Well-Known URIs

Description: This week we catch up with Chrome 86's handful of security-related improvements. We touch on several recent ransomware events and on the consequences of not logging free WiFi users in France. We look at the results of an amazing bit of hacking of Apple, give an update on the enduring Zerologon threat, introduce the revenge of DNT with legislation-enhanced GPC, and describe another renewed attack on undecryptable E2EE now by seven countries. Then, following a bit of SpinRite and GRC forum news, we're going to add the concept of IANA-registered well-known URIs to our bag-of-tricks knowledgebase.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-788.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-788-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with lots to talk about. Chrome gets 86'd. Make good money hacking for Apple. And why three French cafs just got busted for not logging WiFi access. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 788, recorded Tuesday, October 13th, 2020: Well-Known URIs.

It's time for Security Now!, the show where we protect you and your loved ones, online and off, with this guy right here, Mr. Security & Privacy himself, Steve Gibson. Steve, I had to laugh. After 5G, the most repeated phrase in the Apple event today was "security and privacy, security and privacy, security and privacy." Clearly they see that as a real selling point. And I guess they're probably right.

Steve Gibson: And Leo, it's funny you should mention those two topics because we have some things to say today about questions and solutions and issues in security and privacy.

Leo: Oh, how nice. How clever.

Steve: As a consequence of two stories that we'll talk about, Chrome 86, which is not something that you want to 86 because it's good. Those who spend any time in the food services industry may recognize that "86" is a special number, largely within that community. And also, what was it, it's another feature that we'll be talking about. Both stories tie into something that we've touched on a couple times. But it turns out, and I was unaware of how much it had grown, and so we need to add this to our bag-of-tricks

knowledge base. There is a formal registered, as in IANA registered, set of well-known URIs which the more I look at this, the more bullish I become. So that's the title for today's podcast, well-known URIs, Security Now! #788.

But we're first going to catch up with Chrome 86's handful of security-related improvements. We touch on several recent ransomware events, just briefly but to kind of keep our toe in the water. I had a couple people write and say, you know, Steve, you seemed self-conscious about talking about ransomware. But that's what the podcast is about. Okay, but yes, not to excess. Also some bizarre consequences of not logging free WiFi users in France. We'll look at the results of an amazing bit of hacking of Apple, their cloud-based services, and what resulted from some good guys taking a close look at that. Also an update on the enduring Zerologon threat.

We're going to introduce the revenge of DNT with new legislation-enhanced GPC. Also another renewed attack on undecryptable end-to-end encryption, now coming from seven countries. Two more have jumped in. Also I have a little bit of update on SpinRite, something on the GRC forums news, and then we're going to take a look at this really interesting addition to the Web API IANA-registered well-known URIs, which I think everyone is going to find interesting. Oh, and we do have sort of a mind-blowing Picture of the Week. Just for those who didn't already think that hard drive technology had already totally jumped the shark, boy, this picture will convince you.

Leo: Yeah, yeah. All right, Steve. You want to take a look at this picture now?

Steve: This is just mind-boggling. Everyone's wondering how, well, people who have been in the industry for a long time certainly, how it can possibly be that our hard drives are storing the kind of density that they are.

Leo: I bought a 16TB hard drive the other day. 16 terabytes.

Steve: Yeah, 16 trillion 8-bit bytes.

Leo: It's amazing.

Steve: Unbelievable. And they're not expensive.

Leo: No.

Steve: I mean, it's just - it's crazy. So get a load of this. This is a picture of the actuator system which Western Digital is using, and presumably others also. We're all sort of familiar with the idea of a voice coil mechanism that rotates the arms out onto the disk surface where at the end is a head mounted on the end. And you could have like stacks of platters that have these arms on both sides. So that was then. The problem is there's a lot of inertia, and all the heads have to be positioned in exactly the same place. So what this thing does, where we are today is a compound three-stage actuator where you still have what is now considered gross positioning by the voice coil, which sort of swings the whole head tower in and out on the drive to get the heads...

Leo: Is that how they used to work? Or, I mean, was that the whole thing in the old days?

Steve: Yeah, that was all that we used to have in the old days. Now, because tracks have become so close together, and because vibration - we've talked about, like, remember the guy screaming at his hard drives and suddenly throughput dropped? It's because just the acoustic energy of the sound waves hitting the side of the drive was enough to push the heads off-track. I mean, it's like they could be used to detect earthquakes in China. I mean, they are so sensitive now.

Leo: Wow. Wow.

Steve: So what they've done is they've now got two additional actuating systems. Out near the heads is the so-called milliactuator, which servos like the last half inch or so, and then at the head is a microactuator which does the third degree of positioning. So imagine the technology that goes into this thing. It's like, oh, yeah, okay, we've got triple actuators. Yeah, but you have to actually run them. I mean, you have to figure out whose job it is.

Leo: It's mindboggling.

Steve: At which stage. And the fact that you've got the head tower, which they all share, but notice that the milliactuator and the microactuator are per head, which means you can now be individually servoing all 18 heads on this nine-disk platter independently in order to keep them all on track. It's just, to me...

Leo: Amazing.

Steve: And it's cheap. It's not a fortune to buy one of these things. Yet we just sort of take it for granted, oh, yeah, I got 18TB. That ought to last a while. It's like, oh, my god.

Leo: It's the story of technologies. Although I know you and I, well, I don't know about you, but I thought the year 2000, 20 years ago, I thought, well, there's only a few years left in hard drives, and we'll be all solid state or something else by 2010.

Steve: Oh, we're going to have - remember we talked about it, those really cool wafers that Star Trek had. We're just going to have wafers. Just stick the wafer in the slot.

Leo: But they've managed to really get a lot more life out of these things.

Steve: Unbelievable.

Leo: You know we just did the Apple iPhone event. That processor is 5 nanometers, and one processor has 11.8 billion transistors. It's got six cores, four GPU cores. It's the size of your pinky nail in your phone, 12 billion transistors. It's mindboggling.

Steve: And runs on a battery.

Leo: Yeah, and runs on a battery, yeah.

Steve: It used to be that the air conditioning was larger than the computer that you had.

Leo: Right, right, right.

Steve: Unbelievable.

Leo: It's great. I mean, we live in amazing times. We really do.

Steve: Yes, children, this podcast is being operated by old people who remember the abacus and actually had to use one once.

Leo: I had a caller on the radio show on Sunday who's 91. Or maybe Saturday. Who's 91. And I was saying, wow, you've seen a lot of changes. I was talking about the phone company. He said, "I remember before there were phones." You know, nobody had a phone. I said, "Oh, my god, let alone a phone in your pocket that can contact anyone in the world for free."

Steve: Well, and we've talked about this before. You would pick up the phone and see if anybody else was on the line.

Leo: Yeah, party line, yeah.

Steve: Because it was often shared. And then you'd flash the hook switch a few times to get the attention of an operator somewhere. Oh.

Leo: Operator. Operator. And she would physically connect a cable that joined you and the person you were calling via a single wire.

Steve: Yeah.

Leo: Wow.

Steve: So going from there to a thousand miles an hour, we have last week's release of Chrome 86. The main theme for 84 was updates to its UI. We got a bunch of user interface improvements there. Then 85 came along, which was mostly about performance and some new API stuff. Last week's Chrome 86, which was dropped into the stable channel on Tuesday - I think they do that on purpose just to get us because I'm always then a week behind - focuses primarily upon enhanced security features, so

apropos for the podcast, and even some additional - although wait till you hear about this API enhancement. I'm not too sure. It's definitely in the "what could possibly go wrong" category because it allows access to the entire file system of its hosting OS.

