SECURITY NOW!

Transcript of Episode #606

Proactive Privacy

Description: This week Steve and Leo discuss another iOS update update, more bad news and some good news on the IoT front, the readout on Tavis Ormandy's shower revelation, more worrisome anti-encryption saber-rattling from the EU, a look at a recent Edward Snowden tweet, Samsung's S8 mistake, a questionable approach to online privacy, celebrating the 40th anniversary of Alice and Bob, some quickie feedback loops from our listeners, an update on my projects, and a comprehensive examination of proactive steps users can take to enhance their online privacy.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-606.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-606-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. It's been a very, very big week. And I'm going to warn you, we were going to talk about how to protect your privacy in this new era where Internet service providers can spy on you any way they want, but we ran out of time. It's a two-hour show as it is. So Steve's going to defer that conversation to next week. But we do have a lot of security news, including Tavis Ormandy's shower thoughts and why it's a good thing he takes a lot of showers. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 606, recorded Tuesday, April 4th, 2017: Proactive Privacy.

It's time for Security Now!, the show where we get together with Steve Gibson, the king of so many things, I can't even just pick one thing. But I will say he is an autodidact and a polymath. And this show is really a great example of how he can find a subject, dive deep into it, and explain what it means to you and me. Hi, Steve.

Steve Gibson: My friend, great to be with you again for 606, Episode 606. Now, you know, 666, that'll be one to keep an eye on.

Leo: Yeah, that'll be a fun episode.

Steve: We've got another year to go. And fortunately we bypassed April Fool's. Because that just...

Leo: Oh, hate it, don't you?

Steve: Yes, I do. It's just, you know. And so you have to have your guard up, especially if you're in this industry, to like make sure - in fact, of course, theregister.co.uk never misses an opportunity. And I didn't even remember. Just, you know, I saw what they did. It's like, okay, fine. But I'm glad we did not have to make that decision.

Leo: It's a minefield. It's, you know, you just never know if what you're reading is true or not. In fact, briefly I thought George Takei was running for Congress. I briefly thought Google had bought Spotify. And I briefly thought that they had released a product called Google Gnome. But none of those are true.

Steve: Yeah, the only good news, I mean, because we have, like, the whole concept of fake news is big in the world at the moment. And so at least April Fool's Day would concentrate it all into one event.

Leo: That's true. Get all the hoaxes together, yeah.

Steve: Exactly. So what did happen that raised a lot of interest in our listenership in the past week has been, and actually I sort of predicted this last week when we got the news of the House of Representatives quickly passing, and without much apparent scrutiny, this rollback of some legislation from October of last year that the Obama administration had put in place, which has upset everybody - I mean, the EFF is all going crazy, and everyone's running around - the idea that ISPs will be able to collect people's browsing histories without their knowledge or permission and use that for revenue-generating marketing purposes. And of course then the Senate, as we expected, said, oh, yeah, we think that's a good idea.

Leo: Good idea.

Steve: Now, you know, this is, as we talked about last week, it's part of Trump's overall rollback of overregulation, which is one of the campaign themes. And so no one has any doubt that...

Leo: Oh, he signed it on Monday.

Steve: Oh, okay, good.

Leo: Signed it yesterday. There was no doubt. He signed it. He didn't make a big deal about it. He didn't show us the bill or anything. But he signed it, yeah.

Steve: Right. So I got all of this feedback from our listeners about VPNs. And I've run across a good VPN review site that I want to take a closer look at before I talk about it.

Leo: Okay.

Steve: But I want to do a - but the problem is that just using a VPN doesn't actually solve any problem because, if you use a VPN, the instant your browser touches the public Internet, Google will lock onto you, know who you are, and simply say, oh, they're now on that IP.

Leo: Right. I've even said you're just kicking the privacy can down the road.

Steve: Exactly.

Leo: Now it's not your ISP anymore, it's the VPN and anybody who sees you emerge.

Steve: Well, and I thought it would be interesting also to - because there's been a lot of misinformation, not surprisingly, in the popular press about whether, for example, Google and Facebook, how they operate as endpoints relative to an ISP. And so the argument being that an ISP has this concentration opportunity because all of your traffic runs through them versus a Google or a Facebook, where you have to go there. Well, that's not all accurate because largely HTTPS traffic is now going through the ISP, which they're blinded to; whereas Google and Facebook have an entirely different profile because, first of all, they're getting your decrypted endpoint traffic. And their little beacons are scattered everywhere.

So anyway, so today's topic, I titled this "Proactive Privacy" because I want to start by sort of taking a step back and looking at the whole terrain of the technology that our privacy is being compromised by because I and our listeners have no control over what legislation is enacted, but we do have at this point lots of tools available to us. So I just sort of wanted to kind of lay a foundation, looking at everything, looking at ISPs, the power of the ISP, the power of high spread endpoints, the power of a VPN, what does it provide you, what does incognito mode on a browser actually do, and, like, how all these little pieces fit together into a solution.

Leo: Good.

Steve: Which is how we'll end this podcast. In the meantime, we've got another iOS update update, more bad news, and some good news on the IoT front. We have the complete readout on Tavis Ormandy's highly productive shower.

Leo: Yeah.

Steve: Two Saturdays ago. And I'm going to be on The New Screen Savers on next Saturday with you, Leo, to talk about that also to that audience. We've got some more worrisome saber rattling coming from the EU, and it's really looking like we're going to get some anti-encryption legislation, probably first there. And again, in this current U.S.

administration, there seems to be little doubt that the same thing will happen here shortly afterward. So that's certainly relevant to us globally.

I want to take a look at a recent Edward Snowden tweet where I think he got it completely wrong. A mistake that Samsung made with the S8, which has gotten some attention. A questionable approach to online privacy. We're going to celebrate the 40th anniversary of Alice and Bob. Some quickie feedback loops from our listeners. An update on the status of my projects. And then, as I started talking about already, a comprehensive examination of proactive steps that users can take to enhance their online privacy and how all of these little bits and pieces interact. So I think a great podcast.

Leo: Good, good. I knew you would do this, yeah. I was counting on it. Of course, you know, we talked about NebuAd and Phorm when that was an issue. We talked about Verizon's supercookies when that was an issue. And I feel like this just gives these companies permission. It's not, I mean, this is no different than it was prior to October of last year. But now they've been told in effect by the government, nah, go right ahead.

Steve: Well, and you know, Leo, my greatest concern is that we're not far from them saying, from an ISP saying, well, if you're going to use our service, you're going to have to put our certificate in your computer.

Leo: Which means they'd be able to see all that encrypted stuff, yeah.

Steve: Which creates them as a middlebox that then allows us to hide nothing.

Leo: I can see that, at least I can see them attempting that, yeah, yeah.

Steve: Yeah. And so the idea that they're being permitted to aggregate this information, it's just a tiny step from that to, oh, and if you're going to be using our service, we need to be able to protect you from the nasty bad Internet, so here's a certificate you're going to have to use.

Leo: Right. All right, Steve Gibson.

Steve: So our Picture of the Week is just - it doesn't really relate to security. It's just I got a chuckle out of it. It shows someone in a bomb disposal getup, you know, to make them bombproof, with the caption, "On my way to pick up the new Samsung Galaxy S8." So burn...

Leo: Literally. Literally.

Steve: Ah, yes. Like I love the Internet.

Leo: Actually, you've got to figure the Samsung S8 will be the safest phone ever because the last thing Samsung can afford is any more problems with their batteries; right?

Steve: Yes, yes. Boy, I tell you. You know, I was flying a little bit, well, I flew up at the beginning of December to do the Christmas special podcast with you and to attend the Christmas party and so forth. And the airports are all proactively warning you against the Galaxy Note 7. It's like, okay, that's just really got to hurt.

Leo: Bad for business, yeah.

Steve: So we're often seeing a pattern with iOS updates where, like, very shortly on the tail of a major update comes a double-point release. And the same thing happened this week. Just yesterday, well, last week was 10.3, and on last week's podcast we sort of just got a snapshot of the massive number of security fixes that were part of 10.3. 10.3.1 dropped yesterday with a little bit of mystery. There's the statement that it is important because - but the only thing that Apple was saying was that an attacker within range may be able to execute arbitrary code on the WiFi chip. There's a stack buffer overflow which this update addresses.

And I dug around everywhere I could look. Nobody is saying anything more than that. So we don't know whether this was freshly introduced by the 10.3 update, or this is something that was just found that didn't make it in time, which seems unlikely. Or whether systems that did not yet update to 10.3 also have it.

