Listener Feedback #182

Description: Steve and Leo discuss the week's major security events and discuss questions and comments from listeners of previous episodes. They tie up loose ends, explore a wide range of topics that are too small to fill their own episode, clarify any confusion from previous installments, and present real world 'application notes' for any of the security technologies and issues we have previously discussed.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-440.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-440-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson's here with the usual great security news. We're going to answer some questions. And he's come up with a way - I don't think anybody's ever mentioned this; in fact, I think it's a complete and utter scoop and a breakthrough - to improve beyond question the fingerprint reading on the iPhone 5s. Stay tuned. He's a wizard. He's a genius. Steve Gibson and Security Now! coming up next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 440, recorded January 28th, 2014: Your questions, Steve's answers, #182.

It's time for Security Now!, the show that protects you, your loved ones, your privacy, your online stuff with this guy right here, the Explainer in Chief, Mr. Steve "Tiberius Rumpelstiltskin" Gibson. Because, yo, because he's the man. Steve, you know, I was just reading, it's so funny, I was just reading, I think it was on Google+, and of course everybody's talking about this port 32647 stuff. And somebody, I can't remember who it was, somebody we all know, said "I completely forgot about ShieldsUP!" and put a link on there to the ShieldsUP!. I guess somebody had mentioned you in an article about testing that port. So you're famous, Steve.

Steve Gibson: Well, and if you just google that integer now, I'm the first link that comes up.

Leo: You own 32764.

Steve: Yeah, that's my port, Leo.

Leo: That's great. We're going to do something a little off-color today.

Steve: Well, we did skip a bunch of Q&As toward the end of 2013. There was just - either we were overrun with news, or there were specific things we wanted to talk about. And so we were missing them. And then when I archived all my email, and I found out that I had 53,000 pieces of Security Now! mail, I thought...

Leo: Oh, wow.

Steve: ...you know, let's answer a few more of those. And we didn't have - there was nothing that was really grabbing me. Actually the last question in today's Q&A is a teaser for next week's show topic. So we do have a show topic for next week. And actually I have a couple backed up, but just need to be able to pull them all together. So we have a lot - we have an interesting grab bag of news. I have a "I just can't believe they did this" nightmare to get into technically with Bluetooth Low Energy.

Leo: Oh, no. Tell me. Oh, no. Don't tell me. Oh, well.

Steve: Yeah, yeah.

Leo: And I presume that you want to talk about that new slide that surfaced yesterday, the...

Steve: Yep, we've got to talk about, well, I don't have much to say. But even that is - we'll definitely touch on it. So we've got more news on the point-of-sale malware. I promised everybody in Twitter that I would share how I managed to make Apple's Touch ID reliable. And I apparently have. It's just I don't - I no longer have touch fade, and there's a trick for that that I will share. CryptoLocker, it turns out, far from dead. There's new versions of that. Of course we have the never-ending NSA news machine. This Bluetooth nightmare that I alluded to. And I found a couple of interesting, actually one very interesting, for our listeners, new security sites. So lots of fun stuff to talk about.

Leo: Steve Gibson, Leo Laporte. Let's begin. Let's launch.

Steve: So we, sort of from the ether, we got the sense that there were going to be more point-of-sale breaches in the future. That is, we know for sure now that that's how Target got themselves compromised. Neiman Marcus has since confirmed that they have also been a victim of infected point-of-sale terminals. And then just a couple days ago Brian Krebs reported what initially started out to be a rumor, that the chain Michaels, which is an arts and framing chain, big chain, I guess sort of in the Southeast, 1,100-branch chain...

Leo: No, we have one here. They're all over.

Steve: Oh, okay. That they appeared to be - we have Aaron Bros., is the one that I think of.

Leo: Same company, yeah. They own Aaron Bros., too, yeah. And they were both breached.

Steve: Well, yes. And so they have now confirmed it. Before Brian could get confirmation, he said that multiple sources in the banking industry say they're tracking a pattern of fraud on cards that were all recently used at Irving, Texas-based Michaels stores, Brian writes, "an arts-and-crafts retailer that has more than 1,100 stores in the United States and Canada. On Friday, KrebsOnSecurity heard from a fraud analyst at a large credit card processor that was seeing fraud on hundreds of cards over the previous two days that had all recently been used at Michaels. The fraudulent purchases on those cards, the source said, took place at the usual big box stores like Best Buy and Target." So there's another one.

And for some reason the number "8" is stuck in my head. I think there was, like, there have been - there's some reason to believe we're going to have more of these. That is, essentially, the underlying configuration of these point-of-sale terminals are Windows XP Embedded, and this malware that was created is essentially - it's chain agnostic. It will happily infect anyone's point-of-sale terminal that's written on Windows XP Embedded. So I sure hope that any other similar companies that are using this technology that may not yet be victims take this seriously because this is clearly something you could preempt if you realized, oh, wait, those are ours. We're using the same point-of-sale terminals that Target is. How is our security? So...

Leo: And a tip of the hat to Brian Krebs because we were all worried when he left the Washington Post, and he said, "I'm going to do it on my own; I'm going to do my own KrebsOnSecurity blog." And he has been breaking these stories. He's been knocking it out of the park all on his own. And good for him.

Steve: Yeah, he's doing a great job.

Leo: Nice job.

Steve: So, okay. Apple's Touch ID. Many people have experienced sort of what's described or feels like a fade of the recognition of their thumb or finger or whatever. And there have been a number of suggestions for fixing that. For example, and I had suggested that you could - there are, I think it's maybe five slots that Apple makes available where you can register fingers. And I said, well, rather than registering different fingers, how about registering the same finger in different slots, therefore essentially giving it more opportunity to find a match.

Well, it turns out that there is a way to what I call "overtrain," to overtrain the recognizer. And so, through some experimenting, it's possible to demonstrate to yourself that the overtraining is actually happening. And what I think will end up being understood at some point after we sort of collectively get more experience with this is that there was the typical tradeoff made between user convenience and recognition percentage. Meaning that, during the typical training, they would probably have loved to

have three times more fingerprints. But this was brand new. No other phone had this technology. This was Apple's first.

And I'm sure the UI guys said, well, if we got more fingerprints, we'd get a better sample. We'd be able to eliminate noise by comparing multiple reads. I mean, we could just do a much better sample if we got more. But the human factors guys said, oh, we just can't ask people to sit there pressing that button all morning. We can't. So someone made a decision, well, okay, we'll take this many samples. And I'm sure they tried it, and it seemed good enough. Well, apparently for some people that's not good enough.

So here's how you can give your phone as much of a sample size as you want. It will never give up on you. It will never get tired of accepting additional fingerprint data. And what I've learned is it absolutely goes far out the curve in terms of recognition. I now never get a miss. So you just go - you open the settings app, the standard little wacky-looking gear thing, Settings. Then go under General, and then Touch ID & Passcode is the next level. And then in there you go into - oh, and I think, when you do Touch ID & Passcode, it requires you to enter your passcode at that point to get into the Touch ID section. So then you go into Touch ID. At the bottom of that screen is a list of the registered fingerprints. And it is training there. So that, if you give it a fingerprint like the thumb that you normally use, you will see that item - it actually, well, it highlights by going dark briefly and then coming back. It also was a training event. You can verify...

Leo: Oh, that's interesting. It's taking more readings just within that spot. Huh.

Steve: Yes, yes. And it never gets tired of doing so.

Leo: Now, how do you verify that it is actually adding a reading?

Steve: The way you can verify is you can do something that it won't read, like say you give it way too far out at the end of your thumb, where it hasn't been trained. And you'll notice, like, do something like that where it won't recognize it and convince yourself it doesn't know the end of your thumb. Then go back to the trained area, and in multiple trainings slowly move forward so that you're essentially expanding the recognized area of your thumb surface. And you can walk it right back out to the same area that it used to not recognize, and now it will.

Leo: Interesting. They don't tell people this.

Steve: None of this is documented. I was just messing around with it.

Leo: How interesting. So you're not in the train - you're not actually training. It's in the screen before you say let's go, let's train.

Steve: Correct.

Leo: And it's keeping track, and it's actually doing training.

Steve: Yes.

Leo: They must have put that there on purpose.

Steve: Yes, well, see, they would like more samples. But someone said no, no, no, we can't - only ask them for 10. Otherwise it's just going to seem like it's broken.

Leo: It's too much. It does. In fact, it's annoying. It takes a while to do it.

Steve: Yeah. But now, so here what they're doing is, they're just sort of sneaking some more samples from you. They're just sort of like, you know, you put your finger on it, and it says, oh, yeah, that's the one. And I assume that if you had, like, multiple fingers trained, then when you put different fingers on, the proper one would highlight. Well, it's also sneaking another sample from you. We can use that in order to just overload it with samples.

Leo: So that's how, if you have multiple samples on there, it says, oh. It highlights the one that you're using. So it says, oh, yeah, I recognize that, I recognize that.

Steve: Correct.

Leo: But it's at the same time recording more data points.

Steve: Yes, yes.

Leo: Very, very sneaky. Actually in a good way.

Steve: Yes. And unfortunately they don't tell anybody that because it would be nice if that was somehow noted, if you'd like to give us more samples...

Leo: How did you find this? You just - by chance?

Steve: Yeah. And then I was - and it raised the question, is this training? And then I worked out a way of verifying that in fact I am training at that point, and it never gets tired.