Leo: Ooh.

Steve: But more on that in a minute. So as usual, your particular Chrome instance may need, as mine did, at both locations, last night and this morning, a bit of encouragement to get it to jump from 85 to 86. I've been using it daily. I kind of run both Chrome and Firefox for different purposes. Even though Chrome was a week old, 86 was a week old, and it had been out there, it still wasn't until I went to Help > About that it came up and said, oh, yeah, I've got some updates to do. It had been running 85 point something or other. And now it's at 86, but only when I went to look. So if anyone's interested, I'm sure it would finally get around to updating itself, but I like to have the current one, as our listeners probably do.

Okay. So last December we were at 79. And the desktop versions of Chrome acquired what they called at the time the "password checkup feature," which would scan the user's saved-in-Chrome and synced passwords for any collisions with known leaked passwords from other sites' data breaches. Chrome renamed that feature "Safety Check" in May, and now this very useful feature has been added with 86 to the mobile editions of Chrome for both Android and iOS. So that's new. Also with 86 we now have built-in support for a particular type of well-known URI. And it was the much more broad and general use case for this feature, which has just been added to 86, that was the first impetus for this week's podcast topic. So we'll get back to that when we wrap up the podcast here in another hour and a half or so.

Chrome users on iOS are also getting a new touch-to-fill feature which Android users have had since July. It's essentially biometric authentication for password-filling on iOS. Chrome detects the site that the user is navigating around and will then automatically prompt the user to autofill passwords if it has credentials saved for the site. Since Chrome will only trigger on an exact domain name match, this has the benefit which Google has been touting, which is of course common for all password managers that match on domain names, of helping to avoid phishing and visually close spoofed site names. And since iOS now offers strong biometric support, Chrome 86 will support that when it's available in the hardware, requiring a quick biometric reauthentication before it will autofill.

And of course we've all had this with our industrial strength add-on password managers for some time, so none of this is news for most of us. And our password managers of course are not only cross-platform, but also cross-browser make, which Chrome of course is not. It's going to keep it in the family. My LastPass works on Chrome and on Firefox, Mozilla, and Safari and so forth. So still for me that's the better solution. But there are many people who have still not made that leap to a third-party password manager. And if we haven't got them by now, it seems unlikely that we will. So certainly building much stronger security into the base browser, which everyone is always going to be using, seems like a useful compromise.

Also in May of this year, with Chrome 83, our desktop Chromes acquired the Enhanced Safe Browsing feature to provide more phishing and malware detection. With 86 last week, the mobile platforms also have both acquired that protection, as well. I was surprised to learn that earlier versions of Chrome, prior to 86, weren't already warning their users when submitting insecure form data. That seems like a no-brainer which we of course on this podcast have been talking about for like a decade. But at least Chrome

86 is now offering that protection, as well. It gives you a clear warning if you are on a secure page whose Submit URL is not secure.

Again, we've been talking about this forever. But it's listed in new features for Chrome 86, so okay. And it will continue with its warning and blocking when downloading insecure assets from secure pages. In 86, executable and archives are blocked by default, while Chrome shows warnings for Office-related document downloads. Again, you can bypass the warning, but it's like, make sure this is what you want.

And we've noted that browser-based File Transfer Protocol - remember FTP - is not long for this world. Its death will be occurring in stages, and that begins now. With last week's Chrome 86, FTP will still be enabled by default for most - and this is weird, but bear with me - stable channel users. It will be disabled for the canary and beta pre-release channels. I said "most stable users" because, oddly, FTP will also be experimentally disabled for 1% of stable users. Okay. So I guess they just don't want to break anything that they may not be aware of, although you'd think that their telemetry would have long ago told them, I mean, I'm sure it did, how many people are actually clicking on FTP links these days.

So if you find that it's disabled for you, and you need it, after you get through asking yourself why - and really do take a moment - 86 will allow you to reenable it from the command line using the --enable-ftp command line flag, should you need to. When we get to Chrome 87, the disabled percentage will be increased to a coin toss with 50% of Chrome's 87 users discovering, well, hopefully not discovering to their dismay that their favorite FTP resources can no longer be downloaded.

And finally, with Chrome 88, FTP will be completely sunsetted. I don't even know if you'll still be able to turn it on. Probably you could do a command line override, if you really, really definitely had to have it for some reason. But they're trying to say no. And we talked about this before. FTP's not a bad protocol, it's just weird to have it in a browser because the number of times you actually need to be looking at FTP servers and downloading things from them seems like it would be a low Venn diagram collision with browser uses. There are plenty of FTP clients around.

And as for that "what could possibly go wrong" new feature, get a load of this one. Chrome 86's new Native File System API, as they're calling it, is activated by default in Chrome 86. Google boasts that it will enable developers to build powerful web apps that interact with files on the user's local device. Now, naturally, you wouldn't want any random malvertising advertisement to have full and unfettered API access to your entire machine. So yes, the new API is blocked and hidden behind a permission prompt to prevent websites from accessing any local files without your authorization.

Leo: Good.

Steve: Yeah, huh? I know. Again, what could possibly go wrong?

Leo: Jesus.

Steve: After a user grants the browser access, this API allows a website to behave like - and this is Google, like, saying "Won't this be great?" Like, okay, like a locally installed app reading, saving, and interacting with files and folders on a user's device. Google expects this new API to be used to power a broad range of bracing interactive web apps -

yeah, brace yourself - such as IDEs, photo and video editors, text editors, and more. There's no end to what the bad guys are going to think of, I'm sure.

Leo: Right.

Steve: So again, you just know.

Leo: It's like ActiveX. It's basically letting the browser run an arbitrary app on your system. With full access.

Steve: Yes. Yes. And we've got WebAssembly now in order to do things fast, and very much like down actually talking to the processor. And you just know that someone is going to want to automate that pesky permission prompt; right?

Leo: Yeah, yeah.

Steve: I mean, who really wants to have to constantly be giving permission? That's just annoying.

Leo: I presume they're not going to let you bypass it. That would be problematic.

Steve: This podcast is never going to end, Leo. They just keep setting us up. So on the ransomware front, we do have confirmation that Carnival Corporation, the largest cruise line operator, has confirmed in last week's FCC filing that the personal information of customers, employees, and ship crews, including their passports, was stolen during an August ransomware attack. They employ more than 150,000 people from roughly 150 countries. And they of course cater to, or at least they once did, over 13 million guests each year.

Although Carnival hasn't disclosed anything about the attack, the cybersecurity intelligence firm Bad Packets discovered that Carnival had multiple potential points of initial entry and compromise which a ransomware attacker might use to get in, specifically multiple Citrix Application Delivery Controller (ADC) devices and Palo Alto Networks firewalls. The Citrix ADC was found to be vulnerable - or ADCs, multiple - was found to be vulnerable to a CVE from 2019, 19781. And of course that firmware was updated in January, but not by Carnival. And the Palo Alto Networks firewalls which we were talking about recently having a 2020 CVE, number 2021 problem, it was patched in June of this year; but, again, Carnival didn't get the memo.

And, you know, to cut them a little slack, it's difficult to imagine any industry that was probably harder hit by COVID-19 than cruise lines. You're not on any recently, Leo, and we know you guys like those.

Leo: Oh, man, if only. But, yeah, it's going to be years before I'd feel safe on one of those.