Anyway, there just isn't anything more known. But I tweeted to my followers yesterday that iOS users should update again. It's another monster. It's 580MB or something like that, and it takes a long time for the devices to churn through it, but seems to be worth doing because what this means, I mean, it's not like a super, super dangerous, they can get you from some foreign land. You've got to be within WiFi range. But it looks like it's an arbitrary code execution if you're within WiFi range of someone's phone, so something you want to take care of.

So it's mostly just sort of a mystery. It's like, okay, well, it'd be nice to know more about this. There is a CVE number associated with it. And this did come from Google's Project Zero that no doubt notified Apple. So maybe they only just found out about it, and they just immediately pushed out an update. So, if so, that's good for them and good for us to get this fixed.

This one got a lot of press because - and I titled this the "We should have seen this one coming department" of Security Now!. And, like, every news outlet covered this because it was just so juicy. And the headlines were similar to "As many as 90% of Smart TVs are probably vulnerable to wireless hacking via rogue TV signals." And this podcast and our listeners could almost anticipate this because we've been talking about how anything that interprets a complex signal is very difficult to secure.

Well, look what's happened to Smart TVs. As we know, they are computers. And they're smart. And so a Swiss cybersecurity researcher, Rafael Scheel, who works for Oneconsult, showed a proof-of-concept attack using two different, fully patched, Samsung TVs. And this is not to single Samsung out at all. That's what he used. And it is extremely likely that most, if not all, Smart TVs are vulnerable. And the scary thing is that they can probably be taken over by broadcasting a malicious video image, that is, you know, in the same way that, for example, the multimedia interpreter in Android phones has had so much problems, because little edge cases were found in its operation.

Similarly, in this case, there is a web server that is always listening in the background. And an over-the-air signal can exploit a vulnerability that exists in the web server in these Samsung Smart TVs to allow root privilege access and an attacker to then get the TV to essentially reach out onto the Internet in order to make the network that it's on vulnerable. So this is not good. So it's not just, for example, the Vault 7 weakness that we learned about, the so-called "Weeping Angel" attack. And we talked about it a few weeks ago, and we said, yes, but that's a physical local attack exploit where you would need to stick a malicious USB dongle into the TV in order for it to take over. Well, now we've got one where just receiving a signal over the air can do it.

Now, this particular exploit used the European digital broadcasting standard, which is DVB-T, which is the predominant digital over-the-air system in Europe, comparable to our ATSC, which is our HD over the air technology. But it's the same. So it's not that DVB-T is vulnerable where ATSC isn't. It's just that this Swiss researcher used what he had handy, which was his own local digital over-the-air system. There's no doubt that you can do the same thing over ATSC and, actually, probably any of the total of four different standards that exist globally.

So this, again, our takeaway is a TV like other IoT devices really needs to be segmented. It's just, you know, where we are in the world today, these devices are not secure. PCs - Macs, Windows, and even Linux machines - have, due to their history and just the nature of the way they've come together over a much greater period of time, have vastly more mature security and, similarly I would argue, vastly more sensitive data on them than, for example, your typical light bulb. And you want to keep them that way. I mean, IoT devices have vastly less mature security and vastly less sensitive data.

So we really have two very different classes of devices. We have very secure systems with much more sensitive data in our PCs and much less secure systems with much less sensitive data and much less need for sensitive data in IoT devices. These separate classes should really be separate security perimeters. They should not be on the same network. The default currently is for them to be on the same network. It's predictable that in the future we will see routers that make strong network segmentation, truly secure segmentation, easy.

But for our audience, for those who are willing to do it, we already know we have the tools now to do that with things like the Ubiquiti EdgeRouter, which can create separate segmented networks where they can't see each other. And the problem is that smartphones are kind of in between. You often want to use, you want to control your IoT devices with your smartphone. Yet it also does have arguably a lot of sensitive information on it. I would suggest that somebody who was really security conscious should consider maybe a retired smartphone to use on the IoT side that you have scrubbed, and you've restarted it from scratch. You've scrubbed it, wiped it completely, and then use that as your IoT interface device over on the IoT network.

Leo: If you're security conscious, you shouldn't be using IoT. Right?

Steve: True. Well, exactly. I mean, our favorite acronym is IDIOT, you know, I Don't IoT.

Leo: Well, yeah. I mean, if you're really that worried, you'd be crazy to use these devices. They don't add that much convenience. The problem is, as you point out, the TV, everybody's got one of those Smart TVs. So that's not really an IoT device. That's just an appliance.

Steve: Well, it is - well, it is.

Leo: Well, no, it is an IOT device. But it's an Internet-connected appliance. And people don't get it for the IOT capability. They get it as a TV, but it just happens to have Internet on it.

Steve: Right. And DVRs are now on the Internet and doing things. And we've seen instances where DVRs are, like, for example, DVRs were being attacked by the Mirai botnet. So there's an example of - I guess my point is that these non-PC devices are less security mature. People do want to have them in their home. And if you are concerned about security, then taking the time to put them on their own network segment makes sense.

Leo: Absolutely, yeah.

Steve: And think about where your valuables are. Largely your valuables are in your PC. And most PCs don't have a need to be messing with your light bulbs. So they could be separated systems. I just sort of think that's the right way to think about it.

Leo: I'm not sure I'd agree with you that the issue is they're not mature. They are certainly for, like, some things. But those Samsung TVs are running Linux. It's not that they're not mature operating systems. It's that they're not paying any attention.

Steve: Right, right.

Leo: Really. I mean, Microsoft had to be forced to pay attention. They didn't pay attention either, at first.

Steve: Yeah. It's expensive to pay attention.

Leo: Right.

Steve: Yeah. The good news is Z-Wave, which is one of the major wireless, low-energy, low-power IoT systems, just on - it was weird. It was on April 2nd, which is Sunday, is the date of their press release. I don't know why they would have a press release dated April 2nd, but two days ago announced what they call "S2," which stands for Security 2. And this is the kind of move which is beginning to move us away from the pure Wild West of IoT. And the problem, of course, is there's already a huge install base of pre-

Security 2 devices.

But this has all of the kinds of things we want to hear. It's state-of-the-art security specification built into the latest SDK. Devices to achieve the Z-Wave compliance seal have to be using this updated SDK. It uses what's known as DTLS, which is something we've talked about in passing, but haven't talked about a lot. That's essentially TLS over UDP, or so-called Datagram TLS, which lowers the power requirements by dispensing with all of the back-and-forth traffic that setting up a TCP connection entails, allowing secure encrypted UDP packets which are quicker and easier and consume less power.

This also requires devices to have public and private keys. So for the first time they're getting asymmetric encryption using, happily, elliptic curve encryption, which also uses shorter keys and has much lower energy requirements because it's much easier to do the math for elliptic curve than the more expensive traditional RSA crypto. And they're using elliptic curve Diffie-Hellman to do key agreement that prevents man-in-the-middle attacks as long as you're able to authenticate the endpoints. And all of this is built into the lower level protocol supported by and supplied by the SDK.

So, again, all of this sounds like exactly what we need, where then third parties who want to simply have inexpensive IoT devices can get all of this just with a single click, essentially, adding their features on top of the SDK, and the underlying underpinnings are secure. So bravo to the Wave Alliance, which is the large group of companies that are part of this Z-Wave base. This is backward compatible, so of course that's a problem because they have to do it, they have to make it backward compatible to make it practical. But it does mean that, at least moving forward, once this SDK is what Z-Wave IoT devices are built on, they get the advantage of finally the kind of, I mean, again, there may be mistakes that have been made. But at least the policy, this represents the kind of security policy that we need in order to move forward. So, yay.

Okay. Tavis Ormandy's fruitful shower, which is really not a phrase I expected to be...

Leo: On the show. We'll be talking fruitful showers today.

Steve: We'll be talking about very fruitful showers. So this was the weekend, Saturday, weekend before last. Tavis, as we know, had an insightful shower.

Leo: An epiphany, he said.

Steve: Yes, an epiphany. So, and it was amazing. So, okay. Here's what happened.

Leo: It was amazing.

Steve: And we only know about Tavis's amazing shower because the folks at LastPass, as they always have done, got on this immediately. I mean, they were in a dialogue on Sunday. So Tavis was impressed. I'm sorry. I wrote "Travis." Tavis.

Leo: Tavis, yeah. Tavis is correct, yeah.

Steve: Yeah, Tavis, yeah. So Tavis was impressed. The 90-day clock barely had a chance to tick before this was already resolved. And we now know all the details because all of us already have the update.

