SECURITY NOW!

Transcript of Episode #694

# The SQLite RCE Flaw

**Description:** This week we look at Rhode Island's response to Google's recent API flaw; Signal's response to Australia's anti-encryption legislation, the return of PewDiePie; U.S. border agents retaining travelers' private data; This Week in Android hijinks; confusion surrounding the Windows v5 release; another Facebook API mistake; and the eighth annual most common passwords list, a.k.a. "How's monkey doing?" Why all might not be lost if someone is hit with drive-encrypting malware; Microsoft's recent four-month run of zero-day vulnerability patches; the Firefox 64 update; a reminder of an awesome train game for iOS, Mac, and Android; some closing-the-loop feedback with our listeners; and a look at a new and very troubling flaw discovered in the massively widespread SQLite library, and what we can do.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-694.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-694-lq.mp3

SHOW TEASE: It's time for Security Now!, our last episode of 2018. Wow, what a year it has been, and we're not done yet. Microsoft says this was the worst year for zero days, and there are more coming in Windows. A flaw that potentially could be disastrous in a software component almost everybody has on every machine you have. And of course a Picture of the Day, just for the season. Plus I might sing a song. Don't be afraid. It's Security Now!, next.

**Leo Laporte:** This is Security Now! with Steve Gibson, Episode 694, recorded Tuesday, December 18th, 2018: The SQLite RCE Flaw.

It's time for Security Now!, the show where we protect you, your privacy, your loved ones, and your security online with the help of the man in charge over here, Steve Gibson of the GRC, Gibson Research Corporation, at GRC.com. Hello, Steve.

**Steve Gibson:** I'm not sure what I'm doing. Am I doing a Vulcan salute, or am I shooting myself in the head with my first two fingers? I'm not quite...

**Leo:** It's kind of crazy, yeah.

**Steve:** Just not really sure.

**Leo:** Last show of 2018, Steven.

**Steve:** It is indeed. Yes, and this is 694, so we'll be into the 700s pretty soon, the 700 club.

**Leo:** Wow.

**Steve:** Wait, what? Anyway...

**Leo:** Huh? Well, and as I pointed out, because you have so widely pointed out that your last show will be 999...

**Steve:** Oh, well, I wrote that in - it's been written in assembly language, Leo. I have no flexibility there. I've only got...

**Leo:** I think it can only do three digits.

**Steve:** I have a three-digit format string, and it's just, you know, when it hits 999, it's like, oh, well, it's been...

**Leo:** I, as many, feel it should go to AAA, but that's another - or A00. But that's another matter entirely.

**Steve:** Been nice knowing you.

**Leo:** I feel like people are going to start counting backwards, like this is going to be, like, once we get to 700, people are going to say, oh, there's only 299 shows left, only 298, 297.

**Steve:** Yup, bottles of beer on the wall. So we'll...

**Leo:** It'll be the countdown, yeah.

**Steve:** So while I was assembling today's news, I got to one of the stories that I had run across and dug into it a bit more. And I thought, ooh, this is really not good. There is a, well, the absolute number one most popular embedded database is SQLite, you know, SQLite. And, I mean, it's everywhere. It's in - I meant to get some...

**Leo:** It's in iOS. It's in every Mac. I don't know if it's on Windows, but it's everywhere, you're right.

**Steve:** Well, yeah. And what happens is that apps bring it with them. I found four instances in my own system. We'll be talking about this at the end of the show. It's the topic, the main title. Well, the title of the show is "The SQLite RCE Flaw." A remote code execution flaw was found that allows execution of an attacker-provided code if they have any way to get access to a SQLite command. And, for example, browsers allow that. Until Google was notified of this, you could crash Chrome and take over Chrome. Chrome uses SQLite. It turns out that it's pervasive. The Chromium engine was using it. But it is, I mean, well, we'll get there.

So I think a really interesting and important way to ring out this new year which has been, you know, that has wrung us out. You and I were saying before we began recording that, without question, the number one main issue that we dealt with in 2018, and it started right off the top, it was like the first podcast of 2018, was the Spectre and Meltdown problems, the idea that the industry learned that the fundamental architecture of our processors was allowing cross-thread, cross-core, basically interprocess leakage of information, which is, you know, it dominated the news this year.

But there's a lot of stuff to talk about. We're going to take a look at Rhode Island's response to Google's recent announcement of their second API flaw; Signal's response to Australia's anti-encryption legislation; the return, believe it or not, you were fortunately in New York, Leo, when Jason and I had to deal with PewDiePie.

**Leo:** Oh, geez. What does he have to do with security?

**Steve:** I know. You're going to find out.

**Leo:** Oh, boy.

**Steve:** The return of PewDiePie we have. We also have news that U.S. border agents - and this really does get me - are retaining travelers' private data.

**Leo:** Oh.

**Steve:** Not just, yes, not just like checking it while you're crossing, but they're not performing post-transit elimination of the data. We also have This Week in Android Hijinks with another interesting exploit for Android and a little bit of a takeaway there. Some confusion that I saw in the industry surrounding the recent WordPress 5 release. We have another Facebook API mistake that I know you've been talking about, this photo API. We have, oh, and Leo, the eighth annual most common passwords list, also known as "How's monkey doing?"

**Leo:** Yeah, really. True.

**Steve:** We also have why all might not be lost if someone is hit with drive-encrypting malware. The observation on Microsoft's recent four-month run of zero-day vulnerability patches. Every single of the last September, October, November, December, they were patching things that were found in the wild, so that's an interesting bit of news. We've also got the move of Firefox to 64 with some new features. A reminder of an awesome train game that I fell in love with. You did, too.

**Leo:** I remember, yeah, yeah.

**Steve:** Many years ago. Available on iOS, Mac, and Android; and I was recently reminded of it. Also we have a little bit of closing-the-loop feedback with our listeners, including a SpinRite song for the holidays. And then we're going to dip into this SQLite Remote Code Execution flaw, and with a takeaway for our listeners. It turns out that it's probably worth doing a bit of an audit on your system to find the instances of SQLite and see if there are updates available. The danger is purely a function of attack surface.

And so, for example, apparently Thunderbird uses it, which is the email client that I'm using. And of course email is all attack surface. It's just one big exposed attack surface. So that's not good. But it turns out it's pretty simple to find the instances of it. And I think people will be surprised how prevalent it is. I mean, the network monitoring app, the little NetWorx thing that I use has SQLite in it. Like, okay, why? Who knows. But anyway, lots of fun stuff to talk about, and a fun Picture of the Week also for the holidays.

**Leo:** You do Pictures of the Week but today I'm going to do a Mug of the Week. This was sent to me by Spencer [Dudzik]. Actually his dad is a big fan of this show and iOS Today and Know How, and his sons watch. Spencer really liked a joke that you told on the show and sent us a mug. Well, he sent it to me. I'll send you one if you want, too. Remember this one? You can't use "beef stew" as a password because it isn't "stroganoff." Oh, ow, oh. So Spencer, thank you. I really appreciate that. It's got a nice green inside, so I'll be using it for my green tea.

**Steve:** Much as you might want to forget that one, Leo, now...

**Leo:** Every day I'll see...

**Steve:** Every time you bring it up to your face.

**Leo:** You can't use beef stew as a password because it wasn't stroganoff. I blame you. I blame you. It's your fault.

**Steve:** I do like a good stroganoff, though, Leo.

**Leo:** Oh, Stroganoff's fantastic. Just don't use it as a password.

**Steve:** No. So I've been sitting on this Picture of the Week for quite a while. It's just fun.

**Leo:** It's festive for the holiday season.

**Steve:** Apropos of, yes, the holidays. This is just a little ditty familiar to people who are familiar with Christmas songs. This says: "He's making a list. He's checking it twice. He's gonna find out who's naughty or nice. Santa Claus is in contravention of Article 4 of the General Data Protection Regulation (EU) 2016/679." Bad Santa.

**Leo:** Bad Santa.

**Steve:** You've got to, if you're making a list...

**Leo:** Four percent of annual revenue out the door, Santa.

**Steve:** He needs to be careful about data retention.

**Leo:** Yes.

**Steve:** And, you know, be careful with what information he does about who's naughty and nice because that could be abused if it fell into the wrong hands.

**Leo:** So funny. So funny.

**Steve:** You never know. So we're going to have to have some sort of new protocol for Santa.

The Rhode Island government - when I saw this, I thought, huh? What? Rhode Island government sues Google after the latest Google+ API leak.

**Leo:** Oh, boy.

**Steve:** And I'm thinking, okay. So clearly not everyone was as happy as I was over Google's conduct in handling their most recent API leak. Of course it was...

