
# The Great DOM Fuzz-Off

**Description:** This week, Father Robert and I follow more Equifax breach fallout, look at encryption standards blowback from the Edward Snowden revelations, examine more worrisome news of the CCleaner breach, see that ISPs may be deliberately infecting their own customers, warn that turning off iOS radios doesn't, look at the first news of the FTC's suit against D-Link's poor security, examine a forthcoming Broadcom GPS chip features, warn of the hidden dangers of high-density barcodes, discuss Adobe's disclosure of their own private key, close the loop with our listeners, and examine the results of DOM fuzzing at Google's Project Zero.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-630.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-630-lq.mp3

---

SHOW TEASE: Equifax is the awful gift that keeps on giving; the world has stopped trusting the NSA; what we thought to be a relatively ham-fisted CCleaner malware attack turns out to be an incredible piece of exploit engineering; and your favorite browser just got fuzzed. Security Now! is next.

FATHER ROBERT BALLECER: This is Security Now! with Steve Gibson, Episode 630, recorded September 25th, 2017 for Tuesday, September 26th, 2017: The Great DOM Fuzz-Off.

It's time for Security Now!. We're like a kinder, gentler, benevolent advanced persistent threat, but for your brain. And of course, in the world of security, there is none more benevolent or more persistent than Steve Gibson. Steve, of course, is the big brain behind Gibson Research, ShieldsUP!, SpinRite, and our coming non-passworded overlords through SQRL. Steve, so good to see you, my friend.

**Steve Gibson:** It will not be an overlord relationship. It will be entirely beneficial. And optional.

PADRE: The funny thing about this is when the whole Equifax disaster first started coming out, SQRL was mentioned quite often as, yeah, you know what, we do need a better way to identify ourselves than some static numbers from a bygone era. And that sort of approach of being able to use dynamic bits of information was something that was pushed forward by a lot of cryptography experts. So I'm thinking, hey, you know what, you all should probably talk to Steve. He's been working on that for a while.

**Steve:** Well, I love the slogan that the project has about that. I just think it's got such a nice hook to it, which is "SQRL gives websites no secrets to keep." And that's key because that's what's always happening with these information disclosures is that essentially our password is a secret. And we make it up, and we give it to a website, and

we say, please keep this a secret. This is our secret. Keep it. Which is why it's so upsetting for people when, like you say, oh, I forgot my password, and they email it to you. It's like, no, that's not how you're supposed to do this. You're supposed to send me a link that allows me to set a new password, not send me my current one in the clear.

So the way SQRL works, the only thing the website has is your public key, and it's only valid for that one domain, for that one site. So they could publish it if they wanted to. It wouldn't do anybody any good. It's not of any use at all because all it does is it serves to identify the user because it's a per-user and per-site public key. And it's used then to verify a signature that they return of a nonce that the server sends. So it makes up a random bit of noise, sends it off, and says "sign this in order to prove you have the matching private key." That's all there is to it. I mean, yeah, a lot of details, which is why it's taken us a few years to get it all nailed down, but it's just there now. So, but, I mean, it's just like the right way to do it.

PADRE: And actually it all fits in with the concepts of security that are going to come up later in the episode, especially when you start talking about the evolving Equifax disaster, of the death of perimeter security. We have to assume that the bad guys are inside the walls now. That's just the era in which we live. And so as you mentioned, the best way to deal with that is to make sure that, when the bad guys get the information, they can't do anything with it. That's where we're at.

**Steve:** And, I mean, that's useful, but it requires an evolution of technology, which is what SQRL represents. The other thing I would say is that - and this came up, I'm just back from several days of a private security conference, which was under NDA so I don't have any need to or desire to talk about what went on. But one of the points that I found myself making was to remind everyone of the importance of monitoring. That is, for example, we know how long the bad guys were rummaging around inside of Equifax, for, like, weeks, many weeks, without them - and I did see one little bit of news, I didn't have a chance to follow it down, but it suggested that that had actually been going on for, like, four months earlier than was reported.

The point is that, yes, you have defenses. You've got your walls and your moats and your multilayers and everything. But you then must also watch what's going on. You must monitor because - and then the idea would be to be able to account for all the traffic that you see. That is, you know, take a look at it and say, okay, what's that? Okay, that's that. Oh, and what's that? Okay, that's that. But if you see something anomalous where you can't ascribe its valid role in your organization, that's, I mean - okay. So obviously, if you brought that to the attention of the security people, they'd have a meltdown. Well, you can't know it's going on unless you're looking. So you have to be looking in order to have the opportunity to see something that is anomalous and then bring it to the right people's attention.

Anyway, so I think - and the point is that, yes, we need to evolve our technology over time. But today anybody who is willing to put the resources in can at least be monitoring what's going on, be watching what's happening, and with the goal of making sure they can explain what they see.

PADRE: That's just a little bit of a teaser. As you can tell, we're a little bit passionate about this because, Steve, I think you can agree with me, it just keeps getting worse. Every week you figure this is it, okay, this is as bad as it's going to get. And then a couple of days later it's just one of these, what?

**Steve:** And we'll be talking today also about what appears to be an increasing focus on state sponsorship. And so we've seen over the course of the last decade the evolution of

hacks from teenagers, who unfortunately are being arrested and now tried for what they did in their earlier years, specifically Marcus. But now we're really beginning to see that this notion of cyberwar and cyberwarfare, which just - I remember it sort of struck me as science fiction. It's like, what? But no, it's real, and it's not good. [Crosstalk].

PADRE: You've got to bottle that anger. You're got to push it way down like a Klingon, let it explode forth.

**Steve:** So the title for today's podcast is "The Great DOM Fuzz-Off" because how could I not name the podcast "The Great DOM Fuzz-Off"?

PADRE: You just like "fuzz-off."

**Steve:** Yes, which is the title of a Google Project Zero blog posting where a new member of the team built - essentially I guess it would be for him a third- or a fourth -generation DOM fuzzer that we'll be talking about, like what's a DOM fuzzer, and produce some surprising results. So that's at the end of the show. But you and I are going to talk about more of the Equifax breach fallout, a look at some interesting - and almost this would have been predictable, but it actually happened - encryption standards blowback from the Edward Snowden revelations. We're going to examine some worrisome news, actually additional worrisome news about the CCleaner breach that we first started talking about last week. More has come to light since, and of course it's not good news. There's also been some reporting by ESET of their discovery of some ISPs deliberately infecting their own targeted customers with malware, which is a new bad bit of news.

PADRE: But it's friendly malware, Steve; right? I mean, it's a good version.

**Steve:** The recipients don't feel that way. Also we need to talk about a warning that has come to light post-iOS 11, that turning off your radios doesn't. Some look at the FCC's lawsuit against D-Link over their poor security. We've talked about that previously. A judge has thrown out half of the case. We're going to talk about that. Broadcom is introducing a new GPS chip with some cool new features that'll appear next year.

Also I wanted to talk just briefly about the dangers of high-density barcodes which has come to light. And I sort of blew this one off. This appeared during the security conference I was at, and I thought, well, that's interesting. But then as I caught up on my Twitter feed, all of our listeners were sending it to me. And this is of course Adobe's disclosure of their own private key, like posting it on their website. Whoops. Then we're going to close the loop with some of our listeners and then get into this "what is DOM fuzzing" over at Project Zero. So I think another great podcast for us.

PADRE: Steve, I have no idea if we're actually going to be able to get through that because that's a lot. That's a lot of material. But we're going to give it a good try.

**Steve:** Especially when I have you on the other end of the phone, Padre. We always engage significantly.

PADRE: I've noticed that. We tend to dwell a bit more when I'm here. I'm just enjoying it. I mean, I really get to geek out on security, and that's why I enjoy doing the show with you.

**Steve:** At least everyone can hear us this week, so that's an extra bonus. Suddenly you appear, and the audio clears up. I wonder what that's about?

PADRE: I've got some pull. With Comcast. Oh, Steve. One week there will be no more bad news out of Equifax. This is not that week.

**Steve:** Well, we have to talk about our Picture of the Week.

PADRE: Oh, that's right, that's right. I always jump over that.

**Steve:** Which was just fun. I had to remove the F-bomb from the quote from the NSA guy, but it's just a great cartoon. It's just three frames. In the first frame we have Apple's CEO holding the phone, saying "In the new iPhone X...." Then in the second frame, "We want to introduce face recognition!" And then in the lower frame there's a guy in a suit and tie with a CRT or display screen labeled NSA on the back, and he's quoted saying, "We love these guys!"

PADRE: Now, over the weekend I actually got some hands-on time with the iPhone X. I got to play with it for about three hours.

**Steve:** I'm afraid if I saw it, I would have to have it. But right now I'm still sticking with my 6.

PADRE: You won't, honestly.

**Steve:** No? Oh good.

PADRE: Here's the thing. It's beautiful. It's really nice, and they've done an incredibly good job on engineering it. I even like the notch. The notch is not a problem. People are mocking the little notch at the top of the screen.

**Steve:** The little rabbit ears? I hate them, yeah.

PADRE: Yeah. But, I mean, it's not like that takes away from the functionality of the phone. Normally that would just be a whole bar of the screen that you wouldn't have.

**Steve:** Right, right.

PADRE: What bothered me, however, was the lack of a home button. So they had to change everything. So now everything is run by swipe. So something you used to be able to do maybe with a long press or a double-tap of the home button, now you do it with, like, three or four swipes. That felt very weird. And I don't use iOS a lot, so it was really noticeable for me. And I'm thinking, if someone is very much into iOS, they're going to not like that at all. It will take so much getting used to. And at some point it means you split iOS. You're going to have two different interfaces. I don't think Apple wants to do that.