Steve: Yeah. In any event, both these vulnerabilities can be used by ransomware gangs as stepping stones to breach a corporate network, allowing them to then move laterally, collecting credentials needed to take over admin accounts and up to and including, as we've talked about, getting into the Windows domain controller.

Also a new trend has emerged which is a little chilling. In hindsight, a not unexpected development in the ransomware scene. The major ransomware network operators are beginning to purchase access to rich corporate networks, much like Carnival, from independent so-called "access providers." In other words, a new layer of specialization is emerging as the ransomware cybercrime methodology continues to mature. Accenture's Cyber Threat Intelligence (CTI) team has released new research on emerging cybersecurity trends. This includes the results of their investigation into the nature of relationships between ransomware operators and exploit sellers.

In their piece titled "Shady Deals: The Destructive Relationship Between Network Access Sellers and Ransomware Groups," they explain. They said: "Ransomware groups are taking advantage of opportunities to purchase network access on dark web forums to quickly compromise networks across a variety of industries and unleash their disabling malware. Network access sellers' expertise lies in the ability to gain corporate and government network access, which they then sell to other cybercrime groups for a handsome profit. These cybercrime groups can use purchased network access to slash the typical difficult requirement of gaining initial access, establishing persistence, and then moving laterally within a network.

"Network access sellers typically develop an initial network vulnerability and infiltrate the victim network to gain complete corporate network access. Once that access is gained, the network access sellers sell it on dark web forums, usually for anywhere between $300 and $10,000, depending upon the size and revenue of the victim. The majority of network access offerings are advertised on underground forums with some or all of the following information," they write: "Generalized victim industry information, for example, private corporation, medical institution, government agency, education, et cetera; country the victim operates in; type of access for sale, for example, a VPN, Citrix, or RDP; the number of machines on the network; and additional company information, for example, the number of employees and revenue and so forth. The amount of information provided can occasionally lead to the identification of the victim.

"Accenture CTI assesses that the network access market has been driven by the increased diversity of ways that data can be monetized. Previously, cybercriminals wishing to make a profit on underground forums primarily targeted financial data due to its ease of monetization." I mean, we've talked about this through the years; right? Like selling credit card information. You get so many cents per credit card number as a function of how old it was, the spread of expiration dates and so forth.

"However, the Nikolay threat group, also known as Fxmsp, popularized selling network accesses beginning two years ago, in 2018, by proving there was a large demand for their service and that regular sales could be highly profitable. Although financial data remains central to underground economies, sensitive Personally Identifiable Information (PII) and company data, or the promise of access to this data, is profitable because this data can be further monetized through direct sale or by holding it ransom.

"Now, since the start of 2020 and the emergence of the now-popular 'ransomware with data theft and extortion' tactics, ransomware gangs have successfully utilized dark web platforms to outsource complicated aspects of a network compromise. A successful ransomware attack hinges on the development and maintenance of stable network access which comes with a higher risk of detection and requires time and effort." So why do it themselves? Let's just buy some.

"Access sellers fill this niche market for ransomware groups. As of September 2020," they said, "we actively track more than 25 persistent network access sellers, as well as the occasional one-off seller, with more entering the scene weekly. Network access sellers operate on the same forums as actors associated with the ransomware gangs Maze, Lockbit, Avaddon, Exorcist, NetWalker, Sodinokibi, and others."

They said: "We assess with high confidence that this ecosystem will continue to thrive so long as reputable, invite-only dark web forums provide the platform on which network access sellers and ransomware gangs can securely exchange goods and services." So, yeah. What we're seeing is the emergence now of specialization. The ransomware market has become so profitable that it will support horizontal integration.

In other words, it's not necessary for the extortionists to be vertically integrated and doing everything themselves. They can afford to outsource the front end work of obtaining access to corporate networks, and they can afford to pay a pretty penny for that access. That allows hackers with network penetration skills to focus only on getting in, without needing to have the skill set to monetize the access that they obtain, because now there's an eager and active bidding market for that access on the dark web. Oh, Leo. Yeah. We got plenty more podcasts where this one came from.

Also just another note, another victim. The largest software company in North America that none of us have ever heard of, Tyler Technologies.

Leo: Eh?

Steve: It services the - I know - public sector. It's got over 1.2 billion in annual revenue, and 5,500 employees. Wednesday, September 23rd, Tyler was hit with a cyberattack by the RansomExx, that's E-X-X, ransomware operators. That's the same team that was behind the recent attacks on Konica Minolta and IPG Photonics. Tyler immediately disconnected portions of their network to prevent the ransomware's spread and to limit their many clients' exposure. CIO Matt Bieri emailed clients, writing: "Early this morning we became aware that an unauthorized intruder had disrupted access to some of our internal systems. Upon discovery and out of an abundance of caution, we shut down points of access to external systems and immediately began investigating and remediating the problem."

So they did succeed in containing and preventing its spread into their clients' networks. Tyler said that they were severely impacted and expected it would take 30 days to recover operations fully. So Tyler paid the ransom to recover their encrypted data, though they're not saying how much they paid. RansomExx is one of the groups known to exfiltrate data before encrypting it, and then threaten to release the stolen data unless the victim pays the ransom. Since many school districts, court systems, and local and state governments in the U.S. are Tyler Technology's customers, the risk of public disclosure of sensitive information is significant in this case.

So this concern may have been a significant factor in the decision to pay the ransom. Again, we don't know how much, but they did say, yup, we paid because we need to get back up and going. And even if we could recover from backups, we can't risk the data that was exfiltrated going public. And I won't enumerate them, but a bunch of school districts have also recently, in the past few weeks, been hit by ransomware, causing disruption to tens of thousands and several hundred schools being taken offline as a consequence.

So this is just kind of nutty. But laws.

Leo: That's all you have to say.

Steve: Week before last at least five - I guess in something of a sweep of some kind - five bar and caf managers in the French city of Grenoble were arrested and taken into custody.

Leo: Oh, I saw this. This is hysterical.

Steve: Yeah. It's nuts. And we'll talk about how actually more nutty it is now than it was a month ago. Anyway, they were arrested and taken into custody for running open WiFi networks at their establishments and not keeping logs of previously connected users.

Leo: Is that a French law? That's a weird law.

Steve: It is a weird law. Now, the reason is that the law says Internet Service Providers. And, okay, well, I guess if you're running open WiFi, then technically you're an ISP. Although, I mean, it just seems like a misreading of the intention behind the whole thing. They were held and questioned. It's like, what? Like, are our croissants fresh? Yes. Do we log the identity of people who use our free WiFi? What? No. Nobody does that. Apparently this is French law. Oh, and this is - get this, Leo - 14 years old. So 2006. The law is numbered 2006-64. They risk a year in prison, a personal fine of up to 75,000 euro, and a business fine of up to 375,000 euro. Which is obviously targeted at ISP companies.

Leo: Yeah, not a bar, yeah.

Steve: Yeah, not some caf that's serving espresso.

Leo: A lot of espressos.

Steve: Yeah. So obviously it makes far more sense for an actual ISP than an open WiFi access point.

Leo: Right. And that's probably what the law was intended for, I would bet. I understand why they might - I don't think it's a good thing, but I can understand why a government might want to track Internet users.

Steve: Yes. And again, the other thing that occurred to me is that, once upon a time, before we had MAC address randomization, which we now have, you could log the MAC address of the device that was hooking to your access point.