**Leo:** And shutting down Google+ even sooner as a result.

**Steve:** Yes. It was the topic of last week's podcast. It's like, bravo, Google. Anyway, within a day of this announcement of the second leak, a class action lawsuit was filed by the Employees Retirement System of Rhode Island (ERSRI), a government-owned investment fund that provides retirement, disability, survivor, and death benefits to state and municipal employees and public school teachers. So I'm thinking, okay. So they're suing Google over an API flaw where there's no evidence that any information was stolen. Because remember Google said, as far as we know, you know, we've audited this. There was no indication that anything happened. But we caught it quickly, we fixed it, thank you very much. So how was anyone harmed?

Well, this ERSRI accuses Google of intentionally misleading shareholders and federal regulators by failing to disclose its Google+ data leaks in a timely fashion. And although the lawsuit was filed immediately following the second leak report, it also cites both this most recent and the October API incident that we covered at the time. So remember we talked about it in October that Google was kind of caught in the wringer because, as I recall, I think it was The Wall Street Journal who broke the news that there was a problem that Google had chosen not to go public with, and so that was a bit controversial. Anyway, the second incident was announced Monday, a little over a week ago, eight days ago, when Google said that - wait, no. Was it a week ago or yesterday?

**Leo:** It might have been a week ago. Yeah.

**Steve:** Yeah, yeah, yeah, when Google said that this API update that they had performed, or an API update had introduced another potential for data leakage which the company patched within a week. Google said that this second leak wasn't abused to harvest user data. But if someone did, the data of 52.5 million users was exposed. So not good, but they fixed it. So again, okay, how is this grounds for a lawsuit?

The ERSRI officials are claiming that Google's recent string of leaks and supposedly delayed disclosures have harmed Google's own company stock value, which has in turn incurred losses to investors such as itself.

**Leo:** No, no, no.

**Steve:** It's like, really?

**Leo:** Really?

**Steve:** Really? So you're going to...

**Leo:** Really? Sue Equifax. Sue Marriott. Sue Yahoo.

**Steve:** Exactly, somebody who, like, really deserves a good kick. Okay. So Rhode Island's general treasurer, Seth Magaziner, said: "Google had an obligation to tell its users and investors that private information wasn't being protected. Instead, Google executives decided to hide the breaches from its users and continued to mislead investors and federal regulators. This is an unconscionable violation of public trust by Google, and we are seeking financial restitution on behalf of the Rhode Island...."

**Leo:** It's not a bad idea to establish a precedent that forces companies to be open about this stuff.

**Steve:** Well, so I wanted to bring it up and focus our attention on it a bit because, as we know, responsible and timely disclosures of mistakes such as this second one by Google, which they caught and killed quickly, then disclosed, is good for our ecosystem. So I'm a little worried that, if we're starting to establish an "admit culpability and be sued for it" precedent...

**Leo:** Oh, that will put it back under the rock.

**Steve:** Yes. And it's finally coming out from under the rock where it's been.

**Leo:** Right, right.

**Steve:** So if companies start being sued the instant they acknowledge that they made and fixed a mistake, where nobody was hurt, over the reputation damage their own responsible disclosure incurs, then it's foreseeable.

**Leo:** That's a very good point.

**Steve:** That companies are more like, I mean, they're going to say, hey, wait a minute, we don't want to expose ourselves to a class action lawsuit. And of course you and I feel the same way about class action lawsuits. The plaintiffs get nothing, and the attorneys - I mean, clearly, if this thing launched a day after, it had already been written. Somebody was ready to just fill in the blank, "Google+," and launch this thing. So anyway, I just wanted to, when I realized what the suit was about, I thought, oh, this could have a chilling effect. And we don't...

**Leo:** So in effect they wanted them to keep it secret because it hurt the stock price.

**Steve:** Yes.

**Leo:** Huh. Huh.

**Steve:** Yes.

**Leo:** Geez, Louise.

**Steve:** Yes. So I guess the only way to resolve this ultimately is maybe eventually, because mistakes are going to happen, is for the industry to mature so that it's understood that fixing and acknowledging an error should not cause reputation damage. It's the reason for last week's podcast celebrating Google's conduct.

**Leo:** Companies should be indemnified against liability for reporting flaws. This is good.

**Steve:** Yeah, yeah. Meanwhile, we have - I love the title of this post. This was from Joshua Lund, who is a sysadmin, programmer, privacy enthusiastic, security fan, writer, occasional cyclist, and one of the Signal developers, who posted at Signal.org last week. The title of his posting was "Setback in the Outback." And I lightly edited what he wrote for the podcast.

He said: "Like many others, we have been following the latest developments in Australia related to the Assistance and Access bill with a growing sense of frustration. The widespread adoption of strong cryptography and end-to-end encryption has given people around the world the ability to protect their personal information and communicate securely. Life is increasingly lived online, and the everyday actions of billions of people depend on this foundation remaining strong. Attempting," he writes, "to roll back the clock on security improvements which have massively benefited Australia and the entire global community is a disappointing development.

"More than eight years have passed since we released the public beta of what is now known as Signal. Throughout the entire development process, the project has faced resistance from people who struggle to understand end-to-end encryption or who seek to weaken its effects. This is not a new dynamic." He says, in a paragraph by itself: "We can't include a backdoor in Signal, but that isn't a new dynamic either.

"By design, Signal does not have a record of your contacts, your social graph, conversation list, location, user avatar, user profile name, group memberships, group titles, or group avatars. The end-to-end encrypted contents of every message and voice/video call are protected by keys that are entirely inaccessible to us. In most cases now we don't even have access to who is messaging whom.

"Everything we do is open source, and anyone is free to verify or examine the code for each release. Reproducible builds" - which is not easy, by the way - "and other readily accessible binary comparisons make it possible to ensure the code we distribute is what is actually running on users' devices. People often use Signal to share secrets with their friends, but we can't hide secrets in our software.

"Everyone benefits from these design choices, including Australian politicians. For instance, it has been widely reported that Malcolm Turnbull, the 29th Prime Minister of Australia, is a Signal user. He isn't alone. Members of government everywhere use Signal. Even if we disagree with Christian Porter, we would never be able to access his Signal messages, regardless of whether the request comes from his own government or any other government.

"Although we can't include a backdoor in Signal, the Australian government could attempt to block the service or restrict access to the app itself. Historically, this strategy hasn't worked very well. Whenever services get blocked, users quickly adopt VPNs or other network obfuscation techniques to route around the restrictions. If a country decided to apply pressure on Apple or Google to remove certain apps from their regional stores, switching to a different region is trivial on both Android and iOS, and popular apps are widely mirrored across the Internet. Some of them can even be downloaded directly from their official website.

"One of the myriad ways that the 'Assistance and Access' [in quotes] bill is particularly terrible," he writes, "lies in its potential to isolate Australians from the services that they depend on and use every day. Over time, users may find that a growing number of apps no longer behave as expected. New apps might never launch in Australia at all."

He finishes: "Technology organizations looking to open offices in a new country could decide that AEST (Australian Eastern Standard Time) isn't such a great time zone after all. As remote work continues to become more prevalent, will companies start saying 'goodbye,'" he writes, "instead of 'g'day' to applicants from Australia, who are unable to sufficiently secure and encrypt their corporate communications?" He says: "This doesn't seem like smart politics, but nothing about this bill seems particularly smart. We remain committed to fighting mass surveillance worldwide. We encourage users in Australia to reach out to their representatives and express their opposition to the Assistance and Access Bill." And it's interesting. I looked around for other reaction to that, and haven't so far found anything.

**Leo:** 1Password posted a blog post. ProtonMail posted a blog post.

**Steve:** Good.

**Leo:** So a number of people have responded. I hope LastPass will say something. But it does raise some questions in my mind. So I have some questions for you.

**Steve:** Okay.

**Leo:** About this law, the Triple A law, which hasn't yet gone into effect. I think it doesn't go into effect till next year.

**Steve:** Until Parliament meets at the beginning of 2019, yes.

**Leo:** And so there's some chance that maybe they'll change it or retract it. And we don't know how strongly, assiduously it'll be enforced. But it says, essentially, that anybody who provides encrypted services must be able to provide unencrypted cleartext versions for law enforcement if they ask.

**Steve:** Yup.

**Leo:** Which means Signal provides encrypted services. ProtonMail provides encrypted services. LastPass and 1Password provide encrypted services. It sounds like they'd have to, you know, LastPass, we just did the ad, says we don't ever have access to your vault. Only you do.