**Steve:** Well, and in fact they've also done that to a much greater degree because I've got it on all my devices as of last Tuesday morning. And so already, because of the difference in screen size, there's a bigger change between the phone and the tablet functionality, things that are gone from the tablet that you used to have, that the phone still has because they couldn't - those UI features, or the UX stuff, couldn't translate. But I'll just say now, I actually have it later down in the show notes, this is the buggiest release of iOS I've ever experienced.

PADRE: It is, very much so.

**Steve:** Every single one of my devices does bizarre, some of them repeatable, like bright flashes of the screen when I'm trying to scroll something. And at one point I had some of the balloons in a message stuck on the screen with other ones scrolling underneath it from a different stream. It's just like, okay. And I've had the little bar of icons, the tray, stuck on the vertical right side of the tablet. And everything else rotated around, but it just stuck there. And it's like, I mean, it's constantly doing broken things. So I do think that we'll see those fixed. I'm sure they're collecting them now, and they'll fix them, and 11 will stabilize at some point.

PADRE: Oh, yeah, they'll figure it out.

**Steve:** But it's a mess. It's never been such a mess. I've never seen a version of iOS that is this buggy out of the gate.

PADRE: But Steve, I mean, you've got to say it with me: It's not a bug, it's a feature.

**Steve:** Eh, it's a bug. So, yes. Just referring briefly to Equifax, I picked up an interesting conversation over breakfast the second day of this private security conference that was something that hadn't occurred to me in all the discussions we've had about Equifax. There is, among the upper-end security community, a clear, I don't want to say "knowledge" because it doesn't quite clear the bar of being knowledge. But it is believed that this was China that performed the Equifax attack, that this was not some opportunistic random hacker somewhere, but that this was state sponsored, and that the 143 million of us Americans whose personal information was disclosed are probably not in the kind of immediate danger we would be if the goal were to sell this information on the gray market.

Now, maybe that will eventually happen. But the theory of this is that, if this was China state-sponsored attack, the goal was to obtain the very private, personal identify information of specific individuals, government employees, corporate higher-ups, you know, C-level people, to enhance and increase the power and success of targeted attacks against them.

PADRE: That's interesting.

**Steve:** So rather than all this information just being made widely publicly available, there may be more to it than that. And it may not have the kind of sweeping effect it could have because in some senses, if it's a state sponsor, then maintaining the privacy of that is of more value to them. And also then everyone running around locking our reports, while that's what we have to do, it suggests that there won't be many attacks against those locks as a consequence of this, but rather this was a deliberate personal information database acquisition by a state actor.

PADRE: Now, there's a couple of interesting things about that. One, typically when you hear "state actor" in combination with any sort of attack, it means it was some sort of advance attack. I think that's one of the things about this particular breach that did bring a lot of state actor talk at the beginning because this wasn't an advanced breach. This was not an APT. This was just a company that didn't know how to run security. But that doesn't mean it's not state sponsored.

**Steve:** Right. Exactly.

PADRE: What lends credence to this, however, is I've spent the last two weeks crawling all the usual spots, looking for an increase of credit card numbers and identities on sale,

and I haven't really seen it. It's been sort of the regular flow of traffic. And you would figure, if this was for gain by selling IDs...

**Steve:** And because it's time sensitive.

PADRE: It's very time sensitive.

**Steve:** As people are locking things down.

PADRE: Precisely. And I haven't seen a great increase. So either they didn't get a lot of information or, as you mentioned, this was particularly a targeted attack. They were going after some very specific people, and the rest of us were collateral damage. And again, because Equifax had no controls...

**Steve:** Were just swept up in the acquisition, yeah.

PADRE: But as you mentioned at the beginning of the show, if you're not listening, and you don't have any controls in place to see what's actually leaving your network, then that makes breach recovery so much more difficult because we don't know. I mean, it could literally have been one person or no people that they took the data of, or it could have been all 143 million. That's just - that's unacceptable.

**Steve:** Right.

PADRE: My goodness. And it just, I mean, it just keeps getting worse and worse. We've already heard some of the details about how this happened, how it was a struts vulnerability, how they have at least one server that had default admin and password credentials. We know now that they actually did get hacked back in March, and they didn't do anything about it, so that could have been seen as a dry run for this. We know that their security executives had no business being security executives, one of which was receiving like a $3.2 million salary per year to do nothing, literally nothing. And then of course this last week there was the embarrassing and yet somehow very appropriate advisory that they had been forwarding people to a site that wasn't theirs.

**Steve:** Yup.

PADRE: So, yes, it's all around good news, Steve, I think, yeah.

**Steve:** Well, and in the wake of all of this attention, Brian Krebs reported also another little bit of news, and that is that Experian was not well protecting the PINs of their previously locked customers. I remember very well 20 months ago, after the Anthem breach, when we first discussed the need to lock our credit reports. I and my friends and family - because I turned them all onto this at the time, and I'm sure a subset of our listeners here all did this - we went to the various bureaus and obtained locks and PINs. And in every case the fear of god was put into us about not losing this PIN. It's like, this is it, and you'd better not lose this because there's nothing we can do. And I take that very seriously.

So it's like, okay. So I printed the web pages and took photos of them and zipped them up and put them away and really preserved and protected that information. Now we come to find out that, well, that was not very practical. It's like, oh, well, people are going to lose their PINs. What are we going to do about that? So Experian has a web browser-facing PIN recovery. And unfortunately, you only have to give it things that are always available, if you've got this problem in the first place.

PADRE: Name, social security, address, driver's license, and a little bit of your credit history. Those five bits of information will let you get a new PIN. And it's like, oh, but that's all on the web now; right?

**Steve:** Yeah. So it says it's Experian - it's called the "Request Your PIN," which, first of all, you shouldn't be able to do.

PADRE: No.

**Steve:** The whole point is you don't want anyone - because they don't know it's you. The point is how do you authenticate yourself when you get your PIN to unlock the thing you're trying to lock? So it says: "If you have a security freeze on your credit report and wish to obtain your forgotten or misplaced PIN," you know, they just ought to say, well, you're SOL. But no. They say: "You may request it here. A PIN is needed to remove a freeze from your credit report and to provide a creditor access to your credit report." Right. That's why no one should be able to get it. But anyway, they make it quite easy. You fill out the form, providing information that's readily available - your Social Security number, your date of birth, your name, and your address. Bang. Here's your PIN.

PADRE: You know, Steve, this is the tension between the various audiences that we have. There's the hardcore Security Now! audience, and they understand. They get it, that if you use a new service that says we protected everything and you control all your data, and they have a way to recover your password, then that means they're lying to you. Our audience gets that. However, there's another audience, and that's who this is for. This is security theater. It's to make them feel like they've protected themselves, but they don't get that if there's a way to recover something that is supposed to be unrecoverable, that somehow the service is broken. And how do you get to that second group? You can't really because [crosstalk].

**Steve:** I know. And this is my one worry about SQRL because, as its designer, I have refused every temptation to cheat the system in that way, and I've made it extremely safe. I mean, I've done everything I can to help people not hurt themselves. Like for example when the system, when you create your identity, it gives you a 24-digit rescue code. And that sounds like a lot, but it's only one and a half credit cards long because a credit card number is 16; this is 24. So, I mean, it's not bad. But it makes you write it down. And I'm sure people are going to go, eh, no, and just hit "next." Well, then it says "type it in." And they're like, oh, damn. And they have to go back and actually get it. And it won't let you put it on a clipboard because it's not supposed to be in your computer. So it's like, it won't let me copy this. It won't let me cut and paste it. No. Write it down.

And so, again, it's a two-party system because no one wants to trust a third party. Okay. But that means there's no third party to go crying to when you do forget it. So maybe it's going to end up being that SQRL is just not for everyone. If you cannot handle - all you have to do is write that down. But if you refuse to do that, I'm sorry, there's nothing we can do for you. You just step away from the computer, maybe unplug it, and go back to basket weaving because the Internet is not where you should be.

PADRE: It's funny, it brought up a situation I had over the weekend. I gave my dad a new phone, and he's an Android user, so we were switching him over to the new phone, and he couldn't remember his Google password. And I'm thinking, that's the one resource you use regularly on the Internet. How can you not have that? And then I realized, no, that's actually pretty normal.

**Steve:** It's so sticky now. Yup.

PADRE: Yup. And I'm thinking, yeah, so that actually is a problem because people get so accustomed to their device automatically being authenticated that, when they're asked to remember it, it's not even possible. And I would love to have that solid security because, like you, I believe that helping others with security helps me with security. When people around me are more secure, I'm more secure. That's just how circles of trust work. But at the same time, I can't ask my father to use a service that absolutely positively will never allow him to recover a password that he's lost. So I guess I'm just going to have to live with getting owned every once in a while? Is that it?

**Steve:** Well, the point you just made reminded me of the iOS 11 upgrade process because you are forced to give your cloud identity in order for it to synchronize in. But that's before the rest of the system is up and running, where you have access to your password manager. So you would have to have it somewhere else and then have a password that you have a chance of entering through the frustrating touch keyboard with upper and lowercase and symbols and everything. It's like, god help you if - and so it happens that I do have that. But, boy, if you're someone who has become totally dependent upon your password manager, there are still places where even that can't come to your rescue. It's all messed up.