Leo: So our previous recommendation was to use all of the possible fingers, but do

it with the same finger so that you give it more datasets for that same finger. But this would be a superior way to do it.

Steve: Oh, absolutely, because you have - somewhere there's a geometric model of an individual finger's print. We don't know anything about the way they've built it, but it's doing some sort of recognition and feature extraction, and it builds a feature map. And what you'd like for greater reliability is two things. You'd like that the feature map for the region you use typically to, like, have all the noise removed and to find all the relevant features that are available. And so you get that just by giving it more chances to read. And also, for example, under different conditions, when it's colder, when it's drier, when it's more humid and so forth. So this allows you to just keep going back any time you want to and give it some more samples.

But the second thing you'd like to do is to expand the size of the feature map so that it does incorporate out further out the end of your thumb and around both sides and maybe back further, just so that when you put your thumb down, you don't have to be as careful about giving it exactly the same spot on your thumb. And so this allows you to deliberately expand the feature map by just sort of walking the map out to the periphery of your thumb. And it'll follow you as long as it still recognizes it. It'll say, oh, look, I got 80% of what I really have already seen. And here's an extra 20% slice that I'm going to now extend the feature map out in order to incorporate. And you can play with it, and it all works.

Leo: You have discovered - you've got to call, call the media, alert the media. Nobody knows this. This is great stuff. That's a - what a discovery.

Steve: Yeah, every so often we come up with new stuff on this podcast.

Leo: Wow. All right. Well, I wish I'd known about this a few minutes ago on MacBreak Weekly. I would have told everybody. But I'll tell them next week and give you credit. That's great.

Steve: Yeah, cool. So thanks to Simon Zerafa, our friend and contributor in Wales. He found two more new samples of CryptoLocker which are, well, they're very different. Clearly CryptoLocker, they're one third the size. CryptoLocker used to be - the samples I was already hosting on the malware page at GRC for people who wanted to experiment, they were on the order of 900K, as I remember. These are - I think it was 900. And these are, like, 300K. So it's interesting that they're that much smaller. I don't know if they're just - if they've just run it through a compressor or what's going on. But they are new samples. They are the real CryptoLocker. And they are very poorly detected by any AV right now.

So if we've got people, I know that my samples that I was offering have been very popular. It's just at GRC.com/malware, and you can add .htm if you want. And so I posted those a few days ago, and I wanted to notify people that they are there because I know that we've got people in, for example, in corporate infrastructures and IT that were using those to make sure that whatever AV tools they're using are detecting those; or, if not, see about making that happen. And I think, for example, when I posted this, one of them was only detected by six out of 40 different AV tools. So way back down at the

bottom again. And I remember last week, Leo, you were noting that CryptoLocker, the old CryptoLocker, was now being seen by all of them. So it was being blocked by everybody.

Leo: Yeah, that's kind of how it happens; right? And then they modify it and...

Steve: Yes. And that it is the problem is that it's not the specific characteristics necessarily, but just, unfortunately, signatures that are being used. And as soon as the malware guys see that they're being blocked, they are able to engineer around it. I mean, this is the problem we still have, that unfortunately our model is block what's bad rather than own what's run good. Someday, at some point, that's where we're probably going to end up being. And I've used the example of firewalls.

Leo: A whitelist instead of a blacklist.

Steve: Exactly. And to some degree the iOS Store is a curated applications store. We see examples of stuff sneaking by. There were stories last week we didn't cover because it wasn't really too much on topic. But what was it, it was people were - existing apps were being sold to other companies that were then using them to embed ads in them. And so that I think it was in the Android Store they would update the app, and now suddenly it became like, wait a minute. It's sort of like...

Leo: It's Chrome plugins.

Steve: It's like letting a domain go bad, and now it's turned into just a junkie search engine thing.

Leo: It's kind of worse than that because what happens is, if a Chrome plugin becomes successful, the bad guys come along and offer the developer money to buy it. And the problem is that Google does not, once you have got a Chrome plugin...

Steve: Approved, right.

Leo: ...on the store, they don't then check again. And so you can then push updates to this Chrome plugin at will, including whatever you want, malware, whatever.

Steve: I think maybe I should have covered that last week. Should have looked at it more closely.

Leo: It's pretty nasty, yeah. And I think Google needs to respond to it, frankly.

Steve: Yeah. So I said without evidence last week, I made the comment that the NSA's phone metadata collection was ineffective. I was hearing that from the various

discussions that I follow, but I didn't have any specific evidence. Well, on Thursday, two days after the podcast, the government's review panel that was looking at all this finally issued their statement. They warned that the National Security Agency's daily collection of Americans' phone records is illegal, in their opinion - and of course judges have been going back and forth on this so far - and recommended that President Obama abandon the program and destroy the hundreds of millions of phone records already collected.

Quoting from their report, they said: "In addition to concluding that the daily collection of phone records is illegal, the board also determined that the practice was ineffective. Quote, 'We have not identified a single instance involving a threat to the United States in which the program made a concrete difference in the outcome of a counterterrorism investigation,' and added, 'We are aware of no instance in which the program directly contributed to the discovery of a previously unknown terrorist plot or the disruption of a terrorist attack.'"

So it said: "The NSA should instead seek individual records relevant to terror cases directly from phone service providers under existing laws." And of course that's my only suggestion for something that would be effective is, as I said last week, we require that the carriers maintain records. If they want the privilege of having access to FCC-allotted bandwidth and common carrier status, then what comes with that is the obligation to maintain metadata records and then require individual requests. And I did note also, I think it was Verizon that finally published the number of requests that they had entertained. Oh, lord, it was like 370,000 requests in a year. So that's, what, a thousand a day, about. So it's like, yikes.

But still, arguably, I mean, maybe what that would do is raise the cost of performing this collection to the point where, if it's really not generating any value, they would just stop bothering because it's like, well, okay, we're having to jump through bigger hoops now. So we'll just abandon doing it because it's not providing any value.

And the slide you referred to, the NSA slide that was just revealed, was another one of these wacky-named projects, this one Golden Nugget. And everybody picked up on the news. TechCrunch's headline was "NSA May Want Mobile Data Including Info From Angry Birds and Maps." And of course Rovio, the publisher of Angry Birds, vigorously denied cooperating with the NSA. And I'm sure they're not. It very much looks like we now understand that this is all extrinsic acquisition of data on the Internet. And as I've said, the NSA is full of smart people. If it can be done, and they want it, and they've got the budget for it, it will be done. And so they're just doing everything they can.

And so they've got smart guys sitting around thinking, well, what about social networking apps? Well, in fact that's our next story is that the Facebook update - this is not NSA related, as long as the NSA can't get their hands on it. But Facebook's update to Android now requires permission to access all of your text messages.

Leo: Oy.

Steve: I know. A screen was posted in the story that first brought this to light. And under "App Permissions," that comes up when the Facebook app updates, it says "Facebook needs access to additional permissions," and then it lists them. And under "Your messages" it says "NEW," in all caps, "Read your text messages (SMS or MMS)." And down further, it sort of scrolls off the bottom, but under "Your personal information," also "NEW," this was similarly disturbing. It says "Add or modify calendar events and send email to guests without owner's knowledge. Read calendar events plus confidential

information. Read your own contact card. Modify your contacts, read call log, read your contacts, write call log...." And then it kind of - it keeps going, but scrolls off. And I'm looking at that going, oh, my lord.

Now, so note that here's certainly equivalent plumbing of personal information that a commercial enterprise is doing that echoes what the NSA is doing. So, and my point is it's been noted that, yes, we're all up in arms over what the NSA is doing. But in fact we're giving permission to corporations to do very much the same sort of aggregation and social engineering.

Leo: Yeah, and it's usually a tit-for-tat thing. I mean, these are - so, I mean, you see this all the time in these apps. It's nice that Android gives you the list of new permissions before you accept it, and you can read them.

Steve: Yes. Well, and Facebook cannot knock on your door and say that we think that maybe you're up to no good, and we're going to take you away.

Leo: Yeah, they don't have guns.

Steve: Precisely.

Leo: Don't have tanks. They don't have black helicopters. But the other thing is that the way it's set up, a lot of times you ask for one permission, you get a block of those, and all of those have to be disclosed. That's part of the way the API works. The other issue is - and what I suspect is this is Facebook in effect saying we want to add some new services, or we want Facebook Messenger to use text messaging in that case. And so these are things that are features that people presumably want. And you can always say no and not install it. I mean, that's...

Steve: Right.

Leo: There's an opt-out.

Steve: Although it is an all-or-nothing.

Leo: Yeah. I wish it were more granular, and it's not.

Steve: Yes. It would be nice if they had check - and we've discussed this before - if they had checkboxes on the things they want access to, and if they explained to the user why they want it. There's no "why we want this" as part of that. It's you must consent to this laundry list of permissions so that Android, the OS, will give us access to those aspects that are otherwise sandboxed from us. And so first they don't tell you why, and then they don't allow you to say, well, I'll take this one and this one, but I'm not giving you this one.

Leo: Yeah, I mean, that's troubling, too, because, I mean, it's difficult for them to do because if you say, yeah, I'd like this SMS feature, but you can't have this, the whole feature breaks. And I also understand why they don't give you a lengthy explanation, at least on that page, because nobody wanted to piss people off to say, I don't want to read all this, I just want to - get out of my.... But they should somewhere. Somewhere they should say, and if you want to know more about what we're up to, we have a page here that you can read. I agree, that kind of disclosure would be very valuable. It should be required.

