SECURITY NOW!

Transcript of Episode #629

Apple Bakes Cookies

Description: This week Padre and I discuss what was up with SN's recent audio troubles, more on the Equifax fiasco, the EFF and Cory Doctorow weigh in on forthcoming browser-encrypted media extensions (EME), an emerging browser-based payment standard, when two-factor is not two-factor, the CCleaner breach and what it means, a new Bluetooth-based attack, an incredibly welcome and brilliant cookie privacy feature in iOS 11, and a heads-up caution about the volatility of Google's Android smartphone cloud backups.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-629.mp3

Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-629-lq.mp3

SHOW TEASE: It's time for Security Now! with Steve Gibson, and we are laughing in the face of security armageddon. Equifax protected your security with an executive who didn't know anything about security, the EFF walked out on the W3C, browsers get payments done right, your Bluetooth devices are spying on you, and Apple gives you a cookie. Security Now! is next.

FATHER ROBERT BALLECER: This is Security Now! with Steve Gibson, Episode 629, recorded September 19th, 2017: Apple Bakes Cookies.

It's time for Security Now!. It's the part of the Internet in which we trust you, if you trust no one. Now, the person who trusts the least and therefore is the most trustworthy is Mr. Steve Gibson. Of course he is the big brain behind Gibson Research, ShieldsUP!, SpinRite, and our coming non-passworded overlords over at SQRL. Steve, my friend, it's been a while; and I've got to say I've been looking forward to this day for quite a bit.

Steve Gibson: I have. I know our listeners have. And what's better, we have an extra treat for people. You can actually understand what I'm saying today.

PADRE: Yeah, I noticed you were having some audio issues the last couple of weeks.

Steve: Ooh, boy.

PADRE: What was that?

Steve: Yes. So first of all, it wasn't Cox. They've been amazing and fabulous through this. And what I think happened was a combination of things. I hadn't messed with my own coaxial infrastructure forever because I'm sort of a, you know, if it's not broke, don't fix it. But that sort of decays over time. So I believe that there was a problem from the delivery of the external cable into my residence. I did find an unterminated run which, you know, you're supposed to have a 75-ohm terminating resistor on all of those, so that may have been a problem.

So I think the initial problem was a slowly accumulating problem between - essentially with the cable modem that I had getting a strong enough signal, through no fault of anyone's except just indirectly my own from having left everything alone for so long. But then what we did was, thinking that maybe it was the audio interface, we switched to a much more advanced audio interface. And unfortunately the machine that I was using was an old Win7 box that I think was underpowered. So we had overlapping problems. The audio interface was actually having a buffering problem with the machine getting the data, getting the audio off of the USB bus fast enough for this newer, higher powered audio interface. That overlapped with the networking problem.

So anyway, as everyone can hear, that's all gone now. And I got a lot of, of course, feedback from our listeners saying, oh, my god, your content is great, but it's barely worth listening to at this level of quality. So yes, we know. I apologize. It did take a couple times, a couple weeks to get down to this. But as I promised Leo at the end of last podcast, I said, okay, done. I mean, I'm not resting until this is resolved. So anyway, again, I've switched cable modems a few times. Even the people at Cox who don't know me, like random tech support people that I had to talk to in order to do some of these experiments, were fabulous. So I have got nothing but praise for my bandwidth supplier, and I'm happy that we've got a glitch-free connection now, for now and hopefully into the future.

Oh, and I wanted to mention that I was recording at my side, but the problem was, when I sent the audio up to TWiT, the time base was enough different that my audio kept falling behind or getting ahead, I never - it wasn't really clear which way it was. So my own recording wasn't useful. But since the problem was in the USB audio interface, it wasn't any better either. It was the same as what I was sending up to TWiT. Anyway, problem solved. I appreciate all of our listeners' patience, and we're good to go again.

PADRE: Well, it's a great lesson for anyone who's troubleshooting because what you just described is - it's a perfect storm. It's almost a worst-case scenario where you have an issue that crops up, but it's not one problem, it's actually two discrete problems that masquerade as a single problem. That is the most difficult thing to troubleshoot ever.

Steve: Right, right.

PADRE: And also I had this problem not too long ago when I used a USB audio interface, which I thought would make things much more clean because it's free and it's isolated and it was relatively expensive. But it turned out that because it didn't have its own onboard DSP, it was requiring the CPU to put in a lot of its power to do the calculations. And that's just no good.

Steve: Yup. So this is Episode 629. And I didn't have a title until the very end, as I was pulling everything together. But as I dug into a surprising story, I thought, okay, this one is the title for the podcast, ABC. The title is Apple Bakes Cookies, which we will be getting to later in the podcast. I am so excited about something that Apple has done in iOS 11 which has reciprocally and for the same reason really upset the entire Internet advertising community. And our listeners know that the whole issue of tracking and cookies and third-party management, it's been a particular hobbyhorse of mine for years.

I have, at GRC, I built that whole cookie forensics system in order to generate statistics from the browser configurations of GRC's visitors. It's not something that I ever made widely public, mostly because it was sort of an R&D project. And we've talked about it often. So we will, later in the show, address what Apple has done, which is just - it is a brilliant solution to this problem. And so Apple Bakes Cookies is our topic, but we've got a whole bunch of stuff to talk about.

We're going to talk about, we already have talked about the solution to this podcast's recent audio troubles. We're going to do a little more follow-up on the Equifax fiasco because I heard you on a different podcast, Padre, talking about some additional steps beyond just locking your credit with the various bureaus that I wanted to give you a chance to share with our listeners. The EFF and Cory Doctorow writing for them have weighed in on the forthcoming browser Encrypted Media Extensions issue, the EME issue that we talked about, about a month or two back. And, boy, they're not happy. In fact, I'll just step on this a little bit. They've resigned from the W3C over this.

PADRE: They've taken their ball and gone home.

Steve: We need to talk about that. There is an interesting emerging web browser-based payment standard which has got full adoption by all of the browser vendors that we need to cover. Also, something finally hit me, I don't know why it took so long, but it's another story about the danger and the mixed blessing of using SMS, the Simple Messaging Service, for second-factor because it turns out it's actually not a second factor. Then big in the news was the breach at CCleaner, one of everyone's favorite utilities for Windows, to go in and do a much deeper cleaning of junk we don't need than Microsoft's own little disk cleanup utility. They had a big problem that affected people for about four weeks. There is a new Bluetooth-based set of vulnerabilities. Again, the great solution that Apple has come up with in iOS 11 we'll talk about. And then a few other random tidbits. So I think lots to talk about. And maybe we'll actually finish today. I'm not sure, though.

PADRE: Well, this has always been a unique thing about the dynamic between the two of us. We get so into the stories that we always end up doing maybe half of the content. I think maybe, maybe this time we might get to the full docket. It's a nice set of stories. All right. It's a topic that we've been covering for a while, the Equifiasco, Steve. What have we got?

Steve: Well, first we talk about our Picture of the Week.

PADRE: Oh, that's right.

Steve: It is one that - it's just one that I had around. There was nothing really compelling that synchronized with the stories of the week. So somebody put together a fun mug. We've recently had my mug on a mug that a lot of our listeners have been purchasing. I just got a kick out of this. And someone sort of synthesized a SQRL mug where we've got the logo for SQRL, and then the subtitle, "Protect Your Nuts." So, yes. And in fact it's a little-known side effect is that there is in fact a nonce in the protocol which is required because SQRL works by signing a unique challenge that the server presents. So thus we need a nonce. But we didn't want to call it "nonce" for the SQRL protocol, so it is formally known as the "nut" in the SQRL protocol.

PADRE: Steve, you're having way too much fun with your naming conventions.

Steve: So, yes, okay. Padre, it's your turn. More that we can do about Equifax.

PADRE: Right. So of course everyone by now has heard all the details, and they continue to come out. And Steve, I've got to tell you, some of it is just head-scratchingly bad.

Steve: I know.

PADRE: It's as if someone designed a program to fail - bringing in the wrong people, bringing in the wrong technology, and bringing in the wrong security measures. But what we wanted to do, and what we've done on a couple of the TWiT shows, including Know How and TNT and New Screen Savers and TWiT, is we've talked about some of the strategies to sort of minimize the Equifiasco. Now, I know that you and Leo talked about the credit freeze; right?

Steve: Right.

PADRE: I mean, that's - he advocates it. And the reason why he advocates it is there's a little bit of revenge. Because if you put a credit freeze on your finances, you're essentially telling the CRCs, the Credit Reporting Companies, that they are not allowed to make money off your information.

Steve: Well, and in fact, 30 months ago, after the Anthem breach, that was our advice on this podcast. I created a bit.ly link, bit.ly/freezecredit, which takes you over to the Clark.com page that lays out the details per reporting agency, how to get them to shut down. So, yes, that's certainly the first step.

PADRE: But we went over a couple of different strategies because there are options available to you, some of which are free and some of which are paid. Now, the most well known and the easiest is the fraud alert. Let's actually talk about this because there's three levels of fraud alert. One only applies to U.S. military, and that allows them to put an alert on their account for one year, free of charge, giving them access to two credit reports per agency per year. But the ones that our audience are probably going to run into are the standard credit fraud alert, which lasts for 90 days. And unlike five years ago, you don't actually have to call in. All you have to do is go to the website of one of the CRCs. I would suggest TransUnion, just because it's really simple to use. You do have to give some personally identifying information, which I'm always a little risqu about it, but at least I'm not giving it to Equifax.

Steve: To authenticate yourself to them.

