SECURITY NOW!

Transcript of Episode #555

WhatsApp

Description: Leo and I try to cover all of an insanely busy week's security events and news. A draft of the much-anticipated Burr-Feinstein encryption bill has appeared; news from the FBI on hacking iPhones; browser and Let's Encrypt news; several CCTV malware bits; a bunch of new ransomware; an amazing "You're Doing It Wrong"; and the result of my deep dive into the Open Whisper Systems "Signal" communications protocol that's finally been fully integrated into the world's #1 multiplatform messaging system, WhatsApp, along with two things that MUST be done to get true security.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-555.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-555-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Last week, WhatsApp flipped the switch for encryption for a billion users. But how good is WhatsApp encryption? Steve gives us his analysis, plus all the security news, next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 555, recorded April 12th, 2016: WhatsApp Encryption.

It's time for Security Now!, the show where we cover your security and privacy in technology. And nobody does a better job than Steve Gibson from GRC.com, the creator of SpinRite, sure. But many don't know he was the person who literally discovered spyware, named it, and wrote the first antispyware tool. He's been a security expert and covering security kind of on his own for some time, and of course for the last 10 years right here on Security Now!.

Steve Gibson: Thanks to you.

Leo: Yeah. I'm so glad, you know, one of our most popular shows. And the downloads are going up and up and up. And I guess really with today's climate I'm not surprised.

Steve: Yeah. I mean, it's a mixed blessing. We have a huge show. We're going to talk about the WhatsApp app and its underlying protocol, which has been renamed Signal, in order to simplify it because no one knows how to pronounce the name of the endangered

Axolotl healing newt.

Leo: Axolotl.

Steve: Exactly.

Leo: But it's a good name, A, because you can easily get the domain; but, B, because that's the newt that, as you said, self-healing when its tail gets bit off. It's a salamander or something.

Steve: It says, ah, not a problem.

Leo: No problem.

Steve: Just grow another one.

Leo: Yeah.

Steve: But the news did drop this morning, as expected about [gasp] Badlock, which turns out not to be.

Leo: Wasn't that bad. Oh, good.

Steve: We have a draft of the Burr-Feinstein bill that we've been talking about for several weeks and anticipating; news about the country of Hungary going even further than Burr and Feinstein; news on the iPhone FBI hack; some worrisome news about some innards of Firefox's extension-handling architecture. Let's Encrypt gets a huge new supporter with a gazillion domains. Two weird coincidences - coincidences?

Leo: Yeah, or coinkydinks, depends on...

Steve: Of CCTV malware.

Leo: Ooh.

Steve: A bunch of ransomware news. An amazing "you're doing it wrong" in the U.K. involving their power meters, which is shocking. And then we'll talk about WhatsApp. And there are two things that must be done, which are not default, if you actually want more than an illusion of security. But if you do those, oh my god, I mean, this is like the LastPass of communications. These guys absolutely nailed it. And it's also interesting that it has taken years.

Back in 2013 they were looking at the OTR protocol, the Off The Record protocol that we talked about once. And it's funny because, in digging deep into this, I remembered some things we had covered in the past, like there was this notion of repudiation where you would want to be able to claim that you never sent that message, if it was in your interest to do so. And you'll remember, Leo, that the crazy OTR protocol, like after the message authentication code had been expired, it deliberately published it, like publicly, in the protocol, in plaintext, just to sort of thumb its nose at anyone trying to prove that somebody sent something. They said, no, we're going to make it impossible to prove that by deliberately expiring authentication keys. But after they're no longer of any value for authenticating the current message, we'll publish the old ones so that no one can come along and say, you know, you actually sent this. Anyway, what is now called the Signal protocol, I'm just all, like, revved up because...

Leo: That's awesome, yeah.

Steve: ...I've spent the last few days digging into it. And they nailed it. I mean, it is just - it is amazing what they've accomplished. So that's the main topic, after we deal with an incredible amount of news of the week.

Leo: All right. Well, we'll get to all of that. And by the way, thank you for joining us on Sunday on TWiT to talk about Burr-Feinstein.

Steve: Oh, oh, and I keep meaning to say I was so impressed with your explanation at the end of The Tech Guy show on Sunday. I was watching it because of course that was leading into TWiT.

Leo: As your lead-in, yeah.

Steve: You gave the best, most perfect and correct answer about third-party cookies I've ever heard.

Leo: Oh, good.

Steve: I mean...

Leo: Well, I learned it all from you, Steve, so...

Steve: Well, I was holding my breath to think, come on, Leo, yeah, yeah, yeah. I mean, it was better than I could do because it cut out all of the extra stuff, and it was just - it was right for the audience. It was exactly correct in every detail. I was very proud.

Leo: Oh, thank you. Well, I'm very honored because that's, coming from you, extremely, extraordinarily high praise. And I try, I do actually think about you listening when I talk about this stuff and say, what would Steve say? What would Steve say? Yeah, somebody asked what about cookie deletion. And, you know, we talked about what cookies are, why they're not all bad, and what the real information leak risk is.

Steve: Oh, it was just perfect.

Leo: Yeah, yeah. It's a real privilege to get to do that radio show. Sometimes I think, oh, I want to stop doing it because it's - my weekend's shot. But it's such a privilege to get to talk to normal people and kind of be a conduit from people like you, all of our brilliant experts on the TWiT network, to kind of say to a million - it's a million people every weekend, a million people who are normal, relatively, people, not geeks.

Steve: More normal, yes.

Leo: More normal.

Steve: More normal. They are listening to that show, but still...

Leo: Well, they're a little geeky, anyway. But they're also, I figure, like the people who are going to then be asked by their friends and family, well, what does this mean? And so it's kind of a chance for us to kind of set the agenda, to talk about what things are and aren't, and to do so in a non-sensational way, in a way that - and I am lucky because I have the time to, I mean, I probably spent 10 minutes talking about cookies.

Steve: Oh, and, well, it was the tracking. You got the tracking exactly right.

Leo: Oh, good.

Steve: Which, you know, that's important.

Leo: Thank you, Steve. All right. I appreciate it. Okay, Steverino.

Steve: So I did want to pay homage to the 555 timer because this is Episode 555.

Leo: Yes.

Steve: I mentioned it briefly last week, and a number of people sent me some links to a blog. A guy named Ken Shirriff reverse-engineers classic ICs, classic Integrated Circuits. By that I mean he pops the top off of the lid and then looks at the photomicrograph and works out what the schematic was of the chip. And he's done that both for the later CMOS version of the 555, and for their previous, known as the bipolar transistor version. For anyone who's interested, I put the link at the bottom of the first page of the show notes. I really would recommend anyone who sort of has an interest in engineering, yeah, there's the page you're showing for the bipolar, the earlier version of the 555 timer.

And what was unique about it was that its operation, that is, the functionality it provided at the time, was very simple and very easy to understand. And it turned out you could use it as a building block for all kinds of, like, things that you wouldn't have thought of, not just for, like, creating a waveform at a certain frequency. But, I mean, you could hook it up to popsicles that you put in your mouth to measure the alkalinity of your saliva. I mean, just like bizarre things that were possible. And people just kept coming up with new ideas such that there were books, entire books written about things you could do with this one little chip. It was a little modest eight-pin DIP. It had power and ground and output, and then some little I/O lines that, again, were very simple in what it did.

But it was just this perfect little building block. And I used it like crazy. Lots of people even today are still using them. It still exists because it's just such a perfect little thing to drop in the right place. So for anyone who's interested in a little bit of electricity and electronics, this Ken guy, he's also reverse-engineered, I think it was the 741 that was THE op-amp of the day, and a few other circuits. And he does a beautiful walkthrough of the design decisions and how the part works and so forth. So anyway, I did want to note it on this 555th episode of Security Now!.

Leo: And you've mentioned this before; right? This was the question about timing; right? About how important timing was?

Steve: Yeah.

Leo: Or have you not mentioned this before? It feels like you have.

Steve: I don't know if I've mentioned this. We have talked about timing in other contexts.

Leo: Right, and why that was a significant computer technology.

Steve: Ah. Right. Good memory, Leo. We were talking about, in fact, that was a Q&A after we were talking about architectures, was why do all computers have a clock? That is, what is it about a clock? And the answer is it's about synchronization, the idea of things being done stepwise, like step by step, instruction by instruction, fetch and store. Everything is about what happens when. And so you need a clock in order to have a when.

Leo: I feel like we were also talking about this yesterday. We had James Gosling on.

Steve: Someone, actually a friend of the podcast, Simon Zerafa, said that James was talking about PDP-8s.

Leo: Oh, that was it. That was his, like you, his first computer.

Steve: Ah.

Leo: He cut his teeth on PDP-8s, and he was talking about the instruction set. I brought your name up. I mentioned that your goal was to retire and write an operating system for the PDP-8. We were talking about flipping the switches. You've got to watch it. I think you'd really appreciate it.

Steve: I definitely will.

Leo: And I mentioned the kit.

Steve: So I imagine it's on Monday's Triangulation.

Leo: Yeah, it was yesterday. And I mentioned the kit, you know, the - now, yours, are yours powered by Raspberry Pis?

Steve: Those are not. Those were actually - those three are from the 6100 chip, which was manufactured by Intersil. And it was a PDP-8 on a chip. So this minicomputer refused to die and was so popular that it ended up getting itself integrated on a single chip, the 6100. And Bob, whose last name is escaping me right now, who found a cache of these Intersil PDP-8s...

Leo: Right, that's right, yeah, yeah.

Steve: He said, "I'm going to give them some lights and switches." And so that's that. But the one you got...

Leo: The kit.

Steve: And the kit that I have, just a few months ago, that I haven't even opened, I got another three of them because I thought, you know, if one is good...

Leo: Why not, yeah.

Steve: But anyway, I haven't even opened the box, I'm ashamed to say, because I just, you know, I've got way too much going on. But I will get to it somebody. And that one is powered, that one is basically just an I/O panel for a Raspberry PI. So it's switches and lights that the Raspberry Pi reads and writes dynamically. And then running in software on the Raspberry Pi is a PDP-8 emulator, which exists in open source. There's something called the SIMH, S-I-M-H, project which has captured all of the classic machines in software emulation, for anyone who wants to play, and often still has the OS software and everything else to go along with them.

