Deprecating TLS 1.0 & 1.1

Description: This week we discuss Intel's big Spectre microcode announcement, Telegram not being long for Russia, U.S. law enforcement's continuing push for "lawful decryption," more state-level Net Neutrality news, Win10's replacement for Disk Cleanup, a bug bounty policy update, some follow-up to last week's Quad-1 DNS conversation, why clocks had been running slow throughout Europe, and then a look at the deprecation of earlier versions of TLS and a big Cisco mistake.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-658.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-658-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with lots of security news, including the end of the line for TLS 1.0 and 1.1, and a response to the haters who are mad about 1.1.1.1. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 658, recorded Tuesday, April 10th, 2018: Deprecating TLS 1.0 and 1.1.

It's time for Security Now!. Oh, yeah. Oh, yeah. I know, I imagine thousands of people all over the world saying, "Oh, yeah." Yeah. Of course, it's not really a surprise to them since they pushed the Play button and all that. But still, you're playing the right show. This is Security Now!. Steve Gibson, he's our...

Steve Gibson: The technology has not let you down.

Leo: It has not let you down. The deterministic...

Steve: Unlike the technology we discuss every week here, which pretty much does let you down.

Leo: Yeah, it's letting you down all the time. That's Steve Gibson, GRC.com. He's the creator of SpinRite, the world's finest hard drive maintenance/recovery utility, but also ShieldsUP! and, gosh, so many useful tools and utilities. And he's been spending

the last 13 years every Tuesday with us, just waving his hand like that.

Steve: Look how my hand blurs when I move it really fast, Leo. You can hardly even see it.

Leo: Steve's on the spectrum. Just don't mind him. No, just teasing.

Steve: I'm joining Mark.

Leo: He's joining Mark Zuckerberg in the land of the...

Steve: Woohoo.

Leo: ...perpetually perplexed.

Steve: Ah, yes.

Leo: No, Steve is a lifesaver for those of us who follow technology and want to understand it better and understand the perils of it, and that's what we do each and every week.

Steve: And Leo, this Picture of the Week I have been saving. I've been wanting to find an opportunity to deploy it for quite some time.

Leo: I don't think I can fit it on the screen.

Steve: I had to squeeze it down. I wasn't happy to have to squeeze it down. But you can still see what it is, I mean, very clearly. But we'll get to that in a minute.

Leo: Okay.

Steve: Today Episode 658, for those of you who are counting along, is titled "Deprecating TLS 1.0 & 1.1," which follows on from our recent discussion of the final ratification and the adoption of 1.3 and all the goodies it offers. And this was triggered by a blog posting of DigiCert's, where they were talking about their own discontinuation on April 1st. And they said, no, no, no, this is not a joke. We are discontinuing support for 1.0 and 1.1 of TLS. I don't know whether they yet support 1.3, but they certainly do 1.2. That's where everybody is.

So I want to talk about that because there was one just perfect quote from their blog that I really enjoyed; and I thought, okay, it makes sense now to talk about how we get

out of where we have been. And as we know, it's never easy to stop using something which is working and isn't obviously a problem. Even when it is obviously a problem, we often still just say, oh, well, like v4 of IP, IPv4. We're all still using it. We're not supposed to be because it ran out.

Leo: It ran out.

Steve: It ran dry. There is no more. But, oh, look, I'm still using it, and so are you. So what the heck. Anyway, we're going to talk about Intel's big Spectre microcode announcement; Telegram inching out the door of Russia, or actually being pushed out the door; U.S. law enforcement's continuing push for, in air quotes, "lawful decryption," which actually unfortunately amounts to unlawful encryption, which I just - the way that bill was worded is annoying.

Also we have more state level, local U.S. state level Net Neutrality news. Lawrence Abrams at Bleeping Computer is a fan of a coming Windows 10 replacement for Disk Cleanup I wanted to share, since you and I were just talking about Disk Cleanup and how we recommend it to people. We've got a bug bounty policy update which is sort of interesting, I think very interesting, actually. And something kind of icky occurred to me as I was pulling this all together that I want to share. Some follow-up to last week's Quad 1 DNS conversation. I upset a lot of our listeners who were all Quad9 happy. It's like, wait a minute. Anyway - you changed your quad. Also why clocks had been running slow throughout Europe all year until just recently. Really. That's in our Miscellany category.

And I forgot to put it in the show notes until I'd already published them, so I'll put it right here, right now. Friday the 13th is in, what, three days. That's when "Lost in Space" becomes available on Netflix.

Leo: Yay. Yay. Can't wait.

Steve: So I haven't seen it. I'm not making any recommendations. Those will come next Tuesday, more than likely. But I just wanted to give everybody a reminder of that. And then we'll wrap up by talking about the process, like where we stand as an industry with the various versions of TLS. All of SSL, that's all gone now. So now we're on TLS, and we've got four of them. We've got 1.0, 1.1, 1.2, and 1.3. And we're saying goodbye to the first two. So we'll talk about that interesting update and news for our listeners, oh, and the Picture of the Week when we come back from our first sponsor.

Leo: Can't wait. I'm not sure what the point of it is, but I'm sure we'll find out. I'm sure we'll find out.

Steve: Oh, yeah. Well, I was going to say also, before I forget, this just in from a fan and friend of the show Simon Zerafa. And that is that today's Patch Tuesday updates for Windows 7, which is the same as Windows Server 2008 R2, have known crashes in them.

Leo: Oh, isn't that nice.

Steve: So, whoops. After you install the update, your Windows 7 or Server 2008 R2 is known to leak memory out of its SMB server - the Server Message Blocks, file and printer sharing - and a blue screen. Whoops. A stop error occurs on computers that don't support the Streaming Single Instruction Multiple Data, SIMD Extensions 2, that's SSE2. So maybe not crucial.

First of all, if your chip is SSE2-compliant, then you don't have to worry about blue screening. And if you're not doing file and printer sharing and/or a memory leak, I'm not sure how severe it is, that is, how quickly the memory gets depleted. But I guess this must have happened to Microsoft just as they were getting ready, committed already to roll it out because today is the second Tuesday of April, and we've got patches with known problems. So if you're a little more conservative, you may choose - they have said they're right on this, that they're working on a resolution and will provide an update in an upcoming release, presumably without waiting till next month. So you might want to wait a bit longer for them to get that fixed.

And our Picture of the Week, Leo.

Leo: Yes.

Steve: Someone sent this to me, and I had the same feeling you did. It's like, what, so? Then I realized, wait a minute. This looks like your Ace hardware store, whatever. They've decided that these bolt cutters are high-end expensive. And unlike the rubber mallet to the right, which is just hanging on the J hook, these they're going to lock down with - looks like a plastic-sheathed wire rope.

Leo: Oh, I see what you're...

Steve: It's got padlocks at each end.

Leo: Yeah, yeah. Oh, good idea.

Steve: And so there's no way that a thief could arrange to steal those bolt cutters which are secured by exactly what it is they are designed to cut.