**Steve:** We absolutely know for a fact that they don't.

**Leo:** Yeah. It's trust, what you call "trust no one," end to end. Or another way to put it is end-to-end encryption. So the question is, does that mean they have to then modify their code to sell it in Australia, so that when requested they can provide cleartext? And it's my understanding that it does.

**Steve:** So one interesting thing that I - because I've been thinking about this for the last week. One of the ways the world has evolved is that applications no longer talk to the hardware directly. They talk to an API which the operating system publishes. So it could be, you know, we've often talked about this idea of accessing communications either before they're encrypted or after they're decrypted. Well, there is a common place where the keyboard API exists. The video API, the screen output API, and that's this OS layer.

And so it could be, I mean, I don't know, we don't know how this is going to evolve. But it could be that Android and iOS themselves could provide a pre-encryption and post-decryption interface because, after all, as we've often said, I mean, the user is entering plaintext into their keyboard, and they are viewing plaintext on their screen. That plaintext transits from the encrypted tunnel through the operating system after it's been decrypted.

So what we may end up with is a general design which would - the advantage, first of all, it means that our phones do have this - our operating systems have a designed-in monitoring facility. It isn't a backdoor into any of the encryption. It doesn't weaken any of the encryption. It just gets to it before it's been encrypted or decrypted, much like the user, typing on their keyboard, like a keystroke monitor. And so that's a means by which a universal solution could be found.

**Leo:** With the cooperation of Google and Apple.

**Steve:** With the cooperation of the operating system.

**Leo:** Actually, it wouldn't even be Google. It would have to be the manufacturer because Samsung would have to say, okay, we'll do that. Yeah.

**Steve:** Right, right.

**Leo:** And what we don't know is, yes, that would be a solution, but that would require Australian law enforcement to know and do it, as opposed to going to Signal and saying, no, you've got to do it.

**Steve:** Well, and then there's also the question of storage; right? Because, for example, we know in a point-to-point system there is some storage of received messages at the receiving end. But there's no central storage of prior messages by the provider. I mean, basically, Signal is providing a system that allows two people to interchange messages securely. The only storage occurs in the message stream at each end. So, for example, there doesn't exist, it's not like they don't want to provide it in the case of Signal's being able to provide past message traffic. It doesn't exist anywhere in a third location. So, as I said, I really think...

**Leo:** But wouldn't they have to - couldn't they be forced to rewrite Signal to provide that?

**Steve:** Well, yes. And what was...

**Leo:** That's the question. And what are the penalties? We don't know what that is.

**Steve:** And we sort of heard Signal's...

**Leo:** We just pull out of Australia. We wouldn't do it.

**Steve:** Exactly. If you guys don't want to allow security, we're about security. We are not going to compromise our security. If you don't want access to Signal, I mean, if the only way you'll allow us to be there is breaking encryption, we're just not...

**Leo:** Bye-bye.

**Steve:** We're just going to say no.

**Leo:** Apple wrote a seven-page letter in October to the Australian government, saying it raises cybersecurity concerns, gives the states power to abuse users' privacy. "We encourage the government to stand by their stated intention not to weaken encryption or compel providers to build systemic weaknesses into their products." That's always been Apple's point of view with the FBI and others.

So my email provider - it's a little different within Proton Mail because ProtonMail promises end-to-end encryption. My email provider, FastMail, is in Australia. And it worries me that they may be somehow compelled using this to - the thing is, none of my email is encrypted. It's in plaintext. So they wouldn't be doing anything. And if I want it encrypted, I don't use them, I use PGP to do it.

**Steve:** Right. So you know it's been five years since the summer of 2013 when Edward Snowden surprised us all with what we now have as an awareness of, for example, just how much the NSA's breadth and depth of technological surveillance was in place. And, I mean, it's almost kind of quaint now, Leo, when we think of going to a coffee shop and using Firesheep to intercept a decrypted browser session and impersonate someone. I mean, you can't do that anymore. A lot has changed in, I mean, I would argue our, especially this podcast's, our world has changed in these five years. Encryption has now become, I mean, it is now in place. Back then you only encrypted to log on, just to hide the username and password, and then you dropped back to an unencrypted connection, which is the way Firesheep was able to grab the browser session cookie across all these different services.

So I really think that - here we are at the end of 2018. I suspect that 2019 and 2020 are going to be the years where the slow-to-respond governments, I mean, the governments, as we've been covering, they're not happy with this going dark problem, that they're unable to any longer see into what's going on when they need to. We had the famous fight between the FBI and Apple that we talked about last week with the San Bernardino terrorist when they were unable to get into the iPhone with Apple's help. So, boy, I just think this is really interesting, the next couple years. I think this is going to get settled one way or the other.

**Leo:** Yeah, hope so.

**Steve:** Yeah, because countries don't want...

**Leo:** If a law like this is passed in the U.S., that makes it really challenging because companies might be able to walk away from Australia, but they can't walk away from every nation.

**Steve:** No, no. And I mentioned Google. When China said we need you to filter your results, Google said no.

**Leo:** They walked away, yeah.

**Steve:** They walked away, but then they're coming back because it's like, ooh, it's a pretty big market over there, so...

**Leo:** Actually, the latest news is that, while they developed this Dragonfly, that employees were so upset that they decided to not do it and put it on the backburner.

**Steve:** Oh. Well...

**Leo:** Which is good. That's great. Employees need to stand up.

**Steve:** Yes. And in the U.S. we have this notion of warranted search, which it's in the Constitution, and it's inconceivable to me that our legislature won't decide that, if a court orders communications to be intercepted for lawful purpose, that there doesn't have to be a way for that to be done. I mean, it seems like where we're headed. And again, so far we've just sort of been slipping by, I think.

**Leo:** I like it because you're not dogmatic on this. You recognize the need to do it. And I think you believe, and I think you've convinced me, that there is a way to do it technologically that doesn't harm our general safety and security.

**Steve:** Yes. And yes, the absolutists are right. And from an absolute sense, any anything is bad.

**Leo:** Any weakness is bad, yeah.

**Steve:** Any weakness is any weakness.

**Leo:** Right.

**Steve:** And so it's no longer absolute. But in the U.S. we don't have an absolute guarantee of privacy. We know that the Constitution keeps the king's men from barging in without warrant. But if you get a judge to say, yeah, we think there's reasonable cause to look in there, then, you know, so we don't have...

**Leo:** We need to find a way to do it without compromising everybody, without weakening the entire system.

**Steve:** Yes. And as I have said, in the unique case of the Apple ecosystem, as a consequence of the way Apple has designed their stuff, they could have a per-device key, which they protect, which would allow them to unlock - in this case, for example, it was Farook's phone after the San Bernardino attack, where they said, oh, we have no way to do it. Well, presented with a specific device, they could unlock that specific device if they used that technology, without in any way weakening everybody else, except that, yes, there would be a central repository of everyone's master key.

**Leo:** Well, that's an inherent weakness; right?

**Steve:** It is. It's an inherent weakness. It concentrates the responsibility. Apple doesn't want it. But it may be that the decision is made, sorry, if you're going to provide encryption, we have to have a way in. And there it is. I mean, it may be what we end up with.

**Leo:** Yeah. I suspect it will be.

**Steve:** I do, too. Okay. So Leo, two weeks ago you were spared my first...

**Leo:** PewDiePie, oh.

**Steve:** My first encounter with PewDiePie.

**Leo:** With Pewd, okay.

**Steve:** Yes. And I thought, what? PewDiePie? What the heck? So, okay. So we know that - we discussed it two weeks ago. A big fan of YouTube's sensation and top subscription magnet...

**Leo:** And anti-Semite and racist. Might as well throw that in, too, yeah.

**Steve:** Unfortunately. PewDiePie, whose Twitter handle - the fan's Twitter handle is @HackerGiraffe. He was concerned that PewDiePie, oh no, might be about to slip into second place, eclipsed by T-shirt. No, I mean, T-Series, which is in number two place.

**Leo:** Yeah. Because they're installed on every phone in India. So it's a kind of inorganic way to become number one.

**Steve:** Okay, yeah, that's...

**Leo:** And I think that's what's offensive to PewDiePie's fans.

**Steve:** That makes sense. See, Leo, you're bringing a lot more information to this than I had.

**Leo:** Oh, I know way too much about PewDiePie, believe me.

**Steve:** So @HackerGiraffe zipped over to Shodan to get the IPs of some exposed printers. This is a couple weeks ago. He reported then finding - and this is what we talked about two weeks ago - 800,000 printers publicly exposed on the Internet.