PADRE: Precisely. But once you place a credit fraud, they must report it to the other CRCs. So you only have to do it once. You don't have to go from CRC to CRC. It works for 90 days. You can go ahead and get yourself a free credit report. And here's the thing. You can get a free credit report every year, but that's not cumulative over the CRCs. So I could actually get one every four months, going from CRC to CRC. That's actually an important step.

Now, after 90 days, if you do the online version of it, you can just renew. It's just a simple click, and they'll renew it for another 90 days. There is one for active exploitation of your ID. If you are confident that someone has tried to open up credit in your name, you can ask for an extended fraud alert. And what that does is it works for seven years, and you don't have to renew every 90 days. That works much better.

But the difference between a freeze and a fraud alert is simple. In a fraud alert, anyone coming to one of the CRCs and asking for your credit report, they can get the information, but they are given a warning that this account has been put under fraud alert, and here's a number which you provide that they should call to verify that you're actually the person requesting credit, a new mortgage, a car, whatever. But they can still get the information. Which sounds bad, but it actually can be good.

Let me explain that. In a freeze, a freeze is different because, when you put a credit freeze, no one can access your credit report unless you specifically allow them to. So you have to whitelist any company that wants to look at your credit. Now, you may say, well, that's good. We're all about whitelisting. We're all about trusting no one. Except in modern society a lot of people don't realize how many times your credit report is accessed. And it's not just when you want a loan or a mortgage or a credit card. It could be the next time you try to get a phone, the next time you try to get a job, you try to get an apartment. All of those are going to ask for your credit report. And if you have a freeze, you have to whitelist every single one of those. It's kind of a pain in the butt.

Additionally, the rules for freezes differ from state to state. In some states it's indefinite. And when you put a freeze on, it's frozen until you unfreeze it. In other states it will last for up to seven years. And there can be a fee both putting the freeze on and taking the freeze off. So they can get you both coming and going. We don't tell people not to do a freeze. But at the very minimum, the easiest thing to do is to put a fraud alert. And, yeah, so that's really the start of the spiel.

On Know How, if you take a look at Episode, I think it was 344, we went over a bunch of different apps that can train you to take better care of your credit. A lot of people don't actually do the maintenance, the bare maintenance that they need to do to make sure that their credit isn't being used, that there isn't some fraud going on. And also, Steve, I don't know how you feel about this, but there are so many people, and not just young people, there are adults who don't know the basics of finances. And this makes it so much easier to exploit you when you don't actually understand how your credit works. Have you seen that?

Steve: So the one thing I'm puzzled about, you said that the fraud alert, you place a phone number. So it's the company that is querying them for your credit must call you, except there's no enforcement for that; right?

PADRE: There's no enforcement. But here's the thing. If they opened up a line of credit on an account that had been flagged for a fraud alert, and they did not call that number to verify you, then if anything does happen...

Steve: Then they're liable.

PADRE: ...they're liable. So no company, no legitimate company that's trying to grant you credit would dare not call.

Steve: Well, and what we're trying to prevent is a malicious other party - I don't know whether it's third or fourth or fifth. We've got too many parties going on in here. But we're trying to prevent the case of somebody else applying for credit in our name. So the company verifying that application would be the one to call us to make sure that that's really us who is applying for the credit, rather than that malicious party. So if the system works correctly, that does close that loophole. The person who is having credit applied for in their name gets the notification from a valid credit grantor that this is really them. And so I can see how it works, as long as everything fits together.

PADRE: Right.

Steve: And I guess it's more practical for people who are dynamically needing credit. For example, all of mine have been locked since that podcast 30 months ago, and I've never - so nobody has successfully been able to look at my credit reports. I just haven't needed any. But I'm 62, and I'm at the point in my life where hopefully things have settled down, and I'm not needing lots of credit now, unlike somebody in their 20s or 30s who's actively churning around, jumping around, moving, buying things and so forth, where locking all of the access to your reports is much less practical.

PADRE: Right, exactly. If you're not super dynamic, a credit freeze works really, really well.

Steve: Right, right.

PADRE: But if you are, and you're relatively young, and you're moving around the country a lot, a credit freeze, it's actually a huge hassle because, again, it's every single request has to be whitelisted. And imagine if everything you wanted to do in a modern society required you to call up, give a PIN, and whitelist an organization. It's actually a lot more hassle than people think.

Steve: Well, and probably just renting an apartment requires that your credit be checked because the landlord wants to make sure that you're not a deadbeat. So all kinds of things.

PADRE: Yeah, and getting a job. When I was still working HR for the church, that's one of the first things we did. I mean, we'd check social media, but we also checked the credit report because that kind of tells you whether or not someone's trustworthy. If they declared three bankruptcies, and their credit report is in the garbage, you start thinking, okay, this person doesn't have control over their personal finances. Maybe that's not a position that they should be in. It sounds horrible, but that's a publicly available piece of information. Of course an employer is going to take advantage of it.

Steve: We did also - naturally more information is emerging about Equifax. And one of the worrisome things we discover is that the person in charge of security, the CSO, the Chief Security Officer, was apparently a woman by the name of Susan Mauldin, who is a music major and, the best anyone can tell, has no background in technology or security. And immediately after this happened, her name on her LinkedIn profile - first of all, the last name was just changed to Susan M. And I looked just this morning, and it's completely disappeared. There's also been some evidence from people who have been digging into this that all information about her has been proactively scrubbed from the Internet. YouTube videos have been pulled down, and interviews have disappeared, and blog postings have gone away.

And so it looks like there's a real sort of a CYA effort here on the part of Equifax to bury the fact that somebody whose qualifications for the position were dubious at best is, like, trying to be covered up. And apparently she's also retired from Equifax now. So I think one of the things that I think I saw after last week's podcast was the number of inquiries and lawsuits gearing up to respond to this breach is daunting, to say the least. Although it hasn't affected their stock price very much. I think everyone, you know, the official opinion is, well, this is bad, but bad stuff happens, and they'll survive.

PADRE: We kind of wish they wouldn't.

Steve: Boy.

PADRE: You almost kind of want them to fail just so that the other CRCs will take their responsibilities seriously. I mean, they have one job. They get access to massive amounts of incredibly sensitive information that could destroy people's lives, and their one job is to keep it secure. And in exchange for that, they are allowed to sell it, package it, do the analysis so that they can make money. They didn't take care of the former, they shouldn't get the latter. Isn't that, I mean, that sounds fair. But let's talk about Susan, though, because, Steve, what strikes me about this story is Equifax put more effort into scrubbing the web clean of any references to who she was or what her background was, they put more effort into that than providing the website so that people who are affected could check. That's just mindboggling.

Steve: Well, and there was also - there was some, and I didn't have a chance to pursue this, and there was some controversy about it, and the information seemed to be from kind of a sketchy source. But there was also, you know, we talked last week about how perhaps it was one of the libraries that they were using for their server-side application design. There's also some question about whether they had left their web-based control panel configuration set to a username and password of admin and admin.

PADRE: That's secure, yeah. That works.

Steve: Okay. So - okay.

PADRE: Steve, that's worked on my Linksys router for 20 years. Why should a website be any different?

Steve: That's right. And besides, if it was fancy, then people wouldn't be able to log in when they want to, and it would be some big hassle for everyone. So, yes, I agree.

PADRE: It was reverse psychology. People were thinking they couldn't possibly use a default username and password. So therefore it was safe.

Steve: That's right. They're going to have a super strong one, so don't even bother guessing.

PADRE: Oh, my goodness.

Steve: But you're right. It is disturbing to see them now taking a lot of actions for their own sake, for their own benefit, when they arguably weren't taking that kind of action on behalf of their 144, 143 million customers whose data was escaped. And that has just happened. We haven't started to see the consequence of that. But that's probably what starts to happen next is when people actually, you know, certainly the people who are clued in, listen to this podcast, listen to the TWiT network, people who are tech savvy are recognizing they need to take some action. And of course this did go mainstream. So the news is out there. Still, there will be many, many millions of people who don't actually follow through and are vulnerable to exploitation from this breach. So that's unfortunate.

PADRE: Eric Duckman in the chatroom was saying we've covered this on every TWiT show, can we stop? And I understand that. There's a little bit of fatigue. However, it's amazing, every time we do this, how many people still haven't done the bare minimum. The bare minimum is logging into a website, spending five minutes entering personally identifying information, and putting a fraud alert. And you have to do that because at this point it's a race condition because, if I were to be looking to do something nefarious, I would find where they're selling these lists on the dark web, and I would start putting fraud alerts and putting my own number so I could lock people out of their finances. I mean, that's a nightmare scenario.

Steve: And I think here's how I would respond to that is, yes, we're beating a dead horse. But this is our responsibility. So having done that, and this is the last time we'll mention it unless some other interesting information comes up, but now it's the responsibility of the people receiving this information, whether they want to follow through and take care of themselves or not. So I have no problem coming back to this and saying, one more time, this is the liability that people are exposed to.

PADRE: Right. One quick note because I do have security friends who they kind of bristle anytime we make fun of the fact that Mauldin had a bachelor's degree in music appreciation. That doesn't necessarily mean that she would be a bad security person. I know people who have their degrees in the fine arts, and they are fantastic security people. But I think, if you look into Susan Mauldin's background, what you find out is not only did she not study security, she didn't have any practical experience with it.

Steve: Correct. And I would argue, I would counter the other security people who are bristling by saying, okay, so why is all evidence of her being removed from the Internet? If they're proud of who she was, just leave her there. I mean, we obviously all know her name.

PADRE: Yeah, yeah.