Leo: Good point; right.

Steve: And so as we know, 24 bits gives you the manufacturer of the device. The lower 24 bits gives you a serial number maintained by that manufacturer. So it would be possible, once upon a time, to have used such MAC address logs to identify a specific laptop user and to, like, you know, follow them around where they went or to know if they were, essentially, know who they were.

Leo: To prosecute them. Yeah. I mean, if they were doing child porn or something on the caf's access point, you could track that down.

Steve: Exactly. Exactly.

Leo: But nowadays you're pretty anonymous there; aren't you?

Steve: Well, what Apple announced, to our delight, with iOS 14 was finally true, full, WiFi MAC anonymization. It used to be that, until you were associated with an access point, the iPhone would just give out random MAC addresses because who cared. And when you were associated it would then use the device's physical MAC address. And that was always a concern for privacy advocates. So now here we have a perfect instance of where we have a collision of the privacy interests of the user which Apple is enforcing now much more strongly with iOS 14, and the privacy-busting interests of law enforcement for the sake of presumably protecting their citizenry because now iOS 14 devices, they're not trackable, even if you were to log. So that makes the logs even more useless.

I'm not sure where laptops are in the MAC address randomization land, but one can imagine having had it proven to work by iOS 14, and Apple being as cautious as they were, it wouldn't be long before you start seeing laptops doing the same thing, if some of them aren't already.

Leo: Also, I mean, it's easy to spoof a MAC address. I would imagine, if you're going to do bad things on a public network, you're going to have some sec ops involved, I would think. Maybe not, I don't know.

Steve: Well, now, and what's interesting is that MAC addresses are the things that aren't tunneled in VPNs. So somebody using a VPN would still be exposing their device's MAC address unless, as you said, they went to some measures to deliberately spoof it. And also one could imagine that arranging to do that would not be difficult. Because it was like no, you know, there are - and we've talked about this before. There are like, in some corporate settings there are MAC address filters which enable access. But they're bad for security because anybody can see the MAC addresses that are flying around in the air and just use the same one.

Leo: And honestly, I'm sure this law was intended, as you think, for ISPs, where they know, you know, you pay them. They know your street address. If there's a bad activity, you can subpoena the ISP, I've done this, and get the, okay, well, who was using 70.32.3.4 at 10:00 p.m. Sunday night, and get the address. And that's, by the way, that happens with subpoenas all the time in the United States. I understand that, yeah. But a bar, come on. Especially a French bar. What are you talking about?

Steve: Okay. So what do you get when a team of five talented security researchers settle down during the global pandemic to poke at Apple's online services for three months, during July, August, and September? What you get is some significantly more secure services as a result of their discoveries and responsible disclosures of 55 vulnerabilities, 11 of which were critical. And what they got for 28 of the vulnerabilities that Apple has been able to process so far is a total bug bounty payout of more than a quarter million dollars - 288,500, to be precise.

Sam Curry, the lead researcher, blogged. He said: "When we first started this project we had no idea we'd spend a little over three months working towards it completion. This was originally meant to be a side project that we'd work on every once in a while. But with all the extra free time during the pandemic, we each ended up putting in a few hundred hours.

"In addition to the 11 critical flaws, 29 were high severity, 13 medium, and two low severity," he wrote. They could have allowed an attacker to "fully compromise both customer and employee applications, launch a worm capable of automatically taking over a victim's iCloud account, retrieve source code from internal Apple projects, fully compromise an industrial control warehouse software used by Apple, and take over the sessions of Apple employees with the capability of accessing management tools and sensitive resources." To which I say I'm glad these guys are on our side.

The flaws meant a bad actor could easily hijack a user's iCloud account and steal all the photos, calendar information, videos, and documents, in addition to forwarding the same exploit to all their contacts. To Apple's credit, and likely to their shock and horror, once the flaws were responsibly disclosed, Apple patched the flaws within one to two business days, and in the case of a few of them, four to six hours. And I especially love Sam's description of how this happened. And please take heed, all of you other big companies, because it's Apple who is now so much richer for having dangled and paid those bounties.

Here's how this all began. Sam wrote: "While scrolling through Twitter sometime around July, I noticed a blog post being shared where a researcher was awarded $100,000 from Apple for discovering an authentication bypass that allowed them to arbitrarily access any Apple customer account. This was surprising to me, as I previously understood that Apple's bug bounty program only awarded security vulnerabilities affecting their physical products and did not pay out for issues affecting their web assets."

And, oh, Sam's byline in his blog posting is "Web Application Security Researcher." So if Apple was only paying for flaws found in iOS and their various iDevices, then that would not be the territory of a web application security researcher such as Sam. But then he learned, oh, no, this was a web fault in the authentication bypass that this researcher found, for which Apple paid $100,000.

So the blog, it was a Twitter posting, "Zero-Day Sign-in with Apple, Bounty $100,000," on May 30th, 2020. So that caught Sam's attention. His blog posting continues: "After finishing the article, I did a quick Google search and found their program page where it detailed that Apple was willing to pay for vulnerabilities 'with significant impact to users' regardless of whether or not the asset was explicitly listed as in scope. This caught my attention as an interesting opportunity to investigate a new program which appeared to have a wide scope and fun functionality.

"At the time, I had never worked on the Apple bug bounty program. So I really didn't have any idea what to expect, but decided, why not try my luck and see what I could find? In order to make the project more fun, I sent a few messages to hackers I'd worked with in the past" - four of them - "and asked if they'd like to work together on the

program. Even though there was no guarantee regarding payouts, nor an understanding of how the program worked, everyone said yes, and we began hacking on Apple."

So as we know, bad guys are highly motivated by their own self interest to find high-value flaws in other peoples' work. But it's not only the bad guys who are good hackers. It's the bad guys who are typically the most motivated. Everybody needs to put bread on the table, good guys and bad guys. And much as good guys might want to spend their valuable time fixing other peoples' stuff, they cannot typically afford to. This is why generous bug bounty programs work. It was that story of the $100,000 windfall that caught Sam's attention. And unless it had, we would not be recounting this story, and all of those previously undiscovered problems would have remained undiscovered. Some would have been eventually found, probably. The question is by whom?

So again, this just really says that companies with profiles like Apple's who have a big online presence ought to also have a big online bug bounty. And they ought to be advertising it. I mean, if they want their stuff secured, they don't want bad buys to find them and sell them to Zerodium. They want good guys to find them and sell them to themselves, essentially, sell them back to the company in return for a bounty. So big props to Apple. This is the way it should work. And it just seems so clear that offering bounties is the way you get good talented white hat hackers to take a look at your stuff. We've already seen, you know, programmers, I mean, I'm well aware of it, programmers cannot see their own bugs. I look at code I have written, and it looks perfect.

Leo: Oh, yeah, yeah.

Steve: It's just like, I look at it, and I know there's a problem. I can't see it. And that's why we have debuggers. I'll be, like, single-stepping through the debugger. That's good, okay. That's good, okay. That's good, okay. And I look at the next one, it's like, oh. And then it's like, yup. I meant "jump if not zero," rather than "jump if zero."

Leo: Yeah. There's no one who's ever coded who hasn't had that experience of just staring at your code, saying I don't see what's wrong. I don't see. I can't see anything wrong with it. That's just, you know, that's normal, yeah.

Steve: And a bad guy who wants it to be wrong, rather than our own egos that inherently doesn't want it to be wrong...

Leo: Good point. Good point. That's a very excellent point, yeah. We have an investment in it being right. That's right.

