IoT Infancy

Description: Leo and I first cover the past week's security events, including the collapse of the Feinstein-Burr encryption bill, the result of the Oracle/Google trial, Google's attempts to keep Android in the field up-to-date, an intermediate certificate issued to an Internet appliance maker, lots of bad news about laptop add-on bloatware, and an update on SQRL's development. Then we take the first of two weeks' look at the many problems with our infantile Internet of Things.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-562.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-562-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We're going to talk about the latest news in the security world, and then start to dissect how Internet of Things devices work and how insecure they are. That's why he says they're in their infancy. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 562, recorded Tuesday, May 31st, 2016: IoT Infancy, Part 1.

It's time for Security Now!, the show where we cover the latest security news. We help you protect yourself online, help you become an expert who can help your friends and family and those people around you. People who listen to Security Now! not only are safer themselves, but make sure that their loved ones are safer, too. So this is a show you have to listen to. Steve Gibson is here, the Explainer in Chief. He makes it all possible. Hello, Steve.

Steve Gibson: Yo, Leo. Great to be with you again for Episode 562.

Leo: Yikes.

Steve: As we close in on the end of our 11th year of this never-ending quest for security.

Leo: You know, in 11 years, the one thing I've never asked you - in 11 years. You'd think this would be the kind of thing one buddy would ask another. Spaces or tabs?

Only "Silicon...." You know, I got dead silence when I asked Andy Ihnatko and Rene Ritchie.

Steve: I watched it.

Leo: They hadn't seen "Silicon Valley" yet. That's why. I have to, you know, Lisa goes to bed because it's late. After "Game of Thrones" she goes to bed. But I say, "Honey, I have to stay up because part of my job is knowing what happens on 'Silicon Valley' and 'Veep.'"

Steve: That's right.

Leo: And let's face it, John Oliver, as well.

Steve: Yup.

Leo: So that's part of my research. Sunday night I'm stuck in front of the - glued in front of the tube.

Steve: Yup. I have, as our listeners know, I'm still coding. All of the code that I write is in a DOS box in a text...

Leo: Do you use Brief? What do you use?

Steve: I use Brief, yeah. And Brief actually has its roots in Emacs. It was some guys that were Unix people that originally created Brief sort of in the image. And so there is a LISP-like language which actually glues it together. So it's a completely programmable editor. BRIEF was an acronym that involved bars reconfigurable. So it's like, interactive reconfigurable editor, blah blah, you know, programmers, I don't know what it was. But there was an acronym [Basic Reconfigurable Interactive Editing Facility]. And it was a company called UnderWare.

Leo: I remember it, yeah.

Steve: Yeah.

Leo: Only hardcore people used Brief.

Steve: Dave somebody, can't quite get his name.

Leo: It's gone, though; right? I mean, they don't make it anymore.

Steve: Oh, yeah. It got bought. Brief got sold to someone. But it still works. And but the point is that I was also a devout WordStar user. And so one of the things that I did is to completely reconfigure the keyboard mapping so that it uses WordStar keystrokes, where you hold the control down with your pinkie. And of course the Ctrl key is where it was meant to be, immediately to the left of the "A."

Leo: Yes.

Steve: Not Caps Lock. Whoever...

Leo: But that's what I do the first time I - any new machine, the first thing I do is I remap Caps Lock to Ctrl.

Steve: Exactly.

Leo: Because otherwise I'm going to be typing at y'all in caps sooner or later. They call it "Emacs finger," by the way, your Emacs pinky, from a little overuse, because Emacs, everything's a Ctrl keystroke.

Steve: Right. So anyway, the point is that so not only does Brief know about WordStar keystrokes, but then I created a superset of those. For example, I have one key, actually it's Ctrl-I, that a great programming friend of mine came up with that is so useful. He hadn't ever seen it anywhere, but it was implemented back in the '80s on an Interdata minicomputer, and it pulls the character down from the line above. And it turns out it's just so handy to have that. So I adopted that in Brief. So I have that.

But also, for example, I have one that immediately cleans up the file. It removes any white space from the end of lines. So I can do that. And it says, oh, four lines were cleaned up. It's like, oh, okay, fine. You know, where you just sort of have some spaces at the end of something that are obviously not important. And so this removes them. And, of course, there's a tabification, which is where I was coming with this, is that, if there's ever more than eight spaces in a row, it says tsk-tsk-tsk, and it replaces them where it can. It removes them and puts tabs in, in order to, of course, reduce the source file size.

Leo: You really missed out, though. If you were using the one true editor, you would be able to do all of that and so much more. And you wouldn't even have to - I can't believe Brief Pro is 120 bucks. Emacs is free on every platform. But that's okay. That's okay. The problem with Emacs is you spend...

Steve: I paid for mine 30 years ago, and I'm still using it.

Leo: I know, you got your money's worth.

Steve: And I actually live in it every day.

Leo: You got your money's worth. The problem with Emacs, and I really - this is really the truth, is it is so infinitely configurable, it really isn't an editor. It's just a framework that you never get any work done because you're so busy changing, you know, fixing the UI and making it...

Steve: That's the problem with object-oriented languages is that many...

Leo: Yeah, you're always messing with it.

Steve: The programmers get so caught up in expressing the problem in the language that it's like, okay, great, you know, you've got this fabulous object hierarchy design now which beautifully represents the problem. Have you solved it? Well, no. But, boy, we're going to - as soon as we actually write some code, it's going to be amazing.

Leo: Going to be amazing. But first you have to set it all up.

Steve: It's like, uh, okay, yeah.

Leo: Just so. I'm working on the header files. I'll be done soon. Ctrl KS, save that, and we'll move on to the news of the week.

Steve: Yup. So, yeah, lots to talk about. We're going to talk about the Burr-Feinstein encryption bill, the outcome of the Oracle/Google fight, Google's pressure to increase updates of Android in the field. I got a lot of tweets about the news of a company called Blue Coat obtaining an intermediate CA certificate from Symantec. Some news from some security research, insecurity of laptop add-on bloatware of various kinds. Last week we ran out of time, and I didn't get a chance to talk about what's been going on with SQRL, and so we're going to do that. And I had promised to talk about a report that Rapid7 security did when they looked closely at Internet-connected baby monitors.

Well, it turns out that when I laid everything out, there was just so much to talk about that I thought, okay, I mean, I wanted to talk - I ran across other information about IoT stuff. And so I named this podcast "IoT Infancy," but this ended up being Part 1 of that topic. So we have so much to talk about that I just realized we wouldn't be able to do it justice, and I wanted to. So next week we will do Part 2, where we'll deal with all of the crazy security news that has transpired since then and wrap up this topic by looking at some really frightening details of what Rapid7 found. But this week's first part coverage is sort of the meta view of it because there's some interesting other things that I found that sort of leads us into the look at the details that I know our listeners will find really interesting.

So, lots to do. And at the very end, so that our security-focused people don't have to listen to it, a lot of our listeners have asked for what's going on with this whole Seriphos/Enerphos ingredient of the Healthy Sleep Formula. So I promise to move it to the very end so people can say, okay, fine, stop, and then not have to listen to any of that. Although, frankly, 1,700 people a day are now visiting the Healthy Sleep Formula page, and it's become a thing. So I need to give it a little bit of time, and I don't have anywhere else to do that other than here, so we'll just stick it at the end so people can say, okay, not for me, stop.

So Burr-Feinstein. The good news is it met a quick and swift and complete death in the Senate. Reuters had some great coverage and uncovered some little tidbits that I don't think we would have known about otherwise. The way they expressed it was they said: "After a rampage" - talking about the San Bernardino rampage - "a rampage that left 14 people dead in San Bernardino, key U.S. lawmakers pledged to seek a law requiring technology companies to give law enforcement agencies a 'backdoor'" - whatever that means, and we've talked about how ill-defined important terms are, unfortunately, but, you know, so for whatever that means - "to encrypted communications and electronic devices such as the iPhone used by one of the shooters.

"Now," they wrote, "only months later, much of the support is gone, and the push for legislation dead, according to sources in congressional offices, the administration, and the tech sector. Draft legislation that senators Richard Burr and Dianne Feinstein, the Republican and Democratic leaders of the Intelligence Committee, had circulated weeks ago likely will not be introduced this year and, even if it were, would stand no chance of advancing, these sources said."

The other little tidbits that I pulled out of much longer coverage was that "The short life of the push for legislation illustrations the intractable nature of the debate over digital surveillance and encryption which has been raging in one form or another since the 1990s." What I didn't know was that "The CIA and NSA were ambivalent," meaning, okay, but we know that the FBI wasn't. But the CIA and NSA were, "according to several current and former intelligence officials," wrote Reuters, "in part because officials in those agencies feared any new law would interfere with their own encryption efforts." So they wanted things to sort of be left alone. They were like, eh, we have to be very careful with what we do here, if we implement any new legislation.

And Reuters also wrote that "Half a dozen people familiar with the White House deliberations said they were hamstrung by a longstanding split within the Obama administration which pitted Comey and the DoJ against technology advisors and other agencies including the Commerce and State departments." And then, finally, they said there was "reluctance to take on the tech industry in an election year," which I thought was interesting.

So, yes, the tech industry does have power, and this really does sound like the huge, what, reaction that occurred in the tech sector, which was very vocal among both professional cryptographers and everything that we covered, the EFF and all of the letters signed by the Who's Who of technology and cryptography really did serve to put the brakes on this. And, if nothing else, just it's like, well, we can't deal with that now. So, I mean, it died just completely and totally. Which is really a good sign. It's like not alive in any form any longer. So, yay.