PADRE: That's where we are. But, see, that's why you're here, Steve. You're going to make it better. I think, I truly believe that when people understand security a bit better, and when it gets brought down to a level that they can intake, then they're more likely to make a good security decision. People will still forget, and people will still be lazy, but at least they understand the stakes.

**Steve:** One of the very first acronyms that we developed on this podcast was TNO, Trust No One. And there didn't seem to - there hasn't been any blowback against that. And I think it's because only the people who understood the responsibility of - the other one was PIE, Pre-Internet Encryption, PIE. You encrypt before you put it on the Internet. Only the people who were willing to go to that extra level also understood that there was no recourse, that their encryption system was putting noise up in the cloud, and they couldn't ask the noise holder to please turn that noise back into data. Just no. There's just no way to do that.

And so the problem that I face with SQRL is it is both seductively easy to use, I mean, when you actually have the experience you immediately think, wait, this is secure? I mean, it just - it's so easy that everyone is going to want to use it. Unfortunately, to your example, your dad shouldn't because, if his SQRL identity gets away from him, all of the system for recovering it is in danger. Although I will say that, because SQRL is a proxy for you, that is, you give it permission to authenticate on your behalf, I do require you to constantly reenter your one SQRL password when you begin using it each session. You don't have to do it during your multiple hours of sitting there logging into different places. There you only have to give it just the first few characters to make it easy. And that, too, is deliberate. It reinforces your memory of that one password. That's all you have to give it.

And even so, we've got some weasels - now I'm sorry I used that term - in the newsgroup whose signature line on all their postings is "The SQRL password must be removed" because they're annoyed by that. And it's like, I'm sorry, no. There's a reason for everything I have done. It's been well thought through. And so this doesn't let your dad forget the password because he does have to use it every day, just once, and then it won't be in his way. But he has to keep using it. And so anyway, the system has been designed not to be brittle and to have tons of recovery, but ultimately it's only you. And so we'll see how that works out. Apparently it didn't work out very well for the Identity

Lock PINs.

PADRE: Absolutely not. Oh, by the way, when he says "weasels," he actually means nuclear vessels. There are those who are floating by in the night, and you're doing your own thing, so it's all good. Now, Steve, can we talk a little bit about encryption because you brought up that one of the very early acronyms was Trust No One, TNO, and that makes sense. Except the U.S. government has been saying to our allies, "Trust no one except us," and that's not turning out so well anymore.

**Steve:** Well, and our listeners, our longtime listeners will remember well the controversy that arose around the dual elliptic curve random bit generator which there were four different pseudorandom number generators that were NIST approved. And what was weird is that the slowest of those was the one that RSA had as their default in their BSAFE package, which was their crypto library that they've been selling for years. I've got the documentation behind me on the bookshelf. I owned a set of that in the early days.

And so there was never any proof; but the problem with, for example, algorithms that have magic numbers is that the magic numbers have to be set in stone. Everyone has to agree on them. But they have to have certain properties. And so if someone just hands you a magic number that looks like gibberish, but it's like special gibberish, the question is where did the gibberish come from? That is, did you roll dice all weekend and come up with this? Or did very smart people in the backroom say, ooh, look, we can hide some special bits among the gibberish which will give us an edge that nobody would suspect.

And so in this dual EC, this dual elliptic curve technology were some magic numbers, some gibberish, but there was no explanation of where it came from. For example, say that you ran a counter through a hash, and you took the output gibberish and tested each one until you've got one that met your criteria. That's a perfect example of being able to demonstrate you didn't design this gibberish. It emerged from a hash equal to all the other gibberish, and you just chose one that met the criteria you had. There you could say, ah, we know where you got the gibberish, and we like it because it's as good as anybody else's. Yet instead the NSA just presented their gibberish and said, okay, everybody, take it. And it's like, uh, we're not really sure that's clean gibberish.

So now we have, many years downstream, what has come to light through some reporting that Reuters did, is that after three years of ANSI committee work on some next-generation ciphers, and this is the ISO standards body, there were two encryption algorithms named Speck and Simon which were sourced from the NSA and analyzed and believed to be good. But the NSA was also saying, oh, and you know, in power economy environments or time-critical environments, you might want to use some weaker versions. And we've talked about how ciphers work where they typically are iterative, and so you have to run some number of iterations in order to deeply mix the bits enough that they cannot be unmixed.

And in fact attacks on ciphers are done by deliberately reducing the rounds, the number of rounds, as it's called. And, for example, there was even in Rijndael, our current AES standard, there have been attacks against, like, seven rounds because that hasn't given the bits time enough to get thoroughly mixed, that it is possible to still statistically see some leakage. But you're never supposed to do that. You're supposed to always run 11 rounds or 13 rounds or whatever, depending.

So anyway, what happened is, as a consequence of what everyone now remembers of the Snowden disclosures, where some of the documents that were leaked by Edward Snowden after he left the NSA demonstrated that there were budgetary records which

were seeking funding to insert vulnerabilities into commercial encryption systems. So, I mean, that news is out there. And of course it's made the global community skeptical of accepting designs of what would become critical encryption infrastructure which is offered on a plate from the national security agency without...

PADRE: As it should.

**Steve:** Yes, exactly, as it should. Now, you could argue that in the past we were too trusting and just saying, okay, fine. Now that trust is much more conditional and has been pulled back. And so what happened was the U.S. ANSI standards organization, which was proposing the standardization as an ANSI standard of varying strengths of these two new encryption algorithms, finally agreed, after essentially the global community said no, agreed to remove all of the weaker variants so that only the maximum-strength versions of Speck and Simon, which are believed by academic researchers to in fact be super strong, those are the only ones that have been allowed to make it into the standard because the weaker ones looked okay, but unfortunately trust has been compromised in where they came from.

And again, unless you can clearly demonstrate every bit of the sourcing of something like this so that you can just say, yes, it dropped from the sky and we realized, oh, look how good this is. Or it just truly came out of staring at lava lamps and writing down how many bubbles were coming up over time and using that as your source of entropy. No. If we don't know where it came from, it really needs to be treated with skepticism.

PADRE: When we were developing our own type of encryption for different types of communication, we would actually use atmospheric static. But we had specific rigs that would look past the ionosphere, the mesosphere. So we would actually get intergalactic static. And it was one of these things, yeah, that's crazy random. There's no one that can manipulate that.

**Steve:** That would be manna from heaven; right?

PADRE: Precisely. But, see, Steve, the thing about the story that gets me is it's a tragedy that you've introduced so much distrust into a standards body that is responsible for some serious heavy lifting for security because, now that everyone doesn't trust the NSA - and again, rightfully so because the NSA tried to use the trust that other countries had in them to push forward something that they knew was broken. Now you start to ask, well, if the NSA is doing it, who else on the standards body is doing it? Everyone has an agenda. Everyone's trying to get a leg up. Everyone wants to have that backdoor.

**Steve:** State actors certainly, yes.

PADRE: Yeah, exactly. It would be naive to think that the United States has the only institutions interested in doing this. But once you have that level of distrust, then you don't really have a standards body anymore. You have a very adversarial process.

**Steve:** I think what will probably evolve from this is that we change the way these things are presented, very much using the example I did before where we had accepted an elliptic curve algorithm with magic numbers whose origin was unknown. That just has to be ended. We will never do that again. A bunch of people, a bunch of academic cryptographers from different countries will all get together and watch a random number generator generate random numbers, and they will collectively as a team choose the one that they decide is the best, not because it was - I mean, so that by a priori knowledge it has no special characteristics. That's the way we're going to have to do this moving

forward, very much like the way keying is done of the top-level domains or DNSSEC and things, where, I mean, those secrets are so crucial to be kept pure that it's done in a way that cannot be compromised by any single entity. It cannot be biased.

PADRE: Right, right. And of course that is exactly the type of solution that members of our own governing agencies are very much against. They don't like the math. They like to be able to control those magic numbers.

**Steve:** They don't want to lose control, exactly.

PADRE: Yeah. Because we know from the past that trusting a government entity, especially the United States, with a magic number, or in this particular case of the TSA a magic key, that there's no way that they're going to let it leak out and potentially become public knowledge.

All right, Steve. I know we're not going to have any good news because the next topic is...

**Steve:** This is not the happy good news podcast.

PADRE: At one point we have to have a Security Now! podcast where it's just everything is happy. It's like, oh, my gosh, Microsoft has everything together. They'll never have to be patched again. Apple has a great new iOS 12.

**Steve:** Bug-free, yes.

PADRE: Yeah, bug-free. But, no, bug-free is far in our future because we need to talk about a malware that at first seemed like it would have minimal impact, but over the last two weeks has been building into something quite dastardly, actually. What's going on with CCleaner?

**Steve:** I don't remember now if I shared on the podcast last week my feeling that Piriform, the publishers of CCleaner, were downplaying the severity. I felt when I was reading their own disclosure. I was thinking, you guys are really trying to make this really seem like a non-event. Whereas the more independent coverage, I mean, this was originally found by Cisco's Talos security group. And their coverage was, I think, much less biased. Again, any security researcher is playing up their cleverness at finding something and so that tends to be maybe a little overstated, how bad this is. On the flipside, the company that has been found to be at fault is like, oh, well, you know, yes, 2.23 million people downloaded it, but that was only there for a moment; and when 5.34 was replaced, that all went away. So it's like, okay, but what happened in the interim is what we want to know.