Steve: Okay. So center yourself, Leo.

Leo: I'm sitting on my ball carefully here.

Steve: Over your ball. And listeners, when you hear the phrase "Bluetooth LE 'Just Works' Pairing," when something is called "It Just Works," this is where you're entirely expected to have what we're now calling a "Gibsonian reaction."

Leo: It doesn't pair at all, frankly. It just works in the sense that, if you have a Bluetooth LE app, it works. There's no pairing at all.

Steve: Well, there actually is.

Leo: Oh, is there.

Steve: Oh, yeah. And in fact, I mean, they went to some trouble. For example, I found an interesting page at the nih.gov site that says: "Pairing comprises three phases." Now, we should mention, Leo, what you're referring to is the experience, and you're completely correct. That is, the experience is that it just works.

Leo: Yeah, it's all invisible to you.

Steve: And we've discussed Bluetooth pairing. I've explained carefully exactly how it works and that the way it was originally designed was secure except during a brief sort of theoretical window where, if you really wanted security, you needed to go out into the middle of an empty parking lot somewhere so that you could see all the way around yourself and knew that nobody was sniffing. The reason that's important is there are, for example, very powerful Bluetooth radios with long antennas on them...

Leo: Steve's showing one right now.

Steve: ...which hugely extend the range at which Bluetooth will function. And the hackers know about that. So on this NIH.gov site, they say: "Pairing comprises three phases. In the first phase, the two connected devices" - now, they're not actually

connected yet, but want to be connected devices - "announce their input/output capabilities; and, based on these, they choose a suitable method for the second phase. The second phase has the purpose of generating [what's called] the Short-Term Key (STK), which will be used in the third phase to secure the distribution of key material. In the second phase, the devices first agree on a Temporary Key (TK), by means of the Out Of Band, the Passkey Entry, or the Just Works methods. The Out Of Band method uses out-of-band communication means." Now, that could be NFC, or it could be the typical Bluetooth style where something with a screen shows you something which you then enter into the other device.

Now, of course, that requires that there's a screen on one side and a keyboard on the other so that you're able, you the human are able to move the out-of-band information from one device to the other in a mode where no one eavesdropping can know what that out-of-band information is. That's the point of it being out of band. Or, for example, NFC. Assuming that NFC requires near contact, then that would be good. You sort of tap these things together. That allows them to exchange the very short-range secret, which they then leverage into a longer range secret by going out of band. But that's expensive, too. So what ends up happening often is that Just Works is used.

So continuing the NIH thing, they say: "The Out Of Band method uses out-of-band communications means for the TK" - that's the temporary key - "agreement. In the Passkey Entry method, the user passes six numeric digits" - so that's also out of band, but it's sort of formally defined as six numeric digits. Now, okay, turns out that's not enough digits, as we'll see in a second - "as the TK" - as the temporary key - "between the devices. When none of the first two methods can be used" - I would have written "either" at that point, but that's all right - "the Just Works method is employed" - and this is where we have our Gibsonian reaction - "although it is not authenticated, and it does not provide protection against man-in-the-middle attacks. Based on the TK, and on random values generated by each pairing device, the STK is then obtained by both devices, which leads to the end of the second phase."

So they just sort of gloss over that minor problem with Just Works pairing, correctly noting that it provides no protection against man-in-the-middle attacks. It turns out that it's worse than that because a man in the middle typically means you are intercepting, that is, you are - they mention it's not authenticated. Well, that's true. And the typical man-in-the-middle attack allows the person to insert themselves and, if there's no authentication, then they're able to establish communications against each of the other ends. And the other ends each think they're talking to each other when in fact they're talking to the man in the middle. That's the vulnerability with no authentication. That's what, for example, SSL certificates prevent because the server that you're connecting to is authenticated, and no man in the middle has the certificate that the actual web domain and web server has. So it's the authentication aspect that prevents an active man-in-the-middle from splicing themselves in. It turns out this is subject to passive eavesdropping, which is where the real concern comes in.

So I went then to Bluetooth.org to look at, like, what do the actual Bluetooth guys who developed this say? And under "Association Models," where you're associating the two endpoints, they say: "Bluetooth Smart" - which is their formal name for the low energy technology, which turns out to not be so smart - "uses three association models referred to as Just Works, Out Of Band, and Passkey Entry," as we just learned. "Bluetooth Low Energy technology does not have an equivalent of numeric comparison. Each of these association models is similar to Secure Simple Pairing with the following exception: Just Works and Passkey Entry do not provide any passive eavesdropping protection. This is because Secure Simple Pairing uses" - and here it is - "Elliptic Curve Diffie-Hellman." That's the good way to do this kind of key agreement. That is, so full-strength real

Bluetooth, the Secure Simple Pairing, uses Elliptic Curve Diffie-Hellman key agreement.

Leo: Well, that's good.

Steve: Yes. Whereas Bluetooth Smart, which we now need to put in air quotes, does not. The use of each association model is based on the I/O capabilities of the devices in a similar manner as Secure Simple Pairing. So they sort of gloss over that at this point; but they do note that, for whatever reason, they didn't build Elliptic Curve Diffie-Hellman into Bluetooth Low Energy. Maybe they were just - their target was really, really, really low cost. And we know that these Bluetooth LE things are really, really, really low cost.

So finally we get to the guy that cracked it. And this is actually not even new. This is nearly two years old, I think, that this was done, but somehow just sort of escaped attention. So Mike Ryan put together a paper titled "Bluetooth: With Low Energy Comes Low Security." And after laying down a foundation of, like, all of this, he built a proof-of-concept system based on a standard, essentially sort of a software radio-style Bluetooth dongle, a USB dongle. And so he says in his paper, down on paragraph or section 6, says: "BTLE [Bluetooth Low Energy] features encryption and in-band key exchange." As opposed to out-of-band, which we were talking about, where you deliberately go out of band so that somebody who is eavesdropping in-band, that is able to capture the packets that are being exchanged, can't see what's happening outside of that band, out of band.

But BTLE features, as Mike writes, "encryption and in-band key exchange. Rather than relying on a well-established key exchange protocol such as one based on Elliptic Curve Diffie-Hellman" - which, by the way, for example, is what SSL and TLS now is using, and made it ephemeral so that you're always renegotiating keys because it's easy to do that now, and that's where we get our perfect forward secrecy, so they don't have to worry about anyone capturing certificates and being able to decrypt communications in the future that were stored. Rather than that, he says, "the Bluetooth SIG" - which is the formal definition guys - "invented their own key exchange protocol."

So this is where we go to, yes, we never learn these lessons. We have perfectly good, well-established, secure solutions. But no, we're going to invent another one. We're going to invent our own. So Mike says, using the royal "we": "We demonstrate that this key exchange protocol has fundamental weaknesses that undermine the privacy of communications against passive" - and he has in italics for emphasis - "eavesdroppers." Meaning all you have to do is capture the packets. "We note that the session encryption provided by Bluetooth Low Energy is known to be relatively secure. BTLE uses AES CCM" - now, he's talking about session encryption, not establishment, he says - "against which there are no known practical attacks." So that is to say, once you get a secure key agreed to, then the session is well encrypted. The problem is, how do you establish that initial key?

He says: "Our attack targets the key exchange rather than the encryption itself. Our technique is similar in principle to" - and he quotes a couple - "in which an offline brute-force attack is mounted to recover a secret value when all other values are transmitted over the air." And that is the case here. So he says: "Before establishing an encrypted session, a master and slave must establish a shared secret, known as a long-term key. Under typical operation, a master and slave establish a long-term key once and then reuse it for future sessions. Otherwise, the master and slave establish a long-term key through a key exchange protocol."

And finally he says: "The key exchange protocol begins by selecting a temporary key, a

128-bit AES key whose value depends on pairing mode. The master and slave use this value to calculate a so-called 'confirm' value. Aside from the temporary key, all values used to calculate the confirm are exchanged in plaintext over the air. The confirm value itself is also exchanged over the air in plaintext." And this is the problem with the protocol. Mike says: "We exploit the fact that all values except the temporary key are publicly known in order to brute force the temporary key. As noted, the temporary key value depends on pairing mode. Three pairing modes are defined: Just Works, a six-digit PIN, and Out Of Band."

The temporary key is as follows: For Just Works, it is zero. It's nothing. It's null. For the six-digit PIN, a value between zero and 999,999, padded to 128 bits. So that's where they get this 128-bit key is it is just literally the six-digit PIN padded out to that bit length. And then, if you have out-of-band exchange, for example, we were talking about near field, you would use that to exchange a true strong 128-bit value. So that's going to be super safe. The problem is, if you use a six-digit PIN, well, then you've got a value between zero and essentially a million, 999,999. But if you're just in Just Works mode, then they don't even try. It's just zero.