Steve: So, okay. So we talked a couple months ago about the EME, which has been adopted by the W3C, the World Wide Web Consortium. And I want to follow through on that. I have to read the letter that Cory posted, Cory Doctorow posted on the EFF site yesterday, and then we need to talk about it. So, and he addresses this to Jeff, Tim, and colleagues. Tim is certainly Tim Berners-Lee. I don't know who Jeff would be [Dr. Jeffrey Jaffe, W3 CEO]. Did you have an idea? That just didn't...

PADRE: Yeah, no, I was looking at that. That did not strike any memory for me.

Steve: Anyway, so Cory writes: "In 2013, EFF was disappointed to learn that the W3C had taken on the project of standardizing Encrypted Media Extensions, an API whose sole function was to provide a first-class role for DRM [Digital Rights Management, we know] within the web browser ecosystem. By doing so," Cory writes, "the organization offered the use of its patent pool, its staff support, and its moral authority to the idea that browsers can and should be designed to cede control over key aspects from users to remote parties." So he's sort of laying down their fundamental gripe with the idea of standardizing encryption and media content delivery through the browser.

He says: "When it became clear, following our formal objection, that the W3C's largest corporate members and leadership were wedded to this project, despite strong discontent from within the W3C membership and staff, their most important partners, and other supporters of the open web, we proposed a compromise." And this is important because this is where we came down also on this a couple months ago, when we first covered this.

He writes: "We agreed to stand down regarding the EME standard, provided that the W3C extend its existing IPR policies [Intellectual Property Rights] to deter members from using DRM laws in connection with the Encrypted Media Extensions" - and then he quotes a section of the U.S. Digital Millennium Copyright Act or European national implementations of Article 6 - "except in combination with another cause of action." In other words, and this has always been our take, we must provide an exception that allows researchers and non-malicious attack of this protocol for everyone's benefit.

Anyway, and so I said in my notes here our listeners know quite well how every lesson we're learning shows us how clearly crucial it is that academic security researchers be allowed to have verification oversight over proprietary encryption systems upon which many people depend. This must not be closed.

PADRE: Right.

Steve: So Cory says: "This covenant would allow the W3C's large corporate members to enforce their copyrights. Indeed, it kept intact every legal right to which entertainment companies, DRM vendors, and their business partners can otherwise lay claim. The compromise merely restricted their ability to use the W3C's DRM to shut down legitimate activities, like research and modifications, that required circumvention of DRM. It would signal to the world that the W3C wanted to make a difference in how DRM was enforced; that it would use its authority to draw a line between the acceptability of DRM as an optional technology, as opposed to an excuse to undermine legitimate research and innovation." And you can imagine where this is going because they're not happy.

He says: "More directly, such a covenant would have helped protect the key stakeholders, present and future, who both depend on the openness of the web," and blah blah blah. I'll skip the rest of this because it's more of the same. Basically they are incredibly unhappy, and the EFF has resigned from their participation in the World Wide Web Consortium, where they have long been a member, over the fact that the way this came down, that covenant to protect research was withdrawn. Well, I should say it's been shelved for some period of time, and there's a big concern now that essentially the rights holders' lobbying power has won, and that even the exemption for research is now in danger as this Encrypted Media Extensions moves toward standardization and widespread adoption.

PADRE: You know what worries me about this story, Steve? It's the fact that a lot of news agencies are selling this as, well, the EFF doesn't like the corporatization of this group. That's not really what it's about. As you've been trying to say, the clear analog here is how the CFAA, the Computer Fraud and Abuse Act, has been misused...

Steve: Yes.

PADRE: ...since its inception. I mean, yes, ostensibly it's to keep people from accessing systems that they shouldn't access. But we have seen time and time again that researchers get brought to court because they point out a flaw in a system. They were never attempting to exploit it. They were very open about their research. In fact, many of them offered their research in secret to the company before they went public. And time and time again the CFAA was used to hit them over the head and say, well, no, you accessed our system without authorization and therefore your research is null and everyone should not pay attention to the huge gaping hole that you just pointed out in our security.

What the EFF is saying is let's not repeat that mistake. That was a mistake in the first one. It was an unforeseen consequence. Why don't we write this one so that that doesn't happen? That's the actual story. It's not about whether or not corporations are involved or not. It's about whether or not honest academic research can be had in a digital world. And it boggles the mind why the W3C won't allow that.

Steve: Well, yeah. And the problem is we all know that what the whole EME is trying to do is impossible anyway. We have example after example of the problem being, if you need to decrypt for display, whether it's a DVD or an HDMI video signal, the decryption has to be there in order for the end-user to see the decrypted content. Which means protecting it is impossible. I mean, every single attempt to do this has failed, over and over and over, way back when it was a VHS tape, and they were trying to screw up the vertical blanking interval in order to prevent VHS tapes from being duplicated. You just stuck a little box in there that reconstructed the vertical sync, and then you could copy the VHS tape. I mean, it doesn't work. Yet the content holders keep trying to force this on us.

And I was lamenting this when Leo and I were talking about it a month or two ago, that, for example, as a consequence of the encryption in the HDMI signal, when you switch around, once upon a time you were able to flip sources or flip destinations on your media system, and it was an immediate switch. It would just jump between devices or between sources. Now you sit there and wait for the encryption to do its resync and handshake and everything, and both ends to agree. And then, after 10 seconds, finally you get a picture, if you're lucky. And sometimes it sort of hiccups, and you have to go away and come back again. It's just, I mean, there is a cost to this to the consumer of inconvenience, even for people who are playing by the rules. It's just so frustrating.

And I get it that, if we didn't build this into the browser, then we would always be forced with add-ons, which arguably would be less secure, prone to failure. I'd rather have there be, if we're going to be forced to have encrypted content delivery, and the browser is the container for that, I'd rather have it well done once than every content provider providing their own with lots of opportunities for problems. So again, I think where the EFF came down was exactly right. I hope that their withdrawal doesn't minimize the effort that they're able and the pressure that they're able to bring because we really do need academic researchers and security researchers protected from lawsuits which are too easily launched by media companies that don't want anyone poking around inside their code and their system. We need it for security.

PADRE: Steve, I'm with you. That's what I want. I just don't know if that's reasonable to expect. I mean, we have an analog; right? The analog is all of the discussions we've had, not just in the United States, but also in Europe, on encryption, about whether or not there should be a backdoor that government can access. And we've had experts. We've had computer security experts. We've had mathematicians try to explain as simply as they can why you can't do that; why putting a backdoor into any encryption system will necessarily break it, bust it. Not weaken it, but destroy it. And yet you still have politicians and corporations sort of ignoring that and saying, well, but this is what we need. I see the exact same thing with the W3C here, saying we understand that there needs to be an exception for academic research, but this is what we need.

Steve: Yup.

PADRE: And now that the EFF has withdrawn from this consensus group, what's the next step? There is no next step; right? I mean, they've got no input. The W3C is not going to reverse their course. No one is going to have a voice for the consumer anymore.

Steve: No.

PADRE: That's kind of depressing, Steve. I don't know how to take that.

Steve: Yeah, so Cory's letter ends saying: "We will renew our work to battle the media companies that fail to adapt videos for accessibility purposes, even though the W3C squandered the perfect moment to exact a promise to protect those who are doing that work for them." He writes: "We will defend those who are put in harm's way for blowing the whistle on defects in EME implementations." And finally: "It is a tragedy that we will be doing that without our friends at the W3C, and with the world believing that the pioneers and creators of the web no longer care about these matters. Effective today, EFF is resigning from the W3C."

PADRE: And it's a great letter, and it's an incredible sentiment, and he's absolutely right. I just - I need to know what happens next. I mean, it was the right thing for the EFF to pull out. It does send a message. It does now have a tension, eyeballs on this, maybe even in the mainstream media. But what's the next step? What needs to happen? Is there something that could possibly happen that will reverse course for the W3C?

Steve: My feeling is that things like the DMCA need to be revisited. I mean, that's the ultimate source of all of this problem is that that was improperly conceived. And again, unfortunately, in the U.S. we keep seeing that our politics, the politics of our legislators, are for sale by well-moneyed lobbying groups that have proprietary interests behind them. I don't know how we get on the other side of that. I mean, it is a challenge.

But I think the only solution is to go back to the root of the problem, which is a fundamental misunderstanding about the - again, which we keep seeing demonstrated over and over about the need to create exemptions for academic research because this stuff is hard. And prohibiting anyone from looking at it is not the way to make it better. I mean, we keep seeing that. I don't know how this can be made any more clear, but all we can do is hope and keep sending the message. And speaking of...

PADRE: I've been catching up on my - sorry.

Steve: Go ahead.

PADRE: I was just going to say I've been catching up on my Black Hat and DEF CON talks because I missed them this year, and so I'm watching them on the Internet. And there was one talk that was specifically on the DMCA. And they were saying, what we've realized is that the DMCA has one purpose, and that's to put white hats in jail.

Steve: Yeah.

PADRE: Because black hats, they're not going to be honest about their research. They're not going to share it with anybody. Gray hats, they're working in a datacenter somewhere, so as long as they keep their particular part of the Internet safe, they don't care. White hats, their entire existence is to share knowledge. And it's the sharing knowledge that gets them in trouble with the DMCA.

Steve: Yeah. And if they don't responsibly disclose, then I can understand.

PADRE: Absolutely.

Steve: But when they do, don't put them in jail. It's like that teenager, was it in Budapest, who found that the new e-ticketing system had a problem? I don't remember where it was. And he got a visit a couple days later in the middle of the night and was arrested. It's like, what? I mean, he didn't even use the ticket that he was able to issue by this poorly engineered system, and he brought it to their attention. And they said, oh, bad. We don't want any bad news. We're going to arrest you.