So to recap, the 32-bit installer of the v5.33 of CCleaner was maliciously modified to install and include a backdoor which reached out after it was installed to a remote C2, as the community calls it, a command-and-control server. That backdoor malware was able to receive commands and download additional malware payloads. That we knew last week. And we also know that significantly more than 2 million copies of that, the number I have in my mind, I didn't look it up again, but I think it was 2.23, if I recall, were downloaded. And we don't know that they were run, but people downloaded them. And most people, because there's not an auto update in the free version of CCleaner, which most people use, someone who went to get it almost certainly ran it.

So, okay. So that's everything we knew last week. Cisco has continued, with their Talos

security division, continued looking at this. And they obtained a set of files which were ostensibly from the malware command-and-control server. They were skeptical and careful about trusting the files, as they should have been. But they were able to determine that, to the best of their belief, they were absolutely authentic because among the files were logs which showed their own early traces of their activity with the server. So, for example, they probably set up a cleanroom environment and infected a system and watched it reach out to the server. So they were able to later forensically detect their own earlier initial discovery among the files. So they came to the conclusion that, yes, these are authentic. What we have is real.

Then what they discovered was a bit chilling. They found, in the code on the command-and-control server, 20 domain names of targeted companies, among them Cisco.com and a bunch of others. If anyone's interested, the link is in the show notes, and there's a list of the 20 domains in the Talos blog coverage. And they discovered that, if an infected machine contacted the command-and-control server, the source IP of that incoming traffic was looked up and its domain checked against the list. And if there was a match, the command-and-control server woke up and delivered a second-stage payload.

PADRE: Wow.

**Steve:** Yes. And they also determined that that had been done - and I'm scrolling through my notes here, and I don't see the number in front of me. Let's see. Talos wrote: "The use of domain-based filtering..."

PADRE: Twenty machines, yeah.

**Steve:** Was it 20?

PADRE: Twenty machines.

**Steve:** Yes. Oh, yeah, there it is. More than 20 machines, they've determined, have received the second-stage payload. So now we get a better picture of what was really going on. Some attacker - and again, this is now believed to be state-sponsored because this is not, I mean, this is a big project that was put together. And the nature of the attackees, that is, the targeted domains, also suggests that somebody with state-level interests had specific agendas. So they picked a very popular freeware utility, and I don't remember what the number was. We talked about it last week. The total number of downloads is just astonishing. I mean, I use CCleaner. Everybody uses it.

PADRE: It was over two million.

**Steve:** Yes.

PADRE: In the limited amount of time. And remember it was only one variant of CCleaner, was just the 32-bit version.

**Steve:** Correct.

PADRE: So two million machines to get about 20, just over 20. Now, Steve, unlike the Equifax breach, this feels state sponsored.

**Steve:** But the point I was saying was that CCleaner itself, see, they didn't know they were going to get 2.2 million. CCleaner itself had been downloaded - and that's the number I don't remember - hundreds of millions of times, I mean, because it's so

popular. And so you're right, in this brief window they got 2.23 million downloads. But in terms of the overall plan, the idea was to find a highly popular piece of freeware whose supply chain, and that's the term that you're seeing how, we're talking about a supply chain attack, that is, an attack on getting something to someone else and interfering with the chain of delivery.

So take a highly downloaded piece of freeware. Infect the supply chain so that you are then using a scattershot. Essentially you're infecting everybody who downloads it with a first-stage attack. But you're only actually looking for 20 domains. You don't know you're going to get any, but they did. So you have a list of actual targets. And when from this scattershot, this 2.23 million downloads, just by pure luck and opportunity, some of those people who downloaded that version were among the 20 intended targets. And they got hit with the real McCoy with this second-stage shell code which, again, even though there was both 32- and 64-bit shell code, it still looks like it was a 32-bit attack. So the attackers apparently had reason to believe - or maybe it was vulnerabilities in those attacked machines that were only available in the 32-bit variant.

But still, now we have a better picture for what was going on. This was a big project, and there were specific companies targeted, using a purely random, hopeful, maybe this is so popular that some people working in these particular organizations will be CCleaner users and will, while the infected one is in the supply chain, will download it, will run it. We'll identify them, and then we're going to zap them with our actual malicious payload. Wow.

PADRE: Now, Steve, I'm trying to think back. Now, it wasn't quite a two-stage attack. But Stuxnet is one of the first of these types of attacks where you shotgun to as many machines and clients as you can, but you only activate when you get on the right system and, even then, only when the conditions are exactly right for you to do the most damage. That was an incredibly advanced state-sponsored attack back in the day. That's what this feels like. I mean, if this was just some kids looking to be script kiddies or just some Black Hats looking to compromise financial data, they would want to get as much as possible. The fact that these were looking for specific hosts within specific domains, it screams, first of all, industrial espionage; and, secondly, state sponsored.

**Steve:** Yes. And so you can kind of forgive Piriform's initial statement of saying, well, we looked at what happened, and that command-and-control server was taken down before it did anything wrong. Well, that was not true. But you could forgive them for wanting to fudge in their favor because it also, exactly to your point, it wasn't the case that all 2.23 million people got the big enchilada. They only got the little icebreaker that started this.

And also remember that the risk of discovery is all part of this. So that first stage tried to do as little as it could. It didn't want to get discovered. And that's why it just made a little query out to the command-and-control server, basically to say, hey, I'm here. Did we get lucky? And if not, no, okay, go away, just shut up. But in the case of, oh, yes, you're calling in from Cisco, bingo. And then that's when the big payload was delivered.

So this is the way you organize it, exactly as you were saying, Padre. You organize this so that you don't expose yourself. Absolutely, you keep that exposure minimal so that you can continue operating as long as possible before someone like Cisco Talos or ESET or Kaspersky or any of these companies that are now on the watch for this become wise to it and then basically shut down that particular campaign. There will be others, but this was one.

PADRE: Oh, yeah. This is going to be more common, especially as they start building malware to time exfiltrations because that's the next step. Once people actually start looking like, okay, we need to look at any particular malware, any C&C system, and

actually compare it across multiple domains all across the Internet, it's going to change the way that they look for that. But once they have that down, it will be, well, if I'm going to design an APT, I'm going to have it stay dormant in the system until I get exactly what I want, and then I'm going to exfiltrate all at once. And that is such a difficult attack to defend against because it can be literally over in a matter of minutes.

**Steve:** And we're going to do it at 2:00 a.m. on a...

PADRE: On a Saturday.

**Steve:** On a Sunday morning, yes, exactly, when no one is around, and everyone's asleep.

PADRE: Oh, my. Now, Steve, this type of attack - let's start with the infection. They were able to make it look like authentic software because they didn't actually change the install package. They just piggybacked on it. How easy is that to do? Can I piggyback on basically any install package?

**Steve:** Well, it requires access to the system. So the idea is that you had to have a broken server. But normally what's happening is the installer is not signed, but its payload is signed, because the installer is not produced by the software publisher, it's produced by the installer company. And so it was the case that CCleaner had a valid signature, but the installer was modified to install something in addition to CCleaner. So checking the CCleaner signature, everything was fine. But there was also something that was brought along with it, that was added to the installation package.

And so you absolutely want to protect the security of the supply chain, in this case the download server that is supplying all this. And of course you do want to make sure that you're connecting to the proper download server, which brings us into our next story because ESET - and this is very troubling. ESET, which is a well-known security company, has found instances of a WikiLeaks previously disclosed malware known as FinFisher, F-I-N-F-I-S-H-E-R, which is advanced surveillance malware, capable of snooping on webcams, monitoring and recording keystrokes, microphone audio, and the victim's web browsing behavior.

So it's advanced surveillance malware. And it's like, okay, so that's not good. It was found in seven countries which they have blacked out in their reporting. And they specifically say we're not going to talk about where we found it because I guess they're working with those organizations or companies in the background. But they did say it was found in seven different countries. The disturbing thing is that they also found evidence that it was being delivered to the endpoints, the endpoint users, by their ISPs.

So the ISPs were redirecting their customers' attempts to download WhatsApp, Skype, Avast, VLC Player, and WinRAR. Those apps were found to have been compromised, not the actual supply chain download server itself, that is, not the real one. But what the ISP was doing was performing a targeted man-in-the-middle attack so that, when specific customers downloaded any of those very popular apps, the ISP redirected the download to a malicious source which provided FinFisher-infected versions of those popular applications in order to get them installed onto their systems. So this is the first time we had any knowledge and public reporting of this happening, that is - and we don't know which ISPs and which seven countries. I sure hope it's not ours.

PADRE: It better not be ours.

**Steve:** Yikes.

PADRE: I mean, that's actionable. That's illegal. At the very minimum, that's a violation of the Computer Fraud and Abuse Act. And it actually could be worse because now you could be held by the 1986 Telecommunications Act because you're essentially launching a man-in-the-middle attack against a customer who paid you for service which included security.

**Steve:** Yeah.

PADRE: Wow. And the thing is, that's not hard. I have at least four or five different packages that you buy commercially that do just that. Now, it's supposed to be so you can do network management on your own enterprise segment. You can make sure that if anyone - one of the reasons is you want to reduce the amount of load on your externals. Maybe there's a file that a lot of people inside your organization are downloading, and this is an easier way for you to cache it. It also could be that you are trying to keep people from going to particular sites or getting particular software, and there's plenty of software that stops you from doing that. So this is just the application of totally legitimate software. What makes it illegitimate is the fact that they did it without the users' consent. That's - wow. Steve.

**Steve:** I know. I know. And we've been talking about ISP trust and people talking about using VPNs in order to get through their ISP. And our fallback has been, well, yes. But as long as you're over HTTPS, you're okay. But of course...