So here's what happened. This involved something known as "content scripts." So in a situation where you've got a web page, and then you have a browser extension, the browser extension is code that runs in the extension. But there's still the need to inject some JavaScript into the page itself. And that's called "content scripts" because it's JavaScript running in the page's content. And that's done, for example, I mean, that's something you still have to do.

For example, there might be URLs in the page which are not clickable. So the content script could run in the page, scan the page, look for URLs, and turn them into href links to make them clickable. Or it could be responsible for setting the font size as a function of what device the page is being viewed on. Or, for example, in the case of something like LastPass, it might add fields to forms, or remove them, or check them for security in order to warn the user that, oops, this is a submission that would not be secure.

So those, you know, so they're very useful. But so this is scripting that the add-on injects into and embeds in the page to essentially make little modifications to the page's content to enhance what it does.

So Google's own documentation about content scripts states: "Content scripts execute in a special environment called an 'isolated world.' They have access to the DOM" - that's the Document Object Model which is, for example, the actual page's formal structure. And by having formalized the structure of pages, it's then possible for scripting to understand, to interpret and understand the page and make safe modifications or parse its content and process it. So that's the Document Object Model.

So Google writes: "They have access to the DOM of the page they are injected into" - that is, these content scripts which execute in a special environment called an "isolated world" - "but not to any JavaScript variables" - this is Google saying this - "not to any JavaScript variables or functions created by the page. It looks to each content script as if there is no other JavaScript executing on the page it is running on." So complete isolation, thus called an "isolated world."

And Google says: "The same is true in reverse: JavaScript running on the page cannot call any functions or access any variables defined by content scripts." Google says: "Isolated worlds allow each content script to make changes to its JavaScript environment without worrying about conflicting with the page or with other content scripts. For example, a content script could include JQuery v1, and the page could include JQuery v2, and they would not conflict with each other.

"Another important benefit," writes Google, "of isolated worlds is that they completely separate the JavaScript on the page from the JavaScript in extensions. This allows us to offer extra functionality to content scripts that should not be accessible from web pages without worrying about web pages accessing it." Okay? Except that's not true. That is, everything I just read, uh, not quite true.

Leo: It's a vision they had for it.

Steve: Yes, it would be nice. Okay. So just to step back a bit, all of that, the whole need for that is a direct consequence of what a total cluster you-know-what JavaScript is. I mean, it is a catastrophe. It is an abomination. But it's what we have. As we know, it began a long time ago with Netscape that wanted to make pages more active. And they also wanted nonprogrammers to be able to do this. Just they didn't want to have to declare variables, not even to say, okay, is this an integer? Is this variable going to contain an integer or going to contain a string? They said, uh, let's just kind of have it figure it out. And, oh, I mean, it's just - so sort of it's trying to be everything for everyone. And as a consequence, it's just - it's an abomination. But it's what we've got.

So as a consequence it's sort of - it's self-typing. It's got something called "garbage collection," the so-called "automatic language," where if the interpreter figures out and it's its responsibility that you're no longer using a variable, then it garbage collects it because the variable becomes garbage, and so it frees the memory that it's associated with. I mean, it's just it's an incredible nightmare. But it's what we have.

So, for example, there's no notion of what's called "namespaces." In a formal, well-designed language, you've got this concept of a namespace. And having separate namespaces allows code to use a certain name, like a variable "n," and for different code existing in a different namespace to also freely use whatever variable names it wants, like "n," and not have those two different n's refer to the same thing. But JavaScript doesn't have that. There is no namespace concept in JavaScript. So for a long time, and even today, code that's going to run in a page may very well stomp on the variables which the page uses, unless it's extremely careful not to.

So consequently there's this notion, there's all kinds of hoops have been created that code has to jump through. One of the techniques is called "closures," where you're able to, like, enclose all of your JavaScript within a single variable, if you can believe it, and just worry about that one variable being unique, and then the object-oriented things that occur within this closure, I mean, it's just a catastrophe. So along comes this, okay, we're going to create isolated worlds.

Okay. Well, it turns out there is a place where worlds collide. And this is what occurred to Tavis on a Saturday morning, March 25th, in the shower. He was no doubt ruminating while he was lathering up about the code that he'd been reading. You know, he'd been going through LastPass's code, looking at it, and going, "Okay, okay, hmm, hmm, hmm, hmm," you know, as Tavis will, looking for anything that seemed wrong. Well, one of the problems in JavaScript is you do not need to define a variable. You can, you know, and my code does. But you don't have to. So the idea is that the first time you use it, the JavaScript interpreter goes, oh, and kind of looks at what you're assigning to that variable and goes, oh, that looks like a number. Okay, fine. We'll be an integer today. Or, oh, that looks like a string? Okay, fine. And it also freely mutates them back and forth as needed.

It turns out it's handy to be able to ask JavaScript if a variable has been defined, that is, are you currently aware of a variable by this name? So you can use the "type of" function to ask for the type of a variable, like, are you an integer? Are you a string? And you can even say, are you undefined? So one of the things you can do is ask whether a variable has been defined or not.

So it turns out that the LastPass authors used the undefined property as a means of turning on their own content, their injected content, in order to enable it or not. That is, they were using this notion of these completely separate worlds, these isolated worlds. And so throughout their content scripts, which they inject into the page, they use "type of" and then a variable, whether or not it's equal to "undefined." So that's a way for them to determine whether some secure code should be allowed to run or not. If the thing is not defined, then don't do this because they're using that as a switch.

What hit Tavis in the shower was that one of the ways, one of the things that the isolated worlds do have in common is the Document Object Model. And although a page cannot directly define variables that would collide, the isolated worlds do share visibility into the same Document Object Model. That is, there's only one page. So even though the worlds may be isolated, the place where they have visibility, the place where they intersect, essentially, is in all being on the same page, that is, having the same Document Object Model. And it turns out that the Document Object Model can assign properties to objects in the DOM, which has the side effect of defining the variable. And so, believe it or not, somewhere during his rinse cycle, Tavis realized, holy crap, LastPass's approach is to use the undefinedness of their variables to turn code on or off.

Leo: Oh, yick.

Steve: Yes.

Leo: That's awful.

Steve: Yes.

Leo: Do you think Joe Siegrist did this? Or is this something that was done later?

Steve: I have no idea.

Leo: That's sloppy.

Steve: Well, it's JavaScript.

Leo: No, but there's no excuse. I mean, as soon as you say "undefined," I mean, that...

Steve: And that's why I'm saying it is completely valid.

Leo: It works.

Steve: Because this is how horrible JavaScript is.

Leo: Well, it lets you do that. But then it's incumbent on you...

Steve: No, it's formally...

Leo: It encourages you to do that, yeah.

Steve: Yes, exactly. It's formally something that people do. And I don't know, I'm sure, Leo, that you've used systems enough that you've run across apps where something will say "NAN," and it's like, what? NAN? What's NAN? Well, that's Not A Number. And that's literally the way JavaScript tells something that you're misusing. And sometimes that pops out on the UI. I mean, JavaScript is truly just a horrific abomination. But it's what we have.

Leo: Right.

Steve: So Tavis got a hold of the LastPass guys and said, "Guess what, guys, it's actually possible for the DOM, code in the DOM to assign a property to a DOM element which will cause a variable to no longer be undefined. That turns code on in your code that you never intended to have turned on unless you were in control." And they're like, "Holy crap."

Leo: Yeah. See, that's what's sloppy. You know, you're taking advantage of a side effect. I mean, that's sloppy, I think.

Steve: Well, I think it's brilliant on Tavis's part.

Leo: Well, to figure it out, yeah. No, but I'm saying it's sloppy for LastPass to have used that.

Steve: I disagree.

Leo: Really?

Steve: I don't mean to be [crosstalk].

Leo: It's colloquial in JavaScript.

Steve: Yes, exactly. I mean, it's what everyone does. And it just - but should they not have done it? Well, yes. Did they regret it? Yes.

Leo: Yes.

Steve: And how many instances do they have to fix? More than 3,000.

Leo: They did it a lot. They did it a lot.

Steve: Yes.

Leo: This was a colloquialism they liked.

Steve: Oh, yes. It wasn't one place. It was, I mean, and again. And so that demonstrates that this is sort of - this is what you do in JavaScript. So, and everybody could look at that and see nothing wrong with it. That's really my point. And notice that it, I mean, on one hand, I guess I'm, you know, yes, they wish they hadn't done it. Now they are saying they're setting all of those variables to false, and they're checking the Boolean-ness of them, you know, is it true or false. Or they're setting it to negative one, or they're doing something so that there isn't this undefined-ness any longer. But anyway, again, hats off to Tavis for, like, just ruminating on some code that he saw and going, oh, wait a minute, you know, that's not safe to do.