Leo: He was talking about loading the bootloader by flipping the switches in the front and how the really good guys could go, from memory, [vocalization], and they'd be ready to go.

Steve: Yup, because you're doing it all the time.

Leo: Yeah, yeah, you're doing it all the time.

Steve: In fact, the first thing you would check when the machine crashed or died was, did the bootloader survive? Because it was up at the very top of memory. And oftentimes it would just stay there, so that you'd be doing other things, and it'd be time to, like, load another paper tape. And it's like, okay. Is the bootloader still there? Or do I have to put it in again?

Leo: You would enjoy this. Of course Gosling the inventor of Java.

Steve: Yeah.

Leo: It was a great conversation.

Steve: And I want to make sure people understand that Java itself is not, like, the Java...

Leo: Right, it's not the problem.

Steve: ...is not the problem.

Leo: Right.

Steve: It was that it got stuck onto the web, like it got stuck into browsers as, oh, wouldn't this be neat. And it's like, no. This is not, I mean, it's the same problem with Flash. Flash, if it was just not used on the web, it would be fine. It would be a way of animating stuff and doing simple little scripted apps and stuff. But when you put something that powerful in a browser so that people you don't know can put their code in it, and it runs on your machine, what could possible go wrong? So, yeah. Java, I mean, I've got Java pretty much installed everywhere, but no browser plugins because it is a very nice cross-platform solution. And it's huge in corporations. Even, I mean, today it's not going away. It's a great, it's a powerful state-of-the-art language.

So the big reveal. We've been waiting for two weeks because this was announced two weeks ago as going to happen on April 12th. And of course it occurred to me, wait a minute, that's Tuesday. And Microsoft's involved, so that's probably tied into Patch Tuesday. And at 10:00 o'clock Pacific time, a little after 10:00, and I set up my web page change bot to watch for changes at 8:30 this morning because it was expected at 9:00 o'clock Pacific time, I think it was 17:00 UTC, and it was an hour later. And so I jumped on it the moment they updated the Badlock dot - I don't remember whether it was com or org. Anyway, that page. And it's like, yawn. As we hoped for the industry's sake, it was not a big deal.

They say there is a possible remote DoS on a publicly exposed SMB server. And as I have commented for the last two weeks, who has a publicly exposed SMB server? Now, we do know that a search through the Internet did find some wacky media server that there were several hundred of in the U.S., but a bazillion of over in Russia. So maybe that's a problem for them. But even users of corporate resources who are operating remotely are all doing so through their corporate VPN these days. And so that protects that completely.

There's also a potential man-in-the-middle vulnerability, meaning that somebody who could intercept an SMB connection - and they talk about ARP spoofing kind of interception, meaning you'd have to be in the network anyway, that not all of SMB protocol is equally well encrypted. And so there are some things to patch.

Now, Microsoft was yawning, too. Simon again found the Knowledge Base article from Microsoft corresponding to this problem. It almost looks like Pi - 314, it starts out, but then it kind of goes off of Pi to 8527, 3148527. And so Microsoft says: "This security update resolves a vulnerability in Microsoft Windows. The vulnerability could allow elevation of privilege if an attacker launches a man-in-the-middle attack. An attacker could then force a downgrade of the authentication level of the SAM and LSAD channels and impersonate an authenticated user. This security update" - wait for it - "is rated important for all supported editions."

Leo: Kind of important.

Steve: Important. So, you know, maybe tomorrow.

Leo: Not really. Just a little bit, yeah.

Steve: Yeah.

Leo: Sort of important.

Steve: So unfortunately these guys, they even acknowledged the Heartbleed folks and credited them with the style of prerelease hype that they adopted and tried to create a great logo because everyone thought, oh, Heartbleed had a great logo. So anyway, unfortunately, a great logo doesn't always mean a bad vulnerability. And in this case it's like, eh. Microsoft says, yeah, it's kind of important, you know. It's like, maybe. So anyway, Badlock, yeah, fix it. Patch when you want to. But it's difficult to imagine a scenario where any admin would be running around with their hair on fire after learning about it this morning. They couldn't have SMB exposed publicly. No one really can. And internally, well, you're trusting the people on your Intranet to - they're typically your employees, presumably. But certainly there are some scenarios where you could have an adversary in the connection.

So this is good to fix. I'm not saying it's nothing. But it's nothing like the last few major, you know, like Bash is bad in every version of Linux ever made because it turns out we're using that behind the scenes, and we didn't know it, or anything like Heartbleed. So, yeah, this is not on the same scale.

However, what is upsetting is, although it isn't law, and we need to remember that, is Burr-Feinstein.

Leo: It's not even - it's really just a discussion draft. And apparently another one has surfaced. According to Mother Jones, there's another discussion draft.

Steve: And do we know how it compares? I haven't heard that.

Leo: I have it here. I'll have to look. But go ahead. Why don't you talk about - and I will see if I can...

Steve: Yeah. So what the Burr-Feinstein legislation says is, unfortunately, exactly what I was predicting. And that is that, on demand, anyone who is encrypting something must provide it in plaintext, must have a means to provide it in plaintext. They're not legislating the means. They're not requiring a backdoor. They're not weakening encryption. And that's the most important point I want to get through this discussion is this isn't, per se, weakening encryption. But by being able to compel decryption, it certainly does weaken privacy.

So we need to, I think, just from a technical standpoint for this podcast's sake, and its listeners, I want to draw a distinction between this notion of encryption technology and encryption policy. So what this is addressing is encryption policy. And it's saying that anyone who is encrypting - and that's the breathtaking part of this, too. We discussed this on Sunday's TWiT. I meant to have the PDF up so I could quote from it again as I did on Sunday. Yeah, here it is, I found it. Because in the discussion draft, as it's called, they talk about the "covered entity." The covered entity is someone, for example, in the position of Apple, who is providing a smartphone technology and device and communications infrastructure which provides encryption.

So this is just breathtaking in its sweep. They define it formally as the term "covered entity" means a device manufacturer, a software manufacturer, an electronic communications service, a remote computing service, a provider of wire or electronic communications service, a provider of a remote computing service, or any person who provides a product or method to facilitate a communication or the processing or storage of data. So that last phrase, "any person who provides a product or method." I mean, that's the world. I mean, that's everybody.

And then the other part that we highlighted on Sunday, which is sort of stomachchurning, is they describe this as, like the main line, the first line, "to require the provision of data in an intelligible format to a government pursuant to a court order and for other purposes." So this is under court order, a government. And I was like, what do they mean, "a government"? Like this is a U.S. law, so it's only going to pertain to U.S. entities.

Leo: But it could be a county or city government.

Steve: That's exactly right. They define - and this is just horrifying. They define, for the purpose of this document, the term "government" means - and I'm reading from the draft - "the government of the United States and the government of the District of Columbia or any commonwealth, territory, or position" - oh, I'm sorry - "or possession of the United States or an Indian tribe or of any state or political subdivision thereof." Meaning Andy in Mayberry can say, hey, we want you to decrypt this phone. I mean, the mayor, who's a subdivision of the county, that's a subdivision of the state, can say, oh, yeah, we want you to decrypt this for us. So anyway, that's where we are today. Again, not law, just discussion.

The last thing I heard, Leo, was that something had been sent back to the executive branch for further examination. What we heard a couple weeks ago was that the President was not going to weigh in on this either way, although of course we did hear that disturbing rhetoric from SXSW, where he talked about us fetishizing - I can't say it.

Leo: Fetishizing.

Steve: Thank you.

Leo: Fetishizing.

Steve: Fetishizing.

Leo: When you say it that way, it really sounds terrible.

Steve: Wow, yeah.

Leo: Do not fetishize your phone.

Steve: Yes. And in fact what he's saying is we're fetishizing - okay, I can't say it. We fetishize...

Leo: We're doing that thing to the phone again.

Steve: Our encryption is what we're fetishizing.

Leo: Yeah, fetishizing. We're fetishizing privacy, frankly. I mean, that's the logical conclusion.

Steve: Well, and that - yes, yes. And so that's the thing I - if our audience gets anything, it's just I just want to draw the distinction. This is compelling a breach of privacy. Now, the ivory tower academics go crazy, saying there's no way to do that without weakening security. Well, tautologically that's true. But in fact, as the WhatsApp system is going to demonstrate later in this podcast, which is just - it is a master work of crypto technology. It is absolutely possible, if people want to design a system that securely provides that facility, for it to be done. People are saying no, no, no, it can't be done. Sorry, it can be done.

But the problem is we get - that's a distraction. I think we want to stay focused on the policy question, at what level and to what degree are we as a society, in the U.S. at least, going to decide where this lands? To what degree do we want a government order to allow our privacy to be breached, to be opened, to be decrypted, however you want to put it. That's where we should focus our attention because the technology can be done, if people want it to be done. And so it will be interesting to see how this goes. We don't know what Obama is going to say.

Leo: And they haven't even proposed it as a bill.

Steve: Right, right, right, right. And now of course we've got a wacky election for the President. I don't know where this even falls along partisan lines. To me it doesn't feel like a partisan issue, Republicans versus Democrats. It feels way bigger than party and any kind of ideology. This is, I mean, it should be bigger. Were you able to find anything new?

Leo: No, this Mother Jones article is just - it's not even showing the text, just saying - it's just kind of pointing out that the staff - it says: "Senators Dianne Feinstein and Richard Burr apparently have very unreliable staff, as yet another discussion draft of the national security bill they're jointly sponsoring has been leaked to the press. They really need to tighten up their operation." But they don't - it doesn't - it looks, I mean, it's just another draft of the same thing, I guess.

Steve: Yeah. And so what we see is the intent. What this shows is the intent, which is - and that's - we've seen it in the rhetoric over on the law enforcement side, where they march out terrorism and child pornography and all of that, and say how can it be that Apple is encouraging terrorists to use their encryption technology? It's like, oh, lord. Okay. Unfortunately for their argument, so are all of the law-abiding citizens who would like the benefit of this technology for our privacy. So anyway, I do have an acronym, or an abbreviation, to go with this. Of course TNO has now been - is legend for Trust No One. It occurred to me that the proper abbreviation for this, the whole discussion that sums it up is DOD, Decryption On Demand.