Leo: The greatest virtue, really, in our system of government is that it moves incredibly slowly.

Steve: Mm-hmm. Yes.

Leo: It's just really impossible to get anything done.

Steve: I'm often glad now that I'm 61 because they won't be able to screw it up that badly...

Leo: You're safe for the next 40 years.

Steve: ...I figure in the time I have left. Yeah. And it does move slowly.

Leo: Yeah. So any big change like that, you know. I'm really curious. I bet you it is, I think you're right, that it's probably the lobbying of the tech industry that had the most impact on this.

Steve: The feeling I got from reading Reuters' coverage was that there was enough - I don't want to use the word FUD because that's famously Fear, Uncertainty, and Doubt. But arguably, the politicians don't understand the technology. And so the technologists said this is impossible. What you want is impossible. And I would argue, and everyone has heard me argue, that's not true. But it turned out to be great that that's what got said because, like, the politicians who don't know one way or the other said, oh, maybe it is. I mean, the smart people are saying we can't do this, even though we don't know why, so maybe we shouldn't. Yeah, so that was good.

Also there was another good piece of news, and that is that - I think it was Thursday of last week, so a couple days after our coverage last week of the whole Oracle/Google API mess. After three days of deliberation, the two-week-long trial concluded with the jury unanimously deciding in Google's favor that Google's use of the API fell under the umbrella, sort of the get-out-of-jail-free card of copyright law, of fair use. And so of course Oracle immediately vowed to appeal.

Oracle's general counsel, Dorian Daley, in a written statement, said: "We strongly believe that Google developed Android by illegally copying core Java technology to rush into the mobile device market." Continuing, Dorian says: "Oracle brought this lawsuit to put a stop to Google's illegal behavior." Eh, well, no, no. They'd like to have $9 billion, please, because Google did what Oracle failed to do. Anyway, that's me editorializing. Dorian said: "We believe there are numerous grounds for appeal, and we plan to bring this case back to the Federal Circuit on appeal." And observers suspect that they will.

Now, I was a little curious about, okay, what exactly is fair use? And so I found a nice little piece at Stanford Law that sort of explained it a little bit. Stanford Law wrote: "In its most general sense, a fair use is any copying of copyrighted material done for a limited and 'transformative' purpose, such as to comment upon, criticize, or parody a copyrighted work. Such uses can be done without permission from the copyright owner. In other words, fair use is a defense against a claim of copyright infringement. If your use qualifies as a fair use, then it would not be considered an illegal infringement."

So Stanford continues, saying: "So what is a 'transformative' use? If this definition seems ambiguous or vague, be aware that millions of dollars in legal fees have been spent

attempting to define what qualifies as a fair use. There are no hard-and-fast rules, only general rules and varied court decisions, because the judges and lawmakers who created the fair use exception did not want to limit its definition. Like free speech, they wanted it to have an expansive meaning that could be open to interpretation."

So Stanford said: "Most fair use analysis falls into two categories: commentary and criticism, or parody." And of course this would fall under, essentially, commentary. And so Stanford said: "If you are commenting upon or critiquing a copyrighted work - for instance, writing a book review - fair use principles allow you to reproduce some of the work to achieve your purposes," which is where Google is. "Some examples of commentary and criticism include quoting a few lines from a Bob Dylan song in a music review, summarizing and quoting from a medical article on prostate cancer in a news report, copying a few paragraphs from a news article for use by a teacher or student in a lesson, or copying a portion of a Sports Illustrated magazine article for use in a related court case."

So anyway, that's clearly the position that Google took was that we didn't copy all of Java. We copied a tiny portion, that is, these 37 APIs that Oracle is complaining about. And we reimplemented them. We wrote them ourselves, but it was necessary for compatibility to duplicate this portion of the Java standard. And Oracle, you know, Sun never had a problem with it. Oracle didn't have a problem with it until way downstream when it had been proven to have - when Google's use created a huge amount of value to their particular implementation.

And so I guess, in watching the way the industry has responded, there is a sense that this original judge - this is Judge Alsup, whom we've talked about. He was also the judge who gave the jury instructions prior to their going away for three days of deliberation. And some people have said it was a broad interpretation of fair use that this judge, who's been on - I wouldn't say he's been on Google's side, but we've talked about him. We talked about him in 2012 when this thing first - when the case first happened. He's the guy who taught himself Java in order to really come up to speed and understand this. His original ruling was that this API, that an API was not subject to copyright. And then Oracle got that overturned on appeal.

And so then this two-week-long recent trial was Google taking the position, okay, if API is subject to copyright, then our use is not an infringement of that copyright because it was fair under the fair use doctrine. And once again, Judge Alsup's court said, right, we agree. And so we can imagine that Oracle's going to say, well, we don't, and we're going to get this overturned on appeal again. So we're still, once again, the decision is in Google's favor, and it does seem like Oracle is not going to give this up.

Meanwhile, there was an interesting story that Bloomberg covered about Google's increasing pressure on their industry partners for keeping Android up to date. They had an interesting graphic in the story that I put into the show notes here, showing that, whereas 84% of devices running iOS9 are using the latest version, only 7.5% of devices running Android are running Marshmallow. So there's a dramatic difference in the support for the latest versions of the operating systems.

What Google is - and so Google is trying to change the way the current smartphone ecosystem works. And we've talked about this extensively, the fact that, whereas mainstream operating systems, notably Windows and Mac, have essentially moved updating into it being done by default on a schedule, interacting with the user, saying we're going to update at a certain time, and updates are pushed out monthly and so forth. The problem is that the smartphone change has come to an industry that wasn't using smartphones.

And of course all of this echoes sort of the IoT problem, too, that we'll be talking about later in this podcast and have been talking about in the last several podcasts. And that is that the smartphones are being sold by cellular carriers that have traditionally sold feature phones or dumb phones or flip phones or phones that didn't carry with them the burden of being kept updated. And so on the one hand they want the revenue from selling a high-value, expensive, high-demand product. But they don't want the responsibility that inherently comes with selling a computer, an Internet-connected computer.

And all of our experience tells us that Internet-connected computers that are inherently incredibly complex, using layers of software that other people wrote, pulling it all together, grabbing pieces from every direction and cramming it into a ROM and then saying "Here you go," there's going to be bugs, vulnerabilities, problems. These things have to be updateable. And so the problem has been that there isn't that same drive to update. And so Google is, like, struggling to come up with a way to change the way this industry is behaving.

So one of the things they're doing is using much more forceful tactics. Google has drawn up a list that now ranks the top phone makers by how up-to-date their handsets are, that is, out in the field, based on security patches and operating system versions. Google has shared this list, or did share this list, with their Android partners earlier this year and has discussed making it public to highlight those manufacturers who are proactive and publicly shame vendors that are tardy in keeping their handsets up-to-date in order to put pressure on these companies.

And this is behavior, I mean, this is sort of a strategy that we've watched Google implement in various other areas of the industry. Google understands that people don't make change unless they're forced to. And so there is tension that Google is creating with their downstream users of Android in order to put pressure on them to come up with approaches to keep their devices more up to speed.

In the story that Bloomberg put out, and I didn't put it here in the show notes because there was just too much to say, but they did mention that Verizon claimed it cost them hundreds of thousands of dollars to perform their own testing of all of this material that they were getting from Google, that Google was pushing them to push out to their customers. And Verizon was feeling that they had a responsibility not to break their own customers' phones or there would be hell to pay. And so this constant flow of patches isn't something they wanted.

Verizon doesn't want that because it creates a huge burden for them to deal with it. And so they've just been saying no, you know, we want to leave things alone. And they're trying to shorten that process. Sprint has cut their time down from several months by a few weeks, trying to be responsive. But the problem is there isn't the mechanism in place. These companies are not used to being computer operating system vendors. And Microsoft has years of experience doing this. Apple has years of experience doing it. The truth is these are very sophisticated operating systems that are being treated like appliances. And of course this tune is familiar to our listeners because this is the whole IoT problem.

Now, one of the things that Google is doing is changing their behavior a little bit. Because they're seeing that they're unable to get the newest version of their operating systems out into the field, like only 7.5% of all Android devices are running Marshmallow today, so what they're doing is they're reducing their own reliance on this update process which isn't functional. New features, such as, for example, the Allo messaging service, are not being packaged as part of a new OS. Rather they're being packaged on purpose as

standalone apps so that Google can offer these to users of Android that their vendors haven't patched, haven't updated, that aren't current, so that they are able to provide them without the carrier being in the way.

And, for example, Instant Apps, which was unveiled at Google I/O recently, deliberately works on phones running versions as old as Jellybean, which came out in 2012. So what that means is 95% of the current Android user base will be able to get access to, for example, the Instant App technology. So Google's recognizing they're just not getting their partners to update, so they're deliberately making their newer things available, not as part of a later generation OS, because then no one's ever going to have them. And this way people are able, 95% of the current Android user base is from Jellybean or more recent.

So anyway, I thought that was some interesting look at the problem, that another aspect of - essentially we can think of this as a flavor of IoT, where these things are smartphone appliances. Users just want them. They just want them to work. And they want them to be secure and to be kept up to date. But unfortunately they're coming through an ecosystem where that hasn't been the case. And they don't want it to be the case. And so, as we've said on this podcast before, that has to change. These are computers that people have in their pockets.

I got a bunch of hubbub raised when the story was covered that Symantec had given Blue Coat Systems an intermediate certificate authority cert. Now, I immediately recognized Blue Coat because I had referred to them years ago when I did the certificate fingerprinting work at GRC. GRC's certificate fingerprinting talks about the idea of checking whether GRC sees the same certificate fingerprint as a user does because a certificate fingerprint cannot be spoofed. And if there's a man in the middle, if there's any kind of connection interception - which is a thing we're talking about all the time now on the podcast. For example, employers may deliberately, in order to deep scan HTTPS connections, they need to do that.