PADRE: It jumps off, yeah.

**Steve:** We know that's not absolutely true. If this is a state sponsor, we know that it has to be the case, with so many CA root certificates in our system's trust stores, you know, four or 500 of them - maybe it was eight. I can't remember the number now. But, I mean, it was like a ridiculous number, where our browser trusts anything signed by any of them. There is no way we can believe that every state government who wanted to cannot have a certificate made for any site they care about.

And so if they provide a server which has spoofed TLS certificates for WhatsApp and VLC and the rest, and Skype and so forth, then all the ISP has to do is rewrite the IP packet headers that are, even though it's encrypted, even though it's HTTPS and TLS, all they have to do is redirect the traffic to the malicious server which is set up to answer a TLS connection correctly with a fraudulent but valid HTTPS certificate, and a user can download WhatsApp, Skype, VLC, whatever, with everything looking fine over an HTTPS connection, and still be getting a fraudulent, malicious payload.

PADRE: I got a question about this in my show email from Know How, and they wanted to know if using something like a Tor server would prevent this. And I'm like, well, Tor jumps off. At some point it has to jump off the encrypted node, and it's now, if it jumps off on a client that is connected to an ISP that's doing this, then it's a false sense of security because you're only as secure as your most insecure link. Now, if I have a direct encrypted connection to the server from which I am pulling the file, then I can say with certainty that, yes, I control what's being put onto my computer, and I control what's being pulled from that server.

But unless I actually own the server, and I know all the segments that I'm jumping through to get to that server, then I have to assume that it's insecure, even if it does say HTTPS. Which is kind of scary; right? Because we've been telling, I won't say the "less

schooled," but we've been telling the less tech savvy that as long as they see the little lock in the browser bar, that they're safe. And in reality that's no, not so much.

**Steve:** Well, and we need to remember that we get two things from those links. We get encryption, and we get authentication. So encryption is easy. We always get encryption. And, for example, Let's Encrypt, that gives you encryption, but Let's Encrypt is so heavily abused these days by spoofers that it's like, well, it's not really that well authenticated, and they don't try. They're only trying to be the weakest form of so-called DV, Domain Validation certificates. And that was the goal. That was the original goal was just we just want to encrypt everything.

Unfortunately, with that comes the presumption of identity, that is, you're also authenticating the server endpoint. And that turns out not to be true, I mean, largely in the case of Let's Encrypt; but also that's a weak assumption when you've got somebody with sufficient power like a state actor that has the ability to just synthesize whatever certificate they want because they have control of a trusted certificate authority just by virtue of the fact that they are a government.

PADRE: Right. And also they have the power to actually reach into the infrastructure and make sure that you're routed in a way that they want.

**Steve:** Yeah.

PADRE: Which at that point it's game over. They're mentioning in the chatroom that, too, it's time to make sure you always check your hashes. And, yes, that's very, very important. Make sure it's SHA, MDA. Go ahead and check the values after you've downloaded a file. But even that, would that have stopped the CCleaner attack? Because the actual install file was still fine; right? It was the piggyback file that was infecting.

**Steve:** Certainly it depends upon what hash you're checking. And also notice that I've always been kind of skeptical about this whole "here's the hash of the file" because you're getting the hash from the same page as you got the infected file.

PADRE: Exactly. Still have to trust somebody, yeah.

**Steve:** So any hacker worth their salt is going to fix the hash to match the infection. So it's like, oh, look, the hash checks. Well, now you're even worse off because now you think you're extra secure, and you've verified the hash, and so you're going to give that to all your friends. So, no.

PADRE: Yeah, yeah. It's nice. In theory it is a great way to protect yourself. But again, as you're mentioning, as we are getting more and more into the age where the really sophisticated hacking is being done by state actors, A, it's not easy to detect; and, B, once you detect it, there's not a whole lot you can do unless you own all the infrastructure between the resource you're trying to access and your client.

**Steve:** Yeah. Well, let's go to something...

PADRE: More we can control, Steve, because I think we're putting our audience into a tailspin. How about this? You've got a brand new iOS 11 device. It's got all these great features. I love the little dashboard. I love the fact that Megan Morrone showed me how I could set up my iOS device so that it doesn't let me use it while I'm driving, which I love. But it also does this other thing which, again, is not a bug. It is officially a feature. And that is, when you turn off WiFi and Bluetooth in iOS 11, it doesn't actually turn off WiFi

and Bluetooth. What's this about?

**Steve:** That's right. That's right. So, okay. From early days of the podcast we've talked about how annoying it is that iOS continually turns Bluetooth back on because, as we know, Bluetooth is a short-range inter-device communications link, which is very useful. If you have a hands-free phone in your car that is Bluetooth-enabled, if you've got a Bluetooth headset or earphones, I mean, there are certainly use cases for it. But many people don't have that. And so there's nothing for their phone to Bluetooth to. And standard security practice, and we were just talking about - what was the blue attack last week?

PADRE: BlueBorne.

**Steve:** BlueBorne, exactly. We were just talking about how, yes, if you don't actively need something, especially if it's a radio, turn off the radio. And so our advice had always been turn off Bluetooth if you don't know you need to have it on. And so it was annoying that iOS constantly turns it back on again, even though it requires your deliberate act to turn it off. They think they know better. Well, with iOS 11, this presumption has grown in strength. One of the cool features, and I was using it during my recent travels, is that you're able to swipe up from the bottom. If you swipe up a little bit, you get sort of an application switcher. You get your little launch dock.

If you keep going, that takes you to - they call it the Control Center, which you used to be able to get by swiping up from the bottom, but this is different now. It's also got an application switcher. You're able to customize its contents using the Control Panel to change what's there. And I did add a few things and took some things away to make it the way I wanted it. And there you're able to do things like turn the brightness up and down, turn the flashlight on, go quickly to the camera, rotation lock, other stuff like that.

And there's a little set of icons that allows you to put your phone, in the case of a phone, or your tablet into airplane mode and/or turn off WiFi, Bluetooth, and the cellular connection. And so it's very handy. They made it, like there it is. You can get to it quickly. Swipe up and tap tap. What comes to light a week later is that, well, no. If you turn them off there in the right-in-front-of you convenient and easy location, it doesn't turn them off at all. It disconnects their connections that you have to things, giving you the impression that it's turned them off, like fooling you. Just like, okay, it's off. I mean, and the icon goes dark, everything looks off.

But they're not off. They're not off at all. They're not stopping communicating. They're not reducing power, which is the other reason you turn off unused radios is you want power conservation. No. It just disconnects things so like your speaker goes doo-doo-doo and says, oh, yeah, look, Bluetooth's disconnected. Yes, but your Bluetooth is still running on your phone. It just disconnects.

So as people began to realize what was going on, they got upset. Apple was forced to produce a formal explanation. And what they said was, well, yes. All we're doing is disconnecting things because we want - we, Apple, have decided for you that we know better. And so AirDrop, AirPlay, the Apple Pencil, the Watch, then there are continuity features like Handoff and Instant Hotspot and Location Services, we think those are more important than you being able to turn your stuff off when you want. So we're going to ignore you when you turn off WiFi and Bluetooth and have all those things still work.

The good news is you cannot - well, the bad news is you cannot do it from the Control Center. The good news is you can still do it from the Control Panel, the original root of control, if you go to the little rotating gear icon, go in there. When you turn them off

there, they are actually truly off. No more broadcasting, no more receiving, no more potential security vulnerabilities which otherwise exist constantly, thanks to things like BlueBorne - although in this case Apple was not a victim of those attacks. Everybody else was. So just a heads-up to our listeners. You don't save power, you don't get security if you turn these things off from the Control Center. You've got to go the root Control Panel in order to get what you thought you were getting from the Control Center.

PADRE: Right. And we've got people in the chatroom who are mentioning what you did, which is why Apple did this. And we know why Apple did it. And there is an engineering decision behind it. They believe that you want to stay connected to peripherals, and the average user will not want to lose connectivity to the keyboard and the Pencil and the Watch that they might be wearing. And I get all that. But here's the thing for me, Steve, the fact that they put the icons for Bluetooth, the universally recognized icons for Bluetooth and WiFi, and you can gray them out by pressing them, just like you would in iOS, just like you would in Windows or OS X or Android, and it doesn't do anything.

**Steve:** Right.

PADRE: That part, that's the betrayal to me. That's the whole, well, if you weren't going to turn it off, why even put those icons there? That makes no sense.

**Steve:** Yes. For example, they could have had a three-state button where you press it and it gets a ring around it or something that says, okay, other things that Apple doesn't make are disconnected. But Apple-trusted things are still there. And then you hold it longer, or shake it or do something, and then it goes, like, totally dark, which is, okay, now, sorry, you can't type anymore on your Bluetooth keyboard. Do that a couple times, you'll figure out what's going on. But no. They just said - and get this. The Bluetooth spontaneously turns itself back on at 5:00 a.m. I have no idea why.

PADRE: Because at 5:00 a.m. you - we what? Wait, I'm sorry, I didn't hear that part. Seriously?

**Steve:** Yes, yes. At 5:00 a.m. Bluetooth, bing, comes back on. Every morning.

PADRE: That's a very curious decision. Wait, wait. But if I turn it off in settings, it's off off; right? That's like the actual way it powers down.

**Steve:** Yes. If you turn it off in the Control Panel, it's shut down. But if you do the Control Center sleeper, it wakes up at 5:00 a.m. And also apparently if you wander too far. It realizes, oh, he's somewhere else. We maybe should turn on to look around again. So there's both location and 5:00 a.m.-ness, which completely overrides what you've done, just because.