Leo: Ah, I like it.

Steve: Because that's what we're talking about, the idea that, yes, the Internet is going dark. And you guys had a great discussion after I left the conversation on Sunday's TWiT about the metadata because it is the case that law enforcement is drowning in surveillance technology. I mean, there's more surveillance capability today than there has ever been. I don't know about where you guys live, but any street corner here in Southern California has four cameras mounted on every streetlight pointing in all directions. And, I mean, and lots of networking going on, and all kinds of surveillance.

So anyway, DOD, Decryption On Demand, that's really what this comes down to is at what level does a need need to rise to in order to force a company to decrypt, if there is any. Maybe we decide collectively there is no level, that privacy should be absolute. Again, I'm going from the Constitution, which does not guarantee absolute privacy. It just guarantees reasonable privacy against search and seizure and does use the court system to provide search warrants that allow law enforcement to breach someone's privacy. It seems to be that, again, that's - I'm not an attorney or a constitutional scholar. But I'll be surprised if we don't end up with something like that. And the good news is people are going to go kicking and screaming. But I think we don't have to have a larger loss of security and encryption and privacy than what the law decides we're going to end up with.

But this is the U.S. Hungary's government has gone crazy. MappingMediaFreedom.org had an article - which is a good thing because I can't read Hungarian - and they linked to it. And I thought, well, maybe it's English because sometimes you get lucky. It's like, no. I don't even know what Google Translate would do with this. Anyway, the article in Mapping Media Freedom, the headline was "Hungary: Government plans to criminalize the use of encrypted services."

So in their translation of the Hungarian news, they wrote: "The Hungarian government plans to criminalize the use of applications for encrypted communication. The measure is part of a new anti-terrorism legislation package put forward by the Interior Ministry and was first presented on the 31st of March by Janos Lazar, the Minister heading the Prime Minister's Office. If the package is implemented in its present form" - again, pending legislation, so this isn't law yet, but, again, shows intent - "anyone caught using encrypted software can be punished by two years in prison. The providers would be obliged to ensure access to the content of the encrypted messages, and they would have to provide the identification data of the users as well as the IP address used for registration. Failure to comply qualifies as a misdemeanor and is also punishable with a two-year prison sentence. The anti-terrorism package also contains provisions regarding an increase of surveillance in public spaces and enables the Interior Ministry to prohibit mass events." Wow.

Leo: Yeah, you always get these reactions when there's, you know, because of the refugee crisis in Hungary. You always get these kinds of overreactions.

Steve: Yeah. Yeah.

Leo: It's not a law at this point.

Steve: Yeah, right. But it does, it says, okay, we're just going to outlaw encryption.

Leo: Yeah, crazy.

Steve: Can't use it in this country.

Leo: Crazy. I could see that happening here.

Steve: Yeah. So there's been additional news coming directly from the mouth of my favorite person.

Leo: Me?

Steve: FBI Director James Comey.

Leo: Oh, James Comey.

Steve: And longtime listeners will remember that I had us play his blatant lie to Congress into the podcast a few months before Snowden revealed it to be exactly that. When he was directly asked by a senator on the Intelligence Committee, who had sent the list of questions that he would be asked days before, and his staff had vetted them, and he was fully prepared. And he was scratching his head and says, "No, Senator, we are not performing any mass data collection on U.S. citizens, not wittingly." Ugh. Anyway, yes. He said last Wednesday...

Leo: It was Clapper, not Comey, I think.

Steve: Oh, wait. You're right.

Leo: We've got enough nitwits in the intelligence community...

Steve: Sorry, sorry, sorry. Clapper, James Clapper.

Leo: ...that it's easy to confuse General Clapper.

Steve: Oh, yes. Thank you, Leo.

Leo: Clapper, Comey...

Steve: Everything I just said is not this guy.

Leo: No, Comey's said equally stupid things. But he...

Steve: Yeah, he's on the same team.

Leo: Yeah, he's on the same team, yeah.

Steve: So he said that the government had purchased, quote, "a tool" from a private party in order to unlock the iPhone used by, of course, as we know, one of the San Bernardino shooters. Quoting Comey, he said: "The people we bought this from, I know a fair amount about them, and I have a high degree of confidence that they are very good at protecting it, and their motivations align with ours."

Leo: I hope they're better than the FBI is.

Steve: Well, yeah. And it seems to me their motivations are commercial.

Leo: Yeah. They're not aligned at all.

Steve: Exactly. And the FBI's are law enforcement.

Leo: Yeah.

Steve: And they sold this tool, for commercial benefit, to the FBI. And if anybody else wants a copy, here's the price. It probably has six digits. But there's nothing aligned about it.

Leo: Good point.

Steve: So Comey also said that the purchased tool could only be used on, quote, "a narrow slice of phones" that does not include the newest Apple models or the 5s. So not the 5s or the 6. Comey said the government was currently considering whether to tell Apple how it pulled off the hack. He said, quote: "We tell Apple, then they're going to fix it. Then we're back where we started from. We may end up there. We just haven't decided yet." Well, isn't it nice to have all the cards in your hand. Anyway, so while that doesn't exactly confirm how the hack worked, some of the reporting wrote that the distinction being drawn here may suggest that it's specifically the lack of the Secure Enclave on the iPhone 5c's...

Leo: That makes sense.

Steve: ...A6, yes, the A6 system on a chip that renders the phone vulnerable. And then of course we got the Secure Enclave with the A7 SoC appearing in the 5s and subsequent phones, which does make sense. I did see just this morning and didn't have a chance, it was on the right-hand column of the Hacker News, that a company was claiming they were closing in on a hack for the 6. But closing in is way different than having it. So you either, you know, you don't have it until you do. So it's not clear what "closing in" means, except for their marketing.

Some disturbing news about my favorite browser. A beautiful piece of reverse-engineering was done on the architecture of the Firefox add-in ecosystem. And the chilling phrase, when I read it, I said, "Oh, no, no, no, no." The chilling phrase is that all JavaScript extensions installed on a system share the same JavaScript namespace. And it's like, no, no, no, no, no. That just can't - that can't be possible. What that means is that it's like all the extensions are in one JavaScript file and coexist, meaning they can all see all of the variables and routines that they each have.

Leo: What? That's not good.

Steve: Yes. That's horrible. And, now, the JavaScript namespace has always been a problem. It's one of these things where JavaScript has outgrown itself so that there's, for example, there's a book called "Closure" which is one of the techniques. There's a way that you can wrap mature JavaScript applications with many different methods inside one big method that contains them all in order specifically to create a hidden namespace. But unless you do that, and unless you're really careful, this is a problem. I mean, it is, for example, it's a problem when JavaScript that is coming from different places stomps on each other's verbs and method names and so forth, and procedures. It is a problem with JavaScript that it doesn't have clearly divided namespaces. So the exploit is known as CrossFire. And, for example, NoScript numbers among the top 10 most popular Firefox add-ons, and nine of the top 10, including NoScript, are vulnerable to this kind of exploitation.

Quoting from this whitepaper, and I've got links to it in the show notes, the researcher wrote: "Despite the abundance of research focusing on the security of browser extensions in isolation, to the best of our knowledge the possible interactions between multiple browser extensions have not been well-defined from a security perspective. In particular, the Firefox extension architecture allows all JavaScript extensions installed on a system to share the same JavaScript namespace, hence making it possible for an extension to invoke the functionality or modify the state of others. This problem has long been recognized as a namespace pollution problem that can introduce errors if multiple extensions define identical global names. However, its impact on security has not been studied so far." So these guys did that.

Now, of the top 10 most popular extensions, number one is Adblock Plus. And they could find no problem there. But Video Download Helper they found 13 different problems; Firebug, one; NoScript, seven; DownThemAll!, 19; Greasemonkey, 20; Web of Trust with the maximum at 34. Flash Video Downloader had five; FlashGot Mass Downloader had eight; and Download YouTube Videos had two. So those are the top 10; nine of those had one or more, in some cases 34, different problems. The Mozilla people responded, and they said [clearing throat], "Yeah, we're..."

Leo: Yeah. Yeah.

Steve: "This is a problem."

Leo: Yeah, we know about it, yeah.

Steve: And as we talked about a couple weeks ago, Firefox wasn't attacked during the most recent competition of...

Leo: Pwn2Own, yeah.

Steve: Yeah, exactly, Pwn2Own, because it's like no - it's regarded as a soft target. Now, for this to be exploited, you would - and what they did, they have some proof of concept. You need to get a malicious extension into the browser. Mozilla has both automated and human extension verification, extension auditing. So they have an automated process that looks for API usage. And then somebody reads through it. Well, we know that both are prone to failure. We've discussed the failure modes of both of those problems at various times in the last couple years.

So what happens is, because it's possible for the malicious extension to maliciously commandeer functionality in another extension, the malicious extension itself doesn't have to have the obvious ability to do anything wrong because unfortunately what these people are not checking for is cross-extension access, that is, this kind of cross-extension accessibility, due unfortunately to this common JavaScript namespace collision.

Leo: And you probably remember this about seven years ago, that the NoScript guy used this to modify Adblock Plus to whitelist NoScript in Adblock Plus.

Steve: Right.

Leo: Do you remember that? I don't know if you remember that. There's a whole apology on his blog. "I'm sorry. I shouldn't have done this. I will regret it forever." This was seven years ago.

Steve: Yeah, Giorgio.

Leo: Yeah. And which may explain why Adblock Plus is on the list as having no exploitability. I have a feeling they hardened themselves against this; right?

Steve: Exactly. They put themselves - they wrapped themselves up so nobody could have access to their stuff from the outside. I bet you're right, Leo. I bet that's why they are the one exception is they, like, okay, no.

Leo: No.

Steve: I have to say it can be a real nightmare. Like I said, there's a book on "Closure" which is, I mean, the main way this is done in JavaScript because it's like, it makes LISP look easy by comparison. Actually, there are some similarities.

Leo: Yeah, Closure's from LISP. That's a LISP technique, widely used LISP technique, yeah.