Now, the sanctioned way of doing that is for the appliance on the perimeter to install its certificate on all of the machines within that employer's network - essentially, that it becomes a certificate authority. And so, just as we do now, where we have CA certs in our machines telling us which CAs to trust, if you are working for a company that has one of these appliances, all of the machines within that Intranet would have that CA cert added to them in order to allow that appliance to essentially create certificates on the fly.

Okay. Well, what Symantec did was to give this Blue Coat Systems their own intermediate certificate. And that's bad. So what this means is, because everybody is trusting Symantec certs - remember, they purchased VeriSign, the certificate business, the CA business from VeriSign. So all systems the world over have the VeriSign cert in their root store. So what a third-party having an intermediate certificate from Symantec means is it's very different than this appliance operating on a controlled perimeter, like a corporation. What this means is that this appliance does not need to put its certificate in all of the systems downstream from it because, since its cert is an intermediate certificate authority, it's able to create certificates which will automatically be trusted because Symantec has given it permission to do so.

Now, when I went looking for some coverage of this, I thought, okay. I want a snarky take. So of course I went to TheRegister.co.uk, which is where you go for snark. And they wrote: "Blue Coat sells network equipment that does just this kind of espionage." I jumped down into their coverage. "The gear intercepts connections to websites and strips the encryption away so secured communications can be monitored. This is useful for corporations that want to keep tabs on their staff at work.

"Unfortunately, Blue Coat's HTTPS-snooping products have been used by repressive regimes to spy on activists online and quash dissent. To tear away the encryption and peek inside people's packets, Blue Coat's man-in-the-middle gear masquerades as legit websites. And this is so much easier to pull off when the manufacturer is an intermediate certificate authority because it will have the flexibility to generate trusted certificates as required. It paves the way for seamless surveillance by Blue Coat-built equipment." All absolutely true.

The Register wrote: "We asked Blue Coat how it planned to use its new powers, and we were assured that its intermediate certificate was only used for internal testing and that the certificate is no longer in use. The two firms [meaning Symantec and Blue Coat] said in a statement: 'Symantec has reviewed the intermediate CA issued to Blue Coat and determined it was used appropriately.'" Okay, whatever that means. And then they also said: "Consistent with their protocols, Symantec maintained," wrote Symantec, "full control of the private key, and Blue Coat never had access to it." Okay, well, that's not how this works, so that can't be true.

And then the statement continues: "Blue Coat has confirmed it was used for internal testing and has since been discontinued. Therefore, rumors of misuse are unfounded." It's like, well, okay, except that the only way this works is for Blue Coat to have a private key which is trusted by all the systems downstream of it by virtue of its certificate being signed by Symantec. I mean, that's what this was. So for anyone to say that Symantec maintained full control of the private key, well, they may have maintained full control of theirs. But the way an intermediate CA works is that it has a private key which it uses to sign the certificates it mints on the fly.

So anyway, what's important to take away is for Blue Coat to do what it wants to - that is, it is a provider of appliances that do this. But we should remember that all they need in order to do this transparency is for any certificate authority out of the many hundreds, I think it's 1,100, that we implicitly trust, any one of those can do the same thing that Symantec was caught doing. And so the way they got caught was that someone looked at the certificate that Blue Coat was using, and apparently this was not being used for testing. This was somehow publicly found.

But we should remember that any government that wants to do this just needs to have one of the certificate authorities under that government's control. I mean, China has them. Any government certainly has certificate authorities that they can instruct to produce an intermediate certificate and sign it, please. And then that intermediate certificate gets installed on the egress points to their network, or ingress points to their network, or to their country, and they're able to decrypt all the traffic. And no red flags are raised by anyone's browser because there's a chain of trust back from that intermediate certificate, back to a trusted CA, a root certificate authority that is already in the trust store of everyone's computers. So this is the mixed blessing of the whole public key infrastructure is it relies on trust. And unfortunately, over time, it's gotten very top-heavy, or very trust-heavy.

Okay. So there was an interesting study conducted by Duo Labs, and my title for this was "Bloatware Insecurity Continues to Haunt Consumer and Business Laptops." And Leo, I've heard you talking about the whole bloatware problem, the problem of just all laptops containing some add-on software...

Leo: All Windows laptops.

Steve: Right.

Leo: That's not all laptops, thank goodness.

Steve: Yes. Thank you. I'm glad you clarified.

Leo: Really it's Windows is the - yeah, right.

Steve: Right. Well, and in fact I haven't mentioned that, when it became clear that Microsoft was saying that they were going to abandon support for older hardware, I did update myself to a state-of-the-art, it's a fourth-gen Carbon X1 Lenovo. And the first thing I did was strip it and attempt to reinstall Windows 7. Windows 7 would not install on that laptop...

Leo: Oh, really.

Steve: ...because it already had Skylake, and Lenovo wasn't providing those drivers separately. So, and I think I heard you mention that you were having problems installing Linux on your brand new HP.

Leo: Yeah. I get weird errors. Maybe that's it, yeah.

Steve: That's the reason is that - now, what'll happen is, in time, Linux will catch up with drivers for that hardware. But the hardware is newer than our operating systems. And anyway, so what I was forced to do was to roll back to the original Windows 7 that came on this Carbon X1 fourth-gen laptop. And then I went in and surgically removed all of the nonsense, all of that Lenovo crap that is added to it, which is all of their own, we're going to keep all of this stuff updated.

And that's the point of this story is that this Duo Labs took a look at Acer, Asus, two different Dell laptops, HP, and two different Lenovos, and looked at the security of it. And I have another graphic here in the show notes. For example, many of them do not transmit the manifest, that is, they have an XML manifest which their update managers use in order to query what's changed. A couple of them do use TLS, but many of them don't. So that's going back and forth in the clear with no security over it.

The researchers published a report on 10 new laptops, all running either Windows 8.1 or Windows 10, including some Microsoft Signature Edition machines that are supposed to be bloatware free, but still include some of these components. All of the updaters specify their own update manifests, where the system grabs a remote XML file over HTTP. Only the Dell systems update over HTTPS.

So of course that's a concern because anyone who is able to intercept that XML file could change it on the fly. None of the manifests are signed, and they don't use proper engineer practices to make sure the integrity of the manifest is validated properly. All of the manifests include commands to ensure that the updates run properly. Meaning that a bad guy could simply hijack those commands and use that to execute whatever they wish

with system-level permissions. And all of these updaters run with system-level permissions, meaning that they're able to bypass any other security protections on the machine.

So what we're seeing is, again, and this is sort of becoming a theme as we look closely at what happens when third parties are involved in what need to be secure solutions, is that their focus is on making the stuff work, not making it secure. So what Duo Labs found was that every vendor shipped with a preinstalled updater that had at least one vulnerability resulting in arbitrary remote code execution as system, allowing for a complete compromise of the affected machine. Vendors often failed to make even basic use of TLS, properly validate update integrity, or verify the authenticity of the update manifest contents, meaning that they weren't bothering to sign them and validate signatures. So they could be changed at will, and the updating software wouldn't know any different.

Vendors sometimes had multiple software updaters for different purposes and different implementations, some more secure than others. So in one case there was one that was secure, but it was running alongside one that wasn't. And what this does is create a large attack surface which is being created by these ancillary OEM software components, which the updaters are introducing to the system. So unfortunately we have all of this focus on the OS security, where Windows is doing everything it can to keep itself hardened and secure. Yet this unwanted bloatware, which all of these vendors are adding, is making the systems insecure.

So essentially what this says is, when you get one of these machines, you want to - and this is what I've heard you talking about on Windows Weekly with Paul and Mary Jo is just immediately, if you can, reinstall Windows. If you can't, then go in and just get rid of this stuff, which you didn't ask for, and which is doing nothing but creating an exposed attack surface. It looks like, without exception, this stuff is a serious security problem.

Leo: Although I noticed the Lenovo Solution Center gets all green checks. Is the update agent running, though, on all Lenovos? Because, if it is, then it doesn't matter because it gets all red X's.

Steve: Correct. And that's a perfect example. One thing is all green.

Leo: They do one thing right.

Steve: The update engine is running alongside, and it's all bad.

Leo: Oh, lord. But if I wiped it and put vanilla Windows 10 on this HP, Windows is doing it right, is what you're saying.

Steve: Correct.

Leo: Microsoft's doing it right, it's just these third-party solutions.

Steve: Correct.

Leo: Including updaters. And everybody puts updaters on. Everybody does.

Steve: I know. And so what this was a study of was 10 of these systems, are any of them secure? The answer is no, not a single one was done correctly.

Leo: Wow.

Steve: Because, again, security is hard, and their goals are just to make it work. They're under the gun. Their programmers are being told, you know, we were supposed to have this last month. Finish it today.

Leo: Well, I guess I'll be wiping my HP and putting Windows 10 on it until I can - because Windows 10 will work with the Skylake. It's just Windows 7 that wouldn't.

Steve: Correct. Correct.

Leo: And you can probably slipstream something in there that might make it work.

Steve: Yeah. In this case it was - I only purchased it like two months ago, along with - it was when Microsoft said, okay, we're going to stop supporting this. Because all of my laptops are much older.

Leo: That won't happen, by the way, for a couple of years. So that's not the issue. It's just maybe there's some pieces you need in your Windows 7 or something.

Steve: Correct. It was doubtless the mass storage hardware.

Leo: Yes, right.

