.6, and 13.1.5, leaving 12 and 11 unpatched but vulnerable.

We can expect CISA to soon add an alert and a mandatory update for this to their growing catalog of Known Exploited Vulnerabilities. And in fact, they just added five more to that catalog, three for which patches were made available in 2014, one in 2019 and another last year. Yet all five are now under active exploitation even eight years after being patched.

Closing The Loop

Lets Burninate / @letsburninate

Hi Steve, long time listener. I just wanted to share this image when I tried to change my Paypal password and was shocked by this error message for obvious reasons.



The image shows a 'Change your password' form from PayPal. It has two password input fields. The first field is for 'Confirm your current password'. The second field is for 'Enter your new password (Keep your account more secure. Don't use your name.)'. The second field has a red border and a red warning triangle icon, indicating an error. Below the second field, a message reads: 'Your password can only include letters, numbers and these characters: !@#%&^*()'. At the bottom of the form is a blue 'Change Password' button and the PayPal logo.

awk / @adrianteri

RE: Feedback on #869's Moxie's KnockKnock. One can minimize their exposure of things on the internet without punching holes on their #NAT routers. On #833 you mentioned a couple of more options to "overlay networks" other than #tailscale that may enable one even on a home/residential IP to be able to interconnect their devices across the internet even with their IPs changing.

Bob Grant / @bggbp_gopackgo

Hi Steve, thank you for your recommendation of McCollum, can you suggest a good starting point for reading him?

I think I'd start with his Gibraltar trilogy. Michael writes old school hard sci-fi where the joy is in his plot devices. I love being surprised and I have always found Michael's work to be full of delightful moments.

And speaking of Sci-Fi...

Sci-Fi

"Strange New Worlds" — 100% excited about this one. It's a good sign when IMDB's rating is rising over time. It's now up to 8.3/10 and, I think, very well deserved.

"Picard Season 2" rhymes nicely with **"I hate Q"** — God I hate "Q". I always have and it turns out that I still do. He's just an annoying fly in the ointment, and I suppose that I'm not a huge fan of John De Lancie, the actor. This second season is a bit of a mixed bag. There are too many dumb scenes which are drawn out to much, for me to be able to love it like, so far, I love Strange New Worlds without reservation.

"Avatar: The Way of Water" — I stumbled over its trailer when I was at IMDB looking up "Strange New Worlds" current rating. It looks like a seamless continuation of Avatar. And it appears that he has Avatar 3, 4 and 5 in the pipe with #3 already in post production and #4 filming. (The Terminator / Aliens / The Abyss / True Lies / Titanic / Dark Angel / Avatar)

That “Passkeys” Thing

Arstechnica: “Apple, Google, and Microsoft want to kill the password with “Passkey” standard. Instead of a password, devices could look for your phone over Bluetooth.”

BleepingComputer: “Microsoft, Apple, and Google to support FIDO passwordless logins.”

The Record: “Google, Apple and Microsoft to expand support for passwordless sign-in standard.”

The Hacker News: “Google to Add Passwordless Authentication Support to Android and Chrome.”

9To5Mac: “Passkeys in iCloud Keychain could make automatic website login even easier.”

9To5Google: “Google previews how password-killing ‘passkeys’ will work on Android, Chrome.”

All of these headlines popped up last Thursday, May 5th, which was not only Cinco de Mayo, but also World Password Day. And the news of and questions about this new “Passkeys” was also the most tweeted-to-me news item of the past week, with many of our listeners wanting to know what it was and what I thought.

Having spent, as our listeners know, 7 years of my life designing, implementing, demonstrating and proving a complete working solution to this need, I have a good grasp of the problem domain. So I dug into this “Passkeys” news by going to the source, as I always endeavor to, first reading the FIDO Alliance’s May 5th press release which is titled: *“Apple, Google and Microsoft Commit to Expanded Support for FIDO Standard to Accelerate Availability of Passwordless Sign-Ins.”* This was the press release that everyone else was quoting. It appeared that whoever wrote it was being paid by the word, since it went on and on to make sure that its reader would come away knowing that all pre-FIDO systems were bad and FIDO is the cure. At this point it appears that regardless of whether or not it turns out to be the cure, it will at least be the next thing we try. And I’m in the same boat as all of our listeners. We’re all avid users and consumers of the Internet. So we’re all hoping that the industry knows what it’s doing. But, that press release wasn’t going to get the job done. Fortunately, it linked to the description of a FIDO Alliance White Paper titled: “Multi-Device FIDO Credentials.”