So he says, finishing this, he says: "We use a simplistic brute-force algorithm to guess the temporary key. We calculate the confirm for every possible TK value between zero and 999,999. If the master and slave used Just Works or a six-digit PIN, we will quickly find the proper temporary key whose confirm matches the value exchanged over the air." So essentially the confirm is exchanged in plaintext, making this trivial to brute force. And he says: "In practice, we find that a temporary key can be cracked in less than one second on a single-core Intel i7 CPU. This figure could be improved by brute-forcing in parallel and/or using processor-specific AES extensions," none of which they even bother with, making this an absolutely practical attack. Once that's done, then the temporary key is cracked, and you use that to leverage the agreement of the long-term key, meaning that the entire future interaction of those devices is then crackable.

So what does this mean? Well, as we know, LE is often used for things you really don't, whose security you really don't care about, like temperature sensors or all those little tracking badges that people are coming up with now. Or iBeacon, for example, Apple's iBeacon. It isn't actually exchanging any valuable data over that link. It's only saying I'm here, or I'm a pair of red shoes, who knows. It's arguably not very important. The problem is that this is the way these things start.

So, for example, that interesting Coin credit card is also using Bluetooth Low Energy. And the question is, did they do, do they do a full 128-bit temporary key, or are they displaying, and I'm afraid they probably are, a six-digit PIN? And now we know that, if you were ever passively recording the traffic during the pairing of that card, the Coin card, using Bluetooth LE, where you could argue in fact there is valuable information now being exchanged, and they use, for user convenience, a six-digit PIN, that communication can be completely cracked.

So the concern here is that we're going to start seeing applications using Bluetooth LE or Bluetooth Smart, which isn't very, where unfortunately, for user convenience, they've sacrificed the full-strength, 128-bit key exchange, and it turns out it's possible due to the fact that they did not use good Diffie-Hellman elliptic curve encryption. They just used their own protocol, where everything is in the clear, and a six-digit PIN can be cracked in under a second easily, and much more quickly if you develop some software to do so. So listeners beware.

Leo: But it doesn't have to use this less secure system. There are three systems, and some are more secure; right?

Steve: There's one system which gets its security from exchanging a full 128-bit key.

Leo: And that's secure.

Steve: And that's secure. But that requires that you actually arrange to exchange a 128-bit key. The problem is, that's inconvenient. Or, for example, if it uses near field technology, then it's trivial to generate a random 128-bit key on one side and give it to the other. And then that will be out of band. But the problem is that we're all - everyone's trying to minimize expenses. And I'm signed up for Coin, so I will find out how you do the Coin pairing. The problem is Coin uses Bluetooth LE. We know that. And absent high security out of band, you are reduced to using method number two or one, which is either zero, a key value of zero, which you could instantly verify; or you do a brute-force attack, and you're going to guess the PIN in less than a second.

So the problem is it's like, yes, you could pair securely. And if you pair, again, where you are absolutely sure no eavesdroppers can intercept your communications, then you're secure also. Once you've established your pairing, then you've agreed on a secure key. And that's never exchanged in plaintext. But it's that first instance of pairing that unfortunately Bluetooth LE is very vulnerable, much more so than regular Bluetooth because regular Bluetooth uses a strong cipher. It must have been that they said we can't afford the cost to do elliptic curve crypto in a temperature sensor that you're going to stick on the wall. We want Bluetooth LE to have a wide application...

Leo: And who cares if they know what temperature I have.

Steve: Precisely.

Leo: I guess my response to this would be that it may be cost, but I think it also may be looking at how people use Bluetooth. It's been such a pain to pair with Bluetooth that almost every device now just says it's 0000.

Steve: I know.

Leo: They don't attempt, even though the security is possible, they don't attempt it. And that's become de facto how you pair stuff. It's rare that I have to use a number; and, when I do, I'm pissed off. In my car I have to. And I don't care, oh, my god, somebody might actually add their phone to my car's list of accepted phones is not an issue for me. So I think a lot of the times we use Bluetooth, as with a thermostat, it's probably not an issue. It's a big issue if Apple uses Bluetooth LE for its payment system because they don't have NFC on Apple, or Coin, as you used as an example. Of course, I didn't buy a Coin. I thought it was stupid from the beginning.

Steve: Well, and I do, too. But I just need to - I've got to take mine, I have to take mine apart.

Leo: You have to. So I would guess that the people who designed this recognized that there are lots of situations where zero security is appropriate, and it's going to make everybody happy. I use Bluetooth LE for my little slot car set, that Anki that we played with. There's no reason I should worry about a man-in-the-middle attack on a remote control. It doesn't make - it's not an issue.

Steve: Yeah. The TiVo Roamio uses Bluetooth now, and it's very convenient to have...

Leo: You don't want to have to enter a four-digit code to pair your remote control with your TV. That's crazy.

Steve: Yup. And it's funny, too. It comes up in pairing mode. And the first time you put the batteries in and plug the TiVo in, and you press any button on the remote control, they immediately find each other, do this exchange, a Just Works exchange, and now they know each other.

Leo: Yeah. So I think that it makes sense. In fact, I think this is a sensible response to have three levels of security, or really sounds more like two effectively, good security and no security.

Steve: Right.

Leo: And then count on implementations to choose the right thing. And by the way, you could use a QR code if you didn't have NFC. So it's conceivable that Apple and iPhone might pass this out of band.

Steve: That's a very good point. For example, the Coin folks could print a static QR code, a random QR code on the back, or on a sticker. So it's on the sticker. You scan it once, and then you peel the sticker off so that nobody else can - and hide the sticker somewhere safe.

Leo: The places I've used Bluetooth LE, pairing to a scale, pairing to my slot cars, it's been great because there is no - it doesn't look like there's any pairing going on at all. In effect, it sounds like what they're doing is what we would do manually, which is 0000, and we just eliminate that whole process.

Steve: Well, and that's what Just Works pairing is, is literally all zeroes. And so you could instantly crack their attempted encryption. The problem is, and my concern is, there is this presumption that there was an attempted encryption. And so I just wanted our listeners to know, for now and forever, that that's useless, and don't assume you're being protected. And Bluetooth LE is, I mean, it's wonderful. I learned a little bit more about it. For example, I was always sort of curious, how could you have Bluetooth,

because I've always thought of it as being a static association, that is, that it had to be a powered-up link. It turns out that is not the case, that the little TiVo remote control I was holding up a second ago, when you press the button, it sends off a little Bluetooth burst. So they've managed to get this thing so that it's able - the radios come up, they find each other, they handshake, they link, they exchange their message, and they shut down again.

Leo: Isn't that clever.

Steve: Which is just like an IR burst does. It's fabulous.

Leo: So the real, I would say, the real takeaway on this is, on things where you do want it to be secure, you should ask, how is the pairing happening?

Steve: Correct.

Leo: And you're not using the easy one, you're using the secure one, right.

Steve: Correct. And we will find out how...

Leo: It's up to the implementer to do it right.

Steve: We will find out how Coin pairs. And in fact, if Ryan Seacrest ever sends me my keyboard, which, you know...

Leo: Is that supposed to be LE?

Steve: Yeah, it's Bluetooth LE.

Leo: Oh, interesting.

Steve: Yeah. And so the question will be, I mean, there again, I'm sure it just will automatically find my phone, and they'll be happy.

Leo: Yeah.

Steve: Following up a little bit on Precheck, TSA Precheck, because so many - this captivated the interest of our audience because people think, wow, I'd like to have that. I did receive, the day after the podcast last week, so that is to say exactly one week after my TSA Precheck appointment, I received a letter from the happy Pre people. And what I received was my eight-digit Known Traveler number. So that's what I have now. I have a

good-for-five-years Known Traveler number. And I presume, when I am in the future making reservations on airlines, there will be somewhere I can put in my Known Traveler number, and then I'll get - my boarding pass will say TSA Pre.

One of our listeners, Martin Ruby, commented. He said: "Steve, I've been a TSA Pre flyer since the days when it was a test. There are two ways to get TSA Pre. The first is the way you did, by applying for it. The second is by being an airline frequent flyer. If your airline participates in TSA Pre, they can submit your information to the TSA, vouching for you as a Known Traveler on that airline. The difference between the two is when you apply to the TSA, it is good on any airline that participates in the program; and, if it's by the airline submitting your information" - I should have edited this better. It says: "But if it's by the airline submitting your information," he says, "it's only valid on that airline."

Then he says: "I'm a proud owner of SpinRite since the days of OCIPUG." That's an acronym I haven't seen for a while, Orange County IPM PC Users Group. I heard you talking about user groups, by the way, Leo, in the earlier - in MacBreak Weekly. And, yes, those were fun times back then. He says: "...when you would come and present." And he says: "I've used SpinRite many times to fix drive issues. Thank you for a great podcast and a fantastic product. Martin."

And then the one that I got a kick out of, looking through the mailbag this morning to put together our Q&A, was someone who said re TSA Pre screen: "Steve, it's almost certain you have a file and a set of agents assigned to you. Of course they don't need to interview you. The file probably says, 'Smart guy, mostly harmless.'"

Leo: Mostly harmless, I like it.

Steve: "Smart guy, mostly harmless, best not to annoy him."

Leo: And never say "It just works" 'cause he'll get you.

Steve: Yeah. Don't just tell me, oh, yeah, the key agreement, don't worry, it just works. So, two cool security sites. Leo, you're going to want to bring this one up.

Leo: All right.

Steve: HowsMySSL.com, h-o-w-s-m-y-s-s-l dotcom. Type it in quickly before we bring the site down. Oop, and there it us. So we're seeing on your screen, "Your SSL client is improvable." I assume you're using Safari.