PADRE: You're kidding.

**Steve:** And I've seen that. I've noticed it's back on the next day. And before I understood what was going on, it was like, okay, what is this, a bug? No, this is a feature.

PADRE: And I will mention, because they're asking about airplane mode in the chatroom, airplane mode does shut it off, but it also shuts off the cellular radio, which is normally what people don't want to do.

**Steve:** Right.

PADRE: I mean, more than just security, and you mentioned this, Steve, I sometimes need to save power on my device, and so I'll shut down all the radios, the GPS, the WiFi, the whole - shut everything down. And I can do that at a glance on Android, just by dropping down the menu and graying out the buttons. I would be very upset if I learned that those radios just stayed on because Google decided that they should just be on.

**Steve:** And get this. If you have some of those things turned off, then you go into airplane mode, when you come out of airplane mode - you can already guess where I'm headed - they're all back on again. So it doesn't even remember the pre-airplane mode preferences that you had. It says, oh, we're just going to err always in the direction of turning everything on all the time. And when you turn them off, remember, you get warnings. Oh, warning, location services are less accurate now. It's like, yes, go away.

PADRE: I will say that when the next version of BlueBorne comes out, and iOS devices are once again vulnerable, just remember, walk around with your scanner at 5:00 o'clock in the morning, and you'll probably find a few.

**Steve:** That's right. Target-rich opportunity at 5:01.

PADRE: Steve, there's been an ongoing battle in the enterprise/prosumer world with securing the devices that we use every day. And unfortunately there's been some laxity among the manufacturers who don't think that supplying security past the first 90 or 100 days is really of any benefit to them. D-Link had a case filed against it by the FTC where they were going to be held to account for some very shoddy network gear that they put out, specifically routers, wireless access points, and cameras. But now there's a development in the case.

**Steve:** Well, yeah. And we talked about several weeks ago the really atrocious discovery of - I've often said anybody can make a mistake, but people should be held accountable for their policies, the things that are done deliberately, the things that some committee decided. And the D-Link router revelations were the latter of that. They were just evidence of no one caring. When you have fixed and publicly known administrative usernames and passwords exposed to the public Internet, that's unconscionable. I mean, that's someone's decision to do that.

So we've often talked about the dilemma that we have in the industry and wondered how this gets fixed. How do you get a company like D-Link to care? How do you get those deliberate policies to change? Because consumers just look at the prices and the features and how many stars it has on Amazon and go, oh, that looks good, and they click it, and they end up being victims often when what they clicked on has a security catastrophe.

So one solution, which we talked about at the time, was that the FTC, the U.S. Federal Trade Commission that has legal oversight and the goal to protect consumers against this kind of misconduct, filed a suit, a big suit against D-Link as a consequence of their clearly sloppy security. The suit had six claims. And what's in the news that you're referring to, Padre, is that a judge immediately tossed out three of those six. And I looked at it, and first of all I didn't think that was good news because I want a company like D-Link to be held responsible. This may not be the mechanism, but it is a mechanism that in the U.S. we have.

It turns out, though, I agreed with the judge, I agreed with his decision because - and the FTC should have known better. Essentially, the judge was discarding the claims which alleged without proof that there was damage that resulted as a consequence. And that's kind of the gotcha in this kind of situation. The plaintiffs have to have standing. And if

you're alleging damage, you've got to have some. You've got to have some proof. You've got to have some evidence of damage. And the FTC was just saying, "Bad D-Link, bad D-Link." But you can't sue anybody over bad.

PADRE: Right.

**Steve:** Now, you could sue them if we had laws, but we don't have laws. That is, if there was a law that any device that was being sold to a consumer could not contain a fixed static known admin username and login, fine, have a law. Then when they do that, they're breaking the law. But we don't have any laws. We just have "Bad D-Link" and, unfortunately, "Go to your corner." That's just not enough to sue. So hopefully the other three remaining surviving claims will be sufficient to continue this legal case. Maybe the FTC was just throwing everything at the wall, hoping that some would stick. But the judge isn't buying it in this case, saying, well, no. Yes, we don't disagree that they're bad, but there isn't a law against badness at this point. So fix that, and then we can proceed.

PADRE: I think the FTC was feeling the heat from the botnet attacks, of which we believe there were a lot of D-Link devices included. And let me be clear, D-Link is absolutely at fault for a lot of this. Some of the most basic vulnerabilities have existed in their products for over 10 years that they haven't patched. And that's a pretty - yeah, that's bad. But as you mentioned, it's not illegal to make a bad product.

**Steve:** Right.

PADRE: And in fact the judge specifically said, look, this might be, quote, "an annoyance and an inconvenience," unquote. But the FTC had not provided any specific instances of damage, monetary damage, and that's what the law is about. Now, there are going to be people say, oh, well, this judge doesn't understand it. This judge is actually quite good at this. He's been very consistent in his positions. He had a case in July of 2016 against Pokemon Go. Some property owners were saying there was damage because there were portals put on their property. And his thing was, look, unless you can show me what was actually damaged, I'm not going to award you $5 million.

He did the same thing in March of 2017 with the Telesocial case, who said that they were hacked by Orange, and they had some magical math that led up to, like, $100 million worth of damage. And the judge was saying, I don't see a single dollar that's not completely theoretical. And in June of 2015 there was OpenText's suit against Box, and it was the same thing. It was like, okay, you allege that there was a hacking. But before you even try to argue whether or not there was a hacking, where's the damage? We're not going any further until you show me where the damage is.

And Steve, we've argued the other side because we've both talked about people, researchers, who have been caught, and then you have some trumped up charge that's brought against them alleging $50 million worth of damage because you sent an email, and a lot of our employees read it, and it cost time, and that's $50 million. He's the judge you want on cases like that because he'll say, "Show me."

**Steve:** Yes. And we know that there's a problem with, as they're called, "frivolous lawsuits." And so there has to be pushback against that because the system can be abused in this way. So as I said, I agree with the judge in this case. As our listeners know, I am incensed that D-Link is conducting themselves this way. But unless our current laws can be applied to create an environment where this is not permitted, then we need new laws. We need to make it illegal to sell clearly irresponsible security to consumers. When we have that, then that's what the FTC can use.

PADRE: Precisely. It's just our laws have not caught up yet. People are saying in the chatroom that, oh, but there are laws against harm. Yes, but in this particular case harm is defined by monetary loss. And if you don't have monetary loss, then by the legal standing there's no harm. Yes, it's irresponsible. Yes, it's horrible. Yes, you shouldn't buy from D-Link until they fix this. But it's not illegal. Again, the big push is the market sorts it; right? If D-Link makes such a bad product, why are you buying it? Why do you keep purchasing their gear?

**Steve:** Well, and of course I've always often talked about the license agreements that we click through which are complete hold-harmless agreements. So everybody using Windows has given up all of their rights to ever complain about anything. That's what it says in the fine print is no matter what this does, we're not responsible. I mean, cars don't get that luxury. The wheels fall off, that manufacturer's in deep trouble. VW knows all about that with their emissions. So the computer industry is in this weird blessed bubble of, sorry, you're using it, so good luck, even though we got your money. It's incredible.

PADRE: Okay, well, let's move to another story that has both good things and bad things. We'd all like more accurate GPS; right? I mean, I love it when my GPS doesn't have me jump from one side of the city to the next.

**Steve:** Or like you're following your mapping software, and it tells you to get onto the freeway at the next opportunity. It's like, wait, I'm on the freeway. But then you realize, oh, there's a road over to the side.

PADRE: And they think I've, yeah, veered off. No.

**Steve:** And that's where it thinks you are. Anyway, so yes, all this by way of noting that Broadcom has - and this is just fun to know. I just picked this up as I was putting things together, and I thought, oh, this will be just neat to share real quickly. They've produced the first low-power, commercially feasible, next-generation GPS chip. It's the BCM47755. They started sampling it. It will be appearing in next year's smartphones. They're not saying which ones or whose. But in 2018 this next-generation chip will start happening.

The current generation of smartphone, cell phone GPS chips use the original GPS ranging and positioning technology designated as L1. And L1 technology uses a chirp which is lower complexity and longer, and a lower frequency. All of those things, they work, but they have a problem with resolution, that is, your actual location resolution. And they have a problem when you're in an environment with lots of reflections, like you're in downtown of a city, and the signals might be bouncing off of buildings, and that's a problem known as multipath. That is, there are multiple paths by which a satellite signal can reach you due to reflection off of other objects, which tend to confuse things.

So the next generation is known as L5, which uses a much higher frequency, a much more complex burst pattern, which is, as a consequence, much shorter. The shorter burst means that it's over by the time a secondary or tertiary or whatever reflection might reach you. So the phone is smart enough, or this chip subsystem, the GPS subsystem, to only pay attention to the first one. So the higher frequency has a reduced problem with multipath interference in general. The shorter burst further improves that. And the higher complexity, higher frequency signal, which can be carried in a higher frequency carrier, means yo get much better location accuracy - so much so that whereas what we've been using has five-meter accuracy, next year these chips will give us 30-centimeter accuracy. So, wow. Very cool.

PADRE: That is such a jump. I mean, military GPS has had increased accuracy since the start of the system.

**Steve:** Right.

PADRE: But 30-centimeter resolution for a consumer product? I didn't even think that would be allowed.

**Steve:** Yeah. Yeah, it's going to be very cool. Oh, and I forgot to say, half the power.

PADRE: Oh, well, yes.