Leo: This is a subtle one, Steve. I did not get it right away. But if you look at the middle one...

Steve: Yes, that's the perfect one, yes.

Leo: The lock wire is poised to be cut by the bolt cutter. I think I'll take this one, it works nice.

Steve: And you can test it on your way out the door.

Leo: Maybe that's the point, it's a tester.

Steve: That's right. Ah, love it. It's just wonderful. This is a variation on the fox guarding the henhouse. It's like, okay, wait a minute.

Leo: I didn't get it right away, and now I do. That's hysterical. I love it.

Steve: It's just wonderful. So thank you, whoever it was who sent that to me. It was months ago. I capture these, and I save them. I have a place to keep them. And when there's nothing that is salient to the topic of the week, I pull one out of the archive. And this one, I've just been waiting to share this because this is just - this is one for the ages.

Leo: Nice. Very funny.

Steve: Yeah. Okay. So on a not-so-funny note, we have Intel having given up. That's the announcement in the April 2nd, which is apparently their final, revision to their so-called "Microcode Revision Guidance" PDF. I've got the link in the show notes for anyone who's interested. And this will drive an update to my InSpectre app. I imagine I'll just - it's so quick that I'll do it quickly, maybe this evening. And this shouldn't come as a surprise to our listeners, who have heard me every single time talk about how I've been surprised Intel could even fix this problem with microcode.

Well, it turns out they can't fix all their chips, is what this comes down to. Up until this April 2nd chart, they had listed all of the various processors by CPU ID and architecture and family and lineage and everything, all these wacky names they come up with. They had them under categories of pre-beta, beta, production candidate, and production, which were like, okay. Or pending, I think. There was an earlier one for a while, too. It's like, okay, we haven't gotten to that one yet. Or we have it, we're working on it, we're almost going to release it, and we have.

Well, they've added another category: Stopped. And so for their description on "Stopped" they said: "After a comprehensive investigation of the microarchitectures and microcode capabilities of these products, Intel has determined to not release microcode updates for these products for one or more reasons including, but not limited to the following." And they have three bullet points: micro-architectural characteristics that preclude a practical implementation of features mitigating Variant 2 of Spectre. Second bullet point: limited commercially available system support software, which I guess means there's nobody who's willing to push any patch we came up with out into an actual platform. And the third bullet point: based on customer inputs, most of these products are implemented as "closed systems" and therefore are expected to have a lower likelihood of exposure to these vulnerabilities. Which I think that's reasonable.

So while this will be disappointing to those people who do have those processors, first of all, they are the older chips. They're generally older than five years old. So it may well have been actually the case that, even had Intel produced microcode, there were no OEMs around who were interested in bothering to push out the fix. So what we have now with this PDF is a final comprehensive listing of the state of the chips that Intel has. Most of them are in production. There are just a few, I mean, with pages and pages of this PDF, that are in "production candidate," so they're in just the final prerelease state. And

then there is a handful that are "stopped."

So what I'll do, now that we have this, what I'll do is to add that knowledge into InSpectre and incorporate that so that it will affirmatively let people who run InSpectre know whether there is microcode available for their chip somewhere, which presumably Microsoft will eventually incorporate, if they haven't already, into their own microcode, on-the-fly boot-time microcode patch; but also if Intel has affirmatively now said microcode is never going to happen for this chip. So, sorry about that.

Anyway, so we do have sort of the final position. And again, while yes we're unhappy that we've got chips that have this problem, as I've said repeatedly, I'm impressed that the microcode had the ability to be patched at all. I mean, the idea of microcode is not to just make up your own instructions after you've shipped the chip and changed things around. The idea is to compromise on a fully hard-wired design which would require a wafer of silicon probably three inches across, just because the Intel chipset, the instruction set has become so unwieldy over the years, to sort of compromise and have a microcode step which runs faster than the instruction rate so that some of the more complex instructions - and, boy, there are some hairy instructions in Intel land - you would actually implement those as sort of a processor inside a processor, rather than just all brute force with electronics and gates.

So it makes sense. And so I've been impressed that they could do as much as they have, and relieved that for a problem this bad it's even been possible to have an after-sale update to the chips. And I am thankful also that Microsoft has stepped up to the task of dynamically installing that microcode provided by Intel because, as we know, our current supply chain for systems is lacking. It's maybe better, well, maybe it's about the same as it is for mobile devices. I was going to say historically we've always talked about how sad the situation is with old Android phones that are just never going to see a fix. This is, you know, we're in a similar situation with our desktop and laptop systems. I have several that I would love to get patched. I mean, and they're Lenovo. But it's like, eh, they're too old. So no.

But my newer ones, they already got patched. And a couple Dell laptops got patched, too. So I've been impressed with Dell's performance, actually. I think they're one of the most impressive OEMs in terms of following up on this and dealing with even their older devices which they're still keeping in their support loop. So anyway, probably tomorrow InSpectre will get updated. And I did, as I said I might last week, update the DNS Benchmark for the first time. When was it I finished that? In '08, I think, so 10 years. Yeah, it was '08. So I did it, and we'll talk about that in a minute.

I wanted to mention that, as we have been discussing, Russia doesn't like people having conversations inside their borders that they cannot decrypt. So four days ago Russia's telecommunications watchdog agency filed a lawsuit against Telegram, which we've been discussing this was happening, which is within Russia a very popular encrypted messaging app, of course, that we've talked about often because Telegram uses a controversial rolled, I mean, it's bizarre. It's more than controversial. It's just wacky, their own encryption system where just all kinds of stuff is flying in every direction, and no actual cryptographer, you know, cryptographers just scratch their heads and say, what? But Telegram has responded by saying, okay, so break it, and we'll give you some money. But no one has bothered because it's just wacky. Still, it's wacky and no one has broken it, as far as we know.

So this telecommunications watchdog in Russia has asked a Moscow court to rule in favor of restricting access to the service inside Russia's borders - in other words, blacklisting Telegram. So as we know, the FSB, which is Russia's main intelligence service, had

requested access, formally going through the steps. And I doubt that they ever expected to get it. But they formally requested access to Telegram's encryption keys so that the FSB could access encrypted messages sent through Telegram. Telegram, of course, based in the U.K., refused to help. So the FSB filed a lawsuit to compel access to those encryption keys, claiming their need to have access for national security and the agency's fight against terrorism.

So thus far every court that has been asked to rule on this as they go through the machinations has ruled, not surprisingly, in favor of the Russian state. And the final decision came last month when Russia's Supreme Court ruled that Telegram must hand over users' encryption keys to FSB agents without a court order whenever agents requested access. So a very broad request. So Telegram responded through their attorneys that they have no such plans to do that. They are just making, I guess it's a face-saving attempt to fight back, filing a lawsuit against the Russian government at the European Court of Human Rights, saying this is a breach of Russian citizen human rights. That'll go nowhere because ultimately Russia is going to do what it wants to do.