Leo: Safari, yeah.

Steve: Because in fact, in my notes, Safari says it's improvable. Firefox, "Your SSL client is bad." So, first of all, back off a little bit, or I'll back off a bit. What this is doing is something actually I started to do for GRC, but I had other priorities, and everyone would agree. So I thought, well, okay, I'll either leave that to somebody else or do it later. And it's been left to somebody else: HowsMySSL.com.

What this does is it creates a web page based on the details of the initial SSL handshake packet. Remember from our discussions of this that the first thing that happens when your browser connects to a web server is it establishes a TCP connection through the three-way handshake. Then the first packet is a packet saying I want to bring up an SSL tunnel. Here's my SSL versions and my cipher suites. These are all the cipher suites that I know about. So what this site simply does is publish, essentially make it very human-readable, publish the details of what your client browser is offering. So this is sort of the reverse of the SSL Labs test. SSL Labs will test the server on the other end of your browser connection. This one checks your browser.

So Firefox, it says, is bad. I don't think the people at Mozilla will appreciate that very much. And I don't really know that I really agree, either. Firefox gets a "bad" because they support TLS 1.1, not 1.2. It's like, okay, well, yes, there is a 1.2, but 1.1 is fine. But it really upsets this site because listed among the cipher suites, way down at the bottom, but still there, is a suite that they don't like. It's SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA. So in the parlance of cipher suites, it's chosen a set of cryptography which is like, eh, okay, so it's not super spiffy. They said: "This cipher was meant to die with SSL 3.0 and is of unknown safety." Well, yes. It's not of known unsafety. And it's not actually...

Leo: Of unknown safety.

Steve: It's not actually even unknown safety. It's like, okay, 3DES. That's like the Edsel. But still, it's okay. So anyway, Firefox gets a "bad." Now, the reason this is not such - the only way this could be a concern is if you - I guess it would be a cipher suite downgrade attack where a man in the middle saw all of those going off to a server and removed all but that one, and then the server also supported that and then grumbled and said, okay, fine, we'll use that. Well, even then we don't know there's a problem because this is not of known un-security. It's just that, unfortunately, HowsMySSL site is in the business of being critical. And so they said, okay, we don't like this suite any longer.

Opera got "SSL client is bad," the latest version of Opera. And we know that Opera is very secure. But they only support TLS v1.0. And they don't support TLS session tickets. Session tickets are the TLS version of caching. We've talked about SSL caching, where after performing the expensive public key encryption, essentially you resume a session on subsequent connections by saying, hey, here's my session ID from the prior session. If this is still in your cache server, then let's just reestablish our agreed-upon crypto material. And that's very effective. Session tickets is the terminology and the way that's being done for TLS, and Opera doesn't support it. Neither does IE 11.

I got a better grade on IE 11. It says "Your SSL client is improvable." So it's not bad, as is Firefox and Opera. It's improvable. It's like, okay, well, they're all improvable. So they had to come up with a term for, like, okay, not in the gutter, but we're not really happy, either. IE 11 got improvable. Safari got improvable. The only thing that I have seen got - and this is the best they'll give. They'll begrudgingly say that Chrome "is probably okay." It's like, okay. So it gets a "good" on everything, and so it's probably okay.

Leo: Probably okay.

Steve: So anyway, I thought - I knew that our listeners would get a kick out of

HowsMySSL [h-o-w-s-m-y-s-s-l] dotcom. And I've seen other people asking me, Steve, what's going on with that TrueCrypt audit?

Leo: Oh, yeah. I meant to ask you about that last week when we talked about TrueCrypt.

Steve: Yup, and there's a handy-dandy site to answer the question, IsTrueCryptAuditedYet.com. And the answer is not yet. But the project is getting good support. There is progress being made, and it's looking good. So all run together, one word, IsTrue - and, by the way, it's T-r-u-e Crypt, and don't drop the "e" - IsTrueCryptAuditedYet.com. And the answer is not yet. But if you want to, you can make a shortcut or something to check in on that from time to time. And if the status changes, by all means, someone notice that and tweet me because I would love to know.

I mentioned last week, I was working on the SQRL page to document the use of Scrypt. That's Colin Percival of Tarsnap's memory hard password encryption technology, or password hashing is really the better term because it's not reversible. I called it EnScrypt because we EnScrypt the password using essentially an iterative use of Scrypt. I did that because Scrypt, even if you give it a lot of memory, still is too fast. We really want to penalize a bad guy who is doing password guessing. The only known attack on this is trying all possible passwords. And so while, yes, what would really be good is if users chose pure entropy from, like, GRC's passwords page or something, most people don't do that.

And so the only thing I can think of, the only thing feasible, is to make any wrong guess so painful that an attacker - actually I think that SQRL is going to get a reputation instantly for not even being worth attacking because everyone's going to know you can only do one every, well, in the case of a statically stored password, I'm proposing that you have to take 30 seconds to encrypt this in order to export it. And they just won't bother.

But anyway, the point is that what I ended up writing was much more of a tutorial than I started out to write, which I think would be of great interest to our listeners. That is, we have discussed password encryption and hashing and management in the past, like adding salt, and storing the salt with the password, and adding iterations of a hash in order to make it take longer. What I ended up writing on that page - and it's grc.com/sqrl/scrypt, and it's actually page 10 of 18 on the SQRL sort of subsite. You can also just go in the menu, so find SQRL under Research, and then it's page no. 10 is SQRL's use of Scrypt.

I ended up sort of taking the reader through the entire history of how passwords have been handled through time, which I think our listeners would probably enjoy reading. So I recommend it for that. And I think you'll find it interesting also. And I expect this will end up moving into common use. Oh, and I also mentioned that I'd written software. It's there. There's a screenshot of the software. You can download my reference version of EnScrypt, which does this, which is sort of a fun little benchmark. You can say "Run it for five seconds" or "Run it for 30 seconds," and then it will tell you how many iterations of Scrypt your system is fast enough to perform. And that really tells you a lot about the speed of your processor, but mostly about your memory bandwidth, the throughput to main memory, because this is all about busting the caches in order to force main memory access. That's something that GPUs don't have.

GPUs can have a couple gigs on the graphics card, but they don't have high-speed access

to it. They've got high-speed access to local cache, but this uses 16MB of memory, forcing them to go outside of the cache, down in the main memory. And that's what levels the playing field and prevents the acceleration of this password hashing that we see, for example, famously in Bitcoin, that just uses SHA-256. And so now there's ASICs, which are just screamingly fast at doing this. The whole goal was absolutely prevent that from being done with SQRL's passwords. And I think we've achieved that.

And just I had a short little shout-out from a James Cavanaugh who's in Dresden, Germany. He said: "I just wanted to say thanks for recovering my apparently dead drive's data when nothing else could. I tried so many other programs, unsuccessfully. And after all of them, SpinRite's apparent ease of success really convinced me. What a difference. Also intuitive and easy to use." He says: "It just does the job. Great product." So, James, thanks for saying so.

Leo: Well, we all agree with James. All right, Steve. I have in front of me, in my hand, questions from our listener-driven potpourri. Are you ready?

Steve: You bet.

Leo: Shall I fire away? Here we go, starting with Rick Lieb in Monessen, PA. He wonders about a safe way to keep using XP after the April cut-off: Steve, I've watched and read every episode of Security Now! since the beginning. I'm a SpinRite owner. Just upgraded both computers to Windows 7 Pro 64-bit from XP. In order to keep using some software in XP that will not run any other version, I'm using the XP Mode Virtual Machine. This older software does not need to access the Internet, by the way. Once support stops, my plan is to disable networking in the virtual machine so it can't access the Internet. I'm very careful about the media I hook to the computer. So is this a reasonably safe thing to do? Rick Lieb, Monessen, PA.

Steve: Yeah. Of course I'm famously still using XP. And I did note that April 8th, the last day of support, happens to fall on a Tuesday. So it couldn't be more appropriate.

Leo: Maybe they'll Patch Tuesday us one last time.

Steve: It will be a - oh, yeah. It will be the second Tuesday of April, also.

Leo: That's probably why they do it. They're going to give you one more update, and then that's it.

Steve: One final sendoff. They didn't want to do it on April Fool's because that would be a problem. So, yes, you're right, the second Tuesday of the month. And it happens to be it'll be a podcast day. So that will be fitting. But I would say disable it now. Well, no, I guess you still want to keep - you want any updates that XP will get for the next several months. That is, by the way, 69 days from now, just for those of you who are counting. And, yes, once XP is in its final resting state, then you could disable its access to the Internet.

I guess the problem would be if you downloaded new software that required XP mode, that would then be able to leverage a problem with the XP Virtual Machine. What's not clear to me is what mischief it could get up to. Now, again, the bad guys are clever. Maybe they'll figure out some way to do something. But, for example, if you had the typical containment of a virtual machine, then it could mess with your virtual environment, but perhaps not anything outside of that.

So I have a feeling I'll be following in your footsteps because I'm sure I've got software. I'm sure Brief, you know, I'm still using the Brief text editor in a DOS box. And I'm sure it's going to choke. Someone said that Eudora was still functioning under Windows 7 without any trouble. So although I don't think the 64-bit version of Windows 7, which I would want to use because I would want access to more memory, I don't think that's going to run Eudora. But anyway, we'll find out. But certainly, yes, disabling networking would isolate the virtual machine. You may want to turn it on from time to time specifically to receive the Security Essentials updates, because those are going to continue to flow through sometime in 2015, so like maybe one more year worth of those, just because. And then definitely shut it down.