**Leo:** Oh, I remember this, yeah, yeah.

**Steve:** Of which he grabbed 50,000 IPs.

**Leo:** Geez, Louise.

**Steve:** He then used PRET, the Printer Exploitation Toolkit, freely available on GitHub, which gives hackers the ability to access files, damage the printer, or access the internal network. He used it to send out a mass plea for recipients of his illicitly printed page to please subscribe to PewDiePie. And while you're at it, please unsubscribe from T-Series. Which, as you said, if you're in India, apparently you already are.

**Leo:** Right.

**Steve:** Okay. So two weeks later, we're back. This time he grabbed 100,000 printer IPs, sent out another blast over the weekend, and in fact he tweeted that he would be taking a university exam at around the time Monday morning that employees will be arriving at work to potentially find the printouts. Now, I'm a little confused about time zones there because we know that these prints were received globally everywhere.

This printout says: "PewDiePie is in trouble, and he needs your help to defeat T-Series." Then under "What is going on," he printed: "PewDiePie, the currently most subscribed-to channel on YouTube, is at stake of losing his position as the number one position by an Indian company called T-Series that simply uploads videos of Bollywood trailers and campaigns. What to do? Unsubscribe from T-Series. Subscribe to PewDiePie. Share awareness of this issue. Tell everyone you know, seriously. Oh, fix your printer. It can be abused." And "BROFIST."

And I have in the show notes a screenshot of part of the printer page, for anyone who is interested. So anyway, this is, I mean, I certainly agree with point five, fix your printer. And as I mentioned two weeks ago, I can't vouch one way or the other for PewDiePie. I don't know why he has 75 million subscribers at this point. He doesn't seem to be in any great danger because T-Series is at 70 million. As I recall, two weeks ago, they were a lot closer. So unfortunately, I'm afraid that all of the attention that PewDiePie gained as a consequence of this printer blast does seem to have allowed him to pull well ahead of the T-Series folks. So anyway.

Oh, and The Wall Street Journal website was also defaced, as if that wasn't enough, saying that Wall Street Journal - and this is of course not true - Wall Street Journal would like to apologize to PewDiePie due to misrepresentation by our journalists, those of whom have now been fired. We are sponsoring PewDiePie to reach maximum subscribers and beat T-Series to 80 million. Well, that didn't happen. Wall Street Journal fixed it. But anyway, wow. The world we live in...

**Leo:** Just consider the kind of people he's attracting as his fans might tell you a little something about him.

**Steve:** Exactly. Border agents, Leo. Now, okay. So I should mention I saw this last week, the idea that border agents were looking at users' private data. And I thought, yeah, okay. I don't like it, but okay. It's going to happen. Then looking more closely, the other shoe dropped, which is that they are not demonstrating proper behavior with the private data that users who are crossing the U.S. border are being compelled to provide.

You know, in my case, I have absolutely zero contraband of any sort on my phone. But it's my phone. It's private, as are my conversations with friends and family. It's nothing happening. But it's, you know, I guess I feel a bit proprietary toward them.

**Leo:** The good news is you'll never be stopped because you're a white male.

**Steve:** Yes.

**Leo:** Wear a turban, see what happens.

**Steve:** It is true that the percentage of people who are stopped is very low. It's not like 100% of people are having their...

**Leo:** It's very, very low. And I've never had anything but welcome arms when I've come back into the U.S. I've never had - I've been faced with more searches going into Canada.

**Steve:** Yeah.

**Leo:** But that's because I'm a white, you know, male. It's who you are.

**Steve:** Yes. So anyway, there was a report from the U.S. Office of the Inspector General, the OIG, which was published by the Department of Homeland Security website last week, that our own U.S. Customs and Border Protection Agents are retaining the data that they copy from users' devices and not deleting it. So for the past three years, since 2015, when this Trade Facilitation and Trade Enforcement Act, TFTEA, went into effect, CBP, that's the Customs and Border Protection agents, are allowed to carry out warrantless device searches at all 328 ports of entry to the U.S. And so this means that agents are allowed to visually inspect any traveler's devices - smart phones, laptops, and so forth - without any reason. And they're supposed to look for suspicious content of any sort.

Okay, moreover, in 67 of those 328 ports, they're also allowed to copy, and I'd be interested to know what 67, like why? What makes those special? But in those 67, they're also allowed to copy the device data onto a USB thumb drive for uploading onto a search platform called the Automated Targeting System, ATS - yikes - on which more complex searches are carried out against the user's copied data. Now we'll put aside the fact that the license expired for the automated targeting system for about seven and a half months, during which time it wasn't available. But that got renewed, and apparently that little oversight has been fixed.

So according to this recently published OIG report, CBP agents have not been deleting user data from the thumb drives after they've uploaded the data onto this ATS. So the idea is the thumb drive is just being used as a shuttle to transport it to some other machine, where it's then uploaded. And standard procedure dictates that the data be deleted.

The OIG report said: "We physically inspected thumb drives at five ports of entry. At three of the five ports, we found thumb drives that contained information copied from past advanced searches which had not been deleted." So anyway, I just - it bothers me that anyone's data is being copied. And of course it's more worrisome or annoying that after being copied it's not being treated in a responsible fashion in a way that preserves the privacy of the people who just had us poking into their devices.

**Leo:** Don't worry. They're only saving your music. They don't - it's not the data they care about. They just want the music.

**Steve:** Oh, goodness. So we could talk about malicious Android apps till the cows came home. This one just sort of stood out because it reminds me that asking users to be ever attentive while they're using apps is probably a fool's errand. ESET posted their discovery of an Android trojan which steals money from PayPal accounts, even for users that have two-factor authentication turned on.

**Leo:** To your point, your phone knows all that stuff.

**Steve:** Well, exactly. And so what they discovered in November, last month, was malware that combined the capabilities of a remotely controlled banking trojan with a novel misuse of Android's accessibility services, which targets users who use the official PayPal app. So this app is called Optimization Android v1.0. It pretends to be a battery optimization app, does nothing with your battery. After it's launched, it terminates without offering any functionality and hides its icon. So it just basically sort of disappears from your phone. It's now a trojan.

From then on, it watches what the user does with the primary goal of stealing money from victims' PayPal accounts. It requires the activation of what turns out to be a malicious accessibility service. The request is presented to the user as being from an innocuous-sounding Enable statistics service. So a dialogue pops up that says: "Use Enable Statistics?" And then it says: "Enable Statistics needs to, one, observe your actions. Retrieve notifications when you're interacting with an app. Two, retrieve window content. Inspect the content of a window you're interacting with." And so you've got the "okay" or "cancel." So users say okay.

Well, what's just happened is you've given this trojan now access to your screen and keyboard, essentially, which it abuses. It's now on the machine and able to intercept any launch of the Android PayPal app and attempts to maliciously transfer a thousand euros or whatever the local currency is whenever the user logs into PayPal. The use of two-factor authentication doesn't prevent this since the trojan simply waits for the two-factor authentication to complete and the logon to succeed, at which time it jumps in using essentially an alternative API, an alternative UI, that allows it to talk to the PayPal app on behalf of the user, abusing accessibility in order to transfer the funds.

ESET has notified PayPal of the existence of the trojan and of the PayPal account to which funds were being transferred. And the hack will fail only if the user has no balance and no card connected to their PayPal account from which funds could be pulled. So stepping back from this, I think our takeaway is that we are asking for all kinds of features from our smartphones. They've become very smart. And we've seen, I mean, this is just not an Android problem. We've seen instances of this on iOS also where the addition of important features such as those required in support of accessibility can be abused in clever ways that was never their intent.

So the safe use of desktop PCs requires everything we're seeing. The safe use of desktop PCs requires some care about what apps are loaded and from where. So much so that even Microsoft is now moving toward this idea of a store where there will be curated apps that, rather than just this free-for-all of downloading things for Windows from wherever, although Microsoft is having a hard time selling that concept because we all like the freedom that we have now as Windows users.

But I don't think it's fair to ask users to be constantly vigilant in their actions as they use apps. I think it is the case that the only solution is to keep these apps from getting into our machines in the first place. So really the location where we need vigilance is in not installing these things capriciously, I mean, being skeptical of the apps that we install. And of course one of our favorite slogans on the podcast is never install something that something tells you you need. Only install things that you go looking for. So anyway...

**Leo:** Of course those are on third-party app stores, so that's the other thing is to stick with the Play Store.

**Steve:** Yes, yes, yes, yes, yes.

**Leo:** Not that there hasn't been malware on there. But they have to go through a lot more hoops to get that stuff on there.