Steve: Yes, yes.

Leo: Wow.

Steve: I've told the story before about an employee of mine, Jim, who was very bright. He told everybody he was a Gates kid. It's like, okay, fine. And unfortunately he got a little too much attention from his mother, I think. But anyway, I worked with him debugging, I think it was SpinRite 3.1 back in the day. And he would guess what the bug was. And without sufficient information. But he would just, you know, he wanted to be

right. That was his thing. And I would say, "Well, Jim, okay, maybe," I said. "But now there's a problem." And he said, "What do you mean?" I said, "Well, now you have an ego stake in the outcome. But it's just a bug. It's just software. It doesn't have any involvement in this from an emotional standpoint. Now you do."

Leo: Very good point. Thank you, Sensei.

Steve: It's the value of just being able to say, "I don't know. I have no" - I say it all the time. "I have no idea. But I'm going to go find out." And that's the best place to start from.

Leo: Love it, yup.

Steve: So the exploit, Leo, that is going to go down in history, or the vulnerability and exploit, this is going to be one: Zerologon. Last week the Microsoft Security Intelligence Group sent out a tweet: "We're seeing more activity leveraging the CVE-2020-1472 exploit." And they said, "(Zerologon). A new campaign shrewdly poses as software updates that connect to known" - and then they give the name of it - "CHIMBORAZO (TA505) C2" - that's command and control - "infrastructure. The fake updates lead to UAC bypass" - User Account Control bypass - "and use of wscript.exe to run malicious scripts." So again, Zerologon.

So first we had Microsoft's update last week. What we know is that after the initial flurry of immediate action which was leveraging those proofs of concept that got posted, it's now been taken up by the mainstream serious threat actors. It's been incorporated into the campaigns of several well-known ransomware extortionists. And being a fundamental flaw in a core security protocol - remember, this is not an implementation error, this was a protocol error - and given the demonstrated lack of patching vigilance that we keep seeing, it will likely go down in history as one of the more devastating Windows vulnerabilities. And there's more than a little competition in the Windows world for that role.

And speaking of Zerologon, then we have the unfortunate timing of all this, since the FBI and CISA, the cybersecurity arm of the Department of Homeland Security, last week said that they've detected hackers exploiting the Zerologon vulnerability against state and local governments where in some cases the attacks are being used to breach the networks used to support our elections. It doesn't look like it's a deliberate attack on our elections. But unfortunately, Zerologon vulnerabilities exist throughout our government.

And I just wanted to make an aside, also. I recently noted that Florida's early vote counting machines are already up, running, and humming along, since early voting this year has, as we know, blown through all past records. I was gratified to hear mentioned in the reporting of this that none of the machines counting the votes are on the Internet. That was just mentioned in passing. But what that says is that this notion of the danger that we're facing with anything connected to our networks is now like it's oozed out into the public conscience. I mean, everybody knows that this is a problem.

So apparently the machines are not divulging their vote totals. They will not until the polls have closed, in this case in Florida. And then the accrued counts will be taken from them, those will be reset, and then the machines will continue counting additional votes that arrive based on whatever criteria there is. So the good news is these things are not plugged into any network. They are all freestanding. So yay for that.

And FBI and DHS last week said: "This recent malicious activity has often, but not exclusively, been directed at federal and state, local, tribal, and territorial" - for which there's an abbreviation, SLTT (state, local, tribal, and territorial) - "government networks." They said: "Although it does not appear these targets are being selected because of their proximity to elections information, there may be some risk to elections information housed on government networks. CISA is aware of some instances where this activity resulted in unauthorized access to elections support systems; however, CISA has no evidence to date that the integrity of elections data has been compromised. There are steps that election officials, their supporting SLTT IT staff, and vendors can take to help defend against this malicious cyber activity."

So yeah, as I said, the timing of Zerologon is unfortunate because it's still fresh enough that, despite the fact that DHS issued the commandment a few weeks ago that all systems, all government systems shall be patched by - and it was midnight of the Monday before the podcast a couple weeks ago. It still looks like that commandment wasn't taken, didn't reach everybody that it was supposed to. So anyway, yeah. Stop using Windows for anything that's mission critical. Find the most security hardened Unix or Linux around and use that, if possible.

Leo: Wow.

Steve: Wow. And what has been observed is that to gain their initial access - because remember, Zerologon is not an initial access vulnerability, it's like made to order once you get in for lateral movement, and then allows you to end up completely taking over the network. But somehow you've got to first get in. Turns out that hackers are exploiting vulnerabilities, known vulnerabilities, in every case long since patched, in firewalls and VPNs and other products from companies including Juniper, Pulse Secure, Citrix, and Palo Alto Networks. All of the vulnerabilities, Zerologon included, again, as I said, have already received patches. It's just I don't know how long it's going to take for this notion that anything that is networked needs to somehow have a means of getting itself updated. That's just becoming more and more obvious and crucial.

Okay. The revenge of DNT, which of course stood for Do Not Track. We now have something emerging called GPC with legislation. GPC stands for Global Privacy Control, making it easy for consumers to exercise their privacy rights. So there is GlobalPrivacyControl.org, which is the home for this. And this has just been announced. Their press release explains: "With the introduction of privacy regulations such as California's Consumer Privacy Act" - that's the CCPA - "and the General Data Protection Regulation" - the infamous GDPR - "consumers have more rights to limit the sale and sharing of their personal data than ever before. CCPA in particular gives California residents a legal right to opt out of the sale of their data and requires businesses to respect user preferences through a signal from their web browser communicating the consumer's request to opt out."

They said: "While this is great progress, it doesn't amount to much if it's hard for people to take advantage of their new rights. Today, there's no defined or accepted technical standard for how such a web browser signal would work. Without that, users don't have an easy way to express their preferences. Indeed, in his recent testimony before the U.S. Senate, California Attorney General Xavier Becerra explained: 'One provision of our regulations intended to facilitate the submission of a request to opt-out of sale by requiring businesses to comply when a consumer has enabled a global privacy control at the device or browser level, which should be less time-consuming and burdensome.' He said: 'I urge the technology community to develop consumer-friendly controls to make exercise of the right to opt out of the sale of information meaningful and frictionless.'"

So anyway, we have that now. This effort, initially spearheaded by Georgetown Law and Wesleyan University, now also includes The New York Times; The Washington Post; the Financial Times; Automattic, which is to say WordPress.com & Tumblr; Glitch; DuckDuckGo; Brave; Mozilla; Disconnect; Abine; Digital Content Next; Consumer Reports; and the EFF. "In the initial experimental phase, individuals can download browsers and extensions from Abine, Brave, Disconnect, DuckDuckGo, and the EFF in order to communicate their 'do not sell or share' preference to participating publishers. Additionally, we are committed," they wrote, "to developing GPC into an open standard" - and actually that work is well along the way, as I'll explain in a second - "that many other organizations will support and are in the process of identifying the best venue for this proposal."

They said: "We look forward to working with Attorney General Becerra to make GPC legally binding under CCPA. At the same time, we are exploring GPC's applicability and functionality with regard to other similar laws worldwide, such as the GDPR. We are excited about the prospect of empowering people with an easy-to-use tool to exercise their privacy rights."