Steve: Because the original Windows 7, even Service Pack 1, it's way years ago. And so if you can't get that to go, then you're unable to bring it current. And Lenovo is not officially supporting Windows 7 on the modern hardware. And I had to go in and say, no, I don't want Windows 10, I want Windows 7. So I had that downgrade permission, but they're still not officially supporting it with, like, exposed driver packages and everything.

Leo: Yeah. Well, maybe it'd be worth trying to get Linux on here instead of Windows 10, but I'll figure it out.

Steve: So I just did a refresh of the Never10 page. I just wanted to give everybody an

update: 783,503 downloads.

Leo: That's awesome.

Steve: 783,000. It has wildly broken every record in GRC's long history of very popular freeware. We are seeing more than 35,000 downloads per day.

Leo: I can't believe somebody from Microsoft hasn't come to your house.

Steve: A day.

Leo: Mr. Gibson, it has come to our attention that you have blocked nearly a million people from installing...

Steve: Well, this time next week, for next week's podcast, we will be north of a million downloads of Never10.

Leo: Man. Shhh. Steve who? Don't know him. Never heard of him. Unh-unh.

Steve: Yeah. And that's just GRC. All the other download sites now offer it, too. And so they're downloading it like crazy, too.

Leo: Get it from the original source. Let Steve bear the brunt of this. Don't get it from someone that's going to put malware...

Steve: Well, it's nice to have a count.

Leo: Yeah.

Steve: And, frankly, there was one guy who accused Never10 of completely screwing up his system. And, I mean, boy, he was mad. Oh, lord. I mean, he was dropping the F bomb all over the place and screaming at us. And so I answered his email. I wrote back to him, and I said, "Sir, I'm sorry that somehow your machine has been messed up, but really it couldn't have been by Never10. Never10 sets two registry entries..."

Leo: Doesn't do anything. It's simple.

Steve: "...that Microsoft documents and sanctions." And by the way, at that point we were north of half a million. And I said, "Half a million people have used this, and it hasn't hurt anybody else's machine."

Leo: Right, right.

Steve: And I also said, you know - and he wrote like he didn't know who we were. And I said, if you got this from somewhere else, one of the dangers of getting it from somewhere else is that bad guys will use super-popular software as a means of getting malware into your machine.

Leo: Guy Smiley in our chatroom's saying the Major Geeks download site is wrapping it with other stuff. So you really don't...

Steve: A perfect...

Leo: Yeah, perfect example. And they have 16,769 downloads there. So those people are getting - get it from the original source always, always, always. Don't get it from these download sites that just really are kind of piggybacking on success.

Steve: Right, and try to get you to download their manager, you know, their...

Leo: Shameful, yeah.

Steve: Yeah, exactly, and then take over your browser and get the Ask toolbar and other junk. Yeah. Okay. So...

Leo: By the way, just get Linux, Steve. Would you please do me a favor, stop trying to make any of this stuff work and just put Linux on everything. I wish you would.

Steve: Actually, I installed Linux on...

Leo: Good. See, easy; right?

Steve: Yeah, it was. I had to do it because there were some problems with SQRL under Wine.

Leo: So you had to try it, yeah.

Steve: And so, yeah, and so it wasn't fair for me to keep asking...

Leo: No one has written an implementation for Linux? I'm surprised.

Steve: Not natively. No, I mean, it's got to get out there first.

Leo: All right. You feel calm? You feel collected? Your blood pressure's gone down?

Steve: Oh, I just don't have the...

Leo: No, seriously, a lot of this stuff we talk about, it's like just the steam starts coming out of my ears. It's like, how could they do this? It was because of this whole Windows 10 thing that I finally said, you know what, I'm going to put Linux on one of these machines. I've used it for years, but could I use it day in, day out? And now I don't want to use anything else, to be honest with you. Debian is my preferred distro. And you probably heard, I've ordered a kind of high-end Linux laptop as my in-studio system.

Steve: I know, you're panting over it.

Leo: Very excited. It's a 17" laptop. I mean, when I'm at the roundtable for TWiT and MacBreak Weekly and TWiG, I can't use a desktop. I use a desktop in here in my office. But I can't use a desktop. But I want a laptop that's, you know, it's always plugged in. It doesn't need to be portable. So this thing, I'm sure it weighs eight pounds, but it's an i7 and 32GB of RAM, and it's got an NVIDIA 980M GPU. And so I'm - tomorrow I get it. I'll let you know. But it's just, there's something nice about running an open source operating system with no spyware. One of the reasons I don't run Ubuntu - and this laptop from System76 does come with Ubuntu, so I'm going to have to weigh this - for a while Ubuntu was routing your search. Look at that. Nice. Nice. You're using Ubuntu.

Steve: Yup.

Leo: Yeah, you know, I think they're all right. Now, I don't like it that you see there's an Amazon icon there in the dock. That kind of bugged me. And for a while they actually were sending search queries to Amazon. So I kind of got soured on Ubuntu. And then there was the Ubuntu One fiasco. I think they've cleaned up their act. I think they realized people don't like that kind of stuff, not in an open source operating system. But I decided to go back to the distro that Ubuntu is based on, which is Debian. Which, by the way, nowadays is just as easy and straightforward to install on existing hardware as Ubuntu is.

Steve: Yeah, well, and in fact that was one of the - that's an older T500 ThinkPad that I had just...

Leo: Lenovos are great for Linux. Great for Linux.

Steve: Yeah, that I had lying around. And it just, you know, it just went right in. And, I mean, it was flawless. It recognized all the hardware. Networking came up, WiFi,

Bluetooth. I can close the lid, and it puts itself in standby, so all power management is working perfectly. No, it was just - it was a great experience.

Leo: Yeah, it really is amazing, yeah. I mean, we've come so far since the first time I tried to install Linux, which was a Slackware distro in '94 or '95. And it was a nightmare. Not to mention that it was on multiple CDs or maybe even floppies. It was a little bit...

Steve: Yeah, I had to do a few things. There's some controversy on the Ubuntu desktop because the tray was becoming overloaded with icons.

Leo: I don't like it, yeah.

Steve: And so they, like...

Leo: You can change that, obviously. Everything can be changed.

Steve: Yes. And in fact, so, for example, SQRL uses the tray to show its little icon in the tray, and you're able to access settings and do things and change things. So I had to go in and fix that. Of course I had to immediately swap Caps Lock and Ctrl, as we talked about earlier, in order for that to be workable.

Leo: Easily done, though. There's a tweak tool [crosstalk].

Steve: Yeah. Yeah, exactly.

Leo: Yup.

Steve: Yeah. So I was - anyway, so SQRL.

Leo: Yes.

Steve: A bunch of people have asked where we are. With caveats, it's done. So let me give everybody sort of a sense.

Leo: Those are two words I didn't expect to hear out of your mouth. Wow. That's great.

Steve: Well, and - yeah. So one of the things that SQRL has that, for example, that FIDO, arguably some competition for it maybe someday in the future, is the whole SQRL system provides complete identity management, whereas FIDO says that's somebody

else's problem. It's like it's out of spec. The individual implementers will deal with managing identities. For me, that was as much a part of making this thing user-friendly as anything else. And the reason Never10 is going, like, just burning up the wires, is that it's just so simple. I mean, somebody, was it How To Geek, somebody actually referred to me as an "unknown developer." And I thought, well, okay. They may not know me. But still they were recommending Never10 because it's just tiny, and it just does the job, and it works.

So one of the things that SQRL does is, in this identity management, is that there is a - we had to deal with the possibility that someone's private key might get out of their control. The whole concept with SQRL, as our listeners know, and we'll of course give it, again, coverage once this thing is downloadable for everyone to play with, is that you create one master key, and then all of the websites you visit receive a derivative of that based on their domain name. And the derivative is the way they identify you every time you come back. And by signing a challenge that they provide, you prove that you are the owner of the key that they know you by, essentially.

So in a perfect world there is, like, you're giving the website essentially a public key. But the way to think of it is you're giving them no secret to keep. And that's one of the keys. In all traditional login, you're giving them a secret. You're giving them your username and password, and you're needing them not to lose it, not to let it out of their control. Well, we know how well that works. We were talking about 117 million passwords that were published a week ago from an old LinkedIn breach just last week. But the point is that the way the whole - the coolest part of SQRL is that what you give the website is a public key which is unique for that website, derived from your super-secret master key that never leaves your control. So the website knows you and identifies you by that public key.

And so the simplicity of SQRL is that the website sends you a nonce, a random blob. We call it a "nut," of course, instead of a nonce, because it's SQRL. The website sends you a nut. You sign that with the private key which corresponds to that website's domain, which again is derived on the fly from your super-secret master key. So there's only one thing, one secret that SQRL has. The entire system then works off of that. And so every different website gets a different public key. And that's the point is that it's a public key. They could lose it. They could publish it if they wanted to. Doesn't hurt anybody because it's not of any use to anybody else because in order to use it you have to have the matching private key that only the actual originator of that public key would have. So from that standpoint it's very simple.

But the question then is, what happens if that one super-secret private key is compromised? And we just can't say, oh, well, that's not our responsibility. That's out of spec or something. No. We had to deal with it. So in the SQRL system, if you believed that something happened, if you were just - if you were worried, you had some concrete reason to believe that the private key was compromised, that your identity's one super-secret master key was compromised, then we need a means of replacing it that is not burdensome. So SQRL incorporates the concept of a previous key. And your SQRL identity, that is, it's just a file which has a well-defined structure as part of the spec, it can carry both your current and your previous super-secret secret. So essentially you rekey. A user, like if the government got your phone or something led you to believe you could no longer trust in the integrity of your SQRL identity, whatever the reason, you are free at any time to rekey your identity.