So what will happen is, with the next ruling from the lawsuit filed four days ago, when that happens - and everyone's expecting this Moscow court to rule favorably again as it surely will - then Telegram's Internet domains will be added to Russia's official national block list, which all telcos and ISPs are required to abide by, which will in turn shut down Telegram throughout Russia. So certainly at this point no Telegram user within Russia, as a consequence of these gears grinding away as they have been for quite some time, will be taken by surprise when Telegram suddenly stops working for them. They'll have to do something else. So anyway, that's where that is, and I imagine that maybe by next week or the week after we'll just have a short blurb saying Telegram, as is the case in China, no longer functions within the borders of Russia.

The U.S. Senate, meanwhile, is getting set to take yet another run at mandatory lawful decryption of communications. It was just after the San Bernardino terrorist attack where, of course, which we covered at the time, Farook's work iPhone was not easily decrypted, that our senators Dianne Feinstein and Richard Burr, those were the two at the time who pushed some legislation which they authored that never got to a vote because it generated a lot of blowback and critique as a consequence of the fact that it was seeking, essentially, mandated backdoors in popular consumer products.

Well, they're back again. Actually not with Richard Burr, but Dianne Feinstein is, this time with the current Senate Judiciary Chairman Chuck Grassley. So they're getting ready, along with the Deputy Attorney General of the U.S., Rod Rosenstein, to make another attempt at essentially, the way they're putting it, outlawing the use of encryption which the government cannot have access to. So I would imagine it would be under warrant. But as we know, it would follow the traditional U.S. search warrant model.

I don't know how this is going to end. My feeling is that law enforcement and our intelligence services are going to continue to push on our legislators to produce a solution, and that our legislators are not going to agree with academia and cryptographers and industry that there is no way to do this without severely - essentially, as we know what the argument is, there's no way to put in a backdoor that the bad guys cannot also get access to. So we'll see. My sense is we're going to go around and around and around until finally the legislation occurs that will compel consumer products to have a means for providing lawful decryption under a court order. I'll be surprised if that doesn't happen.

And on the topic of legislation, a little better news maybe, the state of Oregon yesterday signed into law their own state Net Neutrality legislation. It makes them the second state

to have done so in the U.S. The state of Washington had previously done so. And Washington's legislation was aggressive and may in fact fail under challenge because of course all of the ISPs are loudly declaring that they're going to fight to overturn this, or to have some sort of federal legislation enacted to prevent states from going their own way.

What Washington state did was they flatly stated that no ISPs or broadband carriers can provide any favorable treatment based on the source of the content that they're carrying. What the state of Oregon did was deliberately designed to withstand a challenge in court, which is to use, again, the power of their purse to say that no state government entity can purchase that bandwidth from an ISP that is not abiding by the same Net Neutrality principles and laws that the FCC is in the process of ending.

And as I was looking through this to sort of get a sense for what was different and what it said, there was one glitch that was unfortunate, and that is that the legislation does acknowledge the possibility that even state agencies might be forced to choose an ISP that violates Net Neutrality when there are no other options. In other words, the purchasing requirements will not apply when an ISP is, in the words of the legislation, the sole provider of fixed broadband Internet access service to the geographic location subject to the contract. And as I've often stated, in my own location here, right in the middle of Southern California, I have no alternative other than Cox and cable. There's no fiber. There's no high-speed copper. I have one broadband provider. And as we know, unfortunately there are many areas in the U.S. where that's the case.

So anyway, certainly what we're seeing is the intention and execution of some pushback from the end of FCC's previous U.S. administration protection for the concept of Net Neutrality. And here I sort of feel like the reverse of this constant go-around with encryption. My sense is Net Neutrality's probably - the principle is probably going to win out over the long term, where I feel less bullish about encryption winding up that way.

Windows 10 has - we've spoken of the upcoming next major feature update, which is still, I guess it hasn't been officially named because it's still being referred to as the "so-called Spring Creators Update." But it's getting ready to happen. That will be adding a feature which Bleeping Computer's founder Larry Abrams is bullish about, so much so that he intends to stop recommending, as he has been, the always-there-since-the-early-days Disk Cleanup utility in Windows. Disk Cleanup will still be present, but Windows 10's Spring Creators Update adds the not quite as catchily named Free Up Space Now option.

So if anyone is on the inside track, or the early adopter track, whatever it is called, you may have already seen it. And certainly, when it is released, the Spring Creators Update, you put the word "storage," you just search for "storage." That'll bring up the panel in Windows 10, and there's a new option there: Free Up Space Now. And what Larry likes is that it, unlike Disk Cleanup, which may sometimes show options for which there is nothing to be cleaned up - that is, it'll show you like zero megabytes or kilobytes savings are available. Unlike that, Free Up Space Now only enumerates those things that can be saved, and you don't have to go through that second step you do with Disk Cleanup that, Leo, you were aware of and mentioned when I brought it up last time, where you have to explicitly say "Clean up system files," click that, and then it goes away again for a long time and then finally comes back and says, oh, look, I found 10 more gig. It's like, yes. So anyway, it's all in one, Free Up Space Now, coming to Windows 10 Spring Creators Update.

Leo: Have you done your taxes, Steve?

Steve: Yes. They're all handed into my CPA. Mine is one of those kind of complex business and...

Leo: Oh, I bet it is, that's right. Anybody who has a business, it's crazy, yeah.

Steve: Yeah. So more than I can handle.

Leo: So I'm just checking here, let me just do "winver" real quickly because I just got an update, but I don't think it - yeah, I'm still at 1709.

Steve: Yup, and I just checked mine while you were talking, and same thing here. I got cumulative update for Windows 10 v1709 for x64-based systems. And in the Update History it shows the feature update. I think I applied that in November.

Leo: Yeah.

Steve: So that's the thing that we're looking for is that maybe it'll be called the Spring Creators Update. Did they actually officially call the previous one the Fall Creators Update ever?

Leo: Yeah. It was the FCU, Fall Creators Update. And they're refused to say a name. I think they don't want to name them. But you'll know if you hit the Windows key, type "winver." It will tell you what version. And if it's 1709, that's the previous version. The new one will be 1803. So you will know that, anyway. But they stagger these rollouts. It starts today, but it won't get to everybody today.

Steve: Can you ask for it tomorrow? If you, like, say, "Hey, I want an update?"

Leo: That's a good question. I mean, if you were an insider you'd get it. But you'd also get the next one, which you don't want. They usually do an ISO version that you can download from Microsoft, but that takes a while. So I don't think there's any way to force it. Not that I know of. Let me look in Advanced Options. No, I don't think so. I'll tell you what, I'll ask Paul and Mary Jo that tomorrow.