Leo: Yeah.

Steve: And I know that LastPass learned a lesson. I hope all other JavaScript authors pay attention to this because, again, this is not something they technically shouldn't have done. This is a consequence...

Leo: Well...

Steve: It isn't.

Leo: It's allowed, and it might be colloquial, but clearly it's a bad practice. Right?

Steve: I don't think so. I mean, again, that's why I would say every JavaScript author writing secure software needs to make sure they're not doing the same thing.

Leo: This wasn't in "JavaScript: The Good Parts," I can promise you. Of course, it's a very thin book.

Steve: No. Remember, yes, exactly, I was just going to say, remember it was one of our photos. There was, like, "JavaScript: The Bible" was two inches thick, and "JavaScript: The Good Parts" was, like, you know...

Leo: Yeah, tiny.

Steve: It was like the Appendix.

Leo: The problem is programs want to be clever. They want to save space. They want to show off for whatever reason. And so this is a clever trick, but it bit them. I just don't think - don't be clever, be explicit. I don't know. I just - I feel like that's sloppy. Now, it does raise some questions. They fixed it on Friday; right?

Steve: If it weren't formally, see, it's in the formal language definition.

Leo: But the side effect of testing a variable to turn on your code isn't.

Steve: True.

Leo: So that's going a little - that's clever. That's saying, oh, you know, I think you should be more explicit about that; right?

Steve: I don't disagree, yes.

Leo: I mean, I understand that the side effect is in the spec. But you shouldn't rely on it to turn on code. Anyway, so but my question - here's the question. A couple of questions come to mind. Well, of course LastPass fixed it. I immediately, though, and I will not put it back, took the Chromium extension out. I am not going to use the autofill feature on LastPass because that's the JavaScript part. I just stopped doing that because I don't trust them, if they did that. That's one bug they've, you know, actually it's the third this month. But that's one bug. But who knows what other cleverness is in there? Right? Doesn't this tarnish LastPass? I guess that's the question.

Steve: Only in the eyes of non-security-aware people. And seriously, I would say that, first of all, what you did I completely understand. But it shouldn't be LastPass, it should be any password browser extension.

Leo: Yes, yes, yes. And that's what I did, yeah. I took them all off, yeah.

Steve: So the lesson I think we are learning is browsers are just not secure enough to be trusted to have all of our passwords. That is, it is too difficult to secure them. So, you know, I'm still using LastPass because I would rather use a piece of software that it takes serious Tavis showers to find problems in. In other words, Tavis has found problems in every password manager he's looked at, and we've talked about other ones. LastPass is just the biggest target, and it's got the largest market share. So it's like, do you fire a person for making a mistake? Or do you recognize that they've learned from their mistake, and they're a better employee for having done that? I mean, and people come down on both sides of that question.

I guess what I focus on is policy. And LastPass I think from day one has had the right policy. They've jumped on this stuff immediately. They fix problems within hours. And so I consider it to be a well-vetted password manager. Someone could certainly choose a less well-examined password manager.

Leo: Well, that's the problem. What do you use instead; right?

Steve: Precisely. But we talked last week about KeePass as a...

Leo: It's open source, yeah.

Steve: Multiplatform, open source. If you're no longer comfortable with integrating a password manager with your browser, then using an external repository that is encrypted and multiplatform, I completely understand that someone might choose to make that choice.

Leo: Yeah. And, you know, the other very important point, and the one I make, see, I'm asking these questions because I have to make a recommendation to neophytes on the radio show and stuff. And it still is far better than what they're doing, which is reusing the same password all the time, or making up passwords that are combinations of birthdates and kids' initials. So it's better to use a password vault no matter what. These are obscure bugs, hard to find, hard to take advantage of.

I still want to make the assertion that it does show a sloppiness in programming that worries me. I understand this is idiomatic JavaScript. It's widely used. But this is a security product. And I worry that it shows a lack of attention to the detail in the JavaScript portions. I'm not going to use the JavaScript stuff anymore. I'm going to use their binary app. Unfortunately, there isn't one for Linux, or even apparently Windows. On the Mac it's easy. They have a standalone binary app you can look up passwords in. Then you have the risk of copying and pasting it. But there's risks everywhere.

Steve: Yeah.

Leo: I just worry, I think, I worry about their JavaScript practices. Do you know what I'm saying? I mean, it seems sloppy.

Steve: I guess my point is we wouldn't know unless Tavis had pointed this out. And he pointed it out because he was scrutinizing it deeply.

Leo: I understand that. But isn't it - maybe I'm wrong, and correct me. It seems like bad programming practice to use the side effect of a test on whether a variable exists to turn on code. That seems like just poor practice. I understand it's idiomatic, JavaScript allows it, and many use it. But it does seem like poor practice. Maybe I'm wrong.

Steve: I think it's just as good, I mean, it's really no different than if you set the variables to false, that is, if you defined them all and set them to false and relied on that.

You can rely on...

Leo: Oh, but that wouldn't have had the bug, would it?

Steve: Not this bug. But it's the fact that you can define a variable through the object model creates...

Leo: Yeah, well, that's another thing. They should use strong typing. You can use strong typing in JavaScript. It doesn't force it, but you can use it. Why wouldn't they use it?

Steve: Well, so I guess my point is that JavaScript formally allows you.

Leo: I understand. But you're writing a security product. And it's well known in the general world - this is why I keep harping on, when kids say what language should I learn, I say learn a functional language. Start with LISP or something like that because it won't allow side effects. It punishes side effects. The compiler - and it has strong typing. You should have strong typing. I don't have a problem with closures or garbage collecting.

Steve: Hey, you're talking to an assembly language programmer.

Leo: Well, I know you don't have garbage collecting. You do it all by hand.

Steve: Talk about strong typing.

Leo: Yeah, right. Well, actually you have no typing.

Steve: Correct.

Leo: You can stick anything in that register. You just have to keep track of it.

Steve: It's entirely up to you, exactly.

Leo: But in a way that actually is good because you know that. You don't rely on some weak mechanism. You know you can't just assume what's in that register.

Steve: Correct. There are no assumptions.

Leo: Yeah.

Steve: They always bite you.

Leo: Yeah.

Steve: So I guess my point is, you know, I'm staying with LastPass; although, again, when I do the math, when I think about an external password vault, it can generate a random password. I can copy and paste it in. Now, the one thing it doesn't protect us from that an integrated password manager does is site spoofing, where the URL is a lookalike URL. So if you go to a spoofed site that looks exactly like PayPal or eBay or something, or Amazon, you'd be more likely to cut, copy, and paste your credentials into that; whereas, if you have an integrated password manager, it's not going to fill the form in, and you're going to go, wait a minute. Why doesn't it recognize this site? And so it would tend to bring that to your attention. And arguably that's still a major security vulnerability that no one has come up with a solution for.

Leo: Let's say I'm listening to this, and I, you know, just on an excess of prudence, decide to move to KeePass or Pass or some other more secure but much less convenient solution. But LastPass, how do I null my LastPass vault on their servers? Should I just erase all my passwords and let it sync? I guess I could do that. I kind of wish there were a way that you - maybe there is a way that we can go to LastPass and say, "Can you please delete everything? I don't trust you."

Steve: Okay, well, there's never been - okay. All of this is endpoint.

Leo: I know. I know, I know. I know. There's never been a breach. I know.

Steve: There's never been a breach. But more importantly, and the reason I originally recommended them, was that the crypto technology gives them zero visibility into our stuff.

Leo: It is Trust No One, right, on that password vault?

Steve: It absolutely is.

Leo: Except that, if somebody did breach their vaults and downloaded it, and it is, you know, it's going to be a target because that's where a lot of people store their passwords, they would have, at their leisure, time to try to crack it. But they're using PBKDF2, and they're using all sorts of salting, yeah.

Steve: Yeah. And actually all of that is done on the endpoint side so that the only thing that's there is an absolutely high-entropy key. So you actually cannot third-party crack...

Leo: You couldn't brute-force it.

Steve: ...the LastPass cloud. So stopping using LastPass and expunging it from your life, that's enough to prevent anyone from getting access to your stuff. The only, as far as we know, the only way in is through one of the endpoints because the endpoints are what have to be able to access that vault.

Leo: Right. That's the vulnerability, of course. And that's why I think it is prudent not to use the JavaScript plugin; right?