PADRE: Right, right. There has to be something like safe harbor for researchers, saying, if you stick to this protocol - so this is what you can do while you're researching, and this is how you responsibly disclose. And as long as you stick to that protocol, you are legally in the clear. That is how you get white hats to be white hats. And that's how you encourage people to be responsible. I mean, all of our shows are about encouraging responsible Internet citizenship. Well, if you want responsible Internet citizenship, you need to provide some rights to responsible Internet citizens.

Steve: Yeah.

PADRE: All right, Steve. Bring me out of this funk. Give me some good news.

Steve: Okay. So I have some good news. Does my audio sound as good to me as yours does to me? I mean, wait.

PADRE: You always sound good to me, Steve.

Steve: My audio sound as good to you as it does to me. It just seems flawless.

PADRE: Yeah. This is the fun part about working with you. It just brings out the best in everyone. All right. So we've got the EFF resigning from the W3C.

Steve: Yeah, we do. However, the W3C did something good. Just yesterday was the announcement that all of our browsers are onboard with a new API, which is known as the Payment Request API. And I thought, uh-oh, what? Because, I mean, there are so many ways this could be done wrong. I thought, okay, what is this? Okay. So the good news is it looks like it's a really good thing. We're all familiar with the annoyance of purchasing things online.

And, for example, I'm happy when I buy something with Amazon that I just say, yes, I want that tomorrow, and magically it appears. Or if there's some used minicomputer on eBay, I go, oh, I have to have that, or a circuit board, or who knows what. And I just, you know, I click on the button, and because eBay is tied into PayPal, it just, you know, something spins around a couple times, and then it says, okay, done. So that's a nice transaction. Of course, those are siloed instances.

Now, we do have, for example, in my case with LastPass, LastPass has my credit card information. And so to some degree it works, if you trust LastPass with that information. Because LastPass is already able to populate form fields, it's able to say, and it does bring up a little extra dialogue, "Are you sure you want this page to receive your payment information?" And you say yes, and then it goes blip and all that, you know, it does as good a job as it can filling in a form. And of course the problem is since there's no standardization of user-facing web payment, there's a problem with automating that process.

So that's what this new standard does. And it looks like they did it, they thought it through, they understood the problem, and they came up with a nice compromise which is lightweight, which is really what we want, this Payment Request API. So the way this works is, if you go to a site, and you navigate to the place where you would normally need to fill out the form, now the site is able to, using this Payment Request API, send something to your browser that says here is the payment information we're requesting. The browser then itself, not an add-on, like just as I was giving the LastPass example, or whatever add-on you have, the browser itself is a database which would contain an instance of this information, obviously securely encrypted and protected.

So the website, this Payment Request API, allows the remote site to say here are the pieces of information we need. The browser checks its database and presents to the user essentially a detail of what the site is asking for and gets the user's permission to forward that to the remote site. So essentially this unifies and smoothes that process. It unifies it because it creates an industry standard query and response mechanism for this class of information where the user is in the loop, saying yes, I want you to provide my relatively unchanging information to that site. You simply say yes. The remote server gets the information, and then it processes all of the payment with its own back end.

So this is just sort of a clean information provision system. It doesn't change the payment flow in any fashion. And, for example, if the site supports PayPal purchases on the back end, it would be able to tell the browser that it accepts credit cards or PayPal. So then the user would be able to say, ah, yes, then that's what I want to use, and you select that option and confirm that you want the information provided, and it just happens. So it'll take a while for this to get supported, first by the browsers and then certainly by the back end.

Since I wrote my own ecommerce system, I fully intend, as soon as this thing is mature, to send that query to visitors who are wanting to purchase a copy of SpinRite in order to smooth the process. And it'll just be much easier to do the same things we've been doing, but essentially solve the problem that no two sites do this in exactly the same way. It will allow a unification of that process. So I think it's very welcome.

PADRE: Okay. So let me break this down a little bit. Correct me if any of this is incorrect. My browser acts as a database, so it's going to have my credit card information, maybe my PayPal account information, address, all that good stuff.

Steve: Zip codes, your address and so forth.

PADRE: Zip codes.

Steve: Yeah, shipping information, billing information separately, yup.

PADRE: All my sensitive information.

Steve: Yup.

PADRE: If there is a request, a payment request from a site that's using this API, they will get a tokenized bit of data that allows them to have what they need for the payment that I select. So instead of having enter your name and your credit card number, it'll say, hey, I've got these four different payment options. Which ones do you want to use? I can click one of those. That automatically gets sent over. But where is the database actually maintained? So let's say I'm using Chrome. I am accustomed to being able to log into my account on any of my computers, and I get full access to everything that browser might contain - the sites I visited, my history, my cookies, et cetera. Will that still work? And if it works, where is the actual database of personally sensitive information stored?

Steve: Well, it is in the browser. And we can think of it as an extension of the "Remember my password for this site" sort of feature. So, for example, if you don't have a third-party password manager, but for example you're using Chrome, and you just say yes, Chrome, remember my password for this site, that's stored. And then if the browser uses cloud sync, then it's automatically synchronized among your other instances of Chrome, wherever you are. So this information would be treated similarly. So it's browser-side data. And whereas right now the browser is responsible for recognizing a username and password field - and as we know, that mostly works, but kind of doesn't.

What these guys have done is they've said, okay, we're going to make this explicit. We're going to remove any ambiguity. We're not going to require that the browser use heuristics in order to realize, oh, look, there's a credit card field. And as we've been talking, there have been some exploits, for example, where fields are being - malicious exploits where fields are being moved offscreen and filled in by the browser behind the user's back in order to allow sites to obtain information without the user's authorization.

So this is a response to that problem, too, so that essentially this will be the way that websites can query the browser in sort of the same way they do so implicitly, with username and password form field recognition. This will surface it into an on-the-wire protocol that allows a browser to say, ooh, this site you're visiting has just asked for payment information. And it offers these options. Which one do you want to use? And then you say "That one," and you say "Yes," and it just happens.

PADRE: You know, if Google is smart here, they will take this payment API idea, and they will just integrate it into Google Wallet because Google Wallet, they've already designed it and semi-abandoned it to be sort of the place where you keep all your credit card and all your payment information. If they made a couple of tweaks, so now that is automatically offered whenever payment is requested through this API, then they make Google Wallet compelling again, and they make it a reason for you to actually keep your payment information. I have a Google Wallet, but I don't think I've updated the credit cards in there probably for five years.

Steve: Yeah. And for what it's worth, Google is 100% behind this.

PADRE: Yes.

Steve: One of the articles that I ran across when I saw this was that Google is gung-ho. They're all - they have a nice page on their site, and they're up and running. So they will certainly be among the early leaders. And I'm sure that - who knows when Microsoft will do it, but they certainly will. And I'm sure that Firefox will be right on it, too.

PADRE: Right. Now, we've got to play devil's advocate here. Is there a downside? Do you see right now an obvious venue to exploit this? Because there will be malicious calls to the payments API, what do they have to do to make sure that this is going to be secure coming out of the gate?

Steve: Well, the obvious attack is on the browser's storage of this. But that's present somewhere anyway. I mean, our passwords are in the browser. If it's remembering our passwords and filling them in for us, then it's in the browser. And password managers, same sort of story. So, I mean, again, ultimately there's a tradeoff that we always face between convenience and security. And so if you want absolute maximum security, you don't let your browser ever have any of this information. You always enter it in every time.

On the other hand, notice that one of the great risks is keystroke logging that we're always facing. And this moves the information away from the UI. That is, it's no longer something - you're not entering your 16-digit credit card number every time you want to purchase something. That happens behind the scenes, which you could argue is a benefit for an obvious class of vulnerabilities, by just having it saying, oh, do you want to use the Visa card ending in 2319, and you say, oh, yes, I do.

Well, nobody watches you do any of this, whether it's keystroke logging or any other malware, gets all that information. They just see, oh, something just got permitted because this is over TLS encrypted between your browser and the remote server. So I don't really see a downside. I don't see that we're opening up a new avenue of attack beyond what we have. And I would argue that just smoothing the way for allowing browsers to securely interact with remote sites for ecommerce makes sense.

PADRE: Right. If you've got a compromised browser, you have a compromised browser.

Steve: Right.

PADRE: Nothing is going to help you from that. But you're right. As long as it doesn't open up a new venue of attack, a new vector, then this is a net good. I don't see anything wrong with this.

Steve: And we have the advantage of the web browser vendors clearly focusing on the security of this. So they're going to do as good a job as possible to make sure that it's secure and safe.

PADRE: Well, I mean, unlike making sure there's an exception for academic research, they have a vested interest in making sure the payments API works perfectly, the experience is good, it's secure, because that's money. I mean, money talks.

Steve: Right, right.

PADRE: Okay, now, Steve, here's a thing. Can we jump in a time machine and go back, say three years? I was a big proponent, when I was still helping out with IT for my organization, of enabling two-factor authentication. And the big thing was, look, you all have phones. We're going to set it up so that you get an SMS to give you the second factor. It's super secure. It's easy for you to remember because you're always carrying your phone. You're never going to forget it. You're always going to have access to that second factor. I can't really say that anymore. I can't tell people to use SMS as a second factor because, as it turns out, it's even more insecure than I thought it was.

Steve: Not only that, but it's not a second factor. That's the crux of the problem. Remember, the whole idea of multifactor authentication is an "and" conjunction between the factors, your password and a second factor, not "or," or not "instead of." And what's happened is, over time, this thing has changed. It's morphed into "I forgot my password, send me a text message." That's not second-factor. That's multiple first factors joined with an "or," rather than an "and." And it's weaker by definition. I mean, it's clearly weaker. If you have more different things you can use to authenticate, and any one of them can be used, then you immediately reduce the security of the whole system to the weakest among them.