What that then means is you carry both. And when you go to a website and perform the SQRL authentication - again, all of this is hidden from the user. All they see is the QR code, and they snap it with their smartphone, or they click it with the mouse. What

happens is the site tells you whether it recognizes you by your current key or, if you have one, your previous key. And if the site recognizes you by your previous key, it behind the scenes automatically updates to your current key. So it forms in crypto terms sort of a ratchet, where it's able to take the secret, the obsoleting secret, the secret that is in the process of being obsoleted, and use that to jump to the current one.

And so the idea would be that essentially that makes this whole process transparent for all SQRL users. They rekey their identity for any reason; and then, as they log onto websites that still know them, that they haven't logged into since, that thus still know them by their old key, it's transparently updated to the new key. So that's the way the system had been from its beginning.

The argument was, okay, well, what if, like, something was really wrong, and you needed to rekey again, that is, have a second rekeying, and SQRL only kept the previous key, and you hadn't visited all the websites that know you by your first obsoleted key before you had obsoleted the key again. And so it was like, well, okay, how far do we take this? And so the point is what I spent, maybe it was - I'm not sure how long ago it was and how long it was. But basically we came to a compromise. And that was that, first of all, remember that maybe many users in their entire use of SQRL will never rekey even once. That is, it'll just work.

So, but again, this has to survive worst-casing. So it's like, okay, we really want this to work, both the potential to become an industry standard, so there can't be any gotchas. So we settled on four, just because we don't want the identities to become too big. They become burdensome. And remember that the only time you would need more is if you were, like, rekeying like crazy such that any site you visited didn't know you by, for example, any of your four previous keyings. Which seems unlikely, given that no one may in fact ever need to rekey even once.

But a good use case, where you would probably want to rekey, is, for example, Stina at Yubico has expressed a strong interest in immediately supporting SQRL, the moment we officially say it's done. And so if you were using it without hardware support, then even I would rekey my identity for hardware. That is, because what Yubico would produce would be an HSM, a Hardware Security Module, where the signing process never left the hardware, so it was virtually then impossible for it ever to get out of your control or be lost. In which case, I could see that rekeying makes sense.

Anyway, so one of the things that had to happen was that the storage spec had to be updated. The interactive spec needed to be changed. And this change has had some repercussions through the spec. So that's all done. And the SQRL client and the demo website, in fact I think Jeff's iOS client is already up to speed. Actually, he beat me there. He had the four previous identities support before I did. So we're current there. And then, as I was going through the client, one of the things that I wanted to fix was the textual input. And I'm only bringing this up because it was like the most recent thing I just finished. And I had them, yeah, here.

So this page the SQRL client prints for you. Again, one of the key concepts of SQRL is that there is no one to ask when you forget your password. That's one of the keys here is any third-party system there is password recovery. You can say to the website, oh, I forgot my password. Send me an email link. Well, that's a mixed blessing, of course, because that means that there is a password, a secret that they're keeping, and there is a process for recovering the password that is subject to exploitation. With SQRL there is no third party. There's no one to go to if you forget your password.

So as part of creating a complete solution, we had to make it practical for there to be no

recourse. And so one of the things that SQRL does is it goes out of its way to help you come up with solutions for "I forgot my password." So one of them is your identity as a QR code. So this is a piece of paper you put away. When you create your identity, you print it out, and you put it away. Maybe you put it in a safety deposit box, or in a shoe box, or wherever your other private papers and things are kept, so that this is your get-out-of-jail-free card. If you only had your SQRL identity loaded in one smartphone, and the dog ate it, or you lost it, or something horrible happened to it, this is your SQRL identity.

Well, one form of that is printed out ASCII. And it is, I forgot how many characters. And actually it varies because here is an identity that has never been rekeyed, and here is one that has been rekeyed. So it's carrying its previous and current key with it. And you can see it's getting longer. Well, there's whitespace at the end of this page because you might have a third or a fourth key which, again, makes it increasingly long. So typically that would, as I said, someone might never rekey. They just may not need to. But we wanted to make it feasible.

So how do we enter something like this, that's long and potentially over their whole lifetime of use getting longer? What I just finished working on last week was a - in Windows we call it a "custom control." A control is something like a button or a dropdown list box or a text field or whatever. Well, I had to create one from scratch because I wanted exactly correct behavior. I wanted the user to, when we print that out, it's in multiple lines with five groups of four characters. So I wanted the grouping to be enforced so that, as the user entered it, it automatically grouped itself. Windows doesn't have that capability automatically. People who've entered those, like, Windows license keys, what Windows has done is have individual text fields, but it's uncomfortable. They sort of - you tab between them. You're unable to cursor between them. It just doesn't work very well. Mine works beautifully.

But the other thing I wanted was I wanted as-you-go confirmation of everything you've done so far is correct. So as you're entering the text into the line, when you enter the 20th character, as you're entering it, the line is red. When you enter the 20th character, it turns green, only if everything you've done so far is perfect. So the first 19 characters you enter are actually the key, part of the keying information. The 20th character is a check character based on a modulus 55 of the hash of all the preceding characters.

And I should mention that the identity is a base - I'm sorry, base56, not 55. It is 56 printable characters with the confusing ones removed. So there's no ones, there's no zeroes, because ones look too much like lowercase "l" and zero looks too much like uppercase "O," of which there aren't any. And so what I did was I took all of the alphabet and removed anything confusing, resulting in - if we had the whole alphabet and numbers, what would we have? We'd have 26 in upper and lower case, so 52, plus another 10, so that would be 62. And so instead we have 56. I've eliminated a bunch of the confusing ones.

And so identities are in this base56 alphabet. And there's a base56 check character. So as you go, these lines turn green. And because there are some people that have red/green colorblindness, I didn't want to depend upon that. So you are unable to proceed on the next line if any previous line is incorrect. So if it won't let you go further, and you can't see that it has not turned green, then your cue is, well, I must have a mistake in the previous line, and you're able then to go look at it again, just inspect it, move the cursor up, fix the character. It's then allow you to proceed. So anyway, all of that's done.

I also then just added, just as a nice little touch, the SQRL icon prints itself down at the

bottom of the page printout. But we got reports from - or I got reports, I say "we" because it's all being done in the SQRL newsgroup on GRC - that that was crashing under Wine when the SQRL client was run on Linux. And so what I just spent the weekend doing was installing Ubuntu 16.04 LTS on a laptop. And I found that that was due to an unimplemented function that I was using. So I coded around that. And at the moment...

Leo: A function unimplemented in Wine, like you were calling something that Wine didn't support.

Steve: Correct. And in fact I run across that a lot. For example, the other problem is that I'm wanting to use Courier New under Windows because Courier is a monospace font, and the serifs make the uppercase "K" and lowercase "k" very different looking. If I removed all possible confusion between characters, we'd end up with six. And so that would make the identity too long because we wouldn't have enough bits per character. So I had to make a tradeoff.

The problem is, one of the other things broken in Wine is something that is called the "font mapper" in Windows, where you're able to specify sort of the generic characteristics you want for a font, and then Windows says, oh, you want - I have one that matches what you've asked for here. And in fact it always gives me Courier. And under Wine it never gives me a monospace font. It just doesn't. That's completely broken in Wine.

So I'm in the process with the newsgroup of identifying universally available fonts. Or, what it looks like I may do is take the FreeMono font, which is one of the free open type fonts, and remove all the glyphs from it that I'm not using. I tried that late last night, and that allowed me, after compression, to only increase the EXE size by 15K. So that would be me bringing my own subset of Courier New, which is in the free version as FreeMono, bringing that in and carrying it in the client, so I'm absolutely guaranteed of what the user is going to see. And I'm a little worried about foreign language fonts in other countries because I want to make this universal.

So that's a sense for this process. I haven't been talking about it because, I mean, while I've been working on it, it's just - it's slow going. I want to get it done. It's got my full time and attention. But again, I'm wanting it to be perfect. And we're very close. When I said "with some caveats it's done." So there's still some Wine problems. Wine's fonts seem to be longer horizontally than the Windows API knows. You're able to ask Windows how long is this font, that is, here's a string in this font. What number of pixels long is it? And I use that in order to trim the size of the controls. Well, Wine trims them wrong, and so the ends of the strings are being chopped off. So I want to give Wine some attention.

So bottom line is it's done. There is some language I need to change in the UI because I was originally talking about creating a new identity rather than rekeying an existing identity. And so the UI doesn't yet have the notion of rekeying just in the language. So I need to do that. But it's been working for quite a while. We, like, kill it when some major upheaval occurs like going from one previous identity to four previous identities. That took it down for a while and had repercussions throughout the code, which I then had to fix. But that's been - it's been up again. And I think we're done.

So there are a number of things like, for example, when you're running the client, and you click on the QR code in order to log in. At the moment, the first time you do it, that launches SQRL. It gets focus, and you can just type right into the - enter your password to prove that it's you in front of the computer because SQRL does still need somehow to authenticate you to it. And then it handles all of the per-website authentication for you.

But it's still necessary to prove you're you, and not somebody who just walked up to your computer.

The problem is, if SQRL is already running and you do this, it launches a new instance which looks to see that it's already running and hands off control to the existing instance. The problem is that Windows deliberately prevents other windows from obtaining focus because that can be abused for annoying pop-ups and things, where something pops up and takes focus away from what you are doing. So at the moment that's something I need to fix.