Steve: I have to say - but again, I continue to feel that, if you are going to, I would use LastPass.

Leo: Yeah.

Steve: Because it's been...

Leo: Over any other JavaScript plugin, right, right, right.

Steve: Precisely. Precisely. Because it has been heavily vetted. If Tavis turned his showerhead on anything else, they would melt like butter. This thing has been, LastPass has been pounded on, again.

Leo: And the good news is hackers don't take as many showers as Tavis Ormandy. So we're safe in that regard.

Steve: And on that note I think we should take a break.

Leo: Great discussion. I really wanted to talk to you all about this because, you know, there are so many questions. But I think I will, based on you and this discussion, I will continue to recommend LastPass to all our users. And even, you know, because convenience is so important to normal users, even the browser plugin. Because they're more likely to use it, generate good passwords and all of that.

Steve: Yeah. And we have yet to actually have an exploit. What we have is every website out there has lost the control of their passwords. But we haven't actually had a single, that we know of, instance of exploitation. So that says, exactly as you say, Leo, for the typical user, LastPass is what you want to use more than not. For the super security-conscious person, I endorse what you're suggesting, which is go to a standalone vault solution and then port the data into the page on your own.

Leo: I like - there's a solution, I wish it were more cross-platform, called Pass, open source solution, that uses PGP keys to encrypt the passwords. So it's on your drive, encrypted with your PGP key. It's a password vault, but it doesn't have any convenience at all. You have to, you know. And we know that clipboards are dangerous. So you'll be tempted to use the clipboard, and that would be a bad thing.

Steve: Oh, and as a matter of fact, a friend of the podcast and someone who hangs out and contributes in GRC's newsgroups, Greg Bell, reminded me last week. I was talking about the clipboard problem relative to KeePass, and he said that KeePass does a keystroke entry.

Leo: Yeah, yeah. Isn't that cool? On Windows, yeah, yeah.

Steve: So that's much nicer, yes.

Leo: That's really nice, yeah.

Steve: And you and I are going to talk about this again on The New Screen Savers on Saturday.

Leo: I know. At less length because it's only an hour show. But we absolutely will. Thank you, Steve, as always. And I don't think I'm alone. I think we all wait till Tuesday, and we go, okay. Got to hear what Steve says on this one. So thank you, I appreciate it. We've got more, lots more to talk about.

Steve: So last week we talked about the U.K. Home Secretary, Amber Rudd, and her post-terrorist attack rhetoric about the need for law enforcement to have access into encrypted communications.

Leo: [Growling]

Steve: I know. And in fact she singled out WhatsApp, although also mentioned Telegram, Signal, and so forth. Now, a couple days later, the EU Justice Commissioner, Vera Jourova, said actually a week ago, last Tuesday, that the entire European Commission is planning to propose new measures this June, so a couple months from now, to make it easier for police to access data on Internet messaging apps such as WhatsApp. Now, of course, we know that the technology has been designed to thwart that. So simply passing a law is just not going to make it immediately so.

Anyway, Jourova said she will announce three or four options, including binding legislation and voluntary agreements with companies to allow law enforcement authorities to demand information from Internet messaging apps "with a swift, reliable response." And this announcement comes as interior ministers from other EU countries have amped up pressure on the Commission to introduce new rules to help police crack through secure encryption and demand private data for investigations.

And the position they're taking is interesting. They're saying that non-legislative measures will be provisional "to have a quick solution" because they recognize that negotiations over EU laws can drag on for years before they're passed, unlike in the U.S. where we now - where laws that we want to pass apparently only take a few days to get signed into law, or to have laws overturned.

Anyway, finally she said that: "At the moment, prosecutors, judges, also police and law enforcement authorities are dependent on whether or not providers will voluntarily provide the access and the evidence. This is not the way we can facilitate and ensure the security of Europeans, being dependent on some voluntary action." Jourova also said that the measures would make it easier for law enforcement authorities to request and access data from online services that are registered outside their jurisdictions. And both the French and German interior ministers have followed suit and said, yes, they want the same thing.

So as we've been predicting for quite some time, it doesn't look to me like the current position that the developers of these applications have taken in the wake of the Snowden/NSA revelations are going to withstand the future legislation. I think that what we're going to end up with is, my guess is, not any notion of a single golden key, something that law enforcement can have to unilaterally decrypt communications. But it'll be the case that individual app providers will have to build some means to respond to court orders in order to give authorities access to specific communications, given a warrant. I'll bet you that's what we're going to end up with here, at least in the U.S. I don't know enough about the way the laws work in the U.K.

Leo: The really good news in the long run is trust the math. It's already out there. Strong encryption exists. It's available, not just from U.S. companies but companies all over the world. And so people who want strong encryption and understand how to achieve it are never going to lose that. But the bad news is people who use Apple Messages or Facebook or WhatsApp and assume that they're encrypted and protected will no longer be.

Steve: Well, I would say they will be - the reason I think this is what we're going to get is it'll be like the search warrants we have now, is we have this notion of illegal search and seizure. You need, for example, a search warrant in order to break into someone's home and have legal access to the contents. Similarly, I think we will see legislation where a warrant is provided to an Apple or to a Facebook compelling them to provide within some range the decrypted communications on a case-by-case basis. So people feel in the U.S. safe against the police knocking their door down. And I think people will feel safe, or should feel safe, that the default is that their communications is not in the clear, easily snoopable by people without legal warrant to have access to it. So it's the way the U.S. and the Constitution we have has been slicing that. And to me, I think that's what we're going to end up with.

And what I guess part of this will be, I guess the other question will be, and that's the point you raised, Leo, what about third-party solutions? That is, will using undecipherable communications be outlawed? That's the other shoe is that it's one thing to say Apple and Facebook and, like, mainstream providers have to provide, have to be able to respond to a search warrant. But what about an individual who uses his own OpenVPN, for which there is no known backdoor, and it's open source, and we trust it as much as anything? What about that? That is, you know, a point-to-point private link that is not from a major provider. As you say, trust the math. We've discussed this before. Will the use of non-decryptable communications be outlawed?

Leo: Well, that's the endgame. It's the only thing you can do.

Steve: Yeah, I know.

Leo: You have to say it's illegal to use WhatsApp. And then outlaw crypto, and only outlaws will have crypto.

Steve: Right.

Leo: So, you know, but given the administration and their attitude towards this, it seems to me, as it does obviously to you, inevitable.

Steve: I think it's a fait accompli. I think it's a matter of somebody writing up a bill, and the House and the Senate will pass it, and our President will sign it, and then there will have to be some technology redesign in order for the companies producing non-decryptable communications to be able to respond to court orders.

Leo: It's inevitable that the White House and Congress will go for this. But remember the intelligence agencies do not necessarily support this. The NSA understands putting backdoors in crypto puts the nation at risk. And they have said this many times. So they will get - I think there'll be some pushback from our intelligence agencies, of all people. The same people who are saying we need backdoors, when the FBI says, "We need backdoors," the NSA responds, "Shut up. You don't want backdoors." There's disagreement in the intelligence community. So it's more - I'm not - it may not be a fait accompli. But it's going to come up, for sure, for sure.

Steve: Well, Apple, to take a case in point, Apple could argue that this puts too much responsibility on them. That is, they've designed a system where they cannot see into their customers' communications. And they don't want to be able to see into it.

Leo: Unfortunately, they can see iCloud, and they easily provide that information to law enforcement. And that's kind of their tit for tat. They're saying, well, we won't let you look at messages, but you can always look at the iCloud. That's their way of kind of appeasing law enforcement.

Steve: Yeah, I mean, you know, law enforcement does not want, I mean, no government ultimately wants their citizenry to be able to have absolutely private communications. They just - they don't.

Leo: They can't spy on them. They can't figure out they're terrorists. They can't figure out what's going on.

Steve: Precisely.

Leo: Now, I have to say this is not speculation. We know what the impact of this is because we tried it once before. The government outlawed strong encryption in browsers, remember, and enforced 40-bit encryption. And to this day we have weak encryption in browsers. We're fighting this battle 20 years later because of this bad policy in the '80s.

Steve: Yup.

Leo: And we're still - a lot of the breaches, many of the breaches we talk about are because of this.

Steve: Yes. Gee, 40 bits? You think that's a secure key? Uh, no.

Leo: The government said, oh, no, no, you can't have strong encryption. Well, and then look what happened. So the problem is, yes, it helps us find bad guys. But it also helps the bad guys find us. The real problem is the phone. Because while I can use - I know how to use encrypted messaging. If my phone is not encrypted, all the stuff that's on my phone is available.