And now we see that, thanks to the problems with the Signaling System 7, which is the international system that glues all of our telecom providers together, due to that it is very vulnerable to interception. So what's happened is we've moved away from using this second factor as additional security to replacement security. And so thus what is in the news recently is that yet another set of researchers have demonstrated how people's bitcoin, I think it was Coinbase Wallet, an account could be compromised by using the SMS second-factor on Gmail for account recovery. It should not be account recovery. It should be additional password verification, the idea being that you want to prevent password being brute-forced. So you say, oh, they know the password, but now they also have to, rather than, oh, they forgot their password, so instead. So yes. It's not a second factor.

PADRE: So it's a 1.1 factor. Maybe not even .1, no.

Steve: It's a .5 because it's less because you've reduced it to any of these can be used. And one of them is, I mean, this is worse than a password.

PADRE: Yes.

Steve: Password you at least have to try, you have to brute force. It might be a really good password, and then they can't get in. Now it's intercept the SMS message on the fly, and that's all you need in order to do multiple different accounts. It doesn't even matter if you have different passwords on your Gmail and your Coinbase account. No, because they're both using SMS. So one single exploit allows them to intercept the text messages being sent out. It's ended up the whole system has just collapsed because it ended up being misused. It isn't multiple factor. It's semi-factor.

PADRE: I've been following a project from a maker that I know. I met him at DEF CON, like six or seven years ago. He has created what is essentially a dirt box using a software-defined radio that is about the size of a laptop, maybe like an old Toshiba laptop, and it's got all - the power supply and everything that he needs to turn this thing on and start intercepting. And so I was asking about, well, so what would you intercept with it? You would intercept calls. He goes, no, calls are boring. I don't care about calls. What I want is I want messages because messages are easier to parse, they're easier for me to filter, and it's easier for me to look for exactly what I'm looking for.

And he couldn't turn the thing on because of course you're getting in trouble any time you want to do that. And he actually is a bona fide researcher. But he was trying to show me how easy it would be for him to, say, walk into an office and just wait for SMS messages to start rolling into his dirt box. And that's a little scary, honestly, because it could happen in real-time. This is not one of those hacks that takes someone time to set up or maybe social engineer themselves into a place where they can get in the middle. This is literally someone walking by your office, and 30 seconds later he's got a data file filled with in-the-clear passwords. That's - yeah. So what's the solution?

Steve: So I began talking about this when I was setting up my Hover account. I moved my domain names over to Hover. And while I was setting it up I wanted maximum security, and they gave me these options. And immediately it was clear to me from our own coverage on this podcast: SMS, bad; the time-based one-time password, good. And so back then, at the time, we talked about this. And I said it is time for us to abandon SMS. That is, it is a vulnerable system which is no longer trustworthy. And at least someone like Hover is using it for multifactor, not alternative factor. But even so, what we want is we want to use the authenticator app that generates a time-based token.

In that case, the sensitive cryptographic information is only exchanged once. It's on your browser. Everyone knows that I like printing out the QR code. I have a little sheaf of printouts of the QR codes of all of my different time-based tokens. And so when I'm setting up a new device, I just run through, literally, the paperwork and snap a picture of each QR code in order to import the keying for the time-based token into the authenticator app, and then I'm good to go. But that's where we need to head.

And the disturbing thing is that, for example, even if you set things up that way with Twitter, Twitter still wants to send you an SMS message just because they're biased towards ease of use. They don't want tech support calls. They don't want people saying, oh, I'm using a new phone, and it doesn't have my authenticator. And it's like, okay, well, then we're going to send you an SMS message. It's like, no, because that's a back way in, a backdoor, essentially, into your password recovery if you don't have the password. So anyway, we need - it's going to take a while because everyone thinks this is, oh, super security. No, it's worse than what we had when you had to brute-force a password. Now it's like, oh, I forgot my password, send me a message.

PADRE: Now, "HiQ1" in the chatroom does point out that you can turn off SMS authentication with Google. And actually I have. I'm not sure how much better the new system is. So the way it works is anytime I log in on a new browser - and I actually have mine set in to time out every three days. So every three days, any device that wants to access anything from Google, so that could be my YouTube account, my Gmail account, it could just be the browser, when I log in, I get a pop-up. It's not a text, but it's a pop-up saying, hey, this device with this name has just requested access. Do you want to grant it? And it's just literally a yes or a no. That's not SMS; right? That's something else. That's some other protocol.

Steve: Yeah, that's their own backend messaging technology.

PADRE: As far as you know, is that secure?

Steve: Yes, because it's TLS, it's authenticated, and it's over the Internet, not the telephone system. And so it's the telephone system that is the problem because this Signaling System 7 is old. And there's no authentication in SS7, none, which means it's completely vulnerable to man-in-the-middle attacks. There's no verification of the endpoint identities among telecom providers. It's like the early Internet. And because we've layered all of the authentication on top of the underlying protocol of the Internet in order to get HTTPS with certificates, but the telco systems never did that.

PADRE: Right. I will say that they also give me a backup because of course the fear is, well, what happens if I break my device, or I don't have my device? My device is stolen? Now I'm locked out. You're going to hate me for this, Steve, because this is about as insecure as you can get. In my wallet right now there is a card that is printed. It has one, one-time-use authentication. Now, you still have to have my username and password, so it's not - it doesn't get you in. But it does...

Steve: That's very good.

PADRE: It does mean that one time, one time I can get in and maybe print out another one-time authentication code. But even if I don't have this device, I can still get into my data.

Steve: Yup.

PADRE: Again, but the wallet's probably not the most secure place to put that.

Steve: I don't know. I mean, Bruce Schneier is famous for saying that you should have complex passwords and write them down because we are good with managing little bits of paper, as Bruce put it. And it's true, you don't want to write them on the underside of your keyboard because that's now the first place everyone looks. But still, no one's rummaging through my wallet. And so, yeah, you have to have your username, your password, and something else. I think your solution is bulletproof.

PADRE: There was a company at CES, and I wanted to do a video with them, but it just seemed so strange. They were making a product specifically for this. So the way it would work is you would print on a very special piece of paper, and it would go into like this laminate pouch. And then you'd crack something, and it now stays in your wallet. It still looks like a business card. To access the code, you had to tear open the seal so you could see it, and the print would fade after 10 minutes. And I was thinking, okay, that's really geeky, super impractical, probably doesn't actually add any security because I can still write it down.

Steve: No, because we all have cameras. We all have cameras. You just take a picture of it.

PADRE: But I thought, that would make for 30 seconds of really cool video.

Steve: Agreed.

PADRE: It's almost the "Mission Impos-" - I wanted it to self-destruct, honestly. I was like, oh, the ink just fades? I wanted it to burst into flames or something.

Steve: Sort of like your MRE heating up the other day.

PADRE: Oh, my gosh. Steve, I have to say this. If any of you are having gastric problems, eat an MRE, and you'll be bound up for a couple of days. They're still better than the ones they had in World War II, but oh, my gosh, there's so much salt.

Steve: Yeah.

PADRE: You know, the next time you come into the studio, we actually have a box of MREs.

Steve: No. No.

PADRE: No?

Steve: No.

PADRE: You don't want to test internal security?

Steve: Our listeners, well, and the industry at large was very upset over this CCleaner breach, which was recently exposed. For a period of four weeks - and the good news is there's a way to tell if you got bit by this. For a period of four weeks, from the middle of August, so a month ago, through September 12th, the downloadable 32-bit version of CCleaner, v5.33, was compromised with the Floxif, F-L-O-X-I-F, Floxif malware, which infects Windows executables and DLLs, backdooring the machine to install additional malware. 2.27 million CCleaner users inadvertently downloaded that 32-bit version of CCleaner v5.33 during that one-month window.

Okay. So the earlier versions were fine. It was updated to 5.34 on September 12th, closing that window. Yet 2.27 million copies of that were downloaded. In the show notes I provide a registry key. So our more tech-savvy listeners, if you look under HKEY_LOCAL_MACHINE, then the software folder, and then Piriform, those are the publishers of CCleaner, P-I-R-I-F-O-R-M. If you have a key under there, Agomo, that's not good. That's the key that this Floxif malware creates. And under there are two data values, an MUID and a TCID, which are used by the installed Floxif infection. So the bad news is that CCleaner does not proactively remove this.

So I just wanted to say, again, I mean, I've got CCleaner. The good news is I didn't download an update or a copy of it. CCleaner does not have an auto update facility, so you do need to get a newer version. Bu you definitely want to make sure, if you think you may have obtained an update or a download during this window from the middle of last month to the middle of this month, then you want to make sure that you didn't get this stuff installed in your machine.

What was found was - and this was found by the Cisco Talos security group. They were testing some of their own antimalware software, and alarms went off, to their surprise, caused by CCleaner. They then dug into it and discovered that the alarm was going off because of this Floxif malware. What happened was that the distribution server was compromised, and this was added to the download.

PADRE: Well, how did they sign it?

Steve: It wasn't. It was added to. So the CCleaner was properly signed, and the signature validates. But there was another chunk that went along with it that was the infection, and it was not signed. But it still you know, someone clicked yes when they were installing it because they thought they were just installing a trustworthy copy of CCleaner, but it was bringing something additional along for the ride.

PADRE: I mean, yes, that's evil, and it's horrible. But props to whoever thought of that because, yeah, I mean, if you download something like CCleaner, which has a reputation, is a legitimate piece of software, it's security software...

Steve: It's a great utility, yes.