Leo: Gregg Keizer writes at Computerworld that Microsoft is going to continue to update the Malicious Software Removal Tool, MRT, through July 14, 2015. So another 15 months of MRT. That's the tool that normally operates completely silently. You can invoke it if you know the command line. But normally it just runs in the background. And it's not exactly antivirus, but it is a removal tool.

Steve: Yes, in fact, I think...

Leo: And of course all third-party antiviruses will see this as an opportunity. In fact, I'm surprised that some company hasn't stepped forward and said, hey, we'll lock you down. I mean, they don't have the source codes, but at least they could, I don't know, offer some sort of service.

Steve: Well, and remember that what we're going to be seeing are problems not unique to XP. There are problems in Vista and Windows 7 and Windows 8 because of the common code base that goes all the way back, actually back to NT. And so what Microsoft is saying - and this is what annoys me. I mean, I understand their need, but they're saying even though the same things we are fixing in Vista and 7 and 8 also affect XP, we're still not going to fix XP for you. We're just going to say no. So it's like, eh.

Leo: And of course you wouldn't want to leave your machine on the 'Net to download the monthly Malicious Software Removal Tool update, but they do offer it for download on the Microsoft site. I think he's smart. I think getting off the 'Net after April 8th is probably the best thing to do.

Steve: Well, and, for example, most people get into trouble by surfing in a browser connected to the Internet. So I would say don't surf in your XP virtual machine. And as he said, the software he needs to keep using, this database or something, has no need to access the Internet anyway. So, yeah.

Leo: No problem. Edward in Daytona Beach wonders about memory hard encryption: The memory hard encryption technique, any known hardware platforms that would wear out flash memory used as main memory? Is there similar potential for wear?

Steve: It's an interesting question that really hadn't occurred to me. I had to think for a second, is there any platform that uses flash for main memory?

Leo: Not that I know of.

Steve: And I'm sure the answer is no. Flash, we know that flash fatigues because writing to flash, as I have explained before, essentially forces - uses high voltage to force electrons across a deliberately insulating boundary or barrier, and in the process fatigues that barrier. So all the flash that I'm aware of is not scratchpad. And you definitely, for example, don't want to set up a swap drive on flash memory because that'll burn it out, too. But mostly it's that it's so slow. So nothing wants to use flash as main memory because it really does take a lot of time in order to write to flash. So it's always used as, I mean, in every instance I could think of, and Leo, you would concur...

Leo: Might be some low, low-speed embedded tool that might use it. But I can't - doesn't seem likely.

Steve: Yeah. And in any event, nothing that you'd be running a SQRL client on. So I think you're going to be safe.

Leo: I think you're safe. So in other words, the memory is not storage. The memory is RAM.

Steve: Yes. And it is volatile by nature.

Leo: Right. Michael in Brisbane, Australia wonders about femtocell security: Steve, I recently disconnected our VOIP and switched to just mobile phones at home. However, my wife has on-call shifts, and we had some black spots in the house where there was no mobile reception. The phone carrier, Optus, provides us with a femtocell - they call it a Home Zone - which I've attached to solve our home's coverage problems. The way a femtocell works, it connects to your Internet service and then is in effect a little cell transmitter in your house. I've also received a new gateway modem/router. I've seen some less than favorable write-ups on femtocells. I was wondering what your take is, since I really don't have the option of not having one. Love the show.

Steve: So I wanted to get your opinion, just sort of from an amazing-how-much-stuff-you-know kind of mode. But what I do know is that there's nothing fundamentally a problem with a femtocell. About four years ago Vodafone in the U.K. had a complete screw-up with their security. And I think that's what really sort of damaged the

reputation is that they basically, I mean, as could happen with any technology, they just didn't do security correct. And it ended up being possible for bad guys to download their own firmware to intercept updates, to update with malicious firmware, to intercept calls and communications. And so that sort of scared people in general about the idea of these little femtocell base stations. But that's four years ago. And, I mean, sure, anyone could make a mistake. We've obviously been talking about how routers have firmware which in some cases is open and created a backdoor listening to the Internet. So mistakes happen. But fundamentally there's really no problem with femtocell base stations from a security standpoint.

Leo: You would know this. Aren't modern cell phones, CDMA and GSM, encrypted...

Steve: Yes.

Leo: ...from your phone, that transmission, I don't know how well encrypted, but it's encrypted from your phone to the base unit; right?

Steve: Correct. It's two things. It's encryption and spectrum...

Leo: Spread spectrum. They're moving around a lot.

Steve: Exactly. And so...

Leo: [Indiscernible] a little chunk at a time on any given frequency.

Steve: Yes. And so it's very difficult for the amateur to intercept that. We do know...

Leo: So that technology would be the same from you to your femtocell.

Steve: Correct.

Leo: Exactly the same because the phone doesn't know. Now, I don't know, once it gets on the Internet, does it remain encrypted? It's obviously no longer spread spectrum.

Steve: Right. And there again, I'm sure it's not in the clear. I'm sure it brings up an encrypted tunnel and does the right thing, does what you would hope.

Leo: But as we know, one should never assume.

Steve: Right. Yeah, it might be Just Works. I wonder if it Just Works when you plug it in.

Leo: [Shuddering] Just Works.

Steve: I'm teasing. There's just really no reason to - I'm sure. Because I did do a little bit of browsing, and I see nothing except for, like, four years ago where there have been problems. And we would know if there were, like, contemporary femtocell security problems.

Leo: I think it's safe to say it's probably as secure as your cell phone communications are anyway.

Steve: Yes, yes.

Leo: No less secure.

Steve: Yes. And the weak link is going to be, as you spotted, Leo, the weak link is going to be that first radio gap between your phone and the base station. And, boy, you'd rather have it be a 10-foot link...

Leo: Yeah, in your house.

Steve: ...than a 10-mile link.

Leo: Yeah, exactly. John Kirby, Davis, California says, "I miss Hamachi." We all miss Hamachi.

Steve: Yeah.

Leo: LogMeIn's not free anymore. They discontinued...

Steve: And we don't mean the tuna.

Leo: No. Yes, we get plenty of that. Actually, Steve does not. He no longer eats tuna.

Steve: No. No.

Leo: As you may have heard, LogMeIn has discontinued the free version. That's fine. They have the right to do that. Unfortunately, I use the free version because it passes through all the security hoops I'm looking for. But I also depend on it for the

central hub aspects, so I can connect my home computer even as its IP address changes. It's nice to be able to connect with my mom's or sister's computers a couple of times a year to make sure everything is still up to snuff. Then he uses it for that. So this brings me to a request for a Security Now! show, which is remote desktop clients which work with dynamic IP addresses. I've used VNC and Microsoft's Remote Desktop client many years ago, but both of these required a known IP address. I also realize that GoToAssist/GoToMyPC is a show sponsor, so I cannot ignore them, although I have no experience with them, only because I haven't needed to yet.

I can answer that question. They use NAT traversal. They're using a third party, so it is not necessary to have a fixed IP address. In fact, I would guess in a lot of cases, well, I'll let you answer. I know this will not be a quick, five-minute Q&A, as each of these solutions will undoubtedly need some background research before you can give any formal review. Thank you for the wonderfully entertaining and enlightening show. I look forward to it every week. John. For what it's worth, I'm staying with LogMeIn.

Steve: So when I saw this, I thought, wow, yes. I saw so many tweets in the last few days, people bemoaning the change that LogMeIn made, essentially discontinuing their free service. And I think, as you say, Leo, or actually as John said, it's their right to do what they want to with the service.

Leo: Well, it's hard to do this stuff for free, frankly.

Steve: Well, yeah. And they're a publicly traded company. They've got shareholders. I'm sure they're looking around for more revenue. It's like, okay, where can we get some more money? And it's like, well, those freeloaders who are still using our version of Hamachi, it's time to cut them off. And no doubt John will be going from free to paid, so this is a perfect example of the LogMeIn strategy functioning. The problem that any other solution will have, as you mentioned, Leo, is that, first of all, many people's IPs change - not often, typically, unless they're always disconnecting and reconnecting their Internet-facing device. Like some DSL systems are, like, actually disconnect completely and then get an IP when they essentially do the equivalent of a DSL dial-up. So there you're going to get very rapid IP change.

But, for example, I use - Sue, my bookkeeper and office manager, we use her IP as one of several factors of authentication for her access to GRC's servers because a TCP connection, can't spoof that. And so maybe every three months or four months she'll say, hey, I can't connect, and so I'll get her IP from her email headers and connect to GRC and just update that IP, and then she has connectivity again. But what that does is that shows me how infrequently her router's typically technically dynamic, but practically static, IP changes. So in many cases you could open ports through a router in order to allow remote incoming access to a remote desktop server.

But the beauty of Hamachi was that they were operating a very robust NAT traversal server so that all of the Hamachi clients phoned out through their local networks behind routers to the central hub. And then you created an account with Hamachi. Then of course the other amazing thing at the time was that he was using 5-dot IP addresses, which had never been allocated in the history of the Internet. We weren't running out of IPs back then. I think his name was Dave, wasn't it? I remember we used to call him

"Hamachi Dave," the guy who designed all of this [Alex Pankratov]. And so everybody would have a 5-dot IP, of which there were 16 million, so there were enough of them for his network. And it was the NAT traversal that just sort of made this thing work. But of course he created a useful service and sold it to the LogMeIn folks, who initially didn't change it much, and then began to change things. And now...