That description reads:

The FIDO standards, together with their companion WebAuthn specification, are on the cusp of an important new development: evolutionary changes to the standards proposed by the FIDO Alliance and the W3C WebAuthn community aim to markedly improve the usability and deployability of FIDO-based authentication mechanisms. As a result, FIDO-based secure authentication technology will, for the first time, be able to replace passwords as the dominant form of authentication on the Internet. [What a concept!]

In this paper, we explain how FIDO and WebAuthn standards previously enabled low-cost

*deployments of authentication mechanisms with very high assurance levels. While this has proved an attractive alternative to traditional smart card authentication, and even opened the door to high-assurance authentication in the consumer space, we have not attained large-scale adoption of FIDO-based authentication in the consumer space. We explain how the introduction of **multi-device FIDO credentials** will enable FIDO technology to supplant passwords for many consumer use cases as they make the FIDO credentials available to users whenever they need them—even if they replace their device.*

<https://media.fidoalliance.org/wp-content/uploads/2022/03/How-FIDO-Addresses-a-Full-Range-of-Use-Cases-March24.pdf>

Okay. Now, obviously, this descriptive overview doesn't yet tell us what we want to know. So I dug into the white paper. We get the "Executive Summary", followed by "A Brief History of Online Authentication". Then a section titled "FIDO: Starting from the Top" followed by "WebAuthn Level 3: Bringing up the Bottom." We're now at the bottom of page four and we begin the frame the problem as follows:

FIDO-based solutions can also increase the security of consumer two-factor authentication by providing phishing resistance, regardless of whether those use cases care about hardware-based sign-in credentials or not. However, we have observed limited adoption in this latter category, especially in the consumer space, because of the perceived inconvenience of physical security keys (buying, registering, carrying, recovering), and the challenges consumers face with platform authenticators as a second factor (for example, having to re-enroll each new device; no easy ways to recover from lost or stolen devices). While these drawbacks can make FIDO-based solutions (whether based on physical security keys or platform authenticators) a tricky proposition for users already accustomed to two-factor authentication, they present an even higher barrier to adoption for users who don't (want to) use two-factor authentication at all, and are stuck with passwords.

And so, finally, we get down to it. The White Paper explains:

The FIDO Alliance and the W3C WebAuthn working group are proposing to address these gaps in a new version ("Level 3") of the WebAuthn specification. Two proposed advances in particular bear mentioning:

- 1.** Using your phone as a roaming authenticator: a smartphone is something that end-users typically already have. Virtually all consumer-space two-factor authentication mechanisms today already make use of the user's smartphone. The problem is that they do this in a phishable manner: You may inadvertently enter an OTP on a phisher's site, or you may approve a login prompt on your smartphone not realizing that your browser is pointed at the phishing site and not the intended destination. The proposed additions to the FIDO/WebAuthn specs define a protocol that uses Bluetooth to communicate between the user's phone (which becomes the FIDO authenticator) and the device from which the user is trying to authenticate. Bluetooth requires physical proximity, which means that we now have a phishing-resistant way to leverage the user's phone during authentication. With this addition to the FIDO/WebAuthn standards, two-factor deployments that currently use the user's phone as a second factor will be able to upgrade to a higher security level (phishing resistance) without the need for the

user to carry a specialized piece of authentication hardware (security keys).

2. Multi-device FIDO credentials: We expect that FIDO authenticator vendors (in particular those of authenticators built into OS platforms) will adapt their authenticator implementations such that a FIDO credential can survive device loss. In other words, if the user had set up a number of FIDO credentials for different relying parties on their phone, and then got a new phone, that user should be able to expect that all their FIDO credentials will be available on their new phone. This means that users don't need passwords anymore: As they move from device to device, their FIDO credentials are already there, ready to be used for phishing-resistant authentication.

Note that this change is not a change in the standard—it is a change we expect authenticator vendors to make in their authenticator implementation. There are proposed changes to the WebAuthn and FIDO specifications that would enable a better user experience around FIDO credentials (including multi-device FIDO credentials), in particular for those relying parties that need to serve password-based and FIDO-based users at the same time. The user experience around FIDO credentials would be very similar to that of using a password manager that helps the user sign in, but the level of security is better than even traditional two-factor authentication—all without requiring any additional steps or devices during authentication: Typically, all a user would have to do on a new device to sign into a relying party is to pass the built-in biometric challenge² on the device from which they're trying to sign in (as we'll explore further below).