**Steve:** Right. And in fact this was on the Play Store.

**Leo:** Was it? Oh, I thought it was a third-party store only.

**Steve:** I think, the screenshot I saw I think showed it on Google Play, but I'm not sure.

**Leo:** Okay.

**Steve:** So, but yes, you certainly only get things from Google Play. WordPress just released v5 of their number one website blogging system. And there was some confusion surrounding it. First of all, I thought it was kind of fun. I didn't realize until I saw the name of it and then dug in a little bit that as they have done since v1, which was the Miles Davis release back on January 3rd of 2004, they've named their releases after famous musicians. Version 2.0 was Duke Ellington; 2.1 Ella Fitzgerald; 2.2 Stan Getz. They had John Coltrane, George Gershwin, Count Basie, Charlie Parker, Bennie Goodman. Anyway, 5.0 is the Bebo Valdes, who is, they say, a pioneering Cuban jazz musician.

What was confusing is that in the press just last week, late last week, was the news that, as a consequence of this major v5 release, Google could index users' passwords. And so it's like, whoa, wait a minute. Okay. So looking closely at this, I wanted to sort of, well, first of all, give people a heads-up that there is an important security release, but there is one for v5.0.1. But it had actually nothing to do with this big release of v5. There is a new release of 5.0. If anyone's interested, the big new feature is they have a completely brand new, extremely flexible block-based editor. And is WordPress still a sponsor of the...

**Leo:** They sure are, WordPress.com, yeah.

**Steve:** Yes. So I did want to mention that.

**Leo:** Now, if you're on WordPress.com you don't even have to think about this because they just keep it up to date automatically.

**Steve:** Right.

**Leo:** It's only if you're running a self-installed version of WordPress.

**Steve:** Correct. So the truth is that, although the headlines made this look like this was a problem with 5.0, in digging deeper into this, it looks like the release of 5.0 was asynchronous to a regular security release where WordPress fixed seven different problems. One of the seven was that in an unusual circumstance which required nonstandard configuration that probably actually nobody would ever encounter in reality, it was possible that a search engine might index the initial setup screen where an email address and a default password could be found. But it was like, you know, the tech press really went clickbait overload on this.

So but the important thing is that there was a week after - the release of 5.0 was on December 6th. On December 13th, a week later, was a security release announcement. And it does affect 5.0 just because it affects them all, but it also runs all the way back to the Count Basie release, which was v3.7 back in October of 2013. So for the last five years they have a security update for everything. So I did want to give our listeners a heads-up, as you said, Leo, for people who are running their own WordPress installations, that's it just a standard security update that affects all versions for the last five years.

So despite the way the press handled it, it was not something devastating that, first of all, was even likely to happen for anyone using v5. It just happened to be that they released this security update a week after their major release. The security fixes, as I said, were comprehensive. And for what it's worth, it does look like a nice update to WordPress.

I know you've talked about the Facebook Photo bug. This again is an instance of I think Facebook doing a good job. Last Friday they announced another security incident affecting millions of their customers. For the 12-day period from September 13th through the 25th, a bug existed in one of Facebook's APIs which exposed the private photos, or I should say potentially exposed the private photos of nearly 6.8 million users. Their forensic analysis of the bug revealed that up to around 1,500 apps, which were built by 876 developers, could have had access to the private photos of these 6.8 million users.

I have here in the show notes a link to Facebook's posting. A Tomer Bar is the Facebook employee who last Friday posted under the title "Notifying Our Developer Ecosystem About a Photo API Bug." Facebook was completely forthcoming in this. He basically stated the things I've just stated. And remember that last week's podcast was titled "Internal Bug Discovery." So I will note that, despite Facebook's current residence being a doghouse from their past conduct...

**Leo:** Good way to put it.

**Steve:** Yeah. They, like Google, clearly have an internal red team of some sort whose job it is to find their own mistakes before anyone else does. Rather than faulting them for making a mistake, I salute them, as I did Google last week, for finding and fixing it themselves. This is not a role that anyone wants to outsource to hackers. So he explained that the nature of this was that: "When someone gives permission to an app to access their photos on Facebook," he writes, "we usually only grant the app access to photos people share on their timeline. In this case, this [12-day window] bug potentially gave developers access to other photos such as those shared on Marketplace or Facebook Stories.

"The bug also impacted photos that people uploaded to Facebook but chose not to post. For example," he says, "if someone uploads a photo to Facebook but doesn't finish posting it, maybe because they're lost reception or walked into a meeting, we store a copy of that photo for three days so the person has it when they come back to the app to complete their post."

He said: "Currently, we believe this may have affected" - and then, you know, those numbers and so forth. And he finishes, saying: "We're sorry this happened. Early next week we'll be rolling out tools for app developers that will allow them to determine which people using their app might be impacted by this bug. We will be working with those developers to delete the photos from impacted users." And presumably those apps who use Facebook's tool would be able then to notify their users that they may have been affected by this app or not, depending.

He says: "We'll also notify the people potentially impacted by this bug via an alert on Facebook." Again, so I think they're being completely responsible and acting the way we need our large Internet providers of services like this to act. As a society we are building ever more complex and capable systems, and these systems will be imperfect. I fully expect that some day, because this is science, this is math, we will figure out how to economically make these systems far more correct than we know how to today. But until we get there, the best we can do is find the problems and fix them as they arise. And so here's another example of I think it being done right.

And speaking of monkeys, Leo, oh, the passwords we use. SplashData, the publisher of the password management applications TeamsID, Gpass, and SplashID, has just released their eighth annual list of the Worst Passwords of the Year. Appearing for the first time on the list in 23rd place is "donald," which we haven't had before, and we can imagine the source for that password. Somewhat sadly, our longtime podcast favorite "monkey" has slipped five places, down to 18th most popular; while "princess" is debuting for the first time in the 11th slot.

So what have we? The top two, surprisingly, have not budged. The password 123456, ranked top last year, still in first place this year.

**Leo:** I was going to use "princessdonaldmonkey," but I guess I can't now.

**Steve:** Oh, well, you know, when you combine words...

**Leo:** It's better.

**Steve:** ...you can get a better password. So yes.

**Leo:** 123456princessdonaldmonkey.

**Steve:** That would be good. Well, no. Don't anybody use that. Okay. In number two place, yes, hasn't moved, "password."

**Leo:** Oh, man.

**Steve:** I know. Can you believe?

**Leo:** But if I spell it with a zero instead of an "O," is that okay?

**Steve:** Oh, in fancy h4xx0r fashion?

**Leo:** In h4xx0r fashion. In h4xx0r fashion.

**Steve:** So, and we also have the slightly less popular, yet growing in popularity for those who are slightly less lazy, 123456789.

**Leo:** Oh. Now we're talking.

**Steve:** So you add, yeah, you add that 789, it does take a little bit longer. That's moved up three into third place this year, while 12345678 has dropped one rung to fourth place. Apparently it got kicked out by 123456789's having moved up to number three. Wow. There is real competition on these number ones.

**Leo:** Tight fight, yeah, mm-hmm.

**Steve:** It is, you know, you never know. And in fact 12345, just five digits, retains its fifth place position. So you can always remember, 12345 is the fifth most popular password.

**Leo:** Easy to remember.

**Steve:** Wow. 111111 is new, has just appeared on the list.

**Leo:** Oh. Oh.

**Steve:** Yup. Oh, 1234567, not 5, not 6, not 78, not 789, but just 7, has moved up one. "Sunshine," how happy, sunshine has just appeared on the list. "Qwerty," Q-W-E-R-T-Y, we know where that came from, that has dropped five because it's been pushed down because some of these number runs have apparently run up. "Iloveyou," unchanged, in 10th place. That's nice. "Princess" has appeared in 11th. "Admin" dropped down one.

"Welcome" also down one. 666666, newly appeared. Believe it or not, among all this jumbling, abc123 still in 15th place, just sitting there, not budging. Everything's churning around it, princesses are popping in, and sunshine is new. Abc123 holding at 15.

"Football" dropped down seven. 123123 also holding at 17. And as I said, sadly, "monkey" has dropped down to 18. For someone who's getting really clever, 654321 in 19th place. And then we have the, wow, how could this actually be - oh, I see. You hold the shift key down. It looks like cartoon swearing: !@#$%...

**Leo:** Let me guess the next one. Caret?

**Steve:** Yes.

**Leo:** Ampersand? Asterisk?

**Steve:** You're a h4xx0r. Like, how did you do that? Amazing. The great Kretzkin is among us.

**Leo:** Oh, man.

**Steve:** Yes. And Leo, it's new on the list, that one you just guessed all by yourself. New.

**Leo:** Well, welcome.

**Steve:** "Charlie" is new in the 21st position. And here we have something really clever. This is like one you were going to suggest: aa123456. And what's amazing to me is that not one person did it, but that's the 22nd most popular password. Twenty-third, "donald." Twenty-fourth, "password1." Okay, that's not very imaginative, but it has just appeared. And then, for those who - maybe a browser, oh, no, I was going to say maybe a site objected to "qwerty," but no, because that's the ninth most popular. So "qwerty123."