Steve: Right.

Leo: And I don't - at this point nobody's making phones that have strong - I don't know where we'd get that.

Steve: I didn't put this in the show notes today, but did you see this news that apparently part of this administration's border policy...

Leo: Yeah. If you're not a U.S. citizen coming into this country, they can ask you everything, including how you like Trump, what your phone numbers are in your contact list.

Steve: And they're now saying your social media account passwords.

Leo: All of your passwords to your accounts. All of that.

Steve: What?

Leo: Or you don't get into the country.

Steve: Ooph.

Leo: We're not going to have a lot of tourism in the next few years, I don't think.

Steve: No, in fact it's already dropped off. Well, speaking of dropping off, Edward Snowden tweeted something that I just thought, okay, Edward, that is so wrong. He said: "Huge," sounding like you know who.

Leo: Huge.

Steve: "Huge: USG confirms cyber offense funded at 9x the rate of cyber defense." Then he says: "Wonder why we can't stop foreign hacks? This is why." And I'm like, no, it's not. It's the most ridiculous thing I've ever heard. Okay. So offense and defense sound like symmetric things; right? You attack or you defend, like one for one. But they're only reciprocal in literally a one-on-one situation. When it's many to many, it changes it completely. So my point is that it is entirely different to attack a single remote entity than it is to defend against all possible attackers attacking all possible targets that you're responsible for.

So, I mean, I would argue that offense is a lot more sexy, and so it gets dollars. But the problem is there isn't a good - we don't have a solution for defense. I mean, in the same way that I said years ago that I would have a nervous breakdown if someone said I had to be in charge of Sony's security. It's like, no. You can't secure that. Well, so this notion that the reason we can't stop foreign hacks is we're not, as Edward suggests, we're not spending equally on defense as offense, that's just nonsense. We can't stop foreign hacks because we can't stop any hacks. I mean, because...

Leo: Geez. Oh, man.

Steve: ...our systems are porous.

Leo: Couldn't you spend some money on, like, protecting the electrical grid or something?

Steve: Oh, I mean, there are definitely - yes, I completely agree. There are things we are, for example, we're conspicuously not doing that we should, that those in the know are begging for money for, that they're probably not getting. And so I certainly would agree with Edward that the idea that we're not spending as much as we should to defend specific aspects of our infrastructure, I would agree with completely. But I just - the idea that he was putting offense and defense on the same footing in the Internet world just sort of seemed crazy because, again, it's easy to attack. It's incredibly difficult to defend.

Leo: Did you read, I don't want to distract you too much, but James Woolsey's opinion piece in The Hill about the real threat that North Korea would explode an air burst, use EMP to disable our grid.

Steve: Yes.

Leo: Causing riots within just few days as grocery stores and everything else went down. There was actually a response to it, a good response, I thought, from Popular Mechanics. But Woolsey is head of a group that is exploring the risks of EMP, electromagnetic pulse.

Steve: Good. I'm glad somebody is because that's, ooh, not good.

Leo: He was a CIA director in the '90s. And then the article is co-written by Dr. Peter Vincent Pry, who is chief of staff of the Congressional EMP Commission. I didn't even know there was one. But I'm glad.

Steve: They're actually - there's a non-nuclear EMP gun. And if the Portable Dog Killer hadn't done the job, then, you know...

Leo: But that's the funny thing about EMP. It wouldn't harm the dogs.

Steve: No. In fact, I've seen this EMP gun. You can just cause a car to stall. You just shoot this thing at it, and the car just, you know, it just stops.

Leo: I think ultimately all of the things we talk about on this show are going to be small potatoes compared to the real-world risks we're facing.

Steve: Yeah. So Samsung made a mistake with the S8 by allowing a person's face to unlock the phone.

Leo: Ooh, yeah.

Steve: You know, it took, like, what, minutes...

Leo: No, they did it at the demo. At the event somebody did it. They took a picture of themselves with their camera.

Steve: Seconds.

Leo: And showed it to the - but in their defense, Samsung did say it's a less secure method than the fingerprint [crosstalk].

Steve: Well, and thus is the problem. They have facial recognition, fingerprint, and iris security. And so what they've done is they've mixed toy security with good security and, as a consequence, muddied the waters. And I think that's a bad mistake. It's like - so what's the point of having your face unlock the phone if your daughter can wave the phone in front of you while you're sleeping and unlock your phone? And so it's either don't lock your phone, or have good security for unlocking your phone, rather than this basically toy security.

And so at this point - and so my point was that what they've done is they've introduced ridiculous security, which everyone is attacking, and said, oh, yes, well, that wasn't meant to be secure. It was meant to just sort of be fun. Well, okay. But what you've now got in the first week after launch is a huge amount of bad press and people laughing at your brand new phone for having bad security. So just it was a bad idea.

Leo: Yeah. But it's fun.

Steve: Yes. Yes, it is.

Leo: You know, there's a similar technology in Windows. But it has two cameras. It has a dimensional camera that can measure depth. And then you can measure the depth of an eye socket, you know, things that a picture wouldn't fool.

Steve: Correct. And what I was thinking was imagine having you, like, having to say something it shows you. Then you get speech, and you get animation of the person's face, saying what it's asked you to say, which would be - it would raise the difficulty of spoofing much higher.

Leo: Yeah, yeah.

Steve: But they didn't do that.

Leo: Actually, my banking app does that. It says "Blink."

Steve: Nice.

Leo: Yeah.

Steve: Nice.

Leo: Isn't that clever?

Steve: Nice. So, and here's another kind of, I don't know, questionable solution. But I know it's going to appeal to some of our listeners. So I wanted to share it. It's called Noiszy.com, spelled strangely: N-O-I-S-Z-Y dot com. N-O-I-S-Z-Y dot com. It is a Chrome extension which hides your actual Internet traffic among noise that it generates.

So the Chrome extension describes itself as, it says: "They're listening. Make some noise. Whatever you do online, you leave digital tracks behind. These digital footprints are used to market to you and influence your thinking and behavior. Congress has voted to allow ISPs to collect and sell your online information without your consent.

"Erasing these footprints, or not leaving them in the first place, is becoming more difficult and less effective. Hiding from data collection isn't working. Instead, we can make our collected data less actionable by leaving misleading tracks, camouflaging our true behavior. We can resist being manipulated," they write, "by making ourselves harder to analyze, both individually and collectively. We can take back the power of our data.

"Noiszy," I don't know how you pronounce it. "Noiszy [N-O-I-S-Z-Y] is a browser plugin that creates meaningless web data digital noise. It visits and navigates around websites from within your browser, leaving misleading digital footprints around the Internet. Noiszy only visits a list of sites you approve, and only works when you turn it on. Run Noiszy in the background while you're working, or start Noiszy when you're not using your browser, and it sends meaningless data to these sites as long as you let it run. This meaningless data dilutes the significance of your 'real' data by creating a campaign of misinformation. You become more difficult for an algorithm to understand, market to, or manipulate. You can outsmart the filter bubble."

To use Noiszy in Chrome, open a new tab. Click the Noiszy icon. Choose the sites you want Noiszy to randomly browse and click Start. Noiszy randomly chooses from your list of sites. Then it randomly chooses from and clicks on links within those pages, choosing only onsite links that don't open new windows. There are delays between clicks of about a minute. This makes web traffic appear to be more real and engaged. After about two to five onsite clicks, Noiszy randomly chooses another site from the list and repeats the process. This continues...

Leo: It seems like such a bad idea.

Steve: ...even when your tab is in the background, until you close the tab or click Stop. So there it is, folks.

Leo: You're going to really annoy people. I hope you don't have bandwidth caps, that's all I can say.

Steve: That's right. Don't do this on your cell phone. Don't leave it running by mistake. Anyway, I got a bunch of people saying, hey, what do you think? I don't know. I think maybe we will wrap up this podcast by talking about the proactive things you can do to use the 'Net without being tracked, rather than try to throw a lot of garbage at the wall and hope that the stuff you care about isn't figured out. Because, frankly, it probably will be.

Leo: Yeah. We're in an interesting seesaw battle; aren't we? On we go, Steve.

Steve: So Mikko Hypponen, who is F-Secure's chief security research officer...

Leo: We had a great Triangulation with him. He is really, really neat. I love him.