For these multi-device FIDO credentials, it is the OS platform's responsibility to ensure that the credentials are available where the user needs them. (Note that some companies are calling FIDO credentials "Passkeys" in their product implementations, in particular when those FIDO credentials may be multi-device credentials.) Just like password managers do with passwords, the underlying OS platform will "sync" the cryptographic keys that belong to a FIDO credential from device to device. This means that the security and availability of a user's synced credential depends on the security of the underlying OS platform's (Google's, Apple's, Microsoft's, etc.) authentication mechanism for their online accounts, and on the security method for reinstating access when all (old) devices were lost. While this may not always meet the bar for use cases that require physical key level security, it is a huge improvement in security compared to passwords: each of the referenced platforms apply sophisticated risk analysis, and employ implicit or explicit second factors in authentication, thus giving two-factor-like protections to many of their users. This shift from letting every service fend for themselves with their own password-based authentication system, to relying on the higher security of the platforms' authentication mechanisms, is how we can meaningfully reduce the internet's over-reliance on passwords at a massive scale.

[In other words, they're saying that we will rely upon the user authenticating to their own device — smartphone or desktop — with biometrics or whatever, rather than authenticating to each remote site individually. That sounds familiar.]

Syncing FIDO credentials' cryptographic keys between devices may not always be possible, for example if the user is using a new device from a different vendor, which doesn't sync with the user's other existing devices. In such cases, the existence of the above-mentioned standardized Bluetooth protocol enables a convenient and secure alternative: if the FIDO

credential isn't readily available on the device from which the user is trying to authenticate, the user will likely have a device (e.g., phone) nearby that does have the credential. The user will then be able to use their existing device to facilitate authentication from their new device.

So, it appears that what this press release and these so-call "passkeys" (which the White Paper explains don't actually have anything to do with FIDO) are, is just the introduction of cloud syncing among devices to facilitate the transport of one's collection of FIDO credentials from one device to the next. The other piece is that the FIDO Alliance appears to have formally given up on the idea that we're all going to go out and purchase a hardware FIDO token when we all already own a smartphone that can serve the same purpose. The use of a possibly available Bluetooth link allows one's smartphone to be used to authenticate to a website on a desktop that does not contain a FIDO authenticator with one's credentials. (Just for clarity, that's what SQRL provides for with a QR code and the smartphone's camera.)

And, yes, speaking of SQRL, I know that the heads of everyone who understands SQRL is exploding right now because FIDO still falls very far short of providing the complete solution that SQRL offers. But having moved from simple usernames and passwords to password managers and multi-factor authentication and then to OAuth 3rd-party authentication, we're now going to get FIDO, though it will apparently be popularly called "Passkeys." From the samples I've seen online, it appears that it will still be necessary to first identify oneself to the web site being authenticated to. So FIDO with "Passkeys" replaces the password but unfortunately, not the username. So it will continue to be somewhat more cumbersome in that way, too.

The way FIDO's crypto works, is that it randomly synthesizes a public and private keypair for each and every website the user wishes to authenticate with, and it gives that site the public key to retain while the FIDO authenticator stores the matching private key for each future use. So, it's this collection of individual private authentication keys — now being called "Passkeys" — that Apple, Google and Microsoft will be obtaining and synchronizing in the cloud for their users. This provides for same-platform cross-device FIDO credential synchronization which is crucial for FIDO since each new website authentication creates another public/private keypair. And it provides for credential recovery in the event of device's loss, which is certainly needed.

As we know, I went a different way with SQRL. SQRL uses a single master key which can be printed and stored safely. From that one key, it deterministically synthesizes unique per-site public and private keypairs based upon the website's domain name and, like FIDO, it gives each website the public key to use for future authentication. But unlike FIDO, there is no growing collection of randomly synthesized per-site private keys that need to be retained and cloud-synchronized among devices. So there's no need to back up a large collection of private keys to the cloud, or anywhere. The only thing a SQRL user ever needs for their identity to be secure and fully recoverable for all websites is one piece of paper.

Overall, it appears that this was all mostly a World Password Day-timed press event without much technology to back it up. We're not getting SQRL, we're getting FIDO. And that means we need cloud synchronized "Passkeys" to make FIDO's use practical.