**Leo:** Oh, there's improvement, yeah.

**Steve:** Rounds out the list in the 25th position.

**Leo:** Nice. Strong.

**Steve:** So SplashData - I wanted to say "Slapdash," but no. SplashData estimates almost 10% of people have used at least one of the 25 worst passwords on this year's list. So 10% - no, certainly zero of our listeners, none of our listeners, except maybe "monkey," but actually we know that one of our listeners has used "monkey" in the past. And apparently nearly 3% of people have used the worst password: 123456. Not even going as far as adding a 789, Leo, which would have made it really not much harder to get.

**Leo:** Where do they get these?

**Steve:** This is from leaked lists of passwords, which they aggregate over the course of the year and then sort and then provide for our entertainment on the podcast.

**Leo:** They take the adult words out.

**Steve:** Oh, oh, yes, yes. I was just going to say, and I have that in the show notes here. We should note that passwords leaked from hacks of adult websites were not included in this report, doubtless being NSFW. That would make for a different list.

So, oh. Bleeping Computer had two stories last week that I thought were very interesting, and I wanted to absolutely put it on our listeners' radar. Just because some malicious drive encryption is done right, and because it would be trivial to do it right, doesn't mean that all of it has been done right. Far from it, apparently. There were two stories that Bleeping Computer had. One was titled "How to Decrypt the InsaneCrypt or Everbe 1 Family of Ransomware." I've got the link in the show notes for anyone who's interested.

And they wrote: "InsaneCrypt or the Everbe 1.0 Ransomware is a family of ransomware infections that were based off an open source project. This ransomware family is distributed through possibly spam and hacking into Remote Desktop Services, but that has not been confirmed.

"The good news," they write, "is that the variants of this ransomware family can be decrypted for free using a decryptor created by Michael Gillespie and Maxime Meignan. In order to use the decryptor, a victim just needs to have" - I thought this was really interesting - "an encrypted file and an unencrypted version of the same file." This has got to be the lamest encryption [crosstalk].

**Leo:** Got to be. You can reverse the hash, in other words.

**Steve:** Right. This can be typically achieved through the sample pictures provided by Windows. I thought that was clever.

**Leo:** Oh, yeah. Those will be encrypted, and you'll be able to get another copy of those easily, yeah.

**Steve:** Exactly. The second story: "How to Decrypt Hidden Tear Ransomware Variants." They said: "If you've been infected by a Hidden Tear Ransomware variant, then you're in luck" - well, kind of. But, you know, I mean, it's never good to be encrypted at all. But given that you have been, this is the one you want to be encrypted with because it's not really, apparently. "You're in luck," they say, "as a program called Hidden Tear Decryptor has been created [also by Michael Gillespie, who apparently does this] that allows you to recover your encryption key without having to pay the ransom."

Hidden Tear is the name of a ransomware family whose full source code, apparently not very well engineered, was published on GitHub. This allowed hackers to download the source code and create their own ransomware variants that could be used to infect victims. Due to the source code's wide availability, there are many ransomware variations under different names that utilize the same Hidden Tear codebase. As the original code was decryptable, this means all other ransomware created from the same code are decryptable, as well.

Some, get this, some of the Hidden Tear variants supported by this tool include: 8lock8, AnonCrack, Assembly, Balbaz, BankAccountSummary, Bansomqare Wanna, Blank, BloodJaws, Boris, CerberTear, CryptConsole2, CryptoKill, CyberResearcher, Data_Locker, Dev-Nightmare 2xx9, Diamond, Domino, Donut, dotRansom, Executioner, Executioner2, Executioner3, Explerer, FlatChestWare, Frog, one I can't say on the podcast, but we're in the F's.

**Leo:** In the F's, yup, yup.

**Steve:** You can imagine, uh-huh. Gendarmerie, Horros, JobCrypter, Jodis, J-Ransomware, J-Want-To-Cry, Karmen, Kraken 2.0, Kratos, LanRan, Lime, Lime-HT, Luv, Matroska, MireWare, MoonCrypter, MTC, Nobug, Nulltica, onion3cry, OpsVenezuela, Paul, PayOrDie, Pedo, PGPSnippet, Poolezoor, Pransomware, Predator, Qwerty, Random6, Random6 2, Randion, Ransom - anyway, it just keeps going. We're only to the R's. So the point is there's a lot of this stuff out there. First of all, who knew? But secondly, there's a very good chance, if you get hit by Donut or PayOrDie, or Kraken, or Sorry - there was one called "Sorry" - then by all means you, your friends or family, check in with Bleeping Computer because, as we know, they're on top of this stuff. Maybe there's a simple way to get yourself decrypted, which would be wonderful because apparently there's a lot of poorly written crypto malware out there, thanks to there being a dumb one on GitHub. So yay. Maybe that's pushing the bad stuff out of the way.

I mentioned at the top that Microsoft is encountering a run of zero-day vulnerabilities. Last week's Patch Tuesday closed a zero-day vulnerability that was being actively exploited by two different state-sponsored cyberespionage groups who are also behind the zero-day that Microsoft patched the month before in November. And before that, in October, Kaspersky Lab discovered a zero-day being used for an elevation of privilege by the FruityArmor state-sponsored cyberespionage group.

**Leo:** What state is that, FruityArmor?

**Steve:** I know, I know. Remember, I don't think they name themselves.

**Leo:** Oh, okay, good. Oh, that's a relief.

**Steve:** You know, they're like names we give them. And it's annoying when the same group is known by 12 different names, as we've seen before. So in this case, yes, it's FruityArmor. And before that, in September, Microsoft patched a zero-day that was in use by hackers who were using it to install and spread a backdoor. So mostly these appear to be high-end, highly targeted, state-level espionage attacks. Their discoverers, or the discoverers of these zero-days, don't want to have them discovered. I mean, the users of these zero-days don't want them discovered because they're too useful for targeted attacks. So they are not being used in widespread campaigns. You know, they're really being kept under the radar. We don't know, when they're found, how long they've been in use.

And given that two different zero-days were both discovered and fixed in the most recent Patch Tuesdays, one wonders how many as yet undiscovered zero-days are also in use because, again, we don't know that they are until they're found. And they're not being sprayed around the Internet. They're not being tweeted by hackers who are having a bad day. They're being developed by the likes of the NSA and other countries' equivalent services for the purpose of getting into other people's computers.

So this spate of zero-days affecting Windows, our increasingly old and creaky operating system, is a bit worrying. Windows is coming under increasing pressure, and it's not holding up as well as we would like or hope. And in fact, Leo, I listen to you with Paul and Mary Jo on Wednesdays, and it's a little sobering.

**Leo:** Oh, yeah. Oh, yeah.

**Steve:** They're not happy any longer.

**Leo:** Unh-unh.

**Steve:** They're really saying, wow, you know, what the heck is going on up there in Redmond? Okay. Our last little bit of news is that Firefox 4 was released. An interesting note is that Firefox's full distrust of the Symantec SSL/TLS certificates had been postponed from the previous release of 63 until finally now, Release 64. Remember that Firefox brings along its own certificate store. Unlike Chrome, for example, which relies on Windows' underlying root store, Firefox does its own. It uses its own security suite. And so it's up to them when they were going to finally roll out full mistrust, that is, no longer accept a certificate from any website using those certs. They postponed it because up until finally now, their telemetry had indicated that too many Firefox users were still encountering websites that were still using the Symantec certs. So, boy, I can't imagine why anyone wouldn't just re-up with DigiCert. And finally I guess everyone has. So there's that change.

Also there is something known as a Contextual Feature Recommender, which is a bit of a tongue-twister, the CFR. The good news, you can turn it off. But the idea is that - and if any of our listeners who are Firefox users are annoyed by this, you can turn it off. It's a new feature that will display recommended extensions that correspond to a particular website you're visiting. So, for example, if you're at Facebook, there is a Facebook Container extension which this thing will say, hey, you know, there is an extension to Firefox that's available for Facebook. There's also an enhancer for YouTube and To Google Translate. You can, however, if you don't like that, uncheck the "recommend extensions as you browse" under the Browsing category of Firefox options. So, eh.

Also a new feature is Multiple Tab Selection. You can either mark a range of tabs with shift, or toggle tabs to select multiples with control-click. And once chosen, you can bookmark them as a group, move them, or pin them. Also they've added Native Windows 10 Share Support so that you're able to - I guess Windows 10 has a web page sharing with a variety of applications installed. And so to access the feature, you click on the little dot dot dot menu in the address bar, then click on the share button, which will open the native Windows 10 sharing dialogue. Not being a big Win10 user yet, I haven't encountered that myself.