Steve: So anyway, I don't think this is, I mean, I'm not leaving Firefox. I'm looking for - I hope there are the resources available to rearchitect it. As we said a couple weeks ago when we were talking about Pwn2Own, they really do need - it's time to say, okay, time to restart. It's because Microsoft, yeah, Microsoft bit the bullet and abandoned their IE codebase and started over, like from scratch. I mean, imagine if you could do that now, knowing today what we know, versus still having IE6 code lingering in IE11, and it causing problems. It would be current standards. It would do everything right. You would have a team trained up on security. I mean, anyone writing a browser today, security is number one. I mean, I would put that behind standards compliance. Make it secure first; and then, yes, we have standards now. So implement as much of that as you can. But, boy, it's got to be secure.

And as we've learned in that Pwn2Own competition, Edge, the new browser from Microsoft, it won. It had the fewest problems. And specifically because Microsoft did bite the bullet and just say, okay, we may have been second, because the Mozilla browser was first, and then IE was second, I think. No, there were some text browsers, too, before that, too. But still, in terms of browsers that are still around, it was just time to say goodbye.

Coincidentally, yesterday, we did get an update to Firefox, 45.0.2, up from 0.1, although it didn't do anything about this, and it wasn't a huge change. It was just an incremental update.

Leo: Some might say this is the brain damage that came from C and has been propagated through these kinds of procedural languages ever since.

Steve: Yeah. And I also think it's just there wasn't the focus that there originally...

Leo: But namespaces, no, you know, computer science has understood the necessity of separating namespaces for a long time. It's pretty fundamental. You don't want to clobber somebody else's variable because you use the same name.

Steve: Yeah. I don't think this is about C, though. I think this is about JavaScript being simple. The goal was let's just create a simple scripting language to sort of automate, to do some stuff on the client side. And what's happening is it really just - it's become an incredibly powerful tool. And so I think it outgrew, I mean, this is the problem. And you guys, for example, have talked about this with iOS. I completely agree with you that the iOS UI that started out as a simple, obvious, easy-to-use thing is becoming cumbersome, with all kinds of hidden behavior, because they keep cramming more stuff into a framework that wasn't designed to hold it. Similarly, JavaScript has outgrown its roots, and it just hasn't done it gracefully. Now, Leo...

Leo: Yes.

Steve: Need to go to steve.grc.com.

Leo: Wow, you're getting fancy with the subdomains here.

Steve: Steve.grc.com.

Leo: Have you ever used a subdomain before? That's a new one on me. Oh, you've got a blog.

Steve: Well, okay. So, and it's a WordPress blog.

Leo: Nice.

Steve: Now...

Leo: HTTPS, baby.

Steve: Yes, baby.

Leo: That's nice. With Let's Encrypt; right?

Steve: Yes, and you can look at the security certificate, and you will see that it is a Let's Encrypt certificate.

Leo: All right. Let me look at the details here. Valid certificate, server certificate. Let me view it. I like it because they give it a nice little gold stamp, very beautiful, from tls.automatic.com. Of course that's the parent company. Let's Encrypt Authority X3. Nice, Steverino.

Steve: Yeah. So with this move they began a couple months ago, actually in January. And as of yesterday, or I'm sorry, no, as of the 8th, 4/8, April 8th, they announced that it was everywhere on the mass number of domains. All the people with custom domains, anything hosted by WordPress.com now has HTTPS. So all of the logon tokens, the session cookies, all of the - well, the works. And so it's just a major nice big step in terms of total domain coverage for Let's Encrypt is all of the WordPress blogs.

Leo: Huge, huge. Fantastic.

Steve: And I did nothing. Nobody did anything. It just all just like, bing, okay, now you have security. And so you could use https:// instead of http to refer to your blog at WordPress. And it works. And you've got security.

Leo: Woohoo.

Steve: So, yay, nice, nice move forward. This is the way it should be.

Leo: Yeah, no kidding.

Steve: Okay. So another beautiful piece of reverse engineering. A researcher, Rotem Kerner, he wrote on his blog: "Since there are so many vendors who redistribute this hardware/software, it is hard to rely on a vendor's patch to arrive at your doorstep. I believe there are more vulnerabilities being exploited in the wild against these machines. And therefore your best shot would probably be to deny any connection from an unknown IP to the DVR services. And so I will leave you here with a list of vendors who are selling some of the CCTV DVR rebranded gear, a company called TVT."

What this guy found was he was setting up a system for a friend, and it was a multicamera CCTV DVR security system. And when he brought up its web interface, because it had a web server in it, he didn't - he was like, it seemed to be missing some of the controls that he was expecting to see. So he's a little on the techie side, to say the least. So he looked at - he did a view source, suspecting maybe that there was some mangled-up CSS that was preventing the display of the control UI that he was expecting to see. What he found at the bottom of the page was - I'm trying to think. I think I'm confusing two stories here, I'm sorry, the next story, because we have two stories that are very similar.

This first guy found some malicious code in the firmware of this commercial DVR system which enabled remote code execution. And this wasn't obviously malicious, but it was at least a mistake. And he did a search on the 'Net because this device opens port 81 and also 8000. And he found, using Shodan, more than 30,000 of these systems exposed publicly. So thus his advice to limit the exposure with a firewall to known IPs. So there are 30,000 of these commercial CCTV DVRs. And looking at them, I mean, he has an example of looking out the camera of some store, which is completely visible now thanks to this vulnerability.

What it turns out is that there are so many of these because the 70 different brands are all the same device. There is a single Chinese company, this TVT, that produces a white-labeled generic solution, and 70 different companies slap their own name on it and resell it under their own brand. They are all the same device, and they all have the same firmware, and thus the same vulnerability. So anyway, there's really no takeaway for our listeners except that anyone who is using these systems, where it's like, oh, yeah, we have username and password. Logon anywhere from the Internet, and you can get access. Uh, that's probably insufficient security. And we're talking about, of course, a major privacy breach. We're talking about something potentially publishing, not only the current image, but the past, because these things record, of all the cameras that it has hooked up to it.

The argument has been made in some of the coverage of this that this is now the way criminals are casing the organizations and retailers that they want to attack before, I mean in the physical world, that they want to break into is they just sit there and monitor that company's CCTVs to look at protocols and procedures and how things operate and where the manager puts the keys when he's through using them and so forth. So it's just chilling that there could be this many systems, all from one company, all with a remote code execution vulnerability. In his walkthrough - it's a nice piece of work - he goes through the process of establishing root with full remote terminal access and the ability then to download his own code package and execute it. So basically complete takeover of the device. And he said they are typically BusyBox based, and this family of CCTV DVRs was BusyBox based. So he was right at home hacking into it.

The other story is the one where someone was setting up a system for a friend which he purchased from Amazon. And so the headlines were "Beware: Even things on Amazon come with embedded malware." Now, the link to the Amazon device shows "product no longer available." So Amazon quickly said whoops and took it off the market. I was curious to see what it would cost because these things are not inexpensive. But so he wrote: "I needed a simple set of good outdoor surveillance cameras for a friend's home. Like everything else I buy, I turned to Amazon. I found what seemed like a great deal for a set of six POE" - that's Power Over Ethernet - "cameras and the necessary recording equipment. Here's the link." And then there's the link.

He says: "When trying to get the cameras to work on my friend's machine, I simply logged into the admin web page and went to configure it. First of all, something seemed a bit off. The interface showed the camera feed, but none of the normal controls or settings were available. Being one of those guys who assumes bad CSS, I went ahead and opened up developer tools. Maybe a bad style sheet or style was hiding the options I needed. Instead, what I found tucked at the bottom of the body tag was an iFrame linking to a very strange-looking host name, www.brenz.pl, a well-known malware distribution domain."

So essentially this brand new, purchased by clicking a link from Amazon - of course this is not Amazon's fault. This is a CCTV system, and we just got through talking about 70 of them that have a different problem. This one, for whatever reason, how it got in there we don't know, but there is just an innocuous-looking iFrame which of course is embedding a Fetch of whatever this Brenz.pl wants to place in the frame. Very much in the same way that malicious advertising gets onto someone's browser, here it's whatever this malicious malware site chooses to stick into your CCTV viewing or admin webpage.

So, again, the retail channel is seeing these kinds of problems. And unfortunately I think we're going to need to remain vigilant about all IoT stuff. The problem with these is they're inherently connected. They are on the 'Net to do their job. Many of the IoT systems are, too. And it creates an exposure. We just got through getting firewalls up in front of our PCs after we finally figured out we were not capable of adequately securing them to allow them to operate without a firewall. We know that. Everyone knows that. And let's just go further and put a NAT router in front of our whole network because lord knows what's going on within our network. We need that security. Yet here we are saying, oh, but, wow, look what I can do. I can, you know, I can change my home temperature from the movie theater before I get home. It's like, yes, because something outside your home is able to reach into your home.

Leo: And so can someone else.

Steve: Yeah, exactly. So there was initially bad news about a new type of ransomware called Petya. And that's what it calls itself, P-E-T-Y-A. And it came into the news because what it does, it operates different than other ransomware, where we've covered - everybody knows what things like CryptoLocker do. They encrypt a subset of your files, a painful subset, a deliberately painful subset, but a noncritical for operation subset. That is, your docs, your spreadsheets, your PDFs, all the things you care about that you have probably created or you've downloaded, but the OS still runs. And of course the point of that is you need your OS in order to contact the bad guys to send them Bitcoin to get the key to do the decryption.

Well, the Petya author or authors decided to take a different tack. First of all, rather than encrypting the files, they encrypt what's known in the NTFS file system as the MFT, the Master File Table. Well, the MFT, it's clever, actually, because it's all the files in the partition, in the volume. The filenames; the timestamps; the stream names; the list of cluster numbers where the data streams, that is, the files, reside; their indexes; their security identifiers; the file attributes - read-only, compressed, encrypted and so forth. Basically it is the MFT, you know, it's like in the old days where we talked about the file allocation table, the FAT, and the directory structure. The file allocation table knew how the clusters were linked together. And the directory structure pointed to the first cluster of the file.

This is the equivalent, rewritten and updated dramatically, for NTFS file systems, which all Windows systems are based on today. If you encrypt the MFT, you don't have to encrypt the files. On the other hand, the system is dead. It's gone. Nothing will run. And so this thing also, it replaces the master boot record with a big, scary red-and-white skull-and-crossbones painted in dollar signs on the screen.