PADRE: Great utility. And you attach something to it, they're still going to think it's CCleaner. They still see that it's signed. And anything that pops up during the installation process, they're going to assume it's legit, it's valid.

Steve: Right, right. So anyway, the takeaway is, if you think you may have been exposed, you can check that registry key because this Floxif infection will create a key there. And if you've had CCleaner from before and haven't updated it through that window, you're probably fine. And so, yikes. But it happens. The Piriform guys, I wasn't really happy with their disclosure. They worked a lot to minimize the threat. They believe nobody was compromised. It's worth mentioning that you don't want this thing in your machine.

But what the Talos guys found was a dynamic domain name generator so that this would be in the future looking for domains which would be registered for the command-and-control server, where it would connect up. The server was brought down. It is believed that none of these infections were ever able to contact the mothership in order to get instructions and then download other things. But you definitely want this Floxif stuff off of your machine. So everyone should just make sure, if you think you may have been open to downloading it during that time, that you're sure it's not there.

PADRE: And as far as I can tell from the bulletin, it is only the 32-bit version; right?

Steve: Yes, yes, for sure. Oh, and there is a cloud version that is also susceptible. So it's the CCleaner Cloud had the same - during the same window of opportunity, that same one month, either of those could have been a problem.

PADRE: Now, JammerB, if you go ahead and go to my screen, so this is - I just reformatted this machine, so I don't have it yet. But you would say it would be under LOCAL_MACHINE and then software. If you have CCleaner, there would be a new key here for Piriform.

Steve: Piriform, yes.

PADRE: And what am I looking for again?

Steve: And then under there is Agomo, A-G-O-M-O. Normally under Piriform is just CCleaner. But there'll be a second entry, Agomo, A-G-O-M-O, which is not from Piriform.

PADRE: Got it.

Steve: And that'll have two other keys, an MUID and a TCID, values.

PADRE: And that tells you that, yeah, it has installed itself.

Steve: You've got to do some cleanup. And you want to look around for, like, Floxif remover stuff. Probably Malwarebytes or something will scrub it right off.

PADRE: Question, Steve. Do you know where I could get a copy of the infected file? Because I'd love to install it and just see what it does, maybe just watch the traffic, see if it's phoning home.

Steve: I don't know, but I'm sure it's around.

PADRE: That could be fun. That could be a Know How.

Steve: Yeah. So one more, and then we'll do our last break. A security research company, Armis Incorporated in Palo Alto, has found and responsibly disclosed some time ago, which is why we're now learning of it, a series of very worrisome Bluetooth-based compromises and exploits. They're calling it BlueBorne, sort of as in airborne, but in this case using Bluetooth. Our listeners know for years my advice on the podcast has been always turn off radios you don't need. And I know that this advice has been heard because iOS has had this habit of always reenabling Bluetooth whenever you update any iOS device, just like, oh, you must have turned that off by mistake. No. I turned it off on purpose because I don't - there's nothing, no Bluetooth thing that I need my phone connected to, thank you very much. But Apple always turns it back on.

So these guys did a deep dive into the Bluetooth stacks throughout the industry and found and responsibly disclosed a large number of zero-day vulnerabilities that is just - it's breathtaking in scope. Basically, all Bluetooth platforms - including iOS, but not recently, props to Apple - had these problems. So in their coordinated disclosure, Google was contacted back on April 19th of this year, after which details were shared, and Google immediately addressed the problems and pushed security bulletins and updates out earlier this month on September 4th.

Microsoft, same thing, contacted back on April 19th, details were shared, updates were made back in July, and then public disclosures in the middle of this month, like about a week ago. Apple was contacted on August 9th because there were no vulnerabilities, and I think it was from - I think iOS 9 may have been, but 10 and now today's update to 11, Apple was completely clean. Sadly, Samsung was contacted on three separate occasions, in April, May, and June; never received a response. These guys were never able to get a response back from Samsung, whose head is apparently deeply buried in the sand because they're rife with exploitable vulnerabilities in their Bluetooth stack.

Linux, the Linux Project was contacted back in August, around the middle of August, and they jumped on this and fixed their problems. So, I mean, across the board - Windows, Linux, the mobile platforms - all vulnerable to a series of very serious attacks. The Bluetooth stack runs down in the kernel. It's got lots of privileges. There are remote code exploits. And, I mean, it's like as bad as it can get. Also, these attacks work without needing Bluetooth association. So it's just the fact that the radio is on and listening. A non-associated attacker within 30 feet is able to compromise a device that has not been patched that has Bluetooth on. So if you've got a Samsung device, turn Bluetooth off because those jokers have not responded to any of this.

So anyway, I just did want to let everybody know. The good news is it was handled, as long as you're keeping your devices up to date. As long as you're using Google and you're current, and Windows and Linux and Apple's iOSes from 10 on, you're fine.

PADRE: Unless you own a Samsung device.

Steve: Samsung, not so much, yes. Yikes.

PADRE: Well, that's what so nefarious and so impactful about this story. The vulnerabilities that they're targeting are typically the older versions of Bluetooth, which we keep because that's the compatibility. We all have devices from way back when.

Steve: Yup, yup.

PADRE: And so, yes, it's easy for me to update this. In July, Microsoft updated Windows, so that's not going to affect me anymore. Same thing with this. This is a OnePlus 5. This actually came patched, pre-patched. But there are also millions of devices that are using older versions of Bluetooth that will never be patched. Think of how many car stereos have Bluetooth.

Steve: Televisions, televisions. Yes, yes, yes.

PADRE: They will never be patched, ever. Now, what you can do with that depends on how they actually designed the entertainment system in the vehicle into the actual controlled area network of the car.

Steve: [Sounds of distress] Yeah.

PADRE: But, I mean, that's, yeah, this is a vulnerability that will keep giving for decades.

Steve: And this is the problem is that we have a very sophisticated protocol which is hard to get right.

PADRE: Yes.

Steve: And we keep tweaking it and adding to it and extending it, and mistakes get made. But unfortunately, exactly as you said, there is an ecosystem of dynamic patching which affects some of these devices, but certainly not all of them.

PADRE: I actually had a family member who read about BlueBorne, and she called me in a panic. And she says, oh, I have a car, and it's got Bluetooth audio, and I keep my phone on so it automatically syncs up every time. And I said, okay, well, there's a few things. First, turn off Bluetooth on your phone. You don't use it for anything else. And then I was about to say, well, just go ahead and use a 3.5mm jack to get to - and then I realized, oh, you've got an iPhone. Okay, so that's not going to work. Get yourself a dongle, that will work. But then I'm like, the Bluetooth on the stereo system, I was going over the model that she had, it can't be turned off. There is no setting to turn it off, so it will always be on and always be vulnerable. Forever.

Steve: Wow.

PADRE: So she just needs a new car.

Steve: Yeah.

PADRE: Easy, yeah. Fixed, fixed.

Steve: Or an old car.

PADRE: And old car like mine, which just has the 3.5mm jack.

Steve: Like mine, like mine. I have no technology.

PADRE: Well, I've got the little OBD-II bus, and that's plugged into a laptop so I can monitor what's going on with my engine. But yeah, I don't want - especially since we've heard that some of the high-end automakers have done wonderful security things like allowing cruise control and the brakes to be controlled by the entertainment system. Which sounds like a really good idea.

Steve: What could possibly go wrong?

PADRE: All right. Let's get away from BlueBorne, let's get away from wireless exploits, and let's get back to good old-fashioned HTTPS.

Steve: Well, this is just a quickie. Someone, one of our listeners thought I would be interested in this and forwarded it to me, and it's very cool. One of the problems that users of the iOS platform have is that it's so closed. You can't see what's going on. And then there's a reason for that. Most users don't care what's going on. They wouldn't know how to interpret what they saw if they could see more of what's going on. So, fine. Except we're not all those users. Certainly not listeners of this podcast.

There is a very neat extension named Inspect, just I-N-S-P-E-C-T, for iOS devices. I've got the link to it in the show notes. It allows you to inspect the HTTPS certificate of any site you're visiting. So it uses the familiar UI, where you send something somewhere, like you copy the page, you send it to email, you send it to message or so forth. This adds a "send it to inspect." And so what happens is the Inspect add-on captures the URL and displays full certificate information. So anyway, just a cool little gizmo. Again, nothing you need, but there's no other way to get that information when you're just using the standard Apple Safari browser. This allows you to check out the certs of the sites you're visiting. So I just wanted to put it on people's radar because I know that some of our listeners would think that was a cool add-on.

PADRE: I'll actually say that that's a big deal. For years now you've been telling people to inspect the certs, just check it out, make sure everything's on the up and up. If you didn't have a tool to do that, then that's not usable advice. Now you can.

Steve: Well, and sometimes you'll get a warning where it's like a go, no go. It's like, oh, there's a problem, but it doesn't - you get no details. And so how can you make an informed decision about whether you want to use it anyway? For example, what if the certificate expired last night at midnight? So it's like, okay, well, I really want to go to this site, but their cert expired 12 hours ago. So it's probably fine. They just forgot to renew their certificate. So this would allow you to see what the problem was and then make a decision about what your behavior should be after you have that information. So seems useful to me.

PADRE: Well, Steve, this next item is about Android. This actually affects me because I have multiple backup devices I keep on hot standby. Maybe I'll power them up every once in a while to make sure that they still work, make sure that they're updated. But there's a new policy at Google that might make that a bit more important to power up every once in a while.

Steve: Yeah. And I just wanted to, again, sort of in the same way, put it on our listeners' radar. I wasn't aware of this, either. And that is that Google will purge your Google Android device backups if you haven't used them in 60 days. So there is an expiration notice that will show after a couple weeks. It'll warn you that you've only got X, like 54 days left. But if you're not - so the point is that you cannot rely on a backup of your device persisting longer than two months from your last use of the device.