So I dug into this a little bit further and found the spec. The spec is formal and clear. So, for example, for this GPC header, the spec reads: "A user agent MUST NOT" - all caps, and this is in formal RPC syntax or semantics. A user agent like a browser must not "generate a Sec-GPC header field if the user's Global Privacy Control preference is not enabled. A user agent MUST generate a Sec-GPC header field with a field-value that is exactly the numeric character '1' if the user's Global Privacy Control preference is set." And then in the show notes and in the spec, they have an example, which is just an http query: GET space and then the URL that you're getting. Then the host header is :example.com. And the Sec-GPC header is "1."

So they said: "The Sec-GPC is deliberately defined without an extension mechanism. Experience with previous similar headers shows that people tend to rely on string equality instead of parsing the value when testing for their presence, especially when extensions do not yet exist. Such checks would of course fail in the presence of extension content, which would in turn render the mechanism moot. Should extensions prove necessary to this standard, they will need to be implemented through other headers, which may in time supersede this one.

"The Sec-GPC signal sent MUST reflect the user's preference, not the choice of some vendor, institution, site, or network-imposed mechanism outside the user's control. The basic principle is that a Sec-GPC preference expression is only transmitted if it reflects a deliberate choice by the user. What constitutes a deliberate choice may differ between regional regulations. For example, regulations in one jurisdiction may consider the use of a privacy-focused browser to imply a Sec-GPC preference, such as under the CCPA Final Statement of Reasons, Appendix E #73, which reads: 'The consumer exercises their choice'" - by the way, this is the California Consumer Protection law. "'The consumer exercises their choice by affirmatively choosing the privacy control, including when utilizing privacy-by-design products or services,' while regulations in another jurisdiction might require explicit consent from the user to send a Sec-GPC preference."

And they said it should be noted that users' preferences for privacy is well established, and user agents are expected to convey user preferences as accurately as they can. To the extent possible, user agents SHOULD strive to represent user preferences, including if necessary by prompting users to decide whether they would prefer to have their data sold or not. And then this ties into the subject of the podcast: A site MAY produce a resource at a .well-known URL in order for a site to represent the fact that it abides by GPC. A GPC support resource has the well-known identifier /.well-known/gpc relative to the origin server's URL. And of course I didn't finish up talking about this, but Chrome 86 also is supporting another URL that is .well-known that we'll be getting to in a second.

So anyway, the point of this is that so we have a browser able now to clearly transmit as a beacon its users' privacy enforcement desires. We have California and probably the GDPR working to legislate the legal force of this signal. And we have a mechanism by which websites are able to, with low cost to themselves, also signal to anyone who's interested whether or not they support the GPC signal. And so at some point, as legislation rolls forward, it will be incumbent upon sites that are subject to the California Consumer Protections Act and GDPR which are collecting privacy and information about their visitors to affirmatively state through this mechanism that they support GPC, and to then honor the requirements under the regulations that they would be subject to when a user visits their site with the beacon on that they want, you know, they are affirmatively requesting GPC support.

So this all seems like really good news. GPC is not synonymous with the failed Do Not Track header, but it's clearly going to become a flag which will no doubt launch some lawsuits when sites are believed not to be honoring the intentions of their users in regions where users' intentions must by law be enforced. So this is like a perfect mechanism for doing that. They've clearly specified it the way we want so that no expression of preference is, you know, does not send a signal. There is no presence of the header set to No. It's either no header at all; or, if the header is there, it is set to a value of one. And really only because headers have to have a value. They're always a name:value in HTTP queries. And that'll be sent to any website that needs to take a look at this.

So we've often talked, as all of our listeners know, I was very bullish about the idea of a DNT header, which didn't ever have any teeth behind it. This one looks like it's getting some teeth. So bravo. Looks like Mozilla will be adopting it. I don't see how Google can have their Chromium-based browser not do it, so Chromium will get it. And then all the Chromium browsers will get it, and that'll pretty much - oh, and two authors at Apple were the authors of the spec. So it seems clear to me that Safari, not surprisingly, will also be having a switch you can flip on if you want your privacy enforced. So yay.

Two new countries have joined in with the Five Eyes intelligence-sharing alliance. In addition to the U.K., U.S., Canada, Australia, and New Zealand, we now have Japan and India. I just thought I would note, just because I think this is like the ultimate collision here of encryption and government, they've now put out over the weekend another call for industry to provide some form of, unfortunately, people keep calling it a backdoor. That's just so heavily laden, I just wish we could come up with a different word. I mean, I have been careful to call it "subpoena-compatible encryption or decryption."

And basically they say we call on technology companies to work with governments to take the following steps, focused on reasonable technically feasible solutions: Embed the safety of the public in system design, thereby enabling companies to act against illegal content and activity effectively with no reduction to safety, and facilitating the investigation and prosecution of offenses and safeguarding the vulnerable, blah blah blah. You know, same stuff. Now we've got two more countries joining and the volume of the drumbeat increasing.

And as we know, I've often argued that Apple could enable iMessage and FaceTime wiretaps, as could Zoom, because in those instances they are managing the keys on behalf of their users. But even I see no possibility for feasibly decrypting unmanaged applications such as Threema or VPNs. That really would require some sort of sophisticated backdoor in encryption and decryption, and that would be an unmitigated disaster. So when I try to understand why I've been saying something is possible, well, it's something is possible in a constrained environment where you've got a managed encryption platform like iMessage, FaceTime, and Zoom. There is nothing that can be done in an unmanaged platform like with a VPN or Threema or any other encrypted applications.

This comes to light because their language says that they want device encryption, custom-encrypted applications, and encryption across integrated platforms like a VPN to be accessible. And you just can't do that. There is just no way. I mean, it's like I can't even conceptually imagine how that could work. You'd literally have to weaken the encryption so that it would take a huge amount of computing resource that only a massive entity would have. But you'd have to be assuming that it couldn't be duplicated by others.

Leo: Right. Of course it can't.

Steve: It can't. You can't make that assumption.

Leo: So you're saying end-to-end encryption. Is that what you mean by unmanaged is end-to-end encryption, encryption where there is no third party?

Steve: Correct, because Apple is a third party that manages the encryption.

Leo: Right. But as soon as you have no third party, as in Threema or Signal or...

Steve: A VPN.

Leo: A VPN. Well, there's a third - is there not a third - so the VPN server doesn't necessarily see your traffic. Well, it does because it's emerging. But if you had your own VPN is what you're saying.

Steve: As you've often said, I run a VPN where I have a server and a client.

Leo: They've had to compromise that software that you're running.

Steve: Right, right.

Leo: Because as we've also often talked about, with end-to-end encryption, the clear text sometimes, you know, if it's available on a device, all you need to do is compromise the device. You don't need to compromise the encryption.

Steve: Right. And in fact these guys are now saying they want statically encrypted content, that is, the content of an iPhone to be decryptable on presentation of a search warrant. So again, I would argue Apple could do that, only because of the special position Apple has because they have tethered all of their iDevices back to their cloud infrastructure. But freestanding devices that do not have a big daddy or a big brother the way we have with Apple, a benign overlord, there's just no way to do that.

Leo: And as we know, the encryption technology is well understood and is easily transmitted. And roll-your-own encryption is totally doable. It doesn't take a huge amount of sophistication to write your own crypto which has no backdoor in it.

Steve: I did.

Leo: You did. And you're using NACL or some sort of library, so you don't even have to handle the crypto library. You just have to write an interface for it.

Steve: Right.

Leo: And you can verify that the library's uncompromised.

Steve: Yup.

Leo: So this is that argument when they outlaw crypto, only outlaws will have crypto kind of argument; right?