But we're at that level. We're to the point now where, for the next I don't know how long, I'm tempted to say a couple weeks, I will be in cleaning up all of those things. I have a long list of things that I've made notes of, many of which have already been fixed, but I haven't gone back and removed them from my to-do list. So we're very close. It's done. And so I hope soon to be able to say here is, I mean, it'll appear on the website. People can download it, create SQRL identities, play with the demo site that we've had, I mean, this is the thing that a long time ago, Leo, you brought it up on your laptop and showed it on the screen, and I used Jeff's iOS client to log myself into your laptop over Skype.

Leo: Oh, man. I remember that.

Steve: Yeah. So, I mean, the technology, that fundamental...

Leo: That was cool, yeah.

Steve: ...technology has been running for a long time. But I want this thing to be polished because we know I have a tendency, once I finish things, to be only dragged back to them kicking and screaming. Which brings me to SpinRite 6.1, which of course I can't wait to get back to because it needs my attention next.

Leo: Good.

Steve: So that's where we are with SQRL. As far as I know, it's finished. In every meaningful way it could be used today. I need to finish up aspects of the UI, things like there's one dialogue where I keep coming to it, and it finishes its work, and it waits for me to hit Next rather than automatically jumping to the next screen. And, okay, well, it ought to do that. So it's that level. It's just the debris that needs to get cleaned up. So it's been quite a journey.

Leo: It would have been easier to write it from scratch, I think, at this point, but okay. That's okay. That's okay.

Steve: Yup.

Leo: Awesome.

Steve: Okay. So I wanted to share three tweets with people. One was from Pat, who tweeted me from @thetweetguy99. He said: "Steve, I recently listened to your discussion of Chrome vs. Firefox in SN Episode 557, and I was struck by your comment that 'Most users would be better served by Chrome.' I'm a sysadmin for a small company, about 50 employees, and I check all of our end-user machine logs each week." Boy, this guy's a responsible sysadmin. "In reviewing those logs, two things stand out each and every week: First, Internet Explorer is by far the application that crashes the most; and, two, virtually all of the malware events we experience are drive-by infections by legitimate websites hosting malicious ads.

"So I thought, it's time to deploy Chrome to everyone. They've [meaning Google] made it fairly easy for enterprises to deploy - msi installers, group policy templates, et cetera. And so I've begun rolling out test installations of Chrome with our favorite adblocker preloaded, uBlock Origin. We're already seeing positive results, even had one user go back to a website that tried to download malicious code onto his machine via Internet Explorer just 30 minutes prior, and was pleased to see that no installation was even attempted when using Chrome with uBlock Origin."

So anyway, I just wanted to share that, a bit of feedback from the field, somebody saying IE was crashing more than anything else, and all malware events that he was logging were from websites hosting malicious ads. And by switching to Chrome with uBlock Origin, problem solved.

Rob Woodruff - you had a comment, Leo?

Leo: No, go ahead.

Steve: Oh. Hi.

Leo: When you're done. I just was - I'm still working on getting Linux installed on this HP.

Steve: Ah, perfect.

Leo: And here's an interesting thing I just noted. Of course the HP has TPM on it, the Trusted Platform Module. And one of the, you know, I know enough to turn on Legacy Boot and turn off Secure Boot because that always keeps you from installing an operating system other than Windows. But it turns out you also have to reset the TPM if you want to install somebody, you know, something...

Steve: Clean it out.

Leo: Yeah. The TPM must have some information about Windows in it, like don't install any of that other stuff.

Steve: Right. Interesting.

Leo: Yeah, I'm playing with it still. Haven't quite gotten there; but we're getting there, bit by bit.

Steve: Cool.

Leo: Yup.

Steve: So Rob Woodruff tweeted. He said: "Hey, Steve. I listened to the podcast from Tuesday, and Carl in Indiana with a Netgear cable modem probably needs to get a firmware update." Remember, this was where I was talking about - he was saying that he was - this was on the whole issue of constantly needing to reboot his cable modem. And I said, you know, if restarting the cable modem brings you back online, then it's got to be a problem with the cable modem. So anyway, Rob said he listened to that podcast, he says, and he "probably needs to get a firmware update."

He said: "I've seen a lot of Netgear cable modems with Comcast have that exact same problem, whereas the SMC cable modems that Comcast provided prior to the Netgear cable modems work just fine. Comcast worked with Netgear to develop a firmware update that was supposed to alleviate that symptom." He said: "I'm not sure who Carl's cable modem provider in Indiana is, but he might want to check with them to see about a firmware update. Either that, or ask for a different brand of cable modem." But I did hear, not only from Rob, but others, that changing cable modems solved their problems for them.

And finally, I really liked this. I can't pronounce this person's name. S-T-I-J-N Crevits, C-R-E-V-I-T-S, said something that I thought was so sharp. He said: "Steve, don't you think not allowing the filtering of Win updates through APIs is more secure? For one, malware would otherwise be able to block security updates." And that's like one of those head slappers. It's like, of course. Remember, and this of course is relative to Never10 because I have commented that Microsoft doesn't provide a clean means for preventing Windows updates. And I was complaining about it because wouldn't it be nice if Never10 could simply uninstall 3035583 and then prevent its reinstallation? But it's like, duh, that's not good because exactly as he mentions, if there was an API that allowed you to say, okay, don't install these updates, well, that's what malware would do in order to keep itself from being washed away by updates.

So thank you for that little bit of reality. I appreciated that. Obviously, once you hear it, it's like, yeah. That's one thing that you don't want software running in the machine to have control over is the machine's own updates. On the other hand, it does mean that we are pretty much a victim of whatever Microsoft wants to push on us through the Windows Update mechanism, over which we have no good control.

And, finally, a note from a SpinRite user whose name is just - it's Sitbit, S-I-T-B-I-T, in London, Ontario, Canada. The subject was YAST! And I'm not familiar, but maybe that's a SuSE package manager because he said, in parens, (NOT a SuSE package manager), but rather that's his acronym for Yet Another SpinRite Testimonial. And he said: "Hello, Steve and Leo. Love you guys. The show is fantastic, yada yada, et cetera. Just a short and sweet SpinRite story. I had an old 320GB spinning disk salvaged from a laptop that I used to store some of my data. It wasn't terribly important, but I didn't want to lose it, so I had it mirrored to a backup drive. Well, the drive failed, so the directory I had it hanging off of showed up as empty. So my mirroring software dutifully deleted my

backup, thinking the files had been purposely removed.

"Long story short, I caused Steve to hear a yabba dabba doo," which of course is the sound that my system makes when someone purchases a copy of SpinRite. I always am appreciative because that, as I have said, pays the bills around here. And he says: "And two hours later my drive was back in action. Thank you for this wonderful product. Thanks again, Sitbit." And thank you for the great testimonial. I'm glad for a yabba dabba doo and that we could recover the data lost from your drive and bring it back online.

Meanwhile, Leo, how is your Linux install going?

Leo: Eh, yeah, well, I don't know. Something weird's happening. I'll tell you what's happening.

Steve: Yeah, I think it's just - I think the hardware is too new.

Leo: Might be. So I use to install Linux, as most do nowadays, a USB key. And I'm using one. I have a Debian here, an Ubuntu, and a Fedora. And I've used the Debian and Ubuntu keys many times to install, so I know they're good.

Steve: On older laptops.

Leo: On older laptops. And the install starts up. The installer starts up everything. But then at some point it says, okay, where's your CD-ROM drive, which I've never seen before.

Steve: That's exactly - that's it. Because what's happened is it's trying to switch from the BIOS that it was using, it was using the BIOS for accessing the drives.

Leo: Right, right.

Steve: It tried to switch to its own driver. And when it switched to its own driver...

Leo: Yeah, that's what happened.

Steve: ...it could no longer see the install media.

Leo: Yeah. Wow. Wonder how I change that behavior.

Steve: What you might try, you might try switching - oh, you don't have a built-in CD.

Leo: No. I'm trying - the Fedora is a 'Net install. So what I'm hoping is that it maybe will continue the install over the Ethernet.

Steve: Correct.

Leo: Who knows what'll happen.

Steve: Worth trying. Is the PXE installed?

Leo: Well, we'll see. I would think. I don't know. I don't know because, you know what, they don't want you to put anything but Windows on this. That's really the truth.

Steve: No. And that's my - I burned then...

Leo: You put it on a CD.

Steve: Yeah, I burned a DVD. It occupied about half of it, or it was like 1.7GB, I think, on a DVD. And again, it's older hardware. And so it knew how to talk to the device, the actual physical hardware devices on the laptop.

Leo: Haven't purchased a machine with a CD-ROM drive in years, literally, like three or four years. I guess I can - I have a USB drive around somewhere. Probably won't work, either. Anyway...

Steve: No, I think if your USB dongle didn't work, I think a USB CD wouldn't.

Leo: Right, that wouldn't work either, would it.

Steve: But it's worth trying a 'Net install, an over-the-'Net install.

Leo: Yeah, yeah, hoping that'll work. All right. Sorry, didn't mean to interrupt.

Steve: No, I wanted a little bit of a breather.

Leo: Yeah. This is the issue, though, and it's one of the reasons I ended up buying a Linux-first laptop is because these manufacturers, they're working so closely with Microsoft, and they see it as a security issue. Oh, we want to make sure you're installing legitimate Windows 10 on here and nothing else.

Steve: I know.

Leo: But that's, you know, it's a limitation of the hardware.

Steve: It's a foreclosure, frankly, of the industry, unfortunately.

Leo: Yeah. Dell is good. Dell continues, and Lenovo, I think, continues to support other operating systems.

Steve: Well, and I just think, again, I think it's that laptop is brand new.

Leo: Right.

Steve: And similarly...