Again, it's not like the device is going to die, or that everything's going to disappear. But again, your backup will be expunged from Google's cloud after 60 days of non-use. So again, just something to keep in mind. You cannot rely on it being there forever. Which is weird, too, because in doing some investigation of this, people are - it's not like their storage is over commit or over limit or anything. They've got lots of available Google cloud space. But Google's just saying, I don't know, we don't want to store something for I guess a device maybe that they assume is retired. Be nice if it was more like 180 days than 60.

PADRE: Right, right. And I understand what Google's doing because they're probably looking at their datacenters and realizing, over half this stuff are for devices that no longer exist.

Steve: Yes.

PADRE: Do we really need it.

Steve: Exactly.

PADRE: Although, when you wrote down the Google "purge," I was thinking that for 24 hours all apps can be downloaded to your phone, and nothing will be illegal or something like that. It's a really bad horror movie. Wait, did you catch that reference?

Steve: It actually had a sequel. They made a second one. It's worse than the five Sharknado movies. It's like, oh, goodness.

PADRE: But it made money, Steve. It made money, so they'll make another one.

Steve: Yeah. All right.

PADRE: Know what I want to do? When's the last time I did Security Now! with you? It's got to be months; right?

Steve: Has been months.

PADRE: Well, I haven't talked about SpinRite in months, and that's just not right.

Steve: Well, and I thought of you because I got a tweet from someone whose handle is @FoodLovingProg, so I guess that's programmer. And he said: "@SGgrc Your 'stache is back." And it's like, yeah, it's kind of coming back. I'm not sure what I'm going to do. And he said: "Do you reckon I'd be able to recover a broken Nexus 5X phone with SpinRite? Could I get non-encrypted data off it?" And I was reminded of you because, first of all, there's a good chance. And then I was recalling that you had had some success in using SpinRite to recover from a phone. So since you were here, I figured I'd let you remind us.

PADRE: Yeah. So it was with an old OnePlus 1. And unfortunately it doesn't work on all OnePlus 1s. It depends on whether or not you've updated the software because on some of the early versions there was more or less a USB dumb mode that you could put it into, where it literally looked like a USB drive. There was no connector.

Steve: And that's what SpinRite would see, and it would be able to treat it like a drive and thereby recover the lost data.

PADRE: Precisely. Unfortunately, in some future versions, because they added a couple of features, especially in encryption, it looks like a USB drive, but only once you install some adapter software. And if you have that version of the operating system, it just won't work anymore because it no longer looks like a plain USB drive. It only looks like a USB drive if you have that piece of software, and SpinRite can't use that piece of software.

Steve: Right.

PADRE: So same thing for the Nexus.

Steve: So the takeaway would be, if your device looks like a USB storage device, or if you can put it into storage mode so that plugging it into a computer sees it like a drive, then yes. SpinRite will be able to treat it like a drive and recover the data in the way it does.

PADRE: Right. Unfortunately, the way the tweet seemed to be worded is that it's a broken Nexus. And if it's a broken Nexus, then I don't see how you could flip it into USB mode. If it doesn't even power up...

Steve: Ah, right.

PADRE: Unless you want to - you could desolder the chips from the phone.

Steve: No, no, no.

PADRE: Yeah. I mean, SpinRite does many things, but wirelessly reach into dead devices, that's not one of them.

Steve: Well, yes. And I've often said that what people are succeeding in doing, and we're often reporting, is that drives are dying, and they're using SpinRite to bring them back alive. And sometimes they're pushing their luck. It's like, oh, now it only lasts three months and I have to run SpinRite again. Oh, now only two months. It's like, okay, look. Ultimately SpinRite is going to fail at this job because, if the drive absolutely refuses to continue being a drive, it gets the final vote. So yes, by all means, use SpinRite to pull yourself back from the brink. But limit the number of times you do that because ultimately it's just going to say, sorry, we're now a doorstop here.

PADRE: I actually had an email exchange with a fan of the TWiT TV network, and she was upset because she said, you know, "You've been advocating SpinRite for so long, and it doesn't work." And I wrote her back. I'm like, "Well, what's the problem?" And she said, "Well, I've got this drive, and about 18 months ago it had a problem, and on your advice I used SpinRite, and it came back. But every time it comes back it lasts less and less. Now I have to run SpinRite every two or three days." I'm thinking...

Steve: Oh.

PADRE: And I'm like, oh, okay, I think I've spotted the problem.

Steve: Perfect example, Father. Perfect.

PADRE: Again, it's a great piece of software. It's not magic. It just seems like magic.

Steve: It'll save your butt over and over and over, but not infinitely.

PADRE: Right, right. Unlike the Dollar Shave Club butt wipes.

Steve: Oh, I can't believe...

PADRE: That's a callback, Steve. I learned that from Leo.

Steve: Yeah.

PADRE: All right. Steve, we do need to talk about the main topic, and that is Apple's wonderful, brilliant cookie solution. What did they do that has you so excited?

Steve: Okay. This is why I'm excited. First of all, it's been a constant recurring theme, probably for all 12-plus years of the podcast because, as I've explained it, we know why cookies were created. Cookies were created because browser transactions have no memory. That is, a browser says "Give me this page," and the website gives the browser back the page. Then the user clicks on something, and that's an entirely separate event. Now the browser says "Give me this page," and then the server gives it back that page.

So cookies were designed back by Netscape in the beginning, in the dawn of browsing, to create some cross-query connection so that, when the server sent back the page, it also gave the browser a cookie, just an arbitrary token, any string, such that subsequent queries made by the browser would send that cookie back. That created the possibility of session persistence. That's the way, that's entirely the way we are able to log onto a site is that your browser now has a nonce, this token. Then you say, oh, here's my username and password. You feed those in, and now this site associates your account with this token, and you are logged onto the site, entirely because of this mechanism of the cookie.

The problem is, as with the evolution of the 'Net and the cleverness, we introduced the notion of third parties, that is, you and the server being the first and second parties. Now there's a third party because, when the server returns your page, the browser will reach out to retrieve resources from wherever, not only the same server, but also a third party's, many times now, ads. And those ads are able to provide your browser with a token, which the browser will dutifully save and return every time the browser requests an asset from that same domain, from the advertising domain. And that's where we get tracking is that then, as you move around to different first-party servers, which all present ads from the same third-party site, the same cookie gets returned by your browser, which allows the advertiser to track your movements around the 'Net.

And my biggest problem with this is that it's never really been shown that this is useful. I mean, what it is tends to be unnerving because many people have anecdotal reports about how they were over on Amazon looking around for, I don't know what, tennis shoes or something. Then they go somewhere completely different, and they start seeing tennis shoe ads. So it's like, whoa, you know, I mean, it's like we have this illusion that, because this tracking is all underneath, it's behind the scenes, it's unseen, unless you add tracking awareness add-ons to your browser, in which case it's really in your face. You get a real sense for just how much of this is going on.

So this has been the problem that we've been facing for a long time. We've talked about NoScript had some cookie blocking. We've talked about turning off third-party cookies so that ads don't get those. We've talked about the do-not-track beacon, the DNT, which was hopeful for a while, but it was voluntary, and the advertisers didn't want to support it. So that sort of fell by the wayside. Then we've got things like uBlock Origin that gives power users more control. With, what was it, iOS 10 we began to get in iOS apps the explicit support for third-party ad and cookie controls with add-ons.

Okay. So now, as of this morning, all my iOS devices are updated now to iOS 11. It was a busy morning, and lots of multi-gig downloads. Apple has done something amazing. They call it "Intelligent Tracking Prevention." There is a link in the show notes, but I need to take us through this. We have time. And our listeners will understand what this means. So I'm quoting now from Apple's disclosure on WebKit.org of Intelligent Tracking Prevention.

They said: "Intelligent Tracking Prevention collects statistics on resource loads, as well as user interactions such as taps, clicks, and text entries. The statistics are put into buckets per top privately controlled domain," or what they refer to as "TLD+1." Okay. So for example, as we know, TLD is like .com or .gov or .org or .net. TLD+1 is one level down, like TWiT.tv or GRC.com. So it's the privately controlled domain.

"A machine learning model," they write, "is used to classify which top privately controlled domains have the ability to track the user cross-site, based on the collected statistics." Okay. So that means that they've implemented a heuristics monitor which will see this third-party advertising tracking happening. Again, remember, add-ons could do that. Attentive users could do that. Apple has built this in.

Based on the collected statistics: "Out of the various statistics collected," they say, "three vectors turned out to have strong signal for classification based on current tracking practices: sub-resource under number of unique domains, sub-frame under number of unique domains, and the number of unique domains redirected to. All data collection and classification happens on the device." So it's all done locally.

Then they say, under "Actions Taken After Classification" - so first there's this classification. Then they say: "Let's say Intelligent Tracking Prevention classifies example.com as having the ability to track the user cross-site. What happens from that point? If the user has not interacted with example.com [itself, that is, as a first party] in the last 30 days, example.com website data and cookies are immediately purged and continue to be purged if new data is added." That's the key. Meaning that this heuristically determines if you are actively visiting domains. And if you are not, if you haven't for a month, that is, if you don't go to a third-party advertising domain deliberately after a month, those cookies are removed. You stop being tracked by places you don't proactively visit as a first party.

PADRE: Now, Steve, that's an important part because a lot of people don't realize that's happening in the background. Even if I'm not actively using that cookie, that doesn't mean that new information and new data isn't being added to that cookie. In fact, that's a very popular way for advertisers to maintain a cookie indefinitely.