**Steve:** Half the power. I think it's a 25-nanometer process or something. And so they're reducing the size and cut the power in half.

PADRE: Anybody who's ever used Google Maps on a phone and watched your battery just drain, or something like Pokemon Go or any AR game that has to tag their location in the real world, you understand that the GPS chipset actually pulls a lot of power, way more than you think it would because it doesn't actually transmit anything.

**Steve:** But it's doing so much work. In fact, this has a dual-core ARM as part of it in order to do all the background math required in order to make this happen.

PADRE: That's where the power saving is coming from, because you're not tagging the main processor anymore with doing all the triangulation.

**Steve:** Right, right.

PADRE: Okay, that makes sense. All right. Well, hey, Steve, I don't want to jinx it, but I think we just had a positive story.

**Steve:** Oh, good, yes. In fact, we have a few more coming up. Although we do have another caution. I meant to put a picture of this in the show notes, Padre. It is on the page if you click the link in the show notes, so at least our video watchers can see it. But this is just a reminder. One of the things that's happening is that things consumers get are containing high-density barcodes, specifically boarding passes. And there is on the screen now. That is a very high-density barcode. And as an engineer and a techie, I would look at that, and I would think, ooh, there's a lot of information in there. That's more than my gate number and my flight number and my last name. That's so many bits, that could contain a lot.

It turns out it does. So one of the things that people are beginning to do is, because of the ubiquity of cameras and Snapchat and Facebook and everything, where it's so easy to post stuff, people are taking pictures of their boarding passes and putting them online, saying look where I'm going. Well, it turns out, yes, and what you're also posting is, "and look who I am." It's like your passport number, your date of birth, passport expiration, all kinds of stuff which you don't want to be posting publicly is encoded in that high-density bar.

So if you've got any photo manipulation software, fuzz that out. Blur it into a gray rectangle or a gray ellipse, or scribble over it, or do something. Or just snip it out and don't post the portion of your boarding pass that contains a high-density barcode unless you know what it contains because it could have way more information and currently does than you believe you are posting, much more than just is on the face of the

boarding pass itself because it's computer-to-computer information, and they're like, hey, we got four kbits. Let's use them all. Right. Just don't...

PADRE: How easy it would be to modify people's high-density barcodes.

**Steve:** Oh, yeah, true. You could get into some mischief.

PADRE: I mean, it's just a bunch of blobs of ink on my boarding pass.

**Steve:** Right, and no one can see them.

PADRE: I don't want to suggest that, Steve, because someone will create some device that will thermal print a new boarding pass that you could put on people's - no, let's not do that.

**Steve:** Yeah. So we also had, and this is - this kind of went by, and I thought, yeah, okay. But it came to so many people's attention. Again, my policy is always mistakes happen. Policies are what people should be held accountable for. Mistakes, yeah, they happen. Maybe this demonstrates a process failure where you might say, okay, we should have a process to have someone check postings before they're made public. The news is that Adobe inadvertently publicly posted their private PGP key. Whoops.

PADRE: [Sighs]

**Steve:** Yeah. And in backtracking it turns out that they were using a PGP extension for Chrome and Firefox known as Mailvelope. And there's a web page interface where there's three blue buttons. The left-hand one says "Public," the middle one says "Private," and the third one says "All." And the person who did this was wanting to post the public PGP email key, but "Public" wasn't pressed. "Export All" was pressed. And so the extension dutifully posted both: in the first half, the public key; and, in the second half, the private key.

PADRE: Why does that button even exist, Steve? When would you want to post both your public and your private key?

**Steve:** And not only that, why isn't there, like, a warning?

PADRE: Are you really sure you want to do this?

**Steve:** Big, red, "Hey, you're asking for your private key. Are you really...." And then I'm sure the person would have gone, ooh, crap, thank you, thank you, thank you, thank you, thank you. Instead, just cut, copy, and paste, and there it is for the world to see. Needless to say, they took it down. They rekeyed. And now they posted their new public key.

PADRE: This is the security equivalent of drunk texting. Oh, no, no, I didn't want that. That's when you really wish that there was a recall button for that message you just sent out. Wow.

**Steve:** Once it's on the Internet, it's loose.

PADRE: I mean, okay. It's easy to make fun of this. But this could happen to people. People post the wrong stuff all the time. It does really make me wonder about that interface, though, that you've got those three options right there, and it is so easy to

post any of them. That's just bad UI.

**Steve:** Yeah. I mean, I'm sure the Mailvelope developers must be getting feedback that a techie in Adobe let the wrong data be published. You're right. It's unconscionable that this thing doesn't say, wait, you've got to stand up, click your heels together three times, and hold Ctrl-Alt-Del down in order to enable this button. But no.

PADRE: And again, people will say, well, the developer of the software never knew that someone would be stupid enough to - but we're past that point; right? We now have to be at the point where security comes first, not at the end. We're not trying to tie up the things that might be a problem. As we develop a solution, you always have security at the forefront. And when I'm designing that UI, I'm thinking to myself, is it a good idea to have the public key right next to the private key? And the answer is no, of course not. There would be no situation where I would want those posted at the same time.

**Steve:** That's a very good point. Good, yes, that's even better. You're right. The UI should not even offer in that context to put out the public key, I mean the private key. It should be deeply protected so that it's easy to expose and provide your public key, which you're having to do all the time. But the things you never want to do shouldn't be next door to the thing you always want to do.

PADRE: Yeah.

**Steve:** Two quick notes about SpinRite, just very...

PADRE: Oh, yes, please.

**Steve:** They are quick. Vigen Galustian, I hope I pronounced your name right. If not, sorry. Vigen, maybe Vigen? Yeah, I think Vigen Galustian in West Hills, California. He said: "I'm an avid user of SpinRite. It has saved me several times. I hope in your future upgrade you can make it to support USB connection and independent of OS." So I just wanted to say, I did mention this last week, but that's absolutely in the - I want to say "floor plan." That's not the right word. The development - there's a word for that. The development timeline. I'm going to do that second.

First comes AHCI support, which will be native hardware and OS dependent. So it'll be able to run across platforms on Macs and Linux and so forth. So we will allow people to download an ISO that they can burn, and I'll get to the next point in a second about that. So, yes. But because USB is slightly less important than AHCI, which is now the universal standard platform hardware - and I don't want to delay SpinRite, that is, the 6.1, a second longer than I have to. And as I said, I'll be making it available even before its release to our podcast listeners as thanks for their long-term support.

6.1 will happen immediately when we get the crazy performance boost and expanded compatibility. And then 6.2 will be similar low-level hardware UHCI, and there's an EHCI, I think it is. Anyway, it'll support the USB 1.1, 2.0, and 3.0 hardware-level chips to provide the same kind of raw, low-level, high-speed performance over USB links. And that'll be in 6.2 that I will immediately start on. This is all going to happen in a single development continuum. But my intention is to drop out intermediate technology in a highest priority to incrementally lesser priority sequence, all of that being intended to create the next-generation hardware and driver foundation for SpinRite 7.0, which will be where we go crazy with features.

So first is hardware compatibility and performance and OS independence, and then 7 will

be like all the stuff people want for multiple drives, doing multiple drives at the same time, extracting files themselves rather than fixing the surface, lots of file system recovery and reconstruction. Basically 7.0 is where I think SpinRite will stay for a while as we produce all those other features.

PADRE: Of course.

**Steve:** And then Greg - and he put his location as "Somewhere." So Greg is somewhere.

PADRE: He trusts no one. Good man.

**Steve:** Yes. His subject line was: "Will new SpinRite make it easy to run SpinRite off a USB drive?" And I thought, Greg, it is now. So he wrote: "It is more trouble to always have to get out a CD drive to use SpinRite. Sure would be handy if it was easy to put SpinRite on a USB stick and run it from there. The current method for doing this is too complicated for me. Hopefully new version will make this easier." Okay. You put the USB stick in your computer, and you press the drive letter...

PADRE: Yeah, it does it right now.

**Steve:** ...of the USB, and it does it right now.

PADRE: I just did that, like, three days ago.

**Steve:** So anyway, Greg, we're not laughing at you. There are no bad questions. I just wanted to tell you, read the user interface. And when that screen comes up, you'll find that it says to install it on a USB drive, just press the keyboard key for the drive letter, and it'll confirm that's what you want to do because it does wipe out the contents that you currently have. It formats your USB stick for you, making it bootable. And it boots the USB and runs SpinRite. So it's already in there.

PADRE: I will say, though, there are some flash drives out there, typically the really cheap ones...

**Steve:** That will not boot.

PADRE: That will not boot.

**Steve:** Yes, yes.

PADRE: So if you put it on that, it won't boot. And it just means try a different drive.

**Steve:** Right.

PADRE: And unfortunately there's not a really good guide about which chipsets will boot and which ones will not. It's just trial and error. I've separated all - because I have literally hundreds of flash drives from the different events I go to, and I'm sure all of them are safe. But I have two tubs, and one says "Bootable" and one says "Not Bootable" because I'm always making bootable USBs.

**Steve:** Ah, nice.

PADRE: Oh, Steve, I did want to add in one thing because I won't be able to do it next

week. But I have a Brother from - he's Nigerian, but he's living in Zambia. And he runs a school there. And they had lost all of their semester's data and the yearbook pics on a bad drive. And he was telling me about this over the summer. I was in Portland doing that two months of off-the-grid type thing. And so I actually gave him my - in my toolkit I had the USB drive with SpinRite. I said, "Try this once and see if it works." And he wrote back to me just a week after he got home, and he said, "Oh, my gosh, I tried it. It got it back. We copied all the data off. I'm going to go buy a copy now." I said, "Yes, that's what we want."