A little controversial for old folks, Live Bookmarks and RSS Feeds have been removed. They've removed support for both Atom and RSS feed subscriptions from the base browser. So anyone who was actively using RSS feeds will simply need to install an add-on to provide the feature. So you can get it back, it's just not built into the default. And I also talked about a Remove Extension context menu option. I'm sure I mentioned this before, so I must have been talking about it coming. It allows you, if you right-click on any of the Firefox add-on extensions, in the dropdown menu is now a remove extension that makes it just instant to get rid of it.

And my favorite add-on is you can get to it by putting about:performance in the URL bar. That gives you a task manager that shows which tabs are using how much of your processor. So if you have a misbehaving tab which has slowed things down or seems to be consuming a lot of resources, about:performance will allow you to identify it and tell it to knock itself off or just kill it.

**Leo:** Steve Gibson is going to sing for us.

**Steve:** Oh, no.

**Leo:** Oh, no.

**Steve:** So I'm not sure how I'm going to handle that. But first, first Rails, R-A-I-L-S. I was reminded of it. It's been years since we talked about it. I fell in love with this game for my iPad. It's also available on Android and for the Mac platform.

**Leo:** Oh.

**Steve:** Yes. And it's just a great game. It's sort of a puzzle game. Stations are placed on the board. You then interconnect them using a rail outlet with switches of your own design. And then trains are emitted from the stations, and you're responsible for flipping the switches in order to route the trains around to keep everything moving. And it's just...

**Leo:** I haven't played it in ages. I just loved it. We both loved it, I remember.

**Steve:** Yes, yes, yes, yes. It's great. So I just wanted to put it on - it's in time for Christmas, so maybe you could give somebody who has a phone or a pad something fun to occupy themselves with. I mean, and again, all ages. And that video you're showing right now, those are some beautiful track layouts.

**Leo:** Yeah. I have certain patterns that I like to use. But I have to say it gets harder and harder.

**Steve:** Yes, yes. Well, the trains get longer, and it's like...

**Leo:** And things happen, and trains break down, and you just get - I didn't realize, they've added snow, which I like. That's nice. They've got some seasons in there. That's...

**Steve:** Different seasons, yeah.

**Leo:** It's called Rails, R-A-I-L-S.

**Steve:** Rails, yeah. There are a bunch of cheesy ones, so be careful. You want...

**Leo:** There are. This is a category, I guess.

**Steve:** Yeah. Okay. So Kevin Sears, a listener from Eagle, Wisconsin, offers us the Holiday Hard Drive Song, he says, a winter holiday remake sung to the tune "Let It Snow." So I'm not going to sing. But so he renamed it "Let It Go." And he said: "Oh, an aging hard drive is frightful."

**Leo:** [Singing] "But my photos are so delightful. Got no plan you know so, I let it go, let it go, let it go. It doesn't show signs of stopping, 'cause error correction is working. The green light is still on so, I let it go, let it go, let it go. The drive wants to say goodnight, but everything seems to be all right. Bits are aged five years or more, seems slower than it did before. The spinning surface is dying, and S.M.A.R.T. says it's goodbying. I think I backed up a year ago so, I let it go, let it go, let it go. Now bits are just ones or zeroes, trying hard to be our data hero. But soon they'll be unreadable - NO! 'Cause I let it go, let it go, let it" - dot dot dot dot dot.

**Steve:** And it died.

**Leo:** And it died.

**Steve:** And so have our listeners, I'm sure.

**Leo:** I had to do that. You can't just read it. You've got to sing it.

**Steve:** I was discussing this with Lorrie last night, and I thought, I can't do this. And then it occurred to us, oh, but Leo could.

**Leo:** Leo will do it.

**Steve:** So Leo, you stepped up. Thank you for that.

**Leo:** Let it go, let it go, let it go.

**Steve:** Or in the case of SpinRite users, just run SpinRite on your drive and keep those data bits happy.

**Leo:** It's very true, really. I mean, I think it's a good lyric. Nicely done. Well done. Sorry about the singing.

**Steve:** No, no, don't apologize. Someone had to do it, Leo. Better you than me. Mike S. in New Hampshire, his topic was router infection. He said: "I have a Netgear router, an R6300v2, if you care. I routinely clicked on the Router Update > Check for new version link. It told me that there were no updates available. Last week I started seeing very slow response from WiFi, so I checked again, and it said the same thing. I decided to check with Netgear, and their website had updates."

**Leo:** Ooh.

**Steve:** Uh-huh. "I downloaded the update and applied it, and the performance went back to normal. So far it's been good." He says: "I'm thinking that some malware got into it and wouldn't let it check for updates itself."

**Leo:** Oh, interesting. Ah. That makes sense.

**Steve:** Yes, yes, yes, yes. Yes, it totally does. That's what you would expect something nasty to do because it didn't want to get itself flushed away.

**Leo:** Right.

**Steve:** So anyway, I just wanted to share that with our listeners. That's a very good point is that, when you think about it, you can't actually trust the internal "check yourself for updates." It's worth going to the manufacturer's website and verifying. So Mike, thank you very much for that little tip. I'm glad to pass that on to our listeners.

And Chris Peterson in Ham Lake, Minnesota, the subject of his note was "It's even simpler." And I have to agree with him. He said: "Regarding the unauthenticated UDP IoT protocol." That was the CoAP protocol we talked about last week. He said: "Since it's IoT, we want to minimize processing and storage." Agreed. He said: "The IoT device only needs to return a pseudorandom number to the requestor." Okay, now let's step back a bit. Remember, what I suggested last week was that the IoT device could come up with a random key under which it would encrypt the incoming IP to create an access token, which it would return to the requestor, such that every subsequent request coming in had to carry that permission token in order to be serviced. So that was my suggestion.

Chris has a better one. And this is better. He says: "This can be of poor entropy," meaning this random number. So he says: "The IoT device only needs to return a pseudorandom number to the requestor. This can be of poor entropy. The IoT device is then paired with the requestor. Any further requests from that IP must include the same PRN" - the same pseudorandom number - "or they're rejected. A request from another IP resets the pairing, and the process is repeated. We can do this because spoofing requires a response to a fake IP address. This fake address, the victim, won't be responding. A short pseudorandom number of 32 bits or less should be sufficient, thus minimizing traffic sent to the victim. Obviously not 'perfect' security, but it's 'good enough' security, and way better than what they're doing now." And he said "73, Chris, K9EQ."

Okay. So this is clever. My solution was overdesigned because your typical IoT device, first of all, it's not fielding stuff from all over the Internet. My solution would have allowed it to do that, to have no state saved in the device, always returning a unique token based on the source IP of the incoming request. So as IPs all over the place asked for permissions, it would dutifully return the token for them. But that's not the model. Your typical IoT device is going to talk to one thing. And if there were two or three, well, over time this would cause it to reissue a random token and then essentially establish a temporary one-to-one pairing with the most recent requestor of a connection to the IoT device.

Anyway, he's right. No crypto. Simple PRN that you could just take, I mean, I think most chips will allow you to read the number of clock cycles since they've powered up. That would be completely fine. Just send that back and save it along with the IP address so you can check it in subsequent requests. Anyway, Chris, tip of the hat to you. That's simpler and, I think I agree, better. And again, much better than the nothing that we have now in this dumb CoAP protocol that somehow happened when it shouldn't and is now being abused for, what is it, times 50 DDoS reflection attacks, which is just crazy.

Okay. So a critical SQLite flaw which leaves millions of apps vulnerable. I should say "app instances." The Blade Group at Tencent has named this "Magellan." And they write - I've got a link to their full disclosure posting. They said: "Magellan is a remote code execution vulnerability discovered by Tencent's Blade Team that exists in SQLite." And I want to remind everyone again, this is probably in all of our machines. I had four instances in my machine, and that took me somewhat by surprise. So I'll explain what it is, and then I'll explain how to find it.

Okay. So they say: "As a well-known database, SQLite is widely used in all modern mainstream operating systems and software, so this vulnerability has a wide range of influence. After testing, Chromium was also affected by this vulnerability. Google has confirmed and fixed this vulnerability. We will not disclose any details of the vulnerability at this time, and we are pushing other vendors to fix this vulnerability as soon as possible."

So what we have now is the knowledge of a problem, the confirmation of the problem. In fact, there was a page on the Internet which is now kind of irrelevant that allowed you, for Chrome 70, the Chrome browser 70, you could crash any tab by clicking it because it would invoke SQLite that the Chrome browser was exposing to any website that you went to. So that's bad. This was a remote code execution vulnerability.