Steve: Yes. He tweeted: "Happy 40th birthday to Alice and Bob! They were introduced in an April 1977 paper by Rivest, Shamir and Adleman." Those initials are R, S, and A. So Alice and Bob, for those who haven't read a lot of crypto papers, they first appeared 40 years ago in this classic paper as the endpoints in a conversation, or the actors in sort of a thought experiment, as Alice and Bob is A and B, essentially. So, okay. So this is 40 years ago, 1977. I was 22 and, you know, four years out of high school. That's how long ago this was. And I'll just read just the first three paragraphs of the abstract. And what I'm struck by is first how modest-sounding this is, and how, like, how we take it for granted today.

So the paper was titled, and for anyone who's interested I have a link to the original 40-year-old paper in a PDF which is at the AI Lab at MIT's website. It's titled "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," all which they had just invented. The abstract reads: "An encryption method is presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key. This has two important consequences:

"Number one, couriers or other secure means are not needed to transmit keys" - which had always been the case before - "since a message can be enciphered using an encryption key publicly revealed by the intended recipient. Only he can decipher the message, since only he knows the corresponding decryption key." And this was unheard of at the time.

"Number two, a message can be 'signed'" - they have in quotes because this was novel - "using a privately held decryption key." So this is sort of the reverse process. First you encrypt with a public key, which can only be decrypted with the private key. Now you sign with your privately held decryption key. "Anyone," they write, "can verify the signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a signer cannot later deny the validity of his signature. This has obvious applications," they write, "in 'electronic mail'" - that was also new at the time...

Leo: Brand new, yeah. This new thing.

Steve: ... "and 'electronic funds transfer,'" as we used to call it, "systems." So anyway, wow, four decades. And now, I mean, you can't imagine life without this technology.

Leo: It's so brilliant, too.

Steve: Yes, and we've had so much fun talking about how you leverage the concept of asymmetric keys, where they are separate, and they're related, but you can't get one from the other. And you can use them in all these amazing, clever ways.

So to close the loop with some of our listeners, the Fat Vegan Chef, and that's his Twitter handle, said, he said - he sent me a tweet. "@SGgrc Can ISP install a certificate that will give them access to secure traffic unencrypted? If/then browsing securely is" - he said "mute," but he meant "moot" - "and they can data mine all they want." Okay, so, no. If it were possible for an ISP or anyone on the Internet to unilaterally force a certificate on us, the jig would have been up a long time ago. The game would have been over.

What will happen if this does happen is that we will be asked, we will be required by our ISP to download an app. And running the app on our machine will install a certificate into our machine's root store. That, for example, there are, in an enterprise environment, there are means by which the enterprise's Intranet's scripting can do that to your systems without your knowledge or permission. That is, that's part of the privilege. But a browser cannot. It's got to be outside the browser, essentially at the OS level. So it won't be something that can just be done. It will require a proactive deliberate action from ISP subscribers. And I dread the day that that becomes part of what we have to do. I mean, that will - ooh, boy. Let us just hope it doesn't happen.

Simon Zerafa, who frequently contributes to the show through pointing me to things, he quoted somebody that just made me shudder. Frank Denis tweeted: "Support for Random.org as a CSPRNG" - that's a Cryptographically Secure Pseudorandom Number Generator - "was added to libsodium." And then has in parens "(will be the default soon). Note that this requires libcurl built with OpenSSL." And my response to Simon was, "Yuck. What a kludge." Though I suppose in an environment where there is absolutely no good local source of entropy, reaching out would be better than nothing. But it's also easy to see how a spoofed cert and a DNS intercept could redirect to a fraudulent source of entropy.

Anyway, so I just - I haven't looked any closer at this but I need to because libsodium is, or was, a secure library. But the idea of it using data from Random.org, which is a nice source of entropy, but it's over the Internet. And so you can't trust it. Now, I mean, maybe if it had certificate pinning so that it was tied to a verifiable cert, you know, that would help. That would prevent a man-in-the-middle cert decryption. But, boy.

Essentially what libsodium is doing is it must be assuming that it's on a platform where it just cannot generate any high-quality entropy itself. We know that crypto needs high-quality entropy. So maybe as a last resort it does this. But, boy, I hope that anyone using it understands that's not just an immediate win. It's got to be something that you do as a last resort.

Mark Gottselig asked: "Is there an IoT security grade or review website you trust that we might be able to use to find secure IoT devices?" Today I know of no such site, actually because I know of no secure IoT devices. Something like what we were talking about earlier, like this Z-Wave Alliance initiative. The problem is we're just too soon for that. Right now we're still in the Wild West stage, and people are selling these things like crazy.

The good news is there's been a lot of attention focused on the lack of IoT security. And those people who created the first wave are now saying, okay, we're going to have a problem selling this stuff unless we get rid of this reputation that IoT has as being so insecure. That'll happen. Then things will have to settle down. And then maybe we can start taking a look at rating the security of the different technologies that we have.

Isaac says: "If you're so worried about the government might do in the future to weaken crypto, then it's our duty to create it and resist." And the problem is, as we talked about earlier, I completely agree until they make it unlawful. I mean, I'm happy to create good crypto, and I'm happy to resist. But I can't break the law. And that's why I stopped working on CryptoLink years ago because I felt this coming, and I didn't want to invest years in creating a high-quality commercial product that might be outlawed by my own government. So that's the problem. And as you said earlier, Leo, if we outlaw crypto, then only outlaws will use it.

Leo: Well, that's the thing. Anybody who's motivated could figure out how to do it.

Steve: Oh, it's trivial to do. We all know how to do strong encryption.

Leo: You could put it on a T-shirt.

Steve: Yup. Right. Troy Carlson extended our "The 'S' in IoT stands for security," adding "and the 'P' stands for privacy." Which I thought was good.

Dr. Suarez asks: "Does the three-router solution protect privacy from ISPs if traffic is HTTPS and originates from the internal router?" And, okay. So ISPs, no ISPs can currently see into any HTTPS traffic. So the three-router solution is sort of orthogonal to that. It doesn't help or hinder. What we want is our traffic to be over HTTPS. In that situation, ISPs cannot see into our connections, but they can see to whom we are connecting. And the only way to prevent that, as we'll talk about in a second, is using a VPN.

An interesting tip from a listener Jim Clark. He said: "My wife asked me to stop commercials from autoplaying in Firefox. We have Privacy Badger, uBlock Origin, and HTTPS Everywhere," he says, but that still didn't work. He wrote: "I started thinking of all the trouble about NoScript." He says: "I searched about:config and found media.autoplay.enabled was set to true. I changed it to false. She is happy now. Love SpinRite." So, and I tried that. If you're a Firefox user, about:config, as we know, brings up a page with about a bazillion different little tweaks you can do. And then so you use the search box, put in media.auto. You only have to type a few characters, "au," and then up comes media.autoplay dot, and enabled will be set to true by default. If you double-click it, it flips it to false. And then nothing plays.

Now, the bad news is I was then unable to get a YouTube to play, even if I clicked on it. Firefox said, uh, sorry. So for me that wasn't practical. Whatever I'm doing, I guess I'm just not running into things that are autoplaying. Or maybe for me uBlock Origin is blocking them enough. I'm not seeing that happen. But for those who want to up the bar and are Firefox users, this really does shut down media autoplay. Unfortunately, it seemed to me that it also shuts down media manual play, which may be too much for you.

Leo: Isn't that weird. I was just looking to see if you could do that in Chrome, and there is, if you do a chrome://flags, it's kind of similar to about:config in Firefox.

Steve: Right.

Leo: These are experiments. There's one called "Gesture requirement for media playback." I don't know what the gesture is. Then there's media playback and cross-origin iframes requires user gesture. I'm just going to turn it on and see what happens.

Steve: Yeah, good.

Leo: I don't know what the gesture is.

Steve: Let us know next week because I know that it's a constant problem for you.

Leo: It is. Well, I've actually been using an extension that disables it. But like you, I find that it's very inconvenient because, when I want to play video, I have to turn the extension off and refresh.

Steve: Right.

Leo: It's kind of a pain in the butt.

Steve: We talked last week about how there are a new class of bots which are pounding on gift card website portals and draining people's gift cards. And we heard from a listener, @Liquidretro. Jon said: "I just checked some of my business's rebate gift cards. One had been drained due to fraud. Thanks for the heads-up." So it's not just theoretical. It's actually happening.

Oh, and I had an interesting counterpoint to the idea that ISPs might be forcing certificates on us. A listener, Alistair Campbell, said: "I'm skeptical that ISPs will be able to force install SSL certs, thanks to the proliferation of IoT devices with no interface for that." And I thought, ah, that's a very good point. We'll have to look at and see what IoT devices do. Are they not using HTTPS? Or do they have a minimal root store? I mean, they probably don't have in their little tiny chips the 400-some-odd certs that we have. So I wonder what their certificate chains look like, you know, how are they authenticating HTTPS connections, if they are. That'll be interesting to find out.