Leo: Somebody will solve this, in other words, in the Linux community, yeah.

Steve: Yes, yes. It'll be, you know, I'm sure somebody's already - one of the kernel gurus is working on updating the driver to incorporate support for whatever that latest chipset is that is in the HP. Because I had exactly the same problem when I tried to do a clean install of this brand new fourth-generation Carbon is it just said, well, it just lost the access to the mass storage when it tried to switch over to it.

Leo: I may be forced to use Windows 10, and isn't that what Microsoft wants.

Steve: Yeah.

Leo: After all. All right.

Steve: Okay. So IoT in its infancy. I actually came up with an acronym, I-D-I-O-T, which is...

Leo: IDOT?

Steve: I Don't Internet Of Things, and of course that's IDIOT.

Leo: I-D-I-O-T, yes,

Steve: Uh-huh. So one little anecdote I loved was something that I picked up, but I've

been saving it for when we could talk about this. And we've often talked about the CVE, that's the Common Vulnerabilities and Exposures. What CVE is, is an industry-wide list of information, like system, security vulnerabilities and exposures whose goal is to provide common names for publicly known cybersecurity issues. The goal of CVE is to make it easier to share data across separate vulnerability capabilities - tools, repositories, and services. That is, you know, to give us a common numbering for, like, all vulnerabilities in the industry. So what I got a kick out of was - and that's at CVE. Mitre.org. They posted an update, and this is maybe a month or two ago, but I grabbed it because I was just - I'm just, like, oh, my god.

It said: "Update. The recent explosion of Internet-enabled devices known as the Internet of Things, as well as the propagation of software-based functionality in systems, has led to a huge increase in the number of CVE requests we have been receiving on a daily basis. We did not anticipate this rate of growth and, as a result, were not as prepared for the latest surge in requests over the past 12 months as we had hoped. The result has been some of the delay in CVE assignments that the software security community has recently witnessed.

"We recognize the inconvenience that has resulted and are working hard to come up with a solution. Last week we proposed a possible option to our CVE Editorial Board, but some members raised concerns about the approach, and we have withdrawn it from consideration. We're working diligently to come up with a solution that will meet the needs of all the various use cases of CVE." In other words, the Internet of Things has overloaded the long-running existed CVE management system. They cannot provide indexes and numbers and names for vulnerabilities fast enough as the Internet of Things is creating problems for the industry.

So one of the little studies that I ran across when I was putting this together, and this is something else that I've been holding onto for a while, but only for a few weeks, was Network World reported on a BitDefender study. We've talked about BitDefender. They are a well-known security company. They examined four popular Internet of Things devices. One was of course the WiMo products - WiMo is popular enough that probably everybody has heard of them - used to control lights and wall sockets. They found that the switch communicates with the smartphone without authentication. The only thing encrypted is the password, but the password is composed of elements of the MAC address and the device ID, both which are already transmitted without encryption since only the password is encrypted. And that encryption uses a preset knowable static key.

Leo: Oh.

Steve: They informed WiMo six months ago. Nothing has changed. So no traction from the vendor.

There's a light bulb called the Lifx Bulb that they also looked at. BitDefender found that its hotspot function suffered from insufficient authorization and authentication. When setting up this mood effect bulb, as they called it, a "temporary," in quotes, "bootstrap hotspot" is created to manage the initial configuration with the phone. But the problem is that you can later create, anyone can later create an identical fake bootstrap hotspot which then allows a hacker to capture the username and password of the main actual WiFi network that the bootstrap hotspot was only there to allow you to establish.

So the idea is, the way this works is that you have an existing WiFi network, and this Lifx bulb wants to get onto the main network. So what it does is it establishes a bootstrap

Open WiFi network that your phone can see. And then you use that to tell it to identify the main network that it wants to get on. You then give it your username and password, and you use this bootstrap hotspot to communicate to it because of course it has no otherwise, like, mouse and keyboard and screen and so forth. Problem is, anybody can come along later, bring up that bootstrap hotspot, and query it for the username and password. Not good. Once again, after six months of having been notified of the vulnerability, its manufacturer has taken no action.

There's something called the Link Hub which is a GE product. The GE Link light bulb hub is used for remote-control lighting, lacks any transport encryption when configuring through the hotspot, so the data is transmitted in cleartext. And once again, after six months, no activity.

The fourth thing that BitDefender looked at seemed a little better. There's something called the MUZO Cobblestone audio receiver, which did fare slightly better than the others in that some of its vulnerabilities were repaired after BitDefender notified its manufacturer. However, the initial issue was scary. And even after it was fixed, there's still a problem. So the device created a new WiFi hotspot that was never disabled. So again, this is this problem of how do you bootstrap these things onto the network? And it's funny because I had my best friend set up a WiMo. And he said, "Oh, yeah, it was so easy to set up." And I'm like, "Okay, well, how did you get it on your WiFi network, exactly?" And he said, "Oh, you know, I just used my smartphone and gave it my WiFi password."

And I had never looked at this process, but it seemed to me this was a chicken-and-egg problem because how could he give it his WiFi password if it didn't know his WiFi password because it wouldn't be on his WiFi. Now I understand. These devices set up an open unauthenticated temporary hotspot, which the phone, you know, probably called WiMo or something, which the phone then gets on, and that's how you communicate with it. Unfortunately, that feature is, like, still there. These things don't shut it down, probably because they're worried, oh, what if you change your WiFi password? Then you're going to have to get back on the hotspot.

So they're creating insecure hotspots which are inherently linked through them to your main secure network, making them insecure. And in fact, in the case of this MUZO Cobblestone audio receiver, believe it or not, it never shuts down its hotspot, and runs an open telnet service with the userID of "admin" and password of "admin" statically present, which allows open access to the household's original WiFi network, bypassing any need for credentials.

Leo: Well, it saves some time.

Steve: So unfortunately, that's a snapshot of the state of the art of IoT devices. We can't say that they're all a problem, but the huge majority of them really do seem to be a problem. And so taking a meta view, stepping back from the details a bit, these first-generation IoT devices are trying to do the impossible. They're trying to be, they're pretending to be a limited-use, purpose-specific appliance, with at the same time having all the sophisticated communications and connectivity power of a general-purpose computer hidden inside. But they're also trying not to have, not to present any of the responsibility baggage that all of our experience has taught us necessarily comes along with any powerful, connected, general-purpose computer.

And in fact, at the beginning of their report that we will cover in more detail next week in

Part 2 of this, Rapid7 put it perfectly. They said: "For our purposes, we can think of a 'Thing' with 'Internet' as simply any device, regardless of size, use, or form factor, that contains a CPU and memory, runs software, and has a network interface which allows it to communicate to other devices, usually as a client, sometimes as a server. In addition, these Things tend not to resemble traditional computers. They lack a typical keyboard and mouse interface, and they often have a user interface not centered around a monitor or other text-filled screen. Finally, these devices are marketed and treated as if they are single-purpose devices, rather than the general-purpose computers they actually are.

"This last distinction is often the most dangerous one to make when it comes to deploying Internet of Things devices. In his keynote address to the Chaos Computer Club titled 'Lockdown: The Coming War on General-Purpose Computing,' Cory Doctorow makes the case that, with today's technology and current computer science thinking, we cannot yet create a computer that is anything other than general purpose. End users may have devices that are nominally prohibited from performing certain actions according to the manufacturer, and those manufacturers sometimes go to great lengths to foil modification efforts. In the end, though, it is not possible to build and sell a computing device that cannot be coerced into rebelling against a manufacturer's intentions."

And so my own take is any system based upon a stored program must be able to have that program changed when bugs and security vulnerabilities are found. We all want that. We need them to be fixable. But that same need for the "ware" to be "soft" inherently opens the device to abuse.

So next week we'll look at some details of what Rapid7 found when they looked closely at baby monitors. And, boy, I know that our listeners will find it really fascinating.

Leo: Sounds like it's going to be worse than that, even.

Steve: Oh.

Leo: IoT, it's in its infancy. And, well, if you want to know a way, I guess, to use IoT, Steve has in the past described a more secure way. You don't want to give it access to your home network.

Steve: Oh, everything, every - yes. Everything we are learning says, I mean, you absolutely have to create a securely segmented network where...

Leo: Is a guest network sufficient? Problem is, then it's not really on your network, which makes another issue.

Steve: Right. That's a problem, too, is that, I mean, you certainly - we will show next week how anybody with a baby monitor, any of these that Rapid7 looked at, I mean, the horrifying insecurity, and the horrifying lack of care that the manufacturers have. What we see are companies producing feature-laden monitors that are virtually devoid of security. Meaning that anywhere, anyone in the world can be looking at your baby sleeping, or wherever you have aimed this camera. I mean, they're just - it's horrifying. And they don't care. They're selling functionality. They're not selling security.

And they all say, oh, you know, encryption this and secure that. And it's just nonsense. But the buyer doesn't know. They just say, oh, look, it said it's secure. Okay, click here and purchase. So there the vulnerability is not that it, I mean, you don't want it on your network, certainly, because lord knows if it's got an exposed telnet server that anybody can log into elsewhere in the world using admin and admin as the username and password, you don't want that on your network. But you really don't want it exposed to the Internet at all.

Leo: Yeah. Wow.

Steve: Yeah.

Leo: And I'm still, just for an update, no. I think you nailed it. So this is what happened to you. It gets to a certain point, and then it just switches the drivers.

Steve: Correct.

Leo: And it says, "I can't read that key anymore."

Steve: Correct.

Leo: And of course I don't have any USB 2.0 ports on this thing, or any other kind of legacy way of doing it.

Steve: Right, right.

Leo: Yeah. All right.