Steve: Cool. I'll be listening. So, okay. It was an interesting article that CyberScoop picked up suggesting sort of a different side to the bug bounties issue. And in pulling the pieces together, I came up with my own, maybe even kind of more worrisome problem. So since they're sort of tempering my own recent bullishness about bug bounties, I wanted to share it for some perspective. So last Wednesday CyberScoop published an article titled "The bug bounty market has some flaws of its own." It got its inspiration for the article primarily from a woman who founded Microsoft's own bug bounty program almost five years ago, Katie Moussouris. She was subsequently the Chief Policy Officer at HackerOne, that's the commercial bug bounty-managing service, and has now founded her own firm, Luta (L-U-T-A) Security.

Well, it happens that today, Tuesday, April 10th, is the first day of the three-day CYBERUK 2018 Conference which is being hosted by the U.K.'s national Cyber Security Centre, which is part of GCHQ. And Katie was there this morning presenting, and somebody was tweeting some of her presentation. And she was also then subsequently tweeting up a storm afterwards.

Her home page, that is, the Luta Security home page has a banner that says, "#1 Solutions Architects for Vulnerability Disclosure or Bug Bounty Programs." And in fact the homepage currently announces that the U.K. government announced at the CYBERUK Conference - the one she's at and was presenting to, she was in the first track - that its new National Cyber Security Centre, the NCSC, is partnering with her firm to invite a select group of security practitioners in the community to participate in what they call the historic first U.K. government pilot for vulnerability coordination. So the U.K. is beginning to sort of move in this direction also.

And, finally, her page concludes with sort of the footer, "Bounty Smarter, Not Harder." I looked through the slides that were presented, a bunch of her tweets. And the upshot of it all is that there is some concern that more of the industry than just Katie share that there could be some downside to these big monster quarter million dollar bug bounties, which major players in the commercial industry have recently been announcing. And essentially it creates a distortion where an award for finding a problem is often larger than the annual salary of someone whose job it is for there not to be any problems.

And Katie's position, after years of her involvement in the industry, is that it is a mistake for major companies to simply solve their problem by offering a massive reward for finding problems in the wild; that there's a danger of sort of just focusing less on prevention in the first place rather than curing them after the fact. As we know, many of these massive companies are cash rich. And so it's easy to - and you can also sort of see there's a PR upside to it. You know, that's been part of our motivation for covering this, is it's like, oh, look, Microsoft is really serious about wanting their bugs to be stomped on. They're offering - and of course Intel, too, in the case of these side channel attacks - a quarter million dollars through the end of the year for anyone who finds any more of these.

So that makes headlines. It makes a big splash. The concern is that it's easy to throw money at that, and it also tends to sort of maybe create some perverse incentives. So in an interview with CyberScoop, Katie said of bug bounty seeking that, first of all, it's not cut and dry, that motivations vary among hackers, but most are driven by a combination of three factors: financial compensation, which is most obvious, I guess; peer recognition, which is certainly the case and has been probably a major motivator prior to the big cash awards; and then also just the pursuit of intellectual happiness, loving what they do.

There is a book that was just published by MIT Press in academia where the editors of a series of articles to which Katie was a contributor, they did an analysis of the market for software flaws and revealed, I guess somewhat unsurprisingly, that this so-called "defensive bug bounty market" is very stratified with a small number of highly skilled individuals essentially winning the lion's share of the rewards. So it is not uniform. In the dataset they analyzed, the authors found that a small number of key sellers, as they call them, of discovered bugs are finding the overwhelming majority of all the bugs. So not uniform distribution, highly spiked.

In one dataset provided by the guys at HackerOne, that as I mentioned we've talked about them a couple times now, the site and service that manages bug bounty programs for other corporations, just 5% of bug hunters were responsible for finding 23% of the

flaws. Which actually it's not like 5% found 95% of the flaws. So 5% found almost a quarter of them. So, yes, it's not uniform. It's also, however, not massively skewed incorrectly.

So I guess perhaps the best advice is not to give up your day job; that it's probably going to take many, many hours of poring through disassembled code or maybe writing your own custom fuzzing pentesting code. So it ought to first and foremost be something you enjoy doing. You ought to think of it more as a hobby than as a career, some way to fulfill any spare time that you may have, which could have a big payoff if you find something significant.

So with all of that said, while I was thinking about this, it occurred to me that there was another dark side that I don't think anybody - certainly we haven't talked about it. I've never read anything about it, and it hadn't occurred to me until now. But it's a very worrisome possibility that, again, could be created by these quarter million dollar bounties. Okay, so we have a few givens. We know that it is extremely difficult to create flawless code, so much so that, despite the fact that everybody tries, nobody does.

We also know that it's extremely difficult to find flaws by inspection. I've often talked about how, as a developer myself, when faced with a bug, there have been times when I have stared at the code, at a page which apparently has a problem, looking at it and not seeing it. And it's not until single-stepping with a debugger onto the problem that you finally go, oh. I mean, you can't not see it at that point. And it's like, oh. Things like type conversions that are happening by the compiler where somewhere, in some include file somewhere, something was typed to be the size of a byte, and your code is assuming it's the size of a word or a long or something. I mean, those sorts of things where you just...

Leo: That's why you need a compiler. You keep doing assembly language, of course you're going to get bit by that. I'm curious. Do you ever do modern things like test-driven design? Do you write tests for your code, you know, for your function? I don't even know. Do you do functions?

Steve: Yeah, yeah.

Leo: You write little macro functions; right? I mean, I presume that's how you work.

Steve: True. And arguably, Leo, where SpinRite is 14 years old, and the DNS Benchmark is 10 years old...

Leo: There's not many bugs in that thing, yeah.

Steve: There are no bugs. I mean, I have an approach. My approach is incremental creation of the code where I'm testing as I go. I do get tripped up when I think of something later that I wish I had done earlier.

Leo: And add it, yeah, that's how you can screw stuff up badly.

Steve: Yeah, because then you get side-effects. But anyway, so we have difficult to create flawless code, difficult to find problems by inspection. They're just hard to see. And we know that backdoors, okay, so "backdoors" is the term we use typically for deliberately inserted loopholes in a system's security. But as we've previously seen, not all backdoors take the form of a retrospectively obvious thing like a hard-coded password in firmware. We often find that as we have discussed often in routers, where someone discovers, oh, look, password 1234 backwards was left in the firmware of some D-Link or Linksys or who knows what router. I don't mean to pick on any one particular brand because they've all had problems that way.

So the point is there are situations where you discover what is obviously a backdoor that somebody put in there. But we also know that not all backdoors are retrospectively obvious. And we've talked about some. For example, you could have what you might consider a gray door occupying a gray area, for example, as we saw with that dual elliptic curve DRBG, the Deterministic Random Bit Generator, which may have been a deliberate design flaw, designed and implanted into that spec by the NSA. We don't know, but it looked worrisome. And it was there. And they did come up with it. And they then paid RSA a large sum of money at the time for RSA to make that the default of the four, even though it was the slowest of all of them. So, like, it's suspicious, but not proof.