Steve: Yeah.

Leo: The only people who will be inconvenienced are the people who can't roll their own crypto, or aren't sophisticated enough to do so.

Steve: Correct. Correct.

Leo: This is terrible. It's a terrible idea.

Steve: I don't know what's going to happen.

Leo: This basically makes everybody vulnerable, except for people like you and me who can do our own. And everybody else is just, you know, including...

Steve: Well, but again, if we do our own, then we're breaking the law.

Leo: That's true.

Steve: And so I don't want to break the law.

Leo: That's true.

Steve: I want to have good laws.

Leo: Yeah.

Steve: So I'm continuing to work toward the release of our ReadSpeed benchmark. And I thought I should explain that the reason this is taking so long is that since this technology will be moved into SpinRite, it seemed to me that in the interest of minimizing the total delivery time for SpinRite, which is what we all want, I should be developing this code for its final release form for SpinRite's use. So that's what I've been doing. Essentially, everything I'm writing will be ready once we nail any bugs that we find by broadening the testing. It'll just be drop-in ready. I'll be able to move this into SpinRite because of course I know exactly what SpinRite's internal structure is.

So there's like all kinds of stuff that I don't need for the benchmark, like I just solved a problem of associating the BIOS view of drives, which are just BIOS numbers. They're always a hex byte with the high bit set so it's 80, 81, 82, 83, 84, associating that with the physical controller and drive which I'm talking to. And that's necessary because SpinRite still supports BIOS-based drives, for which I won't yet have low-level hardware drivers. And so I don't want drives to appear twice. I don't want them missing. I want to provide and present a coherent view.

So all that's been resolved in the last couple days. So a benchmark doesn't need any of that. But rather than writing this all twice, I'm writing it for SpinRite and then sort of giving it a benchmark wrapping. So anyway, we're really moving along nicely with that. And the GRC forums are continuing also to develop. We've got about 3,380-some registered users, generally about 70 or 80 unregistered people visiting at any given time. The so-called "Off-Topic Lounge" has become so popular that it's clear that a community of sorts is beginning to form. So I'm going to be dividing it up into on-topic subforums for talking about hardware, software, operating systems, a dialogue about the Security Now! podcast and so forth. And even health and sci-fi, which are other interests of mine just because people want to talk about it.

So anyway, all of this is moving forward, and of course its intention is to be ready to support GRC's product stuff, whether free or commercial. So the ReadSpeed benchmark will be first.

And speaking of sci-fi, briefly, Ryk Brown has just dropped book 15 of the second series - the second series, Leo - of 15 of his truly fabulous Frontiers Saga series, which I recommend without reservation. It's wonderful, pleasant, pure escapist science fiction featuring a cast of very well-formed and diverse characters who you really get to know and appreciate. And it has a lot of humor, which makes it fun. And Ryk Brown is one hell of a storyteller.

So with this book, this wraps up the second series of 15. So this is the 30th book in the series, the planned series of 75 books. And I for one hope he never runs out of steam because this is one of the best, easiest to read, action-packed space operas I've ever encountered. However, what Ryk has run out of patience with is Amazon's Kindle Unlimited service. He's been complaining that he's just not making any money, despite the immense popularity of his novel series.

Leo: Because they're free. They're free to you and me.

Steve: Yes, yes. So anyway, doesn't look like the next 45 books will be available there. He's been talking about finding some other way to distribute his eBooks. And no one cares. If we have to go somewhere else, we will. The fans of this series have made it clear to him that we'll pay whatever reasonable price he wants to charge, no matter where that is. So anyway, I just wanted to - for anyone who's been reading, I know that John has been following the series and liking them, and I've got several family members who are. It's just wonderful, wonderful sci-fi space opera.

Leo: Uniform Resource Locator.

Steve: Yes.

Leo: But I hear URI as kind of a more generic term, Uniform Resource...

Steve: Identifier.

Leo: Identifier.

Steve: And while you were doing that, I just created GRC's shortcut for the week, and it's a perfect explanation of the differences. So grc.sc/788.

Leo: Okay. Should I read it out loud?

Steve: No, just...

Leo: I'll pull it up.

Steve: You type that in.

Leo: Okay. So a visual?

Steve: Yes, it's visual, sorry.

Leo: Oh, good. Okay. Wait a minute, did I get the wrong one? Port Authority database. No, that's...

Steve: Oh, yeah. You don't want that. Grc.sc.

Leo: Oh, .sc, that's why.

Steve: Shortcut, yup.

Leo: I did GRC.com, dummy.

Steve: Yeah, that would take you to port number 788.

Leo: Yes. I typed everything right. Oh, is that what it did? Oh, that's clever. By the way, I replaced all my stuff with Ubiquiti gear at home. And of course first thing you do, you put a new router - a new router, new switches, new every sort of thing. First thing you do, you go to ShieldsUP!, green.

Steve: Nice.

Leo: I've never seen so much green in my life.

Steve: Nice. A field of green.

Leo: It made me feel really good. I thought, oh. I have to say, and I know you were the one who introduced me to this stuff, you've been telling me about the Edge Router X for a long time, this is the big boy. I don't know why I'm having trouble. Grc.sc; right?

Steve: Yup. Forward slash.

Leo: 788.

Steve: 788.

Leo: There we go. Oh, we've got to zoom in on this sucker. All right. I'll make it bigger. Make it bigger. Here we go.

Steve: Yeah, I just found it there on the fly. But it does, it makes this very clear how URN, URL, and URI relate to each other.

Leo: So the URI is the whole thing.

Steve: Yes, it's the...

Leo: The URL is just the page you're going to, minus any additional qualifiers; right?

Steve: Correct. Which there they call the resource.

Leo: So for instance, grc.sc/788?, you know, query equals 53. Everything before the question mark's a URL. The full thing, including the query, is the URI.

Steve: Right.

Leo: What's the URN? Oh, that's minus the protocol.

Steve: Correct. It does not have the URL scheme. So anyway, so URI sort of being the more technically correct generic term.

Leo: It's the full address, everything.

Steve: Of which URLs are a subset.

Leo: Right.

Steve: So, okay. As I mentioned at the top, this discussion about well-known URIs was first triggered by a new feature in Chrome 86. So let me talk about that. What's just been added to Chrome 86 is the so-called well-known URI change password standard. That is, support for that. What Google said was, starting with 86, Chrome's safety check supports the .well-known/change-password standard. This is a W3C standard that allows websites to specify the URL where users can go to change their passwords. So Chrome 86 adding support for the standard means that users would, for example, be able to press a button in the Chrome password settings screen and be taken directly to that page for that site to change their password right away.

In other words, this is a means for a website to universally and in a standards-compliant fashion tell browsers where to take their users who wish to change their password at that site. Which is just too cool. In my mind, it's another of those perfect compromise solutions that provides a startling amount of benefit at very little cost. And we just talked about that, that GPC. GPC is another perfect example. It is also part of the standard. So the beginning of the URL is always - it's always from the root, so it'll be /.well-known. So everything downstream in the URL of .well-known is reserved. That name space within a server is reserved by the IANA. And password-change is a defined token of that, as is GPC. So if a browser pulls .well-known/gpc, that's an automated universal standardized way of learning whether that site supports this emerging GPC standard.