Leo: Oh, that's good.

Steve: Yeah. And so when you get it, it does this and crashes your system, forcing a reboot. And when you reboot, your system dutifully loads the master boot record and displays the skull-and-crossbones with the explanation that your entire system is now hosed, and it is.

Leo: But the data's there, it's just your master boot record's hosed.

Steve: Well, but, no.

Leo: Or your MFT.

Steve: The MFT, right, the MFT's encrypted, and the MBR is also encrypted/hosed.

Leo: But the data is still on the drive unencrypted.

Steve: Yeah, well, the data is there. On the other hand, it's difficult to get to. It wouldn't be impossible without the master file table. But it would still be difficult.

Leo: You'd need a scanner that would go and look at every sector.

Steve: Right. And remember we no longer defrag our drives. And they actually do get fragmented over time. So although NTFS is a more mature fragmentation minimizing system, it doesn't eliminate it at all. So a venerable drive that's been upgraded from 7.0 to 8.0 to 8.1, now to 10, and so it's been around for a long time, and that's an old NTFS partition. Oh, boy, I mean, you'd have a hard time piecing that thing back together again.

So they made a mistake. A few days after this all hit, this news hit on around - actually on April Fool's Day. Now, just today, it turned out that the security community has been looking at it and reverse-engineering it closely, and the author made a mistake. So if you or anyone you know and care about is hit with this, you do not need to pay ransom. The mistake leaves some clues about the key behind. There's now a free utility you can download which will go snag - the first guy that came up with this had a page where he'd worked out the algorithm, but you needed to be doing some deep voodoo like direct disk accessing to go get this sector and this offset to get this byte stream. And so I was like, well, okay. You need somebody maybe above your pay grade in order to find the data that you dropped into this web page, which then had the algorithm that gave you back the key.

Now there is a piece of freeware that you just download and click, and it gets it and presents it. And then you copy and paste it into the web page. It gets back your key, and you're able - it's just like the key that you'd have gotten from the bad guys, and you're all back and happy again. So anyway, lots of coverage about Petya, if anyone's curious for more. Got all the links in the show notes. But basically - and the problem is obviously these people are going to fix that and solve their mistake. So this is a short-lived window of opportunity. You basically just don't want to get hit with ransomware because it's not good. He wasn't being greedy, either. It was 0.8 bitcoin, which is, what, maybe 300-and-some dollars? So it's like, yeah, okay. So, I mean, not that any is good.

Speaking of which, we have some statistics on ransomware. Just last year, about $24 million was paid to ransomware criminals. That is, these guys, just in 2015, made $24 million. And it's pretty much all profit because they have a low-cost delivery infrastructure, unfortunately. Reuters carried a story saying: "The U.S. Department of Justice and Homeland Security last week provided new insights into the impact of ransomware and cyberattacks on public institutions and the public. The DOJ revealed that the Internet Crime Complaint Center, the IC3, had received nearly 7,700 public complaints" - for what good it does to complain to the IC3, you know, it's too late. Anyway, 7,700 public complaints regarding ransomware since 2005, okay, so that's going back 11 years. So going back 11 years totaling $57.6 million in damages. Whereas just in 2015 alone, victims paid over 24 million across nearly 2,500 cases.

So what we're seeing is - so it was 57.6 in 11 years, but almost half of that just in the one previous year. So it's on the rise. And of course for a while we were just talking about, what was it, CryptoLocker. Now it's the ransomware of the day, unfortunately, because it pays. And also unfortunately we were right in predicting that this was what was going to happen because the moment this appeared we said, "Uh-oh, this is really bad," because these guys are going to make a lot of money, and therefore we're going to get more of it.

And speaking of getting more of it, and of ransomware, Adobe just rushed out an emergency Flash update because it turns out that there was a zero-day Flash exploit, apparently only affecting Windows 10's browser's Flash plugin, in a way that was being abused by these ransomware guys. Sophos wrote: "The bug allows an attacker to send booby-trapped content to Windows 10's browser's Flash plugin in such a way that the browser will not only crash, but also hand over control to the attacker in the process." Adobe claims that the latest in-the-wild exploits were only targeting Windows 10 users. It would be wise for Flash fans - isn't that an oxymoron?

Leo: Who's a fan of Flash?

Steve: Exactly, to update the software immediately. Or just remove it.

Leo: Get rid of it.

Steve: Yes, please.

Leo: And why is anybody making a browser that doesn't automatically update Flash for you?

Steve: Right.

Leo: Edge does, I'm sure, so they must be talking about the plugin for IE11 or something.

Steve: Well, no. What it was, was this was before it was known. So Windows 10, it was only targeting Windows 10 for whatever reason, maybe other aspects of the environment; and it was a zero day. So this was found in the wild, and Adobe then did immediately update and no doubt provide the patches to Microsoft to push out through their own browser update mechanism, and presumably Chrome and so forth, too.

Okay. So this is crazy. Okay. These people are not listening to this podcast. So this is our, "Seriously? Boy, are you doing it wrong." In this case, this is an instance where the "spooks," as The Inquirer called them over in the U.K., that is, you know, their intelligence services, the GCHQ, intervened to prevent a catastrophically insecure U.K. smart meter plan. The Inquirer's story goes: "Intelligence agency GCHQ has intervened in the rollout of smart meters to demand better encryption to protect U.K. electricity and gas supplies. The GCHQ barged in after they saw the plans and realized" - okay, everybody get a grip on the wheel. Take a deep breath. Leo, center yourself over your ball - "realized that power companies were proposing to use" - I can hardly read this - "a single decryption key for communications to the 53 million smart meters that will" - and "smart" should be in quotes - "that will eventually be installed in the U.K.

"The agency was concerned that the glaring security weakness could enable hackers, once they'd cracked the key, to gain access to the network and potentially wreak havoc by shutting down meters en masse, causing power surges across the network. The security flaws would have been particularly catastrophic as the U.K.'s" - as the Inquirer describes it - "'Rolls Royce,' as in unnecessarily expensive, smart metering system doesn't just automate meter reading. It enables power companies to engage in power management and even to cut people off remotely if they haven't paid the bills.

"A telecom industry veteran, Nick Hunn, who is the director of WiFore Consulting, told The Inquirer's sister publication Computing 15 months ago that the system designed by the utilities and metering industries was 'fiendishly complicated.' Nick said: 'Too many cooks have ratcheted up the technical complexity to the point where it is no longer fit for purpose. As a result, it's lining up to be the next major government IT disaster.'"

So unfortunately, we're only going to make it more complicated, but it sure is apparently not only overcomplicated, but still not complicated enough. Now, imagine one master key for 53 million smart meters. What could possibly go wrong?

Leo: It's so complicated to have a different password for every device. I think we should just have one.

Steve: As we know, the - well, yeah, exactly. As we know, in order for it to work, it must decrypt the key. So anybody reverse-engineers one of the 53 million smart meters and captures the key and now has access to the entire grid. Unbelievable. One key for them all, to rule them all. Wow. Fortunately, that won't happen. But it does sound like it's a nightmare. And as our telecom industry veteran Nick noted, talking about it being "fiendishly complicated," complexity is the bane of security. If it isn't really simple, so that you can look at it and say yes, this is secure, then throw it away and start over.

A bit of miscellaneous goodies. Hover is my choice of...

Leo: Oh, congratulations. Good choice.

Steve: ...registrar. Yeah. It was the overwhelming favorite among our listeners. And not only because Hover is a...

Leo: The power of advertising.

Steve: Not only because Hover is a past sponsor. And I am completely sold. In fact, when I go back and look at GoDaddy, I think, what was I thinking? And I know what it was. It's that Mark Thompson was praising GoDaddy, but I think it's only because they're in his town. You know, they're based in Phoenix, and so is he. And I should have used better sense. The good news is I only had - actually, I was only in the process of the painful transfer, doing a test transfer of some domains I don't use, that did go through. And it was Network Solutions being deliberately foot-dragging. I mean, they used every possible day of their week, three days and then four days, they could to delay the transfer. Anyway, I'm so happy with Hover.

Leo: Good.

Steve: I am set up there. I am very impressed. So first of all, to our listeners who tweeted - and also the mailbag was full of suggestions, of people saying Hover, Hover, Hover, Hover. Namecheap came up a lot. Everybody has their favorite registrar. But the overwhelming majority were people who had said, you know, just use Hover, and you won't regret it. And I don't. So thank you for that.

I also wanted to note that Friday night, before I quit for the week, I pushed out a major update to Never10.

Leo: Oh.

Steve: Yes. It is now v1.3. And what I would ask is it's kind of important that everybody who used a previous version run it just once because I was sure that I observed behavior of Windows Get Windows 10, the GWX components themselves proactively deleting the pre-Win10 download blob that in many cases may have been downloaded into people's machines, but I was never able to recreate that behavior, despite extensively trying. So then I started worrying, okay, maybe, I don't know, I can't explain it, I'm sure it disappeared.

But the longstanding disk cleanup tool, which is called Cleanmgr.exe, it does provide the option itself, if you select the Clean Windows System Files option. It goes away and then comes back with a bunch more checkboxes. And right in there is, like, Windows Upgrade Pre-Download. I mean, they call it that themselves. So it might have been operator error, where I discovered that, and I was playing with that, and it did what it said it was going to do, and then I thought, oh, look, GWX deleted the blob. Anyway, the blob is 6.5GB.

Leo: What?

Steve: I mean, it's massive. And this was on a Win 8.1 test machine of mine, x64. So 64-bit, 8.1. It downloads about three or so, and then it expands it to 6.5. And it took me, frankly, quite a while to root that all out. It's an entire tree of more than 19,000 files and folders with bizarre privileges and ownership by the trusted installer. And anyway, the point is, 1.3 knows about it and takes responsibility for it. It shows it to you and gives you a one-click delete. So it may very well be that, if you had been, like, saying no, no, no, no to Windows or to Microsoft and their components, until you got Never10, which stopped all of that nonsense, that 6.5 or whatever blob could still be there. So I feel badly about that, so I created 1.3 to fix that.