Steve: Makes sense.

Leo: But it's locked down. See? We can't have it both ways. All right, Steve.

Steve: Okay. So for anyone who's not interested in sleep, you can hit Stop now. I just need to briefly tell all of the people who have been following along what's going on. As I mentioned before - okay. One of the key ingredients of my Healthy Sleep Formula is Seriphos, which was the trade name of something called serine phosphate, produced by a company called InterPlexus. As everyone knows who has been following this, several months ago - coincident, unfortunately, with my going public with the Healthy Sleep Formula, which I don't have a sense for how many people we've helped, but lots - the InterPlexus company decided to discontinue their old formula. Unfortunately, the new formula doesn't work.

And so what I did, I guess it was maybe a week ago, as soon as the new one came out and reports surfaced that it wasn't working, was I put up a review on Amazon to warn people that this new one was changed and that anyone - the problem was there were, like, some 208 glowing five-star reviews on how fabulous Seriphos was. Unfortunately, unless people were warned, they would be purchasing the new Seriphos, believing it was the old one. So I just wanted people to say - I wanted to alert people that the formula was changed.

Well, InterPlexus was not happy with me, and I received a phone call last Friday from France, where the medical director of InterPlexus was traveling, to express his unhappiness. And I got harangued for an hour by this guy, trying to tell me how pure the ingredients were, and that they bought the best possible components and minerals from Germany, and this new formulation was incredibly high quality. And I said, yes, that's very nice, but it no longer works. And he apparently didn't care that it no longer worked. He just was focused on how high quality the new ingredients were. So it's like, well, okay, sir. I'm sorry. Anyway, so I updated my review to tell everybody how good the new ingredients were, in fairness to him, but also that it unfortunately appeared to no longer work. Make up your own mind, but I have to go find some other solution.

So I tracked down the people who make Enerphos, E-N-E-R-P-H-O-S. The bad news is the moment I posted that news, that like maybe this was going to be a replacement, they sold out all over the Internet. No, you couldn't get it anymore. And so same problem, again. I found them, and I told them that - oh, and I also talked to the chemist who was the originator of Seriphos. He's the guy that created the original Seriphos and was supplying it to InterPlexus for quite a while, until there was some falling out, and they went their own ways.

So this morning I got an email from the company that makes Enerphos. And Sheryl, who is one of the executives, said: "It was good to talk with you last week. Thank you for reaching out to us and giving us a heads-up on the new demand we can expect for Enerphos. Brian worked all weekend making more." Literally making it.

And so she said: "We have a few hundred bottles en route to Emerson now and expect to be able to ship more by Friday. We'll keep you informed of our shipments to our suppliers, so you can keep us updated on the feedback you get from your readers. Thank you again for your thoughtfulness in contacting us directly." And of course it was self-serving because I would like people who are interested in the Healthy Sleep Formula to be able to get the components of it.

So on the Healthy Sleep Formula site, or page, just google "healthy sleep formula," is a link to Emerson. Emerson is one of their larger suppliers. And right now it's backordered. But they're in the process of getting more stock. So if anyone is waiting for Enerphos, the thing to do would be to backorder it. And I'm sure that this Emerson - Emerson Ecologics is the name of the company - will ship it when they can.

Now, I should say I haven't yet made my own determination about Enerphos as opposed to Seriphos. It just takes time. I've been working on this since last October, and I only went public with it recently. And it's changed a lot of people's lives. And now the problem, of course, is that we can no longer get the Seriphos that I based this healthy, like a lot of this formula was based on. So I don't know what I'm going to do. I need to see whether Enerphos is a replacement. And if it's not, then at least I know who invented the original Seriphos, and maybe I could say, "Hey, Brian, can you make some of the original stuff for us because we need that." Anyway, that's where we are.

Leo: I can see it's not long before you start to grind your own phosphatidylserine in a mill in the back of your garage and start making Steve's [Numinos].

Steve: Well, I love that this guy and company, they're in Fresno, and he made a bunch over the weekend because we need more. So that's nice.

Leo: This is starting to sound a little weird. I'll be honest with you, Steve. Just a disclaimer: Steve's not a physician. Consult your physician before taking any supplements. And if you want to know more, GRC.com. And I got nothing to do with this one. This is all you and the Seriphos family. Okay. What is it supposed to be, actually? I mean, is it actually a formulation of a variety of things, like a secret formula?

Steve: Okay. So there is a - great question. There is a molecule called phosphatidylserine. It's a so-called phospholipid. What that really means is that it's a long molecule. It's got phosphorus in it. And attached to one end is something else, in this case serine. And attached to the other side of the phosphorus is a glycine molecule, to which two long lipid tails, actually DHA, docosahexaenoic acid, are attached. And so this is a very long molecule. And you can get phosphatidylserine on the market.

The problem is that, because it's so big, it isn't well absorbed. And but it's very good for us. It increases cellular membrane fluidity, that is, the actual fluidity of our neural cells, so that they're able to pass stuff in and out more easily. And it turns out that it increases the sensitivity of our brain to cortisol. And so by increasing its sensitivity to cortisol, our brain senses the cortisol in our blood better and thus requests less of it from our adrenal glands. And that's the mechanism by which it reduces this main stress hormone which is one of the mechanisms that we use, that the Healthy Sleep Formula uses for producing better sleep.

The other is that phosphatidylserine helps GABA get into our brain through the blood-brain barrier to relax our brain. But phosphatidylserine is not well absorbed because one of the things that we learn when you're following the whole supplement thing is it's not what you take, it's what you absorb that matters. And so serine phosphate is just sort of the head of the whole phosphatidylserine. It's the serine, the phosphate, and then what's attached to it, instead of glycine and two big long phospholipid tails, or fatty acid tails, rather, is just something small, like magnesium and calcium.

And so what that means is it's a much lighter-weight molecule, much better absorbed. And so that gets into our bloodstream. And then our bodies attach the lipid, the fatty acid tails to the serine phosphate head in order to turn into phosphatidylserine. So it ends up being phosphatidylserine inside of us, but not when we take it. So anyway, yes, Biochemistry 101.

Leo: It's just, I mean, the only thing that worries me is some guy in his garage in Fresno, and this is a completely unregulated industry, so who knows what he's, I mean, doing. You don't really kind of know exactly what you're getting. There's nobody overseeing this at all. Right?

Steve: That's true. And actually this is a snapshot, though, into the supplement industry.

Leo: Yeah, no kidding.

Steve: And all I would say in defense is that the number one cause of liver failure and liver transplants in the U.K., of course, is Tylenol.

Leo: Yeah, which I don't take, either.

Steve: So, yeah.

Leo: And it's regulated, right.

Steve: Yeah, exactly. And of course pharmaceuticals are causing all kinds of problems for people. So, yup. Six to one, half a dozen to the other.

Leo: Anyway, GRC.com is the place to get more on that. Of course SQRL, the Perfect Paper Passwords, and SpinRite, the world's finest hard drive recovery and maintenance utility. Plus this show. Steve has audio of the show and beautifully written transcripts, as well. It's all there. Do you ever document somewhere, is it on the website, the build, your computer build, and what you put into it? A lot of people are asking me about it.

Steve: I know. No, I haven't. It's just a matter of time, Leo. I just, you know, I mean, I would like to, but...

Leo: Take some pictures. This is where crowd-sourcing would be so great. If you like could take some pictures and say tag these parts. And people would do it. They would figure it out. I don't know. I don't know. Just a thought. But we'll find out, I'm sure, at some point. One of many things. It's such a great site, full of lots of information: GRC.com.

We do this show every Tuesday, 1:30 p.m. Pacific, 4:30 Eastern, 20:30 UTC. Please tune in live. Join us in the chatroom. If you have questions for Steve, I guess we're going to finish IoT up next week.

Steve: We're going to do Part 2 of IoT Infancy.

Leo: And we'll do questions the week following.

Steve: Yeah. And I have to say, of the big monster box I built, the reason I'm uncomfortable is that it's not something I could recommend. Like, for example, I chose a, what was the name, it was not ASUS.

Leo: Acer.

Steve: A Gigabyte motherboard.

Leo: Oh, Gigabyte motherboard, that's right.

Steve: But I don't know that it was the right one, or the best one. And I kind of had to fight with it in order to get, you know, I wanted to run 128GB of RAM. I could only run - only - 64. But I could run 128 if I slowed it down. So I would, I mean, you know me. I don't want to recommend something that I don't really know, and I haven't really believed in. So once upon a time, Steve's Dream Machine was a result of really knowing there's no better solution for every single one of these components than what I chose. I just don't know that, you know, for this. This works, and I fought with it for a while to make it what I wanted. But I just - I couldn't ever represent that it's, like, the best solution.

Leo: That makes sense. It's just something you chose. But you know your fans. They want to do whatever you did. They figure it's going to better than whatever they do. Anyway, it's good. You make a good point.

Steve: I shopped intelligently.

Leo: Yeah, you make a good point. Let's see. I mentioned, oh, you can get on-demand audio and video of the show, not only at TWiT.tv/sn, but every single podcatcher, all the world around. Just choose one. Subscribe so you get every week.

Steve: Because we've been there from day one.

Leo: Day one. And at LeoLaporte.com/blog there is an entry with three scripts that will allow you to, with some modification, download every episode of Security Now!, should you want to build your collection. I put those on my blog. Three different fans have written scripts in PowerShell and Python and Bash, I think, something like that.

Steve: Nice.

Leo: Yeah. So that's at LeoLaporte.com/blog. Thank you, Steve. We'll see you next week.

Steve: Okay, my friend, thanks.

Leo: Bye-bye.