And this is sort of like one of the Wild West aspects of the World Wide Web has been that anyone could design a website pretty much any way they chose. There were, like, no rules. The one convention in the beginning was that the root of a domain would contain like the default index or landing page. But even that could be whatever the webmaster desired. Sometimes it was index.html or .htm or default or whatever. And if it were only ever people who were visiting, that would be fine. But it wasn't long before it occurred to someone that bots could be created to mimic people's actions as they browsed a site. Starting from that site's root page, which was a well-known, that was like the one predefined URL for any site, and then following the links from that page just like a person might.

So of course this quickly became known as spidering since a bot was crawling around on the web. And the trouble was the first bots were not very smart. They would sometimes

get lost, do dumb things, start looping, issue dumb and sometimes expensive queries or whatever. And sometimes they, like, would slam a site with queries far too fast for the site to handle.

So the first divergence from that pure "anything goes" ad hoc design of websites was the agreement about the universal robots.txt file, which webmasters all know about. By universal agreement, a text file named robots.txt would be placed on the root of any site wishing to control the actions of any bots that might enter the site. The robots.txt file would specify which areas a bot might go and which regions were off limits to a bot. It even provided for bot-specific behavior by matching rules against the bot's user-agent header. And of course a bot's adherence to robots.txt was entirely optional. But well-behaved bots would, often for their own sake, abide by the file's requests.

So the website's root, followed by /robots.txt, those were the first instances of so-called well-known URIs. But the concept was so useful as we move more toward automation that the World Wide Web Consortium has, as I've said, formalized, standardized, and hugely expanded upon this feature. The work on this began 10 years ago, back in 2010, with RFC 5785, which was titled "Defining Well-Known Universal Resource Identifiers," and its abstract could hardly be shorter. It said: "This memo defines a standard path prefix for well-known locations," and then it says "/.well-known/" in selected Uniform Resource Identifier schemes. Then just this last, actually the May before last, May of 2019, that RFC was formally obsoleted with 8615. That RFC defines the function of all of this.

And basically it's just what I've described. It's a formal, well-known URI beginning with .well-known, which reserves that name space of a server's total URL space, which it's free to do anything else with. And by convention, for example, in the case of the /change-password query, none of these directly return contents. Instead, they always return a redirect to the URL of the site where that resource is located, rather than, for example, being the password-change page. Instead, any query to that password-change URL returns one of the several temporary redirect codes.

The W3C has a GitHub page where they talk about web application security change password. And they said: "Client-side password management software helps improve both the security and usability of websites which require authentication. It improves security by reducing cross-site password reuse, and enhances usability by providing autofill functionality." In other words, a fancy way of saying password managers are good.

Then they said: "Sites currently lack a way to programmatically advertise where a user can change their password. By proposing a well-known URL for changing passwords, this specification enables password managers to help users change their passwords on sites which support it. Servers should redirect HTTP requests for an origin's change password URL to the actual page on which users may change their password by returning a response with a redirect status of 302, 303, or 307, and a Location header. Clients must handle such redirects when requesting a change password URL."

And note that all three of those are temporary redirects. The reason you want to use a temporary is that proxies and some browsers will, trying to be helpful, cache a permanent redirect so that they don't even follow the redirect the second time they get the URL. If they've cached it, they just immediately go to where the redirect pointed them the first time they encountered it, which is a problem because webmasters might want to change, for example, where their change password page is, and then they would change where the change the well-known change-password URL redirected browsers. So you want to make it temporary.

And that's really all there is. The adoption of this would be very cool. Even in the absence of a password manager, a user's web browser could, when you visit a site, much as web browsers now go and get the favicon from the root, right, in order to give the site's icon in the browser tab or in a shortcut or a bookmark if you create one, they could also check to see is a well-known change-password URL defined. And if so, they could light up a little button on the browser's user interface Chrome, or maybe like add it to the dropdown top-level menu for the browser so that when you click on the menu, when you're on a site, it could show you a "Change your password for this site." And if you clicked it, it would immediately take you to that site's password change page which had been set up for exactly this purpose. So just a very cool facility.

And similarly, this GPC has a well-known URL, and there's a whole bunch of them. As I mentioned, they are IANA-reserved URIs. Anybody can go and license one, like reserve and register one that they want to use for their purpose. I scanned through them, and there's just, you know, a bunch of them. Some of them are very application-specific. I've got no idea what many of them are. But what we are beginning to see now is a standardized way, sort of an extension of what used to be robots.txt, but for all kinds of applications, which allows us to better automate the interaction of the clients that we're using with the sites and servers that we go to. So just very cool.

Leo: Well, very interesting.

Steve: Yup. Something just sort of crept up on us without lots of people being aware of it. So I just wanted to put it on everybody's radar.

Leo: Yeah, yeah. Yeah, really cool. Well, we'll probably see those pop-ups at some point; right?

Steve: Yeah. So certainly I would imagine our password managers will adopt it quickly because it provides them with a means of immediately taking the user to a page offered by a site. And we'll go from there, I think.

Leo: Yeah, yeah. Really cool. As always, Steve keeps us up on the latest technology. That's why this is a must-listen show. You'll find copies of the show at his website, GRC.com. That's where you'll also find SpinRite, his bread and butter, the world's finest hard drive maintenance and recovery utility, 6.0, soon to be 6.1. You want to participate, buy today, you'll be automatically upgraded, and you can participate in the testing and so forth. GRC.com. 16Kb versions of this show for people who really have no bandwidth. They're a little scratchy-sounding, but hey, they're there. There's also a transcript. I think that's probably the smallest version of the show, a nice text transcript. You can read along as you listen. He also has 64Kb audio.

We have audio and video at our website at TWiT.tv/sn. That's another way you can get it. Or you can go to YouTube. There's a Security Now! channel on YouTube. But do, if you are listening, watching the YouTube version, hit the Subscribe button, smash the bell, ring the bell, as they say, so that you can be notified when there's a new episode. That's actually one of the great things about podcasts. On-demand versions of this show are available anywhere you get your podcasts. All you have to do is sign up. It's free, you know, get the podcast app. Search for Security Now!. "Subscribe" I guess is the better word, although that always implies payment to me. So whatever it is, press the button. Automatically...

Steve: Register. No.

Leo: Register. That's not it, either. There's no real good word for this.

Steve: Yeah.

Leo: "Subscribe" implies payment. There is no payment. It's free. But do press the "Subscribe" button, and that way you'll have every episode the minute it's available. We are currently at 788. The world-famous 789 will come back next week.

Steve: Yay.

Leo: Yay. Steve, have a wonderful week. Have you seen, there's a new show on Fox, I think you might like it, all about a killer artificial intelligence called "NeXt."

Steve: Ooh, no. I want to know about that.

Leo: Yeah. John Slattery stars in it. And the tech's pretty good. There's a lot of code on the screen. It's Python. But other than that, at least it's not JavaScript. It could be worse.

Steve: What's the name of the show?

Leo: It's called "NeXt," N-E-X-T. And it just debuted last week. So your first episode came out. But you should be able to get that on-demand on Hulu or wherever you get your shows on-demand.

Steve: Cool.

Leo: "NeXt," N-E-X-T. It's a killer AI. Who doesn't love that?

Steve: Ooh. Yeah, I mean, I miss Daniel Suarez. We haven't had a really good...

Leo: I know.

Steve: Now, those were great books.

Leo: This is a little Suarez-y, actually. It's pretty good. I'm enjoying it. Again, on the basis of one episode. Thank you, sir. And we'll see you next week. Have a great week. See you next time on Security Now!.

Steve: Okay, my friend. Bye.

Leo: Bye-bye.