So if you run 1.3, it'll just - either it won't show you anything, which means you don't have the problem, or it'll show you in those two lines below the main little readout screen the exact byte count and object count. You can see screenshots. I have updated new screenshots on the Never10 page, where it shows you how many there are. And if you press the "delete them" or "remove them" button, then you watch them down count back down to zero as it scrubs your system of all of that stuff. And the second new feature was much asked-for among our enterprise listeners, and that is full command-line switch control. Many of our corporate and enterprise listeners wanted to use this, wanted to employ it essentially at log-in time in their log-in scripts, which requires - they wanted it just to be done silently. And so 1.3 gives you a mature command-line vocabulary that allows you to put any combination of four different, or I think it's five, actually, different verbs behind the invocation of the EXE, and it'll do whatever you tell it to on any machine silently.

So a nice little update. I'm done with it. It's been a big hit. I think we're at north of 150,000 downloads now. But do, I would ask people who ran earlier versions, just run -

again, you can of course delete it after you're done. Just make sure that it didn't leave a big burg behind because I told people it wouldn't, and now 1.3 will at least let you know if it's there.

Also, one of the Healthy Sleep Formula ingredients, Seriphos, is completely gone from the world, unfortunately, and apparently thanks to...

Leo: Forever?

Steve: Well, okay. So here's what happened. First it sold out at...

Leo: Do they mine it out of - it's all used up?

Steve: Well, our listeners did. It first dried up at Amazon and iHerb, which are the two that I was linking to. And so people started sending me notes saying, "Steve, can't find it anywhere." So I went over the Life Extension, and they still had it. So I made sure I had my own supply. Sorry, but, you know, I've got to keep doing the...

Leo: Bunch of people jonesing here, my god.

Steve: Got to keep doing the research. And then, and also Swanson Vitamins had some. And so I posted that news on the page, and tweeted it, and those immediately dried up. So now there's no more anywhere that anyone knows about. So I contacted the guys that make it, and I said, "Hey, what's going on?" And they said, "Well, we don't know what happened, but it all dried up at retail." I say, "Well, yeah, I know what happened. But why isn't there more?" And they said, "Ah. We have an improved version which we're about two or three weeks away from having stock on. And we thought we had plenty at the rate they were selling, but something happened, suddenly made them..."

Leo: Somebody, somehow...

Steve: "Somehow, for some reason, everyone wants it now. So for a couple weeks we're out of stock." Now, anyway, so what happens is I found a - someone posted in the GRC health newsgroup, he wrote: "Looks like Steve's HSF popularity has depleted the stock of Seriphos. I can't seem to find it anywhere. I'm going to run out in a couple days, and I don't want to miss a night of good sleep. I had forgotten what it was like to sleep through the night, and I'm not willing to go back. Anyone know of an alternative source online?" Unfortunately, I don't think there is one.

The good news is they are saying they expect to have - they didn't have an exact ETA late last week. But they expected to have at least an ETA soon. So I'll be checking back, and I will update on the HSF page when I have any better idea, and when they do, of when they'll get stock back. And it's supposed to be more bioavailable, which is good, and more hypoallergenic, and not have some of the other excipients that sometimes get stuck into these things. People put various other, like, filler stuff to, like, bind tablets together and so forth. And so it's freer of that. So I think it's actually going to be an upgrade.

Now, I have to say the bad news is people have been trying to go without it and reporting failure. And, yes, it is probably the key ingredient of the formula. And in fact I'm right now working on a new version with fewer ingredients. I'm down to four, and it's working really well. I'm going to try to get rid of another one tonight. So it requires taking more of fewer things, which I think is an improvement for a number of reasons. But anyway, I will cover that on the page when I have some definite answers.

Two more things: There was an amazing article written in The Guardian which I tweeted about. And the tweet was, "What should you eat?" And it is a long story. In fact, The Guardian called it - it's in their "long read" section. I created a bit.ly link for it: bit.ly/sn-sugar. And it is a really nice, accurate as far as I could see, sort of summary of the history of the nutrition misinformation that we have received, and why and how it was so wrong for so long. I got a lot of very positive feedback from people after tweeting it earlier last week, so I wanted to share it with our audience: bit.ly/sn-sugar. Really worth the read.

And, finally, Peter Hamilton did it to me again. He suckered me into a series that he hasn't finished writing yet. My review that I posted on Amazon because the Kindle popped up and said, "Hi, you just finished. Would you like to write a review?" And I said, "Actually, I would." So I wrote: "Okay, I admit it. I'm a complete sucker for Hamilton's work. I've read everything he's written, and not just once. Before you read this, you must read The Void Trilogy. And before you read that, you must read the 'Pandora's Star' and 'Judas Unchained' pair. And really the absolute best place to start would be with 'Fallen Dragon.' I envy anyone who has these amazing pieces of truly wonderful space opera ahead of them. As for this work, I never believe that anything I read of Hamilton's will be able to live up to everything he has already done. How could it possibly? And then it does. This does. You will love it. I guarantee it." And so that was "The Abyss Beyond Dreams."

Leo: I just queued it up on my Audible because it's been sitting there. I bought it some months ago when you first talked about it.

Steve: I did exactly the same thing. And I read one night, when the formula was not working because I was doing what I call one of my "negative tests"...

Leo: That's aptly named for that, actually, yeah.

Steve: Yup, "The Abyss Beyond Dreams."

Leo: "The Abyss Beyond Dreams," yeah. I'm there right now.

Steve: The good news is, for anyone who doesn't want to wait, the second one is due out mid-September.

Leo: That's one of the reasons I didn't start it, because it's a trilogy, and it's only Book 1.

Steve: Yes. And it's like, remember "Pandora's Star"...

Leo: We've been there before.

Steve: ...left us, like, oh, my god, what's going to happen? The good news is this doesn't. It is two books, but this gives you a really nice semi-conclusion. Yes, it'll be nice to see what happens next. But you're not left like with a cliffhanger, where you are really annoyed that you can't keep going. So I would suggest to people, if you like - and first of all, we haven't talked about sci-fi in depth for a while. So there may be listeners who don't know Hamilton. So "Fallen Dragon," it's a standalone fabulous perfect place to start with your introduction to Hamilton.

And remember that I do have a PDF that is sort of my reading list, and that's bit.ly/sn-scifi. Or maybe there's no hyphen. S-C-I-F-I, either with or without a hyphen. Or maybe it's SG. Anyway, I'm sorry. I think it might be sgscifi, maybe sgscifi, which will get you a reading list of Hamilton stuff, who is up there among our favorite authors.

Leo: Right.

Steve: And lastly, since this podcast has got so much material to cover, I wasn't going to do a long SpinRite diatribe. I just wanted to say I saw a nice tweet from Stuart Carroll, public tweet. He said: "After hearing about SpinRite on Security Now! for so long, I tried it yesterday, and it brought my media drive back from the dead. Thanks, SGgrc." And Stuart, thanks for sharing your news.

Leo: Well, that was easy.

Steve: Yeah. It does that, by the way. It brings drives back from the dead, for anyone who's curious.

Leo: Yes, sir.

Steve: So of course the big news is the announcement from Open Whisper Systems that they had finished integrating their communications technology into WhatsApp across all platforms. And, I mean, it is multiplatform. And in fact they've just got - they've expanded the beta to the desktop version. It's been in limited beta for a while. It's now available in open beta, so anybody who wants to get the WhatsApp app for their desktop. Now, in their photo on the announcement they show it on a Mac, and I didn't have a chance to dig into which desktops it's supported on. But at least the Mac. So we've sort of been talking about Open Whisper Systems, and of course Moxie has been a topic and aspect of the podcast.

Leo: As has his hair.

Steve: As has his hair and the fact that he likes to sail, for a long time. TextSecure was the name of an earlier messaging app that they had created. And then there was like - there was RedPhone that they had. And then there was Circle. And there have been different efforts. They've also been moving this forward over time. At one point they looked at Off The Record, the OTR protocol, which we've talked about in the past. There was a nice web browser implementation of OTR which provided encrypted point-to-point communications. But there were some problems with the way it worked because it really needed to be online communications. It required real-time interaction between the endpoints for the exchange of keys.

And what the Whisper Systems guys wanted to come up with was something that would allow sort of the text messaging model, that is, where you could send a message to someone who may be offline at the time; and when they came online, then they would be able to receive the message. So it would be queued somewhere until they were able to receive it. And the security challenges of that were significant. So they were, over the course of years - and that's one of the interesting takeaways, is this kind of super-mature technology, which is what I think they have today, doesn't happen overnight. It takes nurturing, and it takes building a few and seeing how they work and thinking about it because it's often difficult to attack something that's just an idea. It really helps to, like, play with it.

And this is why building stuff is so important. That's one of the lessons that I've talked about often is that hobbyists who create something discover stuff they didn't expect to find. And it's because they didn't expect it that it's useful. Even though you don't know it's useful a priori, it turns out once you do it it's like, wow. I didn't think this was going to happen. And that's the point. So as a consequence of them, like, having many previous things, they've got to a point today where I was just enraptured by digging into the details of this protocol. It, too, has been renamed, as has the app. There's no more TextSecure or RedPhone or Circle. Everything is now Signal.

So Signal amalgamates and pulls all of those previous efforts together as the app. And the unpronounceable name of the protocol, Axolotl, A-X-O-L-O-T-L - and isn't he adorable, Leo? Oh, is this the cutest little thing I've ever seen. He's critically endangered and adorable, probably because Disney stole them all, aquatic salamander. And they called it Axolotl because the salamander, as we talked about at the top of the show, has amazing self-healing capabilities. Being a salamander, it can regrow limbs. And the Axolotl protocol has some self-healing properties.

So I'm going to read two things from two different blog posts where they describe stuff, and then I'll get into my own bullet points of the features. So one of their blog postings said: "To continue eliminating confusion and simplifying everything within the Signal ecosystem, we're renaming Axolotl to Signal Protocol. The implementations have been renamed, so there are open source Signal Protocol libraries available for C, Objective C, Java, and JavaScript in our GitHub repository." So everything I'm going to talk about is also, not only well vetted, but open source, encouraging this to be the messaging protocol of record. They continue: "These have been making their way into an increasing number of communication apps" - as well they should, I say. "And," they write, "we're excited for the future of the Signal Protocol as it continues to spread."