So my point is that you can also have gray doors, which sort of occupy this gray zone, this gray area. So, okay. Offering an award of a quarter million dollars to the "discoverer," and I'm going to put "discoverer" in quotes now, air quotes, of a critical flaw in a mainstream commercial system can also create a powerful incentive for such a flaw to first be deliberately implanted by a confederate accomplice who has code writing access. In other words, whereas the NSA - and remember that we've wondered that the U.S. intelligence services might not have agents installed in major corporations with this kind of access, doing things on behalf of national security. Well, a quarter million dollars is a lot of incentive for some people, and not all coders in all corporations are happy with their jobs.

So my point is it's so extremely difficult to create flawless code when you want to that no one expects it any longer. This tends to remove suspicion of flaws which we've now all come to accept, which in turn means that it's absolutely possible for a very subtle, yet still critical flaw to be deliberately added to existing code in such a fashion - now, you can't go back and edit that code to put the flaw in because, again, we have version control systems, and someone's going to come along, and your name is going to be on the change that was made, and you're going to have to explain why you did that. But when writing a bunch of new code that all appears to be hunky-dory, it's entirely feasible that somebody could deliberately put in a flaw that cannot be seen by inspection, doesn't cause anything to fail, cannot be found casually, and for which a compelling subsequent discovery miracle story could be fabricated with a substantial award at the other end.

And I'm not suggesting that the employee who wrote the code could win the prize for finding it. But this is why you need the confederate operating outside of the company as an independent researcher to, quote, "discover," unquote, this problem and win himself a quarter million dollars in cash. So I was thinking about, as I was running through the nature of these incentives, with projects as big as they are, as many coders as there are, as well meaning as most programmers are, as these bounties become massive, it does create various sorts of perverse incentives. And it just occurred to me that, wow, that might make a great subject for a believable work of fiction. Let's hope it stays that way. Who knows? And we probably would never know because, if it's done well, it would just look like it was a mistake.

I wanted to follow up on last week's discussion of Quad 1 and Cloudflare. I got a bunch of

what I would have to describe as "blowback" from our listeners, who were all happy with Quad9, and happy with my previous happiness of it, and basically saying, hey, what about Quad9? Now you're all happy and jumping around about 1.1.1.1, but I'm all set up on 9.9.9.9.

Leo: Oh, come on. How hard is it to switch? Geez Louise. Oh, I have to go in and change my DNS? Oh, man.

Steve: I know. So...

Leo: Steve, make up your mind. You can't change your mind.

Steve: So I did want to follow up. As I mentioned...

Leo: I don't think you changed your mind anyway. It's just another thing you could use, a different thing to use.

Steve: Exactly. It offers different benefits. And it turns out that Quad9 also offers DNS over TLS. Apparently not DNS over HTTPS. I looked around for that. I couldn't find any sign of that. But DNS over TLS, and I'm going to clarify the distinction in a second. I did want to mention, as I already did at the top of the show, but to embellish a little bit, that as promised, GRC's DNS Benchmark received its first update in a decade because I added 1.1.1.1, 1.0.0.1, and 9.9.9.9. And in looking at the feedback that I had asked from our listeners, most people confirmed that 1.1.1.1 was fast. But my experience with 1.0.0.1 being a lot slower was not generally experienced by others. In fact, I would have to say I was an outlier in that regard, so you ought to discount what I said.

Again, nobody can be sure about the access to their own DNS without testing it. That's the only way to know. So thus GRC's DNS Benchmark, the only way to know. Many people reported that, to my surprise, that 1.1.1.1 was faster than their own ISP's DNS, not to the degree that the chart which was last week's Picture of the Week showed, which was bizarre, suggesting that ISPs were 68ms. I mean, 1.1.1.1 was 7ms, and the ISPs were 8ms, which is reasonable. But for what it's worth, 1.0.0.1 was right there in the running. And there were several reports of the older Level 3 DNS - remember 4.2.2.1, 4.2.2.2, 4.2.2.3, and 4.2.2.4 - that those guys were faster than 1.1.1.1. Again, the only thing that is consistent about DNS is that it is inconsistent, massively, from one user to the next. So the only way to know is to test it yourself. But I wanted to correct the record about 1.0.0.1 being so much slower. I appear to be an outlier there.

And, yes, I had forgotten about Quad9. And remember that this was the service, they're using the 9-dot network, which is IBM's space. And so this was put together by people who wanted to create a secure DNS to filter out bad guys so that, for example, phishing emails, when you clicked on the link in the phishing email, nothing would happen because that domain name would have already been taken out of service by your DNS provider if you were using 9.9.9.9. So if you want that, that's still a good deal. It didn't seem to have the performance, frankly, that Cloudflare's services did. But again, as I said, it's one thing for the Benchmark to say that this is faster than that. Hover the mouse over and left-click on the chart, and it'll show you numbers. And in fact the printout of the summary also shows you numbers. You get a complete numerical printout

with much more resolution to the numbers. So you can see if it's a difference that matters. But at least you have the actual numbers.

Okay. So DNS over TLS versus DNS over HTTPS. DNS over HTTPS is what I spoke of the last two weeks, the so-called "DoH," DoH protocol, which is the one that Mozilla has incorporated into v60 of Firefox, which is now the nightly Firefox and will be released on May 8th. And even then I think you still need to go into about:config and turn it on. But what that means is that the typical Firefox user, with a little bit of configuration - and maybe at some point I'm sure they'll surface it at the UI so you just click a checkbox and say yes, I want to use secure private DNS. That's over HTTP.

As far as I know, Cloudflare is the only provider of that version of secure DNS. I could not find an indication of it from Quad9 or Google DNS. But all three of them - Cloudflare, Quad9, and Google DNS - do support sort of the more standards-based original secure DNS over TLS. That's the one where it runs over port 853. And it's different enough that I think I'm going to give it a podcast soon. It's specified fully, that is, DNS over TLS is fully specified in an RFC 7858. And you have to play some games with the certificate validation because you don't have the same infrastructure in place that you automatically have with HTTPS and web browsers and domain names and so forth. So I'm probably going to do a podcast talking about DNS over TLS because it's clearly a solution that makes sense.

But the HTTPS version supported by Cloudflare is something people could play with now. Cloudflare has open source DNS over HTTPS clients for Linux, Mac, and Windows 32- and 64-bit. There's a page, if you just google DNS over HTTPS, that'll take you to - or maybe say "developers," add "developers" in there. That'll take you to the page where you can download a client for any of those platforms - Linux, Mac, or Windows. You install the client on your system. It sets up essentially a DNS proxy running on the localhost IP, so 127.0.0.1, and defaults to port 53, which is the DNS port. Or you can set it to a different port outside of the lower 1024 because those are reserved for root or privileged access, and so you would need to be able to install it as a privileged service in order to use that.