**Steve:** Perfect.

PADRE: From one IT Jesuit to another.

**Steve:** Interdigital Jesuit.

PADRE: Indeed. Now, that's going to close the loop. And I can't believe this, we might actually get through all the material, which I didn't think was possible.

**Steve:** Actually, no, we're going to skip that. Let's go to the DOM fuzzing.

PADRE: Even better.

**Steve:** Because, yeah, because we've got eight minutes to go, and we are actually going to get through all of our material right on time. So this is the title of the show. And I just, again, I said, okay, if I have the opportunity to call the podcast "The Great DOM Fuzz-Off," how can we not do that?

So an engineer named Ivan Fratric joined Google's Project Zero. And in the past, prior to joining Project Zero, I don't know if he was at Google or where he was. But the way he explained it, he had a history of having explored DOM fuzzing. And we'll explain what that is in a second. And so he had already implemented a number of previous systems. So he decided that, upon joining the Project Zero team, that he wanted to start fresh using all the experience he had acquired to create a next-generation and open-source - this is all up on GitHub - DOM fuzzer.

Okay. So what's a fuzzer, and why are they useful? Well, the Document Object Model, which is what DOM stands for, is this increasingly complex, but beautifully unified means of programmatically accessing the contents of a web page. So you're able - there's this notion of the body of the page, and the body contains sections, and the sections contain paragraphs, and the paragraphs contain content and tags and forms and so forth. All of that has been formalized in a structure, and it's inherently a hierarchy so that you have lists of things. And then those objects contain lists, and those objects contain lists and so forth, forming something that can be - it can be enumerated, that is, it can be stepped through. It can be examined by code. It's a beautifully powerful system, which is what enables today's dynamic web pages and all of the stuff that we're doing.

And of course part of that is JavaScript, which is the default language which can be run on the page to be the thing that looks at the DOM, the Document Object Model, and participate in it, and be part of it also because you can have JavaScript embedded in these objects, or invocations of JavaScript. It's very complicated, but it's a system which now works and is well-defined. The problem is it all uses the "I" word, which we talk about from time to time. It is one big massive interpreter. And as we know, interpreters are hard. Interpreters are difficult to get right. And the reason is that they're built to interpret what their designers build them to interpret. But it's very different to do that

than to have them robust against abuse of interpretation.

Fuzzing is an intriguing idea. That is, and we talked about this, I remember one of the early cases was Eeye Systems in Aliso Viejo, Southern California. They had a whole bunch of PCs, I think they were Windows machines, that they were fuzzing. And they were finding all kinds of problems that had never been found because, after all, an API is an interpreter. It's a set of calls, and you give it arguments, and it does something with the arguments. It interprets what you give it, and hopefully it doesn't crash.

And so what was happening was Eeye Systems, they were crashing all these Windows machines, and then they would look at the log of the fuzzing of just the nonsense that they had given to the machine, and then they would go, ooh, why did that crash? And that's the beginning of finding an exploit is because it should never crash. And so what Ivan did was he wrote a next-generation DOM fuzzer. That is, he wrote technology to create crazy, insane, non-logical web pages containing web content and JavaScript, designed to find crashes, designed to crash the web browser. And that should never happen.

PADRE: Steve, let me ask you this. So the fuzzer is just generating random elements. It's just putting everything in there, trying to see how it will be interpreted. Do I get to control the randomness of those elements? Can I ensure that it's running all the elements that I've tried in the past? Or is this just truly pile everything into a blender and let it stream out?

**Steve:** Yes. It needs to be structured. And in fact in his blog posting he talks about the fact that you could do random stuff, but random stuff isn't going to mean anything. And if you did deterministic stuff, well, it would all mean something, but it wouldn't be random enough. So you have to have a constrained system that still has sufficient degrees of freedom to, like, poke into the corners and find the dark spots and the soft spots and the stuff that hasn't been caught by the authors. And, I mean, these are mature browsers that are in use right now. We're all using them.

So what happened was no browser survived. Everything collapsed under the attack, essentially, the benign probing of this next-generation DOM fuzzing technology. As it happens, Google's Chrome Blink engine was found to have two bugs that were - they were Project Zero. They were zero days. They were unknown. Firefox's Gecko engine had four. The Trident engine in IE also had four. The Edge HTML engine in the Edge browser was found to have six. And the WebKit engine in Apple's Safari browser was the "winner" in that sense.

PADRE: The winner, yeah.

**Steve:** The big loser had 17 different problems. All of these have been assigned Project Zero bug IDs and have been responsibly disclosed to their vendors, and I'm sure are in the process of being fixed. But anyway, it was just really interesting. He wrote: "We tested five browsers" - those five I just said - "with the highest market share: Google Chrome, Mozilla Firefox, IE, Edge, and Safari. We gave each browser approximately 100,000,000 iterations with the fuzzer." That's the other thing. With this kind of automation, you can just do far more than you could ever do, even with a bunch of monkeys typing on the keyboard. I mean, this thing is fast, and so it gives you great coverage.

So it says: "We gave each browser approximately 100,000,000 iterations with the fuzzer and recorded the crashes." Again, no browser should crash when you give it a web page. It's not supposed to do that. When it does, that's a reason to go look and find what it

was that made it crash; what part of the fuzz that it received it was unable to handle.

PADRE: Was there any consensus on a particular fuzz that was causing the most crashes? I'm looking at the numbers, and maybe like 10, 11 I see a couple times.

**Steve:** Ah, good point, yes, repeats of the same debris, yes. So he said: "If we fuzzed some browsers for more than 100,000,000 iterations, only the bugs found within this number of iterations were counted in the results." So it may not have been the first hundred million, but a window of a hundred million somewhere.

He said: "Running this number of iterations would take too long [yeah] on a single machine and thus requires" - and I love the phrase - "fuzzing at scale."

PADRE: Overfuzzing.

**Steve:** Yeah, fuzzing at scale. Overfuzzing, that's good. "But it is still well within the pay range of a determined attacker. For reference, it can be done for about $1,000 on Google Compute Engine, given the smallest possible VM size, preemptible VMs," he says, "which I think work well for fuzzing jobs as they don't need to be up all the time, and 10 seconds per run." So you can spend $1,000. And if you are good enough to write this code - and it's now in the public domain, it's now open source - you, too, can find zero days in popular web browsers.

PADRE: That's scary. That's absolutely scary.

**Steve:** And fuzzing is cool.

PADRE: Now, so the zero bug IDs that they identified in this test, those are just ones that crashed a browser. But if it's listed here, does that mean that they can reliably crash the browser with that bug, or only in a certain sequence?

**Steve:** It means that - okay. Yes. In order to qualify for a Project Zero zero day, it has to be a zero day. I mean, so they opportunistically found it; but then they said, whoa, what just happened? Because - and this is always the case. You threw some monster crap at the browser, and it collapsed. But it turns out it's only one number that had a semicolon or a back tick or something in it that was the culprit in this much larger page.

PADRE: So you rewind the sequence and keep playing back little bits and pieces until you find a particular sequence.

**Steve:** Exactly. Exactly that.

PADRE: That makes sense.

**Steve:** Yup. And then you have found something that a bad guy could exploit if you didn't fix it first.

PADRE: I remember looking for zero days back when I was in high school. And the idea that sometime in the very near future I would essentially be able to spin up the equivalent of tens of thousands of workstations and test against them for basically pennies...

**Steve:** Wow.

PADRE: Yeah. I always thought it was going to be, oh, I'm just going to be working on maybe one or two workstations, and maybe I'll luck into a vulnerability here or there. But this is a blueprint for making zero days in sort of an assembly line.

**Steve:** Yeah.

PADRE: Which, again, scary.

**Steve:** Yeah. And I think what this also demonstrates, we saw this before with Eeye. We've talked about fuzzing a number of times. It is too valuable not to do. It ought to be part of any security-oriented project's development chain, is when you think you've got it all nailed down, throw a bunch of crap at it and see if it survives because, if it doesn't, you need to find out why.

PADRE: Well, Steve, I think our audience has been thoroughly fuzzed by that last story, so this might be a good time to end. My friend, it is always a pleasure to work with you. And again, I miss Leo when he goes away. It's always fun having him in the studio. I learn so much from him. But when he does go away, it means I get to play with you, and that's always a privilege. So thank you very much for letting me participate in what has been a grand pageant of security awesomeness.

**Steve:** Has been. And Father, it goes both ways. I know that Leo is enjoying his vacations. He announced earlier this year he plans to be taking a lot of them.

PADRE: Yes. I'm looking forward to that.

**Steve:** So you are certainly welcome back.

PADRE: Folks, that does it for this episode of Security Now!. Don't forget that we're live here on the TWiT TV Network every Tuesday at 13:30 Pacific time. Steve will always be here to inject you with some healthy, healthy paranoia, which is good in this day and age, and help you understand the wonderful world of security because the more you understand, the less security seems like a black box.

You can also find all of our shows at the show page at TWiT.tv/sn, as well as iTunes, Stitcher, and wherever fine podcasts are found. You can also find high-quality audio downloads on GRC.com, which is also where you'll find everything GRC has created - SpinRite, ShieldsUP!, and, coming soon, SQRL. I'm Father Robert Ballecer, the Digital Jesuit, saying that if you want to keep your data safe going into the future, you're going to need some Security Now!.

**Steve:** Thanks, Padre.