Leo: There's a moral here. It costs money to do things. And I think this is a unique area. There's a lot of places freemium works. It's very big in the App Store. We just saw it from Apple's results, 92% of all Apple apps are freemium. That is, free, but you pay for in-app purchases, for upgrades. It's how you make money. 92% of the revenue in the App Store, I misstated it, 92% of the revenue in the App Store comes from free apps within app purchases. So it's a very successful model. The problem is, I think, the people who use LogMeIn free and TeamViewer are a certain class of geeks who just never pay. And you just can't upgrade them. And I would guess, I have no inside information, that what happened is LogMeIn realized that, that there are some people who are going to pay up. But the people who are using the free service are going to keep using that forever, free. We cannot convert them.

Steve: Right. And it's funny, too, because remember they did another stage in this evolution where it used to be that people could have Hamachi, or LogMeIn version of Hamachi, on servers, and then could network into those servers for, like, remote server maintenance. And one of the things that LogMeIn changed, this is maybe, what, six months ago or so, was nope, can't do that anymore. You've got to be, like, have a logged-in desktop session and be active in order to use it. And so that forced some set of remote server admins...

Leo: It moved them up, yeah.

Steve: Yeah, moved them up, or they just said, screw this, I'm not going to go up that curve.

Leo: I always worry when something I really love is free because free is not sustainable forever. People have to eat.

Steve: Right. Well, and my model is somewhat different. I look at things GRC could do which I would like to do, but which would create a dependence of people on GRC. And I just - I don't - I'm not comfortable with that model. I don't want to, like...

Leo: We don't do it either. We don't do it either. I just, you know.

Steve: Right. So I do everything for free, and...

Leo: And you have one thing that pays you money.

Steve: And SpinRite pays the bills, right.

Leo: And as long as that works, which it apparently does, you're going to keep doing it.

Steve: Yeah. I think - so if you don't want a third party, and the third party is the way to easily solve the router transversal, the NAT transversal problem, then one thing you can do is you can use one of the Dyn DNS services, which most routers now support, where essentially you get a DNS string, a DNS machine name whose IP dynamically tracks the router's public IP. That way you can install in your mother's and your sister's system a means of contacting a server, a desktop server behind your router, which would track your IP. So that link wouldn't break. I think Dyn DNS, I know that there are paid versions of that. But I think there are still free versions, too. So it requires some setup and configuration, but it would solve the problem. But except for things like GoToAssist, GoToMyPC, and similar services, I don't know of anything else.

Leo: Well, there's TeamViewer. I think a lot of people use TeamViewer.

Steve: Oh, great, and TeamViewer, also.

Leo: And that's free, yeah. See, I think TeamViewer might stick around because they charge so much for the paid version that they don't need to convert everybody. It supports this strata of people who will never give you money.

Steve: Yup.

Leo: Johnathan Rabkin, Dartmouth, New Hampshire wonders about suspend to disk security. Actually a couple of people do. I was wondering if, when a computer suspends to disk - that's hibernate, basically; right?

Steve: Yes.

Leo: Yeah, laptops primarily, you could remove the disk, take it out of the computer, scrape the RAM image in the swap partition as a way of accessing passwords and encryption keys. Moreover, I was wondering if that RAM image is overwritten on disk after the machine wakes up, or whether they just leave it around, maybe you could get it later. And that ties into CZ in Washington, who wonders about "Inception" and breaking whole disk encryption. He points to a link at breaknenter.org/projects/inception. He says it appears TrueCrypt, BitLocker, and FileVault are a good deal less useful. So what is this Inception thing?

Steve: Well, Inception was interesting. First of all, it's a copyright violation, I'm sure.

Leo: Oh, but this is where - this is like freezing the RAM.

Steve: Correct. Exactly. But one of the things this confirmed was a conjecture we made quite a while ago on this show, and that was that Thunderbolt was going to have the same problem as Firewire. And indeed it does.

Leo: Because it uses DMA, Direct Memory Access.

Steve: Exactly. Exactly. It is essentially a direct port into your machine's running memory. So as we've talked about on the podcast before, if you can use a Firewire-linked - I think Inception supports Linux or FreeBSD machines. If you can use a Firewire-linked machine, and it's powered on and running - even if it's in standby, often the Firewire interface will still be live - you're able to browse around through the memory. And there are tools available, forensics tools, which have been designed to interpret the binary hash of RAM and locate keys that are in the hash.

Now, Johnathan's question about suspend the disk, this is a big problem, too. And in fact for the first portion of its life, TrueCrypt did not offer encryption of the hibernation file, which was a known problem. That is, if you had whole disk encryption, and you used TrueCrypt and hibernated, the hibernation file wasn't encrypted because it takes just extra technology and hooks, essentially, down into the OS and the kernel, to be able to come out of encryption, where the encrypted file is encrypted and decrypted in a safe way. I remember when they added that, and it was like, oh, good, because that was something that was missing.

But in answering Johnathan's question, yes, if you're not explicitly performing whole disk encryption, and that whole disk encryption includes the encryption of the hibernation file, then the hibernation file lives in the root directory of your boot drive, and it's there all the time. So it's written to when you hibernate, and it is read from after you wake up. And I'm not aware that it is deliberately scrubbed. I think that just the OS says, well, the guy's up and running in memory anyway, so who cares whether there's, like, a second copy of it sitting around.

Leo: And it would be unencrypted. It would just be - it's just a memory dump. They don't process it in any way; right?

Steve: It is an image, yes.

Leo: Yeah, just a dump of memory. Hmm.

Steve: So to get security there, you need whole disk encryption.

Leo: I don't like hibernation. I never use it. It's slow.

Steve: No. Yes.

Leo: Either leave the computer on or reboot and load everything.

Steve: Yeah, or just use standby and remember that you're in...

Leo: Standby's fine.

Steve: Yes, remember that you're in standby.

Leo: Eric Foss, Bakersfield, California has a question about MailStore: I don't know how you got MailStore Home to work with Eudora, but I have hundreds of .mbx files, and clicking on each one ain't gonna happen. I have most of my mail back to 1997 wow - so it would be nice to be able to search it. But, well, if you get a chance, I hope you'll explain how you used MailStore. MailStore was this program that allowed you to search all your old email; right?

Steve: Yeah. Actually, again, it was another popular recommendation of mine which followed a discovery weekend before last. Okay. So I didn't go into it in more detail. But I got so much feedback from people who were like, oh, this is fabulous, blah blah blah, that I thought I'd take a little bit - I'd talk about it a little bit more.

First of all, I used the commercial version also, which - because I was experimenting. I mean, I thought maybe I was going to settle, I mean, I have settled on MailStore. But it turns out I was able to completely work with MailStore Home, but not until I got things settled. I used - the MailStore Commercial version is free for 30 days. And I only needed it for one day. So what I did was I installed the commercial version, and it's actually two portions. It's a server that runs in the background, and then the MailStore client that runs in the foreground. MailStore Home, the free version, has some limitations. For example, you can only create three of one type of profile, as they call them, like archiving three mailboxes. The Server, the commercial version, has no limits at all.

So what I ended up doing was informing the - I installed the Server version. And, by the way, they can cohabitate at the same time. They were both living, the Home version and the Server version, both living on my system at the same time with no problem. And I actually did all of the importation of my many Eudora folders into the server version. And at one point I had, like, 27 or 28 simultaneous folders being encrypted and indexed at the same time, and it handled them with no trouble at all.

Once I had that, I then exported it to a Microsoft Outlook PST file because that was the common cross-version file format that they both supported. And then I used MailStore Home and imported that Microsoft PST file and then re-indexed it in order to create a MailStore Home index. So I sort of came up with a bit of a kluge. Basically I just wanted to do this once. For me, I had email going back, a quarter million pieces of email nearly, going back to '07, so I just needed to get it into MailStore Home once. And once that was done, then, all of my incremental archiving is being done very happily under MailStore Home.

Now, Eric has a slightly different problem because he says he's got hundreds of MBX files. So the solution there is that, in Eudora, you're able to simply drop one folder onto another, and it'll move them over. So I would say create a separate directory tree of all these MBX files, use Eudora to very quickly consolidate them. You can just select all your messages and drop them on a folder; select all, drop them on a folder; select all, drop them on a folder. That'll consolidate all of those mailbox files into one huge file, and then just let either version of MailStore, either Home or the Server, chew on that one MBX file,

creating an index. So that's a way you can - again, you only have to do it once. And you might want to use Server, though, because it will allow you to do 30 things at once. Actually, I think there's - no, 25 was the limit in how many threads it will run at once. So I set it to the limit, 25. But I gave it even more. And as soon as it finished with those, it picked up on the other ones. So it ended up working great.

Leo: John Seybold has a question about SpinRite 6.1 vs. SpinRite 7: As a very long SpinRite user - I think I originally bought v2, I know I bought v5 on a 5.25" disk, and I've purchased multiple copies of SpinRite 6 - I find it odd that you're talking about 6.1 and 6.2 rather than 7 or 8. From your discussions on Security Now! - great podcast - this certainly seems like a major version upgrade, not an incremental release. Why did you call it 6.1?