But the point is you then change your DNS to 127.0.0.1, and you're now using DNS over HTTPS to Cloudflare. So you don't have to wait, my point is, for Firefox to only be making that available for its browser, and not for other browsers, and not for your whole system. If you change your system, or even your router, to 127.0.0.1, then all of the processes in your system will look at the localhost proxy to get DNS. And speaking of routers, it is already built into the pfSense router from v2.3 on, and it will be surfaced at the GUI in 2.4.4. So it turns out even now it is a simple process to add this to pfSense, although I should mention pfSense is supporting DNS over TLS, which is the RFC standards-based solution.

So anyway, I wanted to spend a little more time, update where the Benchmark is, the fact that, yes, Quad9 supports it; Google DNS supports it. If you're a pfSense user, you can use DNS over TLS for your whole network in order to get security and privacy. And I'm sure we'll be talking about this in the future. And I didn't mean to abandon all of the Quad9 people. As you said, Leo, I'm just looking at something new and shiny. And actually it is faster. So if speed really is more important to you than the security that 9.9.9.9 offers, well, 1.1.1.1 does shave a few milliseconds off the queries, again, depending upon your actual performance.

Leo: Yeah. I already have malware protection in other ways. So I just like the idea of having encrypted DNS, that's all.

Steve: Yup, exactly.

Leo: If you don't need the other features of Quad9. Plus kind of I trust - I'll be honest with you. I don't know who Quad9 is run by. I mean, I know who it's run by, but I don't know who they are.

Steve: We know Cloudflare.

Leo: Yeah, know Cloudflare.

Steve: We know their heart's in the right place.

Leo: Nothing against the Quad9 guys, I just don't know who they are. I literally know the Cloudflare guys. I mean, they're sponsors, but John Graham-Cumming's an old friend and all that.

Steve: So.

Leo: Yes, sir.

Steve: This week Europe's electric transmission lobby announced that oven, microwave, and alarm clocks across the continent were finally no longer six minutes behind. What?

Leo: What?

Steve: So, yes. By resolving a grid dispute between Serbia and Kosovo...

Leo: Oh, my.

Steve: ...and running the continental grid at a slightly higher frequency than normal.

Leo: So it is, it is the frequency.

Steve: Yes. Now, it is the case that many systems still rely upon AC for their timing.

Leo: No.

Steve: Because it's more accurate than crystals.

Leo: If it's exactly 60Hz or whatever.

Steve: Well, and see, that's just it is you can count the cycles over a long period of time and adjust. And so what actually happens is that AC line frequency is not constant because, for example, in the summer during peak hours where there's a heavy load, the electric load actually slows down the generators and lowers the AC frequency because there's just too much demand.

Leo: So the frequency's directly tied to how fast the generator is spinning.

Steve: Yes, yes.

Leo: Wow, wow.

Steve: And so what happens is there's, like, really, really accurate time bases, like cesium beam because the decay of cesium atoms is well known. So there are absolutely really good time references. And so the grid, the cycles of current on the grid are counted. And at night the generators are sped up in order to catch up and make up for the slowdown that can occur during the day.

Leo: I didn't know that. So they know that clocks and other timing devices are doing this, so they actually compensate.

Steve: Yes, yes.

Leo: I'll be danged. I'll be danged.

Steve: And so I have a picture in the show notes of a quartz crystal's frequency variation as a function of temperature. And it's pretty good. But because a quartz crystal is itself mechanical, and it's using the natural harmonic oscillation in order to set the frequency, if you heat it up, it expands, gets bigger, and so it slows down. And when you make it colder, it shrinks, gets smaller, and speeds up. So quartz crystals, which are used for timing, they require temperature compensation if you want them to be accurate over the long term. The beauty of using AC is that, while there might be a short-term drift, for example, in the middle of a hot summer day, that gets made up for. So over the longer term, nothing, nothing is more accurate than counting zero crossings of AC.

Leo: So it's actually not an old-fashioned technology.

Steve: Right.

Leo: It's a legit one. I'll be darned. I'll be darned.

Steve: Yes. So get this. Well, when it all goes correctly.

Leo: Well, that's one of the things in Japan. I'm going to Japan, and some parts of the country are 50Hz, and some are 60. So you have to buy a clock for the right region, I guess.

Steve: Right, right.

Leo: Wow, wow.

Steve: So get this, Leo. Between mid-January and early March of this year, a grid dispute between Serbia and Kosovo resulted in 113 gigawatt hours of unmet demand from Kosovo, meaning more demand than there was power. Since Kosovo is part of the Continental Europe Power System, the unmet demand on this 25-country system pulled more power than was available, resulting in a grid-wide slowdown of the spinning generators, which were unable to keep up with demand. This in turn led to a, again, system-wide, a Europe-wide decline in frequency to an average, because Europe is on the 50Hz system, to an average of 49.996Hz. And this in turn caused AC time-based clocks, which were dutifully counting the zero crossings of the current and then dividing by 50 to get seconds, as a consequence of the over demand from Kosovo, time was passing too slowly. And over the span of three months, clocks throughout all of continental Europe lost six minutes. So they were running six minutes behind.

Leo: But, now, these are clocks in a closet somewhere. Most people would have said, oh, the clock's behind, and set it.

Steve: Well, yes. And in fact that's a very good point, Leo. So what happened is, last month the European Network of Transmission System Operators, ENTSO, publicly admonished Serbia and Kosovo for not properly balancing their grids according to previous agreements. The group wrote: "This average frequency deviation, that has never happened in any similar way throughout the Continental Europe Power System, must cease. ENTSO is urging European and national governments and policymakers to take swift action," you know, after three months.

Leo: Swift.

Steve: Yeah, swift. Two days later, on March 8th, the TSO, the Transmission System Operators from Serbia and Kosovo confirmed that they were back to balancing their grids appropriately. And, finally, last week, just last week, so it took another month, ENTSO announced that it had restored the lost six minutes to clocks around the continent by maintaining a slightly higher than normal average frequency of 50.01Hz deliberately for a month. In other words, somebody was counting and, oh, my god, it's been three months, and they knew how many cycles behind they were. So for a month they went fast and

put the lost cycles back into the power grid, and all the clocks came back.

Leo: Unless you'd set your clock.

Steve: Ah, exactly.

Leo: In which case you're fast now.

Steve: In my show notes I have here, ENTSO wrote: "One of the effects is notably that the digital clocks geared by electric frequency are now back on time - that is, as long as you oven-clock owners within the Continental Europe Power System did not change your slow clocks to the correct time a month ago. If you did, now you could be six minutes fast. But at least you're less likely to be late now." So I thank - Ars Technica picked up on that little fun tidbit, which I got a kick out of.

Leo: What a great story. I'm guessing that the reason they did that is because there are clocks, I'm thinking of radio stations and other facilities where they have, you know, they don't manually set the clocks.

Steve: Untended, yes.

Leo: Yeah. And those are probably more mission-critical than your oven clock. So your oven clock's probably not that accurate anyway, so you set it.

Steve: Well, and I'm sure you must have taken things apart the way I did as a kid.

Leo: Oh, yeah.