But Alastair raises a great point. If we have a large proliferation of non-PCs making HTTPS connections, then an ISP would not be able to force our light bulbs to accept their certificate. So that creates, I mean, that might be a deal breaker, then, for that entire concept, which, you know, yay. That would be great because the last thing I want is for my cable modem supplier to be intercepting my communications. That would move me probably to a VPN, I think. That just seems too creepy.

Oh, and one final thing. And somehow this didn't occur to me when I was talking about the difference between SMS second-factor authentication and the time-based one-time passwords, the TOTPs. I was grumbling that SMS is less secure, as indeed it is because every single time it's used you're relying on a cell phone text message to send you a six-digit thing, a one-time password rather than using an algorithm where you only need to exchange that sensitive data once over a certified TLS connection with a browser, which is probably a lot more secure, well, which we know is a lot more secure. And then after that just time and crypto is able to autonomously generate the proper one-time password. Anyway, this listener mentions the reason the companies want SMS security is so that they have your phone number. And it's like, oh.

Leo: Doh. Of course.

Steve: Yes. That hadn't, of course, that hadn't occurred to me. And I made a note here, I've been meaning to mention this for a long time. I can only reply to people through DMs, only because I just don't want to, you know, I've got 57,000 followers, and it just doesn't make sense for me to be tweeting replies to single people on my Twitter feed. So I often will reply using a DM, only to be told that, oh, sorry, that person that you're replying to is not following you.

So I just, you know, I just did want to mention that, if you ask me questions or send me things, I often - I would love to respond, if I can. But you have to follow me, not because I'm trying to make you follow me, but because I can't DM you unless you do. And I do have my Twitter set up so everyone can DM me, whether I follow you or not, because I famously don't follow anybody. And it's funny, too. Because I was looking at Snowden's Twitter feed. Someone sent me that tweet, and I wanted to verify that it actually was from him. And I remembered then, not only does he have three million followers, but he follows only one account.

Leo: What is that account?

Steve: The NSA.

Leo: The NSA.

Steve: The NSA. It's like, okay, Edward. Point taken. I wanted to give our listeners a quick update on SQRL. As I had mentioned last week, the server side is all finished. I'm now working on essentially bundling it up for installation, removal, and update. I'm adding the install, remove, and update, not that it, well, it needs update. But the install/remove is just because it's what people are accustomed to. They're going to download SpinRite into their Downloads directory. I mean, SpinRite. Sorry. SQRL. And then they're going to double-click on it. And so it needs to be able to "install itself," which it actually doesn't need to do because it's just an EXE. There's no, you know, it's just one thing. It's my kind of GRC code, 283K, and it's got all the multilingual stuff built in. It's the full implementation of the SQRL protocol with all of the bells and whistles and features. You just run it.

But most people are just going to click on the link and either download it or run it from GRC, and you could do that, too. So it needs to be able to move itself into the normal place in the program files and to add an entry into the add/remove programs list and, you know, behave itself. So I'm just going to - I'm adding that functionality, and also the ability to check for updates and update itself, either with permission or automatically and so forth.

Once that's done, then we're really close. I will look through to verify all of the various functions and make sure that the user interface jargon is consistent. And then I will, with great pleasure, be announcing it on this podcast for everyone to download and play with, in English only, at first. At that point, while people are playing with it, I will synchronize the online documentation with what we finally ended up with, that is, bring the SQRL web pages current because they've been lagging behind. I've just been working on the code.

And at that point I return to work on SpinRite v6.1 while we see whether SpinRite - I keep saying SpinRite - whether SQRL is able to gain traction. And so I'm still - I'm not going to invest any time in the non-English version. Certainly the website will all be in

English only. But as our listeners know, all of the hooks are in place to make it multilingual. But I'm going to get right back to SpinRite, to work on SpinRite 6.1, because I don't want to invest in multilingual until we know that it's going to work, that it's going to take off, that it's going to succeed.

If it begins to gain traction, after 6.1 is out, then I will be able to relatively easily make it multilingual. And we will use our listeners to help us translate the strings. All of the strings that exist in the product are in a single separate file. So I can simply publish that, and they're numbered. And all of the references in the code are by number. So if we simply translate the strings into their equivalent in other languages, it instantly turns this into a multilingual product.

And as for SpinRite, I'm going to come up with some way, and I haven't quite worked out the details, but I'm going to come up with some way to first make 6.1 available to everybody who has been supporting SpinRite and me through the years, before people can buy new copies of it. So it's my way of saying thanks to everyone who has been continuing to purchase SpinRite and supporting this effort to get to 6.1. I think everybody who has done that should get it first. So I'm going to come up with a way to do that. Okay.

Leo: Nice.

Steve: Yeah. That feels right to me, as a way of saying thanks for everybody's patience and support.

Leo: That's really great.

Steve: So Proactive Privacy. You know, Leo, we're out of time.

Leo: Oh, come on, no, you can't do this to us. Come on, we can be late.

Steve: Really? But we're at two hours of podcast, and we're at 4:00 p.m.

Leo: Yeah, I know, but how long do you think this will take?

Steve: Well, I've got a lot I want to say.

Leo: Twenty minutes? People are going to be frustrated if you don't...

Steve: Yeah, I know. But I just...

Leo: All right. It's up to you. I mean, if you need a half an hour, that probably is too much. And I don't want you to give it short shrift, obviously.

Steve: I really don't.

Leo: We want to really do it.

Steve: I really don't want to. So, shoot.

Leo: All right.

Steve: Let's have less news next week because I really...

Leo: Stop asking questions. Stop hacking things. No more news.

Steve: So I just - I have to say I'm sorry. But we just spent two hours. And I will make time. Next week, no matter what happens, less superfluous stuff because I really - I care about this, and I know our listeners do.

Leo: Yes, yes.

Steve: And, I mean, you know, the power of the major web players, the power of the endpoints, the power of incognito browsing modes, the need for a separate IP address is critical because otherwise you can get...

Leo: There's a lot to talk about, yeah.

Steve: Yeah, there is. There's too much. And I just - I don't want to be in a hurry, and I don't want to be forced to...

Leo: Good. And I want to interact with you on this.

Steve: Good.

Leo: So, yeah, you're right. We shouldn't have to rush through this. So we'll do it next week.

Steve: Okay. I will make a lot of time for it next week.

Leo: That's fair enough.

Steve: Okay, buddy, everybody.

Leo: Although there's plenty of people wish this was a four-hour podcast. I just want to say.

Steve: Just saying.

Leo: It's always an option.

Steve: Just saying.

Leo: We'll give you - I'll give you a whole day.

Steve: Hang an IV over me.

Leo: I do have a quick plug because we did get nominated, I'm very pleased - you often get nominated for the podcast award. The Webbys have added a podcast category, and we are nominated in this podcast category for the first time, which I'm very pleased about. If you go to TWiT.to/webby2017, you'll see the People's Voice Awards. And we're nominated in the Technology category for Triangulation. But I consider it a nomination for the whole network. So go to Podcast Digital Audio, look in Technology, and then you can vote for Triangulation. Forget that other podcast from Vice Media or Marketplace. Vote for me. Vote for me. We'd love, I would love to be able to go up onstage and accept a Webby Award during the first time they've given podcast awards. So it's pretty exciting. So just a little plug, TWiT.to/webby2017.

Steve Gibson, I'll give you a plug: GRC.com. That's the place to go to find Steve and SpinRite, the world's best hard drive maintenance and recovery utility. If you've got hard drives, you really need SpinRite. You can also find freebies there, lots of them. Steve is very generous with his time and his research, and he puts it all on the website, GRC.com. This show, too, audio plus transcripts, really nice transcripts at GRC.com. You can ask questions there: GRC.com/feedback. You can tweet Steve, @SGgrc.

You can also get this podcast on our site, TWiT.tv/sn. We do it live every Tuesday, 1:30 Pacific. We started a little late today. I apologize. It's partly my fault. That's 1:30 Pacific. That'd be 4:30 Eastern time, 20:30 UTC if you want to stop by and say hi during the live taping. But of course the main reason we make on-demand versions is so that you can listen at your convenience, wherever you are. And that's at TWiT.tv/sn and wherever you subscribe to shows. Great show, Steve. Next week, privacy and how to protect it in the new age.

Steve: For sure.

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

Steve: Thanks, buddy. Bye.