And then in one paragraph from a different posting they said:

"As of today" - and that's this most recent one of last week - "the integration is fully complete. Users running the most recent versions of WhatsApp on any platform now get full end-to-end encryption for every message they send and every WhatsApp call they make when communicating with each other. This includes all the benefits of the Signal Protocol - a modern, open source, forward secure, strong encryption protocol for asynchronous messaging systems, designed to make end-to-end encrypted messaging as seamless as possible."

Leo: By the way, Burke has just handed me a note. He's apparently as big a science fiction fan as you are. He reminds me that in "Dune," the Mentats grow in tanks called "Axolotl tanks."

Steve: Oh, nice.

Leo: Go Burke. Nice.

Steve: Burke, I think, yeah, that's...

Leo: Nice reference.

Steve: Yeah, it's up there, yes, very nice. And, boy, they need to because they're strange-looking creatures.

Leo: They're kind of [crosstalk] the spice.

Steve: Yeah. Okay. So we have now, I mean, as far as I'm concerned, this problem has been solved. These guys utterly and absolutely and beautifully nailed it. We have confidentiality, meaning of course that the communications is encrypted. We have integrity, meaning that any message alteration will be detected and will fail the transaction, which is to say that there's a message authentication code as part of this, a MAC, so that the message cannot be altered. They claim authentication, and that's, as I guessed, actually, last week, it's like I wasn't seeing how that was there. And it turns out they know it's not. But it's available.

And so that's one of the caveats we'll talk about. And so it is possible to confirm the identity of the correspondent. But you need to take action to do so. Then they also claim participant consistency. And that's really also important, and unfortunately it defaults to off. So that's the other thing that has to be done manually. I don't know why it's off. It's like, turn it on. But our listeners will. And we'll cover that in a second.

Leo: Yeah, okay. I want instructions. Okay.

Steve: Yes. Then they also claim destination validation, and that's actually just related to the previous two, authentication and participant consistency. They do have forward secrecy, and we remind ourselves that that means that a future compromise of a private key will not allow the decryption of past messages. And they have backward secrecy, which as you would expect is the reverse of that, meaning that a past compromise of a key will not allow decryption of future messages. And that's really just an ephemeral key that we've talked about, a key that's constantly being changed and renegotiated. So nobody who had a key in the past, it doesn't do them any good about the future.

Message unlinkability. The messages are asynchronous. They are independent. They can arrive out of order, and they can be missing, and the whole system still holds together.

And another really cool feature, and Moxie likes this a lot, is message repudiation, which means that it is possible for a - okay. So a nonrepudiation system is unfortunately like PGP, or Threema, where in both cases a public key is used to validate the message signed with a private key. The problem is that private key never changes. That is, that's static for a PGP user or a Threema user. Which means that a recipient of such a message automatically has a guarantee that it came from the sender. The problem is that's a nonrepudiatable guarantee. That is, the sender is unable to say "I did not send it."

Now, it might be a little confusing because you're saying, wait a minute, we want to make sure it's from the sender. But so the way we get message repudiation, which is a very cool feature, which is why Moxie likes it so much, is that the recipient could also forge a message from the sender. Which means that the recipient cannot say to a third party, this guy sent this to me, because that recipient, by virtue of the way the protocol works, can also generate a message which is as equally valid as something that they received.

So when you think about it, that is just incredibly cool. It's what you want if you care about the ability to have message repudiation, which certainly could come in handy because we've got such strength in terms of the crypto system. So what it means is the recipient knows that only the sender or the recipient could validate this message. But the fact that the validation includes the recipient means that the recipient can't claim, well, it means that the sender can claim, with provability, that, no, it's not necessarily the case that I sent that because the person I sent it to can forge it, too. The recipient can forge it. Which is very clever and very cool.

And then finally asynchronicity, the fact that the system is designed so that it does not need to be real time. It works where messages are being queued by a server until a recipient is ready to receive them. So I'm not, for the purpose of this podcast, going to dig into the protocol. Frankly, I can't do it on an audio podcast because I would need some diagrams, because it is really complicated. There's one point, for example, where each of the endpoints has a - they produce an elliptic curve pair, and it's funny how many things Signal and SQRL have in common. Things that I did two years ago are, like, this is the way Signal is working, too, because it's like the right way to do it. They're using the same curve I am, the curve 25519, which is Bernstein's elliptic curve algorithm.

So they create Diffie-Hellman private and public keys, static ones. Then they create another set of ephemeral keys. Then they use a triple Diffie-Hellman protocol where they exchange their ephemeral keys, and they use Diffie-Hellman key agreement three times to agree to generate, to take their private key and the other person's ephemeral public key, get a key agreement there. The other guy takes his private key and the other's Diffie-Hellman public key, gets a second agreement. And then they take the exchanged ephemeral keys and then use that with Diffie-Hellman to get a third, and they concatenate all three of those together in order to get this master session key.

And anyway, like I said, it's a crazy protocol. But what it ends up doing is it eliminates the OTR, the Off The Record protocol, used DSA signing, the Digital Signature Algorithm, which is extremely scary for cryptographers because it's a brittle and fragile algorithm that's easy to get wrong; whereas Diffie-Hellman is just - it's hard to get wrong. So, and that bizarre exchange and triple Diffie-Hellman and concatenation system, a whole bunch of amazing properties fall out of it. And this is what Trevor, who's been working with Moxie, came up with, was that particular key.

But there's this notion of a ratchet. In an interactive protocol, a ratchet is the term where you want to evolve a key that you agree upon. As you send messages back and forth, you ratchet the key forward. And there's this notion of a single ratchet, where the first side sends the first half of a Diffie-Hellman key agreement to the other guy. But until he receives the other guy's half, he can't really do anything. When the other guy sends it back, he sends his half of the key agreement with an implicit acknowledgement of the receipt of the first half. Now each side has new Diffie-Hellman ephemeral public keys which they have exchanged. And they're able then to do a Diffie-Hellman key agreement to obtain the identical next ratchet key.

The problem is that's nice in real time when you're able to keep sending these back and forth and keep evolving keys. But one of the innovations here is an offline ratchet using a hash, where, if the sender is sending successive messages, they really do not want to reuse keys. So they use a hash ratchet, which one side can do just by hashing the old key in order to get the next key. And that allows the keys to evolve single-endedly. And then the way the protocol works, the moment the outstanding first real-time sort of double-sided ratchet can be completed, then they both resynchronize the offline ratchets. And as I said, it is a little mind-bending, but they've got it all working.

And the one other really interesting thing about this protocol is that there's a problem with establishing the first session. That is, in a real-time protocol you could somehow come up with a shared key in real time. But this whole system wants to be able to send somebody you have never sent a message to a secure message. So that the very first one is secure. And the question is, how do you solve that problem?

Well, they did it brilliantly. When you register your WhatsApp or your Signal app, whatever it is, with the central server, your client pre-seeds the server with a hundred of your public keys and gives them IDs. Then what that allows is that allows somebody who wants to send you a message, who has never sent you one before. They're online. I mean, they're sending you a message. They're able to get one of the hundred pre-seeded keys from the server, which you have provided the server ahead of time, in order to bootstrap a secure message.

Leo: It's your public key, so there's no downside to putting that on the server. Or it's a one-time...

Steve: Actually, it's 100 ephemeral public keys.

Leo: Okay.

Steve: So it's not your public key. It's actually...

Leo: It's a one-time key.

Steve: It's actually, exactly, it's a cache of one-time keys. And what they do, then, is they use that to encrypt and place that back on the server until you're online. When you are, that blob comes to you. Well, first of all, you will never allow the reuse of that one-time key. So if you've ever seen it used, actually in use before, you absolutely drop the message. The server itself should never issue it a second time. But again, a recipient could try to reuse it. But they're one-time. So since the server is going to delete it, nobody else will get it. The recipient has no reason to reuse it because it won't be accepted because the originator of that key will remove it from their pending valid ephemeral keys list and won't accept it.

So the system is just, I mean, it is - they nailed it every way from Sunday, with one exception, that you cannot be sure who you're talking to. And this is always the problem. Authentication is the bugaboo. It's the thing, you know, when I was excited about Threema, it's because they put that in front. They said, you know, authentication is critical. And that's what the three dots and the yellow, green, and red, like, stop sign was. It was how much, you know, you only got green, full green authentication if the two devices had seen each other's QR codes and had, like, a physical device exchange because then you absolutely knew that they had exchanged - that the public key you were getting was the public key of the person you thought you were getting it from. That's the problem is this man-in-the-middle problem.

And what's interesting is they know it. The picture, Leo, at the top of this - in the show notes, at the top of our discussion, under the WhatsApp headline, is the photo that they show in their blog posting. And what is it? It's the authentication portion. What they've done is they understand that the one thing they haven't solved - and, I mean, it is the problem. There is not a way to solve this. And so what they do is they provide a QR code or a, yes, sitting down, 60, six zero, digit confirmation. That QR code is a version of those 60 decimal digits. They chose decimal because WhatsApp wants to be multilingual, and no other alphabet is as universal as the decimal digit system which does exist in all the languages that WhatsApp spans.

Leo: Now, what if I - could I put my QR code on the screen? And then people would...

Steve: Yes.

Leo: Because that would verify, I mean, as long as you trusted this, and somebody didn't edit my code into the video.

Steve: No, that won't work.

Leo: It won't work.

Steve: That would work with Threema. That won't work here.

Leo: Oh.

Steve: Because, as I was just about to say, that code contains both of the parties' keys.

Leo: Oh, I don't have a code until I'm in a conversation.

Steve: Correct.

Leo: Got it.

Steve: And so this is used - so you establish a conversation. And so, okay, so what I like about this is that authentication is optional. That is, you don't need to do it. Everything's fine. I mean, and it's almost certain that the world is working the way you expect it is. But you cannot know. You cannot be sure. So what you have to do is arrange to share either the QR code or those 60 digits with the person that you're in a conversation with. That is unique to your conversation.