Steve: The old-style clock motor had this spinning disk on the back of it. And it was synchronized to AC. I mean, it was following AC. And if AC slowed down, so would it. And if AC sped up, so would it.

Leo: The cheap ones weren't smart enough to do averaging or anything. They just figured, well, we'll trust the power grid.

Steve: Well, and again, remember, I mean, over the long term, nothing is more accurate because there actually are people out there...

Leo: That's what surprises me. That's amazing.

Steve: ...counting cycles. They're saying, oh, we're 13 cycles behind after this grueling

afternoon. We'll put those lost cycles in tonight. Isn't that cool? I just love that.

Leo: That's really interesting, yeah.

Steve: So I did get a nice note I wanted to share. It's in the form of a question from J.D. Green on the 9th, which was yesterday. Actually, I guess it was a tweet because I have an @Cybts1. He said: "Hi Steve. Love hearing you on Security Now! each week. On last show you spoke about an email about SpinRite. I was wondering if SpinRite works on thumb drives. I have one that seems to be corrupted, and I would like to recover its contents. Thanks for all your work."

And of course he was talking about that neat note about the guy's mom who had her sequel to her book, her novel that was endangered because her computer died, even though it had been slowing down for a long time, and she probably should have taken a hint. And then remember that she had also a thumb drive that was non-readable when it came to reading it. And I don't know whether the guy who sent the email ever tried to recover the thumb drive.

But for J.D. Green and anyone else, yes, SpinRite works on thumb drives. We have in the past covered a number of listener stories and feedback where their thumb drives that had critical data on it, they just assumed it was solid-state, and it was ber-reliable, and it would not, you know, like they could count on it. And when it came time to read something from it, they couldn't. So the trick with thumb drives is that they need to be recognized by the BIOS when you boot the system into SpinRite. So the only thing you have to do, since BIOSes don't have plug-and-play capability to recognize that a thumb - typically they don't. There are some that do, but typically they don't. You need to have the thumb drive installed when you turn the system on. Then the BIOS will see it, SpinRite will see it, and you can run SpinRite on it.

And as we've seen, we don't know for sure, obviously because we don't know what the nature of the damage was or is to any specific drive, but these drives have the ability to have SpinRite recover their data, and we've covered many stories of that happening. In fact, it's sort of related, but Father Robert had an Android phone that he recovered this way, too, because the phone was able to look like a drive. He was able to put it in that mode, run SpinRite on it...

Leo: Oh, my god, that's awesome. I didn't know you could do that.

Steve: ...and recovered it.

Leo: That's great.

Steve: Yeah, he shared that story with us. So, yeah, very cool. When all else fails, as your last resort. Okay. So, finally, I wanted to talk about winding down the earliest versions of TLS. I talked about this, this was a blog post from DigiCert titled "Deprecating TLS 1.0 & 1.1." And I loved the line from the blog. The author, who's a communications guy who simplifies technology, said: "...but on the Internet there's a big difference between 'nearly dead' and 'dead.'" Which is exactly what we've seen. Time and time again we've seen that old technologies that are not proactively terminated, they're

zombies. They just live on forever.

And so, for example, we finally, of course, got rid of MD4, then MD5, then SHA-1. But, I mean, it took deliberate acts of, okay, we're absolutely no longer going to support this any longer, and then made that decision. So I thought his point was exactly on, which is there's a big difference between nearly dead and dead. It will live on forever unless the decision is made to terminate it. And that's getting ready to happen.

So where are we in the world right now? Well, first of all, we know that the majority version of TLS is 1.2. That is, it was finished in '08, remember, 10 years ago. So everyone's had plenty of time to get up to speed. Even so, it took Internet Explorer five years to get it supported from the time it was released. So there are some instances where people have been slow to adopt.

But as we talked about a couple weeks ago, 1.3 is finalized, so it's ready to go. It turns out that there's very low use of 1.1. 1.1 was just sort of - it was a fix for 1.0. But 1.2, which is an improvement over 1.1, has been around for a decade. So for various historical reasons, because SSL v3.0 and TLS v1.0 are virtually the same protocol, just sort of had a renaming, there's been some hangover of TLS 1.0, more so even than 1.1 because, if you were going to change, you would go to 1.2, which as I said was finished a decade ago. So there's a site, it's SSL Pulse. It's actually a service of SSL Labs. And Leo, you will find it fascinating. It's SSLLabs.com/ssl-pulse. And the guys at SSL Labs maintain a whole bunch of really interesting statistics about the state of SSL, well, actually it's now TLS, servers around the Internet.

Okay. So put some numbers on this, of the 150,000, so 150K HTTPS-enabled sites which this SSL Pulse service at SSL Labs monitors, 88% support TLS 1.0. Okay, 88% still support 1.0; 85% support 1.1. So again, slightly stronger adoption or support for 1.0 than even the newer 1.1. Whereas most of the Internet is actually using, that is, now remember, it's one thing for the servers to support all four versions - 1.0, 1.1, 1.2, and maybe 1.3. What actually gets used is what the client chooses, well, actually the client and server together.

Leo: There's a handshake; right? I mean, there's some sort of conversation.

Steve: Yes. Well, yeah. So support, we need to separate "support for" versus "what's being used." What's being used is 1.2 because the preference is to use the best, and servers are configured to use 1.2 if it's available from the client, 1.1 if it isn't, 1.0 if 1.1 isn't. So there is, you know, you're always trying to negotiate the most recent protocol that both endpoints support. So as a consequence, the fact that browsers have now, for at least five years, every major browser supported 1.2. That's now what everybody is using.

But as I mentioned, IE did not support, even though TLS 1.2 was released in '08, IE did not support 1.2 until five years later with the 2013 release of IE11, which was the first version to support 1.2. But that's five years ago, so now everybody's got that, too. Also, Android versions prior to 5, which was released in 2014, only supported TLS 1.0, which interestingly represents nearly 18% of Android devices still in use today. So a substantial portion of Android in use today is at prior to 5.0 and only supports TLS 1.0.

So Cloudflare, back in our discussion again, which as we know is one of the world's largest CDNs, Content Delivery Networks, has good visibility into what's going on at Internet scale. And they recently shared that about 11% of traffic on their network still

uses TLS 1.0, and a much smaller percentage, 0.38 using TLS 1.1. And of course the rest are up already at v1.2. But still, 11% using TLS 1.0 is a substantial bit.

However, deadlines are approaching. A major change being brought on by the PCI standard, the Payment Card Industry, is happening at the end of June. So exactly at the middle of this year, June 30th, 2018. The Payment Card Industry standard requires the deprecation of TLS 1.0. So any website that wants to maintain PCI compliance must stop supporting 1.0 at the middle of the year, June 30th, 2018. And as we know, most websites already support 1.2. So browsers, for the last five years, all major browsers have supported TLS 1.0, many for even longer than that.