Steve: Oh, and believe me, they're not happy about this. We'll get there in a minute.

So Apple says: "However, if the user interacts with example.com as the top domain" - that is, as a first party - "often referred to as a first-party domain, Intelligent Tracking Prevention considers it a signal that the user is interested in the website and temporarily adjusts its behavior as depicted in the timeline." And there's a shot that you're showing on the screen now, Padre, which sort of explains how this works. And essentially, if the user interacted with example.com the last 24 hours, its cookies will be available when example.com is a third party. This allows for "Sign in with my X account on Y" login scenarios.

In other words, they had to figure out how not to break the log in with Google, log in with Facebook and so forth, the various OAuth flows which are third-party login technologies. I don't like them because they're also, as we've often discussed here, a privacy problem, since those people you're logging on with know where you're logging on with them. That's part and parcel with it. But it is a popular means of authenticating now, until we have something better, and we all know what that may be someday.

"This means users only have long-term persistent cookies and website data from the sites they actually interact with, and tracking data is removed proactively as they browse the web." Yay.

And, finally, under what they call "Partition Cookies," they say: "If the user interacted with example.com in the last 30 days, but not the last 24 hours" - so there's like a middle ground - "example.com gets to keep its cookies, but they will be partitioned. 'Partitioned' means third parties get unique, isolated storage per top privately controlled domain or TLD," that is, tracking is blocked, which is brilliant. So they say: "For example, account.example.com and www.example.com share the partition of example.com."

They say: "This makes sure users stay logged in, even if they only visit a site occasionally, while restricting the use of cookies for cross-site tracking." And then they say: "Note that WebKit already partitions caches and HTML5 storage for all third-party domains." So this is just - this is a wonderful, automatic, background, low-impact, privacy-enforcing heuristic that iOS 11, as of today, brings to the Apple platform. And not surprisingly, advertisers are freaking out. There are six major industry advertising consortiums that have just published an open letter blasting the new tracking restrictions Apple has unveiled and just implemented today, saying that they are, quote, "deeply concerned," unquote. And just two paragraphs I'll read, just so you get a sense for this.

They said: "The infrastructure" - this is the advertisers. "The infrastructure of the modern Internet depends on consistent and generally applicable standards for cookies, so digital companies can innovate to build content, services, and advertising that are personalized for users and remember their visits. Apple's Safari move breaks those standards and replaces them with an amorphous set of shifting rules that will hurt the user experience and sabotage the economic model for the Internet.

"Apple's unilateral and heavy-handed approach is bad for consumer choice and bad for the ad-supported online content and services consumers love." Oh, yes, we love those. "Blocking cookies in this manner will drive a wedge between brands and their customers and will make advertising more generic" - oh, darn - "and less timely and useful." Oh, double-darn.

PADRE: Ninety-nine percent of the time, if you hear a release from any corporation or consortium that talks about consumer choice, they're normally talking about the opposite of consumer choice.

Steve: And notice they still serve us ads. Everything still works. Ads will be there, impressions and clicks and all that. They just won't be able to as easily compile massive portfolios of our identity, which they're selling behind our backs and saying, oh, it's all anonymized. No, it's not. Not at all.

PADRE: Well, that's the thing about the solution. If they're only doing what they are publicly claiming that they are doing, which is serving you more relevant ads...

Steve: This won't hurt them. This won't hurt them.

PADRE: Won't hurt them at all. However, if they're one of these businesses that realizes that the data collection business can actually be more lucrative than the ad-serving business, then yeah, this is going to put a huge dent in their wallet. And guess what, it should. And they shouldn't have that. That should not be a source of revenue for them because I have not given them that permission.

Now, I love this idea of buckets. In other words, yeah, your cookie can still work. It just can't see any of the other cookies. And it can't see any of the other activity. And if you are being honest and saying you just want to make sure that when they come to your site that they get the experience that they've set up, then that still works just fine. Wow, you know, I can't see this approach not being copied by Google and Microsoft. This is going to be the norm; yes?

Steve: The only way around this is if we start seeing redirects to the advertisers, which would be annoying. That is, you click on a link, and the site you're visiting redirects you to DoubleClick.net, which then redirects you back because then that's a first-party visit which is transient, but that would serve to refresh your DoubleClick.net first-party presence. And again, this is Spy vs. Spy or cat and mouse. So if that started to happen, if there was an explicit attempt to work around this, then I'm sure Apple and WebKit would adjust in order to say, oh, those don't count, bouncing through, doing a non-user-driven redirect through what would nominally be a third-party site to allow it to register itself as a first-party domain. It's like, okay, that also we can have technology to prevent that. And as far as I know, maybe they already do. This just occurred to me as a means of bypassing it. Certainly they already thought about this. So maybe it's already protected.

But yes, Padre, I am - this is just yay. And I agree with you. Given that this happens, users are going to want this, and we're going to want it pervasively. And the system will adjust. It's not like, again, as you said, they're leveraging our trust without ever asking for permission, and they are ignoring request after request after request to stop it. They just say no, you can't make us. Well, yes, maybe we can.

PADRE: Yeah, exactly. And the thing about this approach is there's a lot of room for adjustment. They've got that three-day parameter right now, which it might not be long enough. Maybe people are going to find out that four-day or five-day is what they want. Or maybe others are going to realize, hey, you know what, if I haven't visited in 48 hours, it's probably not one of my primary sites, and I don't really care about it. And

Apple has the ability to play with those parameters.

But what I love, and this is what, when I was reading the story last night, this is what told me that they'd actually put serious thought into this. The idea that, even if a cookie receives new information, if it hasn't been actually interacted with by the user in 30 days, then it still gets eliminated.

Steve: Yeah.

PADRE: That means, yeah, Apple looked at what are the tactics being used that really kind of bend the rules, and that was one of them. That was one of the biggest tactics, which is I'm going to refresh my cookie every couple of hours. And then people wonder why things are working slowly or not working at all. It's because you've got a bunch of third-party cookies that are doing things they shouldn't be doing. If I put them in their own bucket, I let them play in their own little world, and they all think they're alone, then those third parties stop getting the information that they want, and they starve, which will cause me to play the world's smallest violin.

Steve: Oh, darn, yes. Oh, darn.

PADRE: Steve, that is brilliant. You know what, sometimes we have Security Now! moments of brilliance that go over everyone's head. But I think everyone can understand this, the fact that you've giving us, or Apple is giving us, more choice on how we handle cookies. The fact that Apple can give us an experience that is far more secure and yet really kind of accomplishes 99% of what we want, yes, that fits into the "brilliant" category.

Steve: Yeah. I mean, it solves the problem. It provides users with more privacy. It breaks this long-term persistent tracking model which has been - and this is - I've always said, this whole third-party cookie thing was kind of a mistake. It's like the idea was the cookie was associated with a domain. It was intended to allow persistent browser sessions. It was intended to allow us to log onto sites, not to allow sites we never visit to track us around the 'Net. So this has always been abusive, but there hasn't been a good solution. And now I think we've got a very useful heuristic. So bravo. As I said, and thus titled this podcast: Apple Bakes Cookies.

PADRE: Steve, it's been an absolute pleasure. And I can't believe this is the very first time ever that I've done the show with you that we have actually reached the end of the show notes, and we've gone through everything in the show notes. That doesn't normally happen. Normally we have to skip a few things.

Steve: Padre, I've learned to scale my ambitions accordingly because I know we're going to have much more fun with each topic.

PADRE: Oh, it's because I slow you down. Be honest. It's all good. Of course, folks, Steve Gibson is the mastermind behind GRC.com. If you haven't been there, then you haven't seen the best part of the Internet. It contains everything from ShieldsUP!, which you're going to have to run. Folks, if you don't know why you should be port scanning your own network, then watch Episode, I think it was 320, of Know How, and figure out what kind of information we can gather from you if you don't mind your own security.

He's also the purveyor of SpinRite, which you've already heard me talk about several times as the number one tool that I keep in my IT toolbox to recover drives that are dying, or maybe even service those SSDs that are going a little bit more slowly. And still anxiously awaiting SQRL. I'm looking forward to this, especially since there's a new renewed sense of purpose for replacing online identity, some way to confirm that we are who we are. Steve Gibson, you might actually end up being far, far ahead of the curve on that one.

Steve: Well, we'll see. We're going to make it possible and then hope that the world understands what we've done.

PADRE: That does it for this episode of Security Now!. Don't forget that we are live here on TWiT.tv every Tuesday at 13:30 Pacific Time. Steve will always be here to inject you with some healthy paranoia and help you understand the wonderful world of security. You can find all of our shows - I'm sorry, go ahead?

Steve: I was going to say, and we get you next week, as well.

PADRE: That's right, that's right. And next week actually, because I've got connections, I've already figured out what's going to be in the news. Next week, Equifax has fixed everything; Microsoft will announce that they're fully patched, and they never have to patch another version of Windows; and, oh, Google is going to announce that every Android phone dating back 10 years will be running Oreo.

Steve: And that would be the last episode of Security Now! then.

PADRE: Ever.

Steve: Because we'll be done.

PADRE: Indeed. Don't forget, you can find all of our shows at TWiT.tv/sn, as well as iTunes, Stitcher, and wherever fine podcasts are aggregated. You can also find high-quality audio downloads at GRC.com, which is also where you will find, as we mentioned, all of those wonderful projects, all those wonderful products that Steve has made over the years to keep you safe. Until next time, I'm Father Robert Ballecer with the one, the only Steve Gibson. And remember that, if you want to keep your data into the future, you need to think Security Now!.