So what they have is simply a Q&A at this point. Am I affected by the vulnerability? Answer: "If you use a device or software that uses SQLite or Chromium, it may be affected, depending on whether there is a suitable attack surface." And of course that's important. The malware has to have a means of executing an SQL command against the database which is present. So it's the embedded database itself where the vulnerability exists. As I mentioned, my email client, Thunderbird, uses SQLite. And of course email is an attack surface. So that's pretty much all it is.

Question two, what is the danger of this vulnerability? "Remote code execution, leaking program memory, or causing program crashes." Three, does the vulnerability have exploit code? They say: "Yes, we successfully exploited Google Home with this vulnerability, and we currently have no plans to disclose exploit code." That is, everything needs to get fixed. Four, what are the conditions for exploiting this vulnerability? They answer: "This vulnerability can be triggered remotely, such as accessing a particular web page in a browser or any scenario that can execute SQL statements."

Five, has Magellan, which is what they call this, been abused in the wild? Their answer: "We have not seen a case of it yet." Six, is there a fix or workaround? And the answer: "We have reported all the details of the vulnerability to Google, and they have fixed the vulnerability. If your product uses Chromium" - and remember there are many other browsers that do - "please update to the official stable version." And so in the case of Chrome it's 71.0.3578.80. And I already have one, my Chrome is a little bit further than that. They said: "If your product uses SQLite, please update to 3.26.0. The CVE number is pending."

So on the 14th, late last week, there was a Crash Chrome 70 with the SQLite Magellan bug. That was at WorthDoingBadly.com. That's obsolete now because, as we know, Chrome is at 71. You want to make sure, though, that your Chromium-based browser is safe. So for what it's worth, it might be worth going to WorthDoingBadly.com and trying this. The site is WorthDoingBadly, all one word, dot com slash sqlitebug. So WorthDoingBadly.com/sqlitebug, and make sure that your browser doesn't crash.

Okay. So SQLite is a library, essentially, that many different things use that, for whatever reason, need to store user data, and even things that I'm not quite sure why they're storing user data. I found it in four places on my system. It was in NetWorx, N-E-T-W-O-R-X, which is an app I've mentioned and I use. I love it. I have it running on my screen all the time because I have it using SNMP, Simple Network Management Protocol, to continuously check the counters, the byte counters on my pfSense router to watch the incoming and outgoing bandwidth. I just kind of like to get a sense for data coming and going in and out of my network so if something seems anomalous, I can go, okay, wait a minute, and go check why a huge amount of data is transiting.

Python, my installation of Python brought sqlite3.dll along. And Adobe Reader - shock - is vulnerable with sqlite.dll. Adobe Reader, even though mine is 9.something, which is not super recent, but it had v1.0.0.1 of sqlite.dll dated 2/27/2009. Also my favorite PDF printer, which is Nova PDF, that has a .NET version, sqlite.interop.dll, also vulnerable, v1.0.82.0 dated 6/8/2018. So that's not very old, but it's not yesterday.

Okay. So anybody can find these who is a Windows user by changing your directory to the root. So do cd\ to go to the root. And then you want to do a recursive dir, a recursive file search for sqlite*.dll. So cd\ to move to the root, and then dir /s, which causes a search through all subdirectories, space sqlite*.dll. Hit Enter, and pretty quickly you should be rewarded with a list of all the files that begin with sqlite. You'll find sqlite.dll, probably sqlite3.dll, and maybe some that have interop.dll, which are the .NET variety.

So this doesn't immediately mean that you're in trouble. What it means is that those apps are using this embedded database for their own storage purposes on your system. It means that the API, that is, the SQL API has a vulnerability that could allow code to be injected on your system and run. The danger - now, for example, Adobe Reader, there's the danger, Will Robinson. For some reason, who knows what it's doing, but of course why wouldn't Reader have SQL built into it because it has everything else, Flash and SQL and oh my god. And JavaScript and stuff that no one needs. But it's there. So if there's a means for a document to invoke this SQL embedded database in Reader, that's a worry because that represents an attack surface.

So I haven't looked at why my install of Python has SQLite 3. Probably it's just that it supports that database engine API, so it's part of the Python library. It doesn't mean that it's itself vulnerable, or maybe it's part of the IDE. Who knows? Anyway, so it's probably worth just doing a little audit of your system by recursing through your drive for sqlite*.dll, see what you find, and then maybe, if it looks like there might be an attack surface exposed, like I would be worried about my Adobe Reader very much. I'll probably actually pursue this over the holidays and have some information at the beginning of 2019 just because that seems like an obvious attack surface. And it might be the fact that I'm deliberately using an old version because I don't want to be the SaaS, sign up and join the Adobe fan club, so I'm using one that's entirely offline. I think it's 9.5 was the last one that was not hooked into the cloud.

But anyway, the SQLite folks immediately fixed this. They have an update. The problem is that that doesn't mean that the libraries that we use are going to get updated or in a timely fashion. If it turns out that there are means of leveraging this into an attack surface, we're probably going to see this in coming weeks and months next year. So I wanted to put it on everyone's radar. It's easy to audit your system and get a sense for what apps you have installed which are using this embedded database.

Again, it doesn't mean it's a problem. It's only a problem if there's a way for a malicious use of that through the app. So it may be Python is a problem. But Python's an example. I mean, so, for example, the SQLite 3 in my Python is v3.21, relatively recent, dated 6/27/2018. Probably it's already been updated, or it will be shortly. So that kind of thing can be fixed. I guess I would say, where possible, where you see that an app is using SQLite, maybe give them a few weeks to catch up, to get their use of it updated, and then update the app so that you're not vulnerable. Anyway, I don't want to...

**Leo:** Scare anybody?

**Steve:** Yes, I don't want to overstate this. But it could become a popular means of attacking people's systems.

**Leo:** It's everywhere. It's everywhere.

**Steve:** The good news is - it is everywhere. The good news is that it absolutely requires access to SQL. There is a web SQL interface. That's what Chrome supports. Firefox uses SQLite, but they weren't exposing the interface. Chrome was. So actually it allowed JavaScript to issue SQL commands to an embedded database on your system. And that was the way in. So, briefly, Chrome could be exploited with this. No longer. So you want to make sure that, if you are a fan of a Chromium-based browser, that you get yourself patched. And again, I'm sure that anyone who is producing a browser already has updates and fixes, just as Google did instantly with Chrome.

**Leo:** Yeah. My Chrome is safe.

**Steve:** Yup.

**Leo:** Good to know.

**Steve:** And Leo?

**Leo:** Actually, this is Safari. Safari's safe, too. Good.

**Steve:** Not only is this the end of the podcast.

**Leo:** The end of the year.

**Steve:** This is the end of the year.

**Leo:** We're done. Next week is Christmas Day, and we're going to have a "Best Of."

**Steve:** Cool. Cool.

**Leo:** Some great stuff. We've never done this before with Security Now!, but I think it's going to be fun. New Year's Day we will do the same because there's no point in - but we'll be back January 8th for a whole new season of Security Now!. Lots of new stuff. There'll be exploits galore.

**Steve:** Who knows?

**Leo:** Who knows?

**Steve:** This same podcast last year, the last podcast of the year in 2017, we would have never predicted that 2018 would have been the year of a disaster of the Intel chipset. And other processors, too. Not fair just to pick on Intel. But, you know, every processor architecture collapsed this year.

**Leo:** Yeah.

**Steve:** Who knows what we have in store for us for next year?

**Leo:** Something wonderful.

**Steve:** We'll be here. We'll be here.

**Leo:** And I am relieved. Thank goodness Steve's here. Find everything Steve does at his website, GRC.com, including SpinRite, the world's best hard drive maintenance and recovery utility; and this show and transcriptions thereof. GRC.com. He's on Twitter. If you want to leave him a message or ask him a question: @SGgrc, or GRC.com/feedback. Do people still use that? Or they just use Twitter probably.

**Steve:** Oh, yeah, yeah. That's where I got both of the notes that I shared, yes.

**Leo:** There you go, GRC.com/feedback. You can also get audio and video of the show from us, TWiT.tv/sn. Subscribe in your favorite podcatcher, you just get it automatically and download it the minute it's done. Collect all 694 episodes. Make sure your collection is complete. Steve, have a great holiday. Have a good two weeks off.

**Steve:** Same to you, my friend. I'm going to enjoy a little time off, and we'll be back to kick some security butt here.

**Leo:** Yeah. Thank you. We'll see you next time on Security Now!.

**Steve:** Thanks, buddy.