So what's finally happening is that other companies have announced their plans to deprecate 1.0. DigiCert already did on April Fools' Day, April 1st, Sunday before last. They disabled support for TLS 1.0 and 1.1 for all their services including their website and API. Another CDN, the KeyCDN, will end support for TLS 1.0 and 1.1, oh, on March 30th. So they already did, too, two weeks ago, as will Cloud.gov. There's a service, Fastly, will stop support of 1.0 and 1.1 on May 8th. Cloudflare has announced its intentions to disable 1.0 and 1.1 on June 4th, so earlier in that same month that the PCI compliance requires that it be ended at the end of the month. And Microsoft Office 365 will support only TLS 1.2 starting next Halloween. So on October 31st support for the earlier versions of TLS ends. So we are seeing a formal winding down.

Now, one consequence, remember, there are still some 1.0 gotchas out there. The DigiCert blog posting noted that there are older, or just not yet updated, development tools which don't yet support 1.2, such as cURL, which are in wide use. And since GitHub was one of the first major services to turn off TLS 1.0 and 1.1, some things that had been working suddenly were broken by this. So GitHub did this in February two months ago, and that revealed some breakage in a number of developer tools which have since been updated. So as we know, sometimes it takes breaking it in order to force something to get fixed. It does look like now that we've got a good, strong, very secure standard, TLS 1.2, that was finished 10 years ago, that we're finally getting ready to say goodbye, forceably as is necessary, to 1.0 and 1.1.

Leo: Tada.

Steve: The deprecation of the earlier two, yup. And then eventually we will be moving to 1.3. Probably not fast, since 1.2 doesn't have any known problems.

Leo: It's pretty good; right? We don't really need - yeah.

Steve: Mostly it's the perfect forward secrecy and some improvement in handshaking, and so connection startup speed. But no known oh-my-god reason to abandon it. Remember that 1.0 was subject to the POODLE attacks and a few of the other problems, which have since been mitigated by browsers jumping through hoops in order to avoid that. So it'll just be good. It's necessary to finally just say no to these older protocols so we can take support for their workarounds out.

Leo: Can you take it out of your browser somehow, like disable it?

Steve: Yes. You can certainly do it at the server side, and there are options in the browser to have your browser no longer offer...

Leo: Say I don't want to accept 1.0 or 1.1.

Steve: Yup. That's the right thing to do.

Leo: So look in your browser config.

Steve: Yeah, just google "disable TLS 1.0 Firefox" or Chrome or whatever your browser of choice is.

Leo: Right.

Steve: And I know, for example, on IE, in the Internet Options there's a set of checkboxes for the various versions. You just turn them off.

Leo: It is conceivable that would keep you from getting on some ancient site.

Steve: It is conceivable, yes.

Leo: But highly unlikely, I think; right?

Steve: Yes. And I would argue, you know, maybe wait a few more months, like wait till there becomes a bit of a groundswell, and then any other consequences would end up surfacing. It's sort of the way I was able to, thanks to DigiCert's amazing service, to get them to make a special cert for me for SHA-1, which didn't expire until midnight, like the very last day I could possibly do it, in order to keep GRC available for people with really old browsers until I had no choice but to move to SHA-256, which I then did a day or two before. So it's the sort of thing where, unless there's a clear problem, you know, we've mitigated the known problems with TLS 1.0 and 1.1. But still, you just finally have to say, okay, no more of this nonsense. We're just going to stop supporting you. Everybody's had a decade to catch up. It's time.

Leo: And by the way, you probably should turn off SSL 3.0, if that's in your settings, as well.

Steve: Definitely. Even I don't have that one.

Leo: Even in your Windows XP? You do not use IE.

Steve: Even I don't have it.

Leo: You don't use IE on Windows XP, I'm sure, because I doubt that that has that as an option. I mean, and most websites, it's not the website, it's the server that they're using.

Steve: Correct.

Leo: And also most all websites are going to be updating their servers fairly regularly. I mean, that's not a hard thing. You don't have to redesign the site or anything, it's just built into the server.

Steve: Nope, nothing. It's all the low-level protocol interaction.

Leo: Very good, Steve. A couple of quick updates. One I've just been checking. According to Paul Thurrott, Microsoft decided not to push out the Spring Update or whatever the hell they're calling it today.

Steve: Spring's sprung.

Leo: Instead they pushed out an update to insiders. So I guess there were some more betas they want to do before they push it out. So it was supposed to be today, the Patch Tuesday for the month of April, but no. Also, for those of you who subscribe to the Security Now! feed, and we love it, and use Overcast.fm as your podcast app, for reasons unknown, earlier today you might have heard a little house music stuck in the Security Now! feed. As far as we could tell, this has nothing to do with us because it only happened if you're using Overcast. But we're working with Marco Arment, its author, to figure out what the heck? And I apologize if you were awoken during your morning commute by a little house music. That is, it's not on our feed. I checked the feed. It's not in the feed. So something happened, maybe an interaction between the feed and Marco's great software. We love it. We love Overcast.

Steve: Ah, those computers, Leo.

Leo: You know? You were talking about bugs earlier? The most difficult thing, of course, is an interaction with the outside world. And you never know what you're going to get from the outside world. You never know.

Steve: Hate that, when you want the computers to actually do real work for you and get something done.

Leo: I don't know what's going on. Poor Marco. He's debugging as we speak.

Steve: Cool.

Leo: Yup. Thank you, Steve Gibson. You'll find Steve and all of the goodness of the GRC Corporation at the website GRC.com. That includes, of course, SpinRite, the world's best hard drive recovery and maintenance utility. Not just hard drives. Android phones. SSD cards. All sorts of stuff. GRC.com. Buy SpinRite because that's his bread and butter. But you'll find plenty of free stuff there, including the world-famous ShieldsUP!. And SQRL is going to be there. And, oh, man, I can go on and on. Perfect Paper Passwords, Password Haystacks, all sorts of great security research information.

Oh, and maybe this show once in a while. You might want to check that out, too. You definitely won't get any house music if you go to GRC.com and search for a Security Now! episode. He offers 64Kb MP3 audio plus nice transcripts by Elaine Farris, so you can read along as you listen. Or search, and that's really the most useful thing. You can search through all the previous 658 episodes and find anything you want and jump right to it. GRC.com.

We have audio and video - video, who'd a thunk it - at our website, TWiT.tv/sn. If you want to watch the show created live, all you have to do is tune in on Tuesdays, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. The stream is at TWiT.tv/live. There's also an audio stream there, if you want to listen. And if you do either of those live things, please join us in the chatroom: irc.twit.tv. And of course it's always a good idea to subscribe. Use Overcast or Pocket Casts or iTunes, whatever it is you listen to podcasts with, and just make sure you get Security Now! each and every week, the minute it's available. Thank you, Steve. We'll see you next time, and have fun with "Lost in Space."

Steve: Oh, yes. "Lost in Space," Friday the 13th, everybody. We'll talk about it next week, I'm sure. Bye.