Leo: Okay. Get your phone out, Jason. I'm going to authenticate with Jason right now because we've already had a little chat. So I would pull up that chat. And then how do I do that? Is that in the menu here?

Steve: I fired it up and got it going at my end, but I've never had a conversation with anyone, so I don't know.

Leo: That's the problem. No one has this. Okay.

Steve: Okay. So, but, well, no, it's WhatsApp.

Leo: I know, everybody has it. But the problem is not everybody uses it.

Steve: The second part of this is that, for some reason, if that ever changes, you are not notified. That is, if a man in the middle inserts themselves, you're not told. But under account security, you can turn that on. And it's like, okay, why is that not on?

Leo: Oh, the alert that it's no longer secure, yeah, by default is off.

Steve: The alert that the identity linkage has been broken, essentially, that is, that thing that you are authenticating has changed. And so they say "Show security notifications," and then they said "Turn on this setting to receive notifications when a contact's security code has changed." Why would you not want that on all the time?

Leo: Yeah.

Steve: Because it should never change.

Leo: Right.

Steve: The messages you send and your calls are encrypted, regardless of this setting, when possible. But everybody needs to turn that on.

Leo: Right.

Steve: So, I mean, since no one's - even if you don't authenticate, there's never a reason for that code to change unless they are setting up a new device, or they've lost their device, and they're replacing it, and et cetera. So anyway, it is an amazing protocol. And, I mean, it does provide encryption. They've nailed it all. But the one problem is you need to verify one time. You only need to do it once.

Leo: So I figured out how. Actually Jason did. I tapped his picture on the chat that we're already engaged in and go to info. No?

JASON: Yeah, that's it.

Leo: Oh, yeah? Info? Okay. Picture, info, and then I scroll, and it says "Encryption," and there's a little key, and there's that. Now what does he - oh, so I press - so he's done that. Now I'll scan your code, Jason. So put your code down here, and I'm going to scan your code. Yeah. Now, pull up the code on my - right, Steve? I would scan my code in his phone.

Steve: Yup.

Leo: And then that would verify that there. And now we are secure.

Steve: Yeah. Well, now you are authenticated.

Leo: To him. And now we want to do the opposite; right? So you press scan code on your phone.

Steve: No, there's actually no need to reverse it.

Leo: There's no need. It's both ways, okay.

Steve: All it was doing was comparing to make sure they were identical.

Leo: Got it. So it's verified, but that was an unnecessary second step.

JASON: Yeah.

Leo: So now it's verified.

Steve: Correct, right. And the other way is to read...

Leo: That long...

Steve: ...the 60-digit...

Leo: Yeah, not going to do that.

Steve: I know it's long. But frankly, the first three digits will be wrong if the code is incorrect. It's a hash.

Leo: Jason said now we're encrypted. But, no, we've always been encrypted. This is just that now I know it's you, couldn't be anybody else.

Steve: Exactly. Until then, there could have been a third party who was intercepting both of your connections, pretending, and essentially impersonating each of you to the other.

Leo: Right, right.

Steve: And this cuts out the middleman. This absolutely verifies there is no middleman. And so for someone who you're wanting to - who you can't visually exchange codes, you can just say, okay, as the picture shows, you know, 173275114426668. Because you'll notice you and Jason will have the same number. If you compare, yeah, they're the same.

Leo: Ah. Got it. That's why you only need to do it once. It's the same QR code.

Steve: Exactly.

Leo: Got it.

Steve: Exactly.

Leo: So Dr. Mom says, so if I'm not - I want to chat with my cousin in Paris. She would have to read the code off over the phone. Or I guess she could send a picture of the QR code. But then you'd have to trust that transport. So that wouldn't work.

Steve: True. True.

Leo: If you have PGP - okay, here you go. You have PGP. She encrypts a message to you with that number in it, and then you can verify. Or a picture, actually, better, a picture of the QR code. A question from the floor.

Steve: Fax it to her or...

Leo: Fax it to her, yeah.

Steve: Or here's one. Alternate. Because this code is set up in 12 groups of five, alternate reading to each other every five digits.

Leo: Wait a minute. Now, here's a question. This is a good one from the audience. It's the same code. I just take a picture of my own code. That wouldn't work, Steve. There must be a different number.

Steve: No, this is for you to - all you're doing is - what that checkmark said was that these are the same.

Leo: Yeah. But couldn't I just take a picture of the QR code and then...

Steve: Yours and Jason's are showing the same code; right?

Leo: Yeah. So I just take a picture of my code on my phone. Yeah, it is the same code.

Steve: So the point is it's for your information. It's not - it doesn't do anything for the...

Leo: I'm not too bright. I apologize. I don't really understand how to do this.

Steve: It doesn't do anything for the app. It doesn't change anything. It's just this allows you to know that you actually have Jason's private key, and he actually has yours.

Leo: Right. So you really wouldn't want to just take a screenshot of your screen and then take a - because that would be - you wouldn't be proving anything. You have to make sure these both match. That's it.

Steve: Yes.

Leo: This could be the bad guy's code on your phone.

Steve: Right.

Leo: So you have to go to tell Jason, hey, make sure these match, and then we know.

Steve: Right.

Leo: Got it. This could be the man in the middle until I verify that we both have the same thing at both ends.

Steve: Yes.

Leo: Got it.

Steve: And in fact, if you had different codes, there is no question that there is a man in the middle.

Leo: Right, because that would have to be, yeah.

Steve: Those codes have to be identical if it's a point-to-point connection. So not only does it prove that you each have each other's actual private key, it also proves, if they're the same, that there is no man in the middle. If they are different, there is a man in the middle.

Leo: Okay. Now, I don't want to be the Windows 10 in the punchbowl here, but there is one point that there's nothing - it's not open source, so you can't - in fact, this is of course what EFF says, as well. There is, you know, seven green checkboxes. And while Signal is open source, and you can verify it, you can't verify WhatsApp. So there's a red check there that's code is not reviewed.

Steve: And in fact one of the things they announced a few months back was they got a verifiable build on Android, meaning that, from the public source, they built a binary identical to what the Android app had, which of course is what we want to see. We want to see that it is absolutely, bit-for-bit, from the source, that it was no source [crosstalk].

Leo: Yeah. You can do that, and you should do that because that verifies the fingerprint. But that only verifies the binary blob.

Steve: Well, and Leo...

Leo: You don't know that there's no NSA code in there. You just take - you have to take their word.

Steve: Right. And, even worse, you made a very good point a couple weeks ago when we were talking about Apple. And this is why I want to - the other thing I want to stress on the podcast is absolute security in the ivory tower exists at the mathematical level. In application, it doesn't. There is no such thing. It is an illusion. And so I would argue that we get too worked up over this idea of why can't I have perfect mathematical security? Because you can't. You're typing on a screen, and you didn't write the firmware yourself, or the software, or design the silicon.

Leo: Or, exactly, the processors. We talked about it last week.

Steve: There's too many, you know, there's theory, and there's real world. So it cannot get any better than this. It doesn't need to get any better. So people should expect...

Leo: Open source could arguably be better.

Steve: No.

Leo: Not perfect.

Steve: No, no. I would say I really don't think that we're seeing any evidence of that because Stagefright is open source. OpenSSL is open source.

Leo: No, no, no, no. Not by nature of being open source, but just you have to trust Moxie Marlinspike and the WhatsApp team that they haven't injected code in here. If it were open source, somebody could at least verify that there's no backdoor in there.

Steve: Right.

Leo: I agree with you there's no - open source isn't, per se, more secure.

Steve: Unfortunately.

Leo: Bugs exist in every form of code. But it does allow you to verify that there's no intentional backdoor in there.

Steve: Right.

Leo: All right. Eh, you know, I mean...

Steve: Yeah, well, okay, Leo.

Leo: There's nothing perfect.

Steve: If you want perfect security, you use a cutout to arrange a meeting in the middle of a park on a bench. Or if you don't trust the person that you're going to whisper to, you meet in a steam bath and briefly flash each other.

Leo: Excellent, excellent tradecraft, Steve. I have a QR code on my butt. Please scan it.

Steve: Unfortunately, technology is only as good as its implementation. And nothing has changed from the days of "Casablanca" and the old spy movies. If you really want to be secure, you've got to meet face to face in a place where you cannot be eavesdropped on, without listening devices, and whisper to each other. That's the way it is.

Leo: Is it Ben Franklin said two could keep a secret, but only if one is dead, I think was the line.

Steve: Right. Anyway, WhatsApp is a beautiful piece of work. Hats off to these guys. We have now what should be an industry standard messaging solution in Signal - open source, cross-platform, multi-language libraries available. There's just no excuse for anyone to do anything else. They nailed it.

Leo: Actually, Ben said three can keep a secret if two are dead. But it's still the math. It's just the math. Steve Gibson, I am so glad that you give this your thumbs-up. And what's remarkable really about this is that, with the flip of a switch, Facebook and WhatsApp gave one billion people, close to one billion active users, instantaneously secure, not just text, voice communication.

Steve: Yeah.

Leo: Wow. Pretty impressive.

Steve: Yeah.

Leo: We do this show every Tuesday, 1:30 Pacific, 4:30 Eastern time, 20:30 UTC. Please watch live, if you can. Love having you in the chatroom. We also had a great studio audience. You probably heard them giggling in the background and asking tough questions. If you want to be in the studio audience, tickets@twit.tv is the email address. Steve will be doing a Q&A, God and Moxie Marlinspike willing, next week. So you can put your questions up on the Twitter. He's @SGgrc.

Or go to his website, GRC.com/feedback. While you're there, golly, there's so many great things to do. Pick up your copy of SpinRite, world's best hard drive maintenance and recovery utility. He also offers audio of the show, and transcripts, written transcripts. I think it's really helpful sometimes to read along. And the show notes are there, too. So if you want to see the pictures we talk about, they're all in there, as well. And links.

Steve: And all the links.

Leo: Yeah. We have also audio and video of the show at our site, TWiT.tv/sn. We also put it on YouTube, and it's on every podcast client and all that. Just search for Security Now! and subscribe. Ten years in the making. You don't want to miss another episode.

Steve: 556 next week.

Leo: Wow, wow. Thank you, Steve. We'll see you next week.

Steve: Thanks, Leo.