Steve: Okay. So I understand what John is saying, and part of me sort of agrees. But I'm fine with the sales that I have of SpinRite. I dislike companies that really leverage their users' upgrades a lot. Now, again, 10 years would not be a lot. But to me this feels like the right thing to do. I am resisting adding features. This is all about speed and compatibility, essentially. So to me, for a speed and compatibility change, it feels right to do this. Also, we will be introducing - 6.1 will run on a Mac. And so that will dramatically expand SpinRite's market additionally. So in that sense it's a new version. It will open up a big new market for SpinRite.

My plan for 7 is a complete rewrite. And so for me this creates a boundary, kind of a clean boundary. The UI in SpinRite 6.0 is set up to do what it does. And it would just be a - it would take too much time, basically, to make it do a lot more. So I can, if I restrict 6 and 6's future to speeding it up and introducing it to all the hardware and to the USB keyboard on the Mac and so forth, that I can do in a short time. And so that's my goal: Get SQRL finished; get back to 6.1, get that out; then tackle USB interfaces with 6.2, get that out. And my point is that then can hold us while I work on 7.

And 7 I start by rewriting the UI. 7 will do all the other things that people would like from SpinRite, like be file system aware, do file recovery rather than sector-level recovery so that, for example, if a drive is dying, it'll pull all the files off of the drive and note the areas that it has trouble, but skip those, and get all the other files that it can, then come back and do data recovery on the files that wouldn't go. And recover a drive in an image format to a different drive, rather than to the same drive. All the kind of things that, yes, now that drives are commodity items, it makes sense to do. That's my target for SpinRite 7. But that's going to take a while. And I can't hold a new SpinRite back for 7. So that's why I decided, do something basically that sort of catches SpinRite 6 up to today. And that'll effectively buy me time to do what I really want to do, which is 7. And there'll be an upgrade fee for that, which I imagine people will be excited about because 7 will go so much further than SpinRite 6.

Leo: That's quite an ambitious plan you've got there, young Stevie.

Steve: That's my plan. And meanwhile we'll be doing podcasts every week, Leo.

Leo: Holy cow. File recovery. Paul White writes to us from 45 25' 5.1312" North, 122 46' 30.5652" West - which, by the way, is about Beaverton, Oregon - had a thought

about CryptoLocker: Listening to SN-439, you said something about the way CryptoLocker works that turned on a light bulb in my head. You said if one plugs a USB drive into an infected machine, on the next cycle that drive would have its files encrypted. If one had two identical USB drives and connected one to the machine and allowed it to be encrypted, could the files on the two drives be compared and - whoa - reverse engineered to reverse the encryption? Happy SpinRite owner, blah, blah, blah. SN best 'cast ever, blah blah blah. Paul in Portland.

Steve: And we know exactly where Paul is, by the way, in Portland.

Leo: Yeah, I actually see a picture of his house when I put that in my maps thing.

Steve: So, okay. So, Paul, no. The good news is that won't allow CryptoLocker to be reverse engineered. And that's a good thing because, if that did work, it would work with all other encryption, and that would be a bad thing because what you're describing is called the "known plaintext attack." And the classic example, for example, and we talked about this when we were discussing crypto years ago, the Caesar cipher was a simple substitution cipher where you substituted letters for other letters. And so what came out looked like gobbledy-gook unless you really looked at it closely.

Now, if you had the decrypted version of an encrypted message, with a simple substitution cipher, you could instantly create the equivalence table that turns plaintext into ciphertext and back. So that's an example of a so-called "known plaintext attack." Now, of course that cipher, the Caesar cipher, is weak because, even if you didn't have the plaintext, you could do a frequency analysis if you knew the language that the plaintext was written in and the character occurrence frequency. You could do a frequency analysis of the plaintext and figure out what the characters were just by how often they appeared in the ciphertext.

But the ciphers that we use now, that CryptoLocker uses, which is AES, and that SSL uses, and that all of our full disk encryption uses and cloud encryption uses, everybody's using, are specifically immune to plaintext attack. That's important because many of the things that we are encrypting have known plaintext, like the beginnings of web pages have standard headers. The beginnings of files have standard file headers. There's a lot that is known. And so if there was a way to map the ciphertext to the plaintext that gave up any information about the key that was used, that would be a problem. And so one of the very most important characteristics of contemporary ciphers, one of the first things people look at is does any information leak about the key that was used in transiting between the plaintext and the ciphertext. And in good ciphers, all the ones we use, the answer to that is a definitive no. Knowing what it was when it was not encrypted doesn't give you any information that's useful about the key.

Leo: It's not reversible. That's what I would have said.

Steve: Okay.

Leo: Wayne in Park City, Utah wonders whether he's missing something: I listened

to your Episode 439 on the Target problems. Am I missing something? Why isn't every authentication server or set of servers or server set or service, why aren't they set to only allow a few bad logins before both logging and alerting the attempt, then blocking additional logins for a few minutes? Thus password guessing is not a viable technique, duh. Yet Windows Server doesn't set this as its default, or even remind admins to change these defaults during an installation. Isn't there a downside, or is there a downside, to making these changes to the defaults? Shouldn't people just prevent that? What do you say, Steverino?

Steve: So this is what I said was the - this is our last question of the day and is our teaser for next week's episode.

Leo: Oh.

Steve: The answer, essentially, is that, while that would inarguably provide greater security, we also know that it would provide headaches for the support people because people who used to have four or five common passwords, they would not remember which one of their four or five common passwords they used, and so they would put a few in, trying to guess, and then the thing would say, oop, sorry, you're out of guesses, contact your administrator or wait some length of time or whatever. So unfortunately, today, these things are still wide open. There was a recent survey done of exactly what the industry's current password policies are, and we're going to discuss that next week.

Leo: Ah, very good, very good. All right, Steverino. Hey, one other story. Have you ever heard of EVE Online?

Steve: No.

Leo: This seems like this would kind of intrigue you. It's a massively multiplayer online role-playing game, MMORPG, where thousands of people engage in space battles with one another. They've got territory, and it's been going on for 10 years. It's a very popular game.

Steve: My word.

Leo: Yesterday they had the largest online war in its history, 4,000 players simultaneously in the same space, battling each other because apparently a battle star neglected to pay its rent or protection money, really, to the - anyway, it's a very complicated reason. But unbelievable, 4,000. I wish I had video. I'm looking to see if I could find it on Twitch or somewhere. But, wow, wow.

Steve: And so it sounds like it's very much like what we did on New Year's Eve.

Leo: Yeah.

Steve: So it's an online starship battle simulation.

Leo: Yeah. Well, except, I mean, if you look at it, it's beautiful. You're really in it. This is a still from the war. I mean, imagine with 4,000 ships converging on one another.

Steve: Oh, lord.

Leo: In a single system. That is pretty amazing.

Steve: Wow.

Leo: Steve is at GRC.com. If he were doing this, he wouldn't have time to do all the other things he does for us, including not only this show, but also all the freebies he posts on his website. ShieldsUP!, you could still test your port 32764. If you go to bit.ly/port32764, you all should do that, make sure your router is not vulnerable.

Steve: Now you can just put 32764 in Google, and the first link that comes up is mine.

Leo: Steve. You'll find him. He's closely associated with the number. And all of this is pro bono. The one thing he does that makes money is his SpinRite thing. So if you don't - if you've got hard drives, you really ought to have it. It's the world's best hard drive maintenance and recovery utility. It's just fabulous.

Steve: You know, we talk almost always about recovery because of course no one's going to get it just because, for what the heck. But there are people who have purchased it, and I thank them, just to say thanks for everything I'm doing. And I would encourage them, run SpinRite on your drives. It will keep them from failing. We've heard testimonials from people who bought SpinRite to say thanks, but just left it on the shelf, waiting for a hard drive problem. And so it's like, well, if you have it, run it, because it absolutely does prevent hard drives from crashing.

Leo: Yeah, absolutely, get it. And don't be put off thinking, oh, I'm going to wait till v7 because I don't want to pay the upgrade fee. It's going to be a long time. Get v6. Steve's a very ambitious fellow, and he's a fast coder, I'm sure.

Steve: That's true. There will be no benefit to waiting. And everyone who gets SpinRite 6 now does get 6.1 and 6.2 for free. That's my promise.

Leo: Seven, we're not talking this winter. Maybe next - maybe sometime in this decade, maybe.

Steve: Well, and that's if no more SQRLs attack.

Leo: Yeah, SQRL is also there, all the information about SQRL. Somebody said, where does Steve put all the health and nutrition stuff he talks about? Is that written up anywhere? Yes, it is, GRC.com. Also 16Kb audio versions of this show; handwritten, human written transcripts of everything he says available there. We have higher quality audio and video of the show, as well, at our site, TWiT.tv/sn. You can also subscribe in your favorite netcast catcher, and it'll just come to you every week. Or watch us live. I think, more and more, some of the fun of this is watching live. And you could do that by visiting us on, what is this, Tuesdays at 1:00 p.m. Pacific...

Steve: What day is it?

Leo: I have no idea. I'm very confused because of this new schedule. Tuesdays, 1:00 p.m. Pacific, 4:00 p.m. Eastern time, 21:00 UTC on TWiT.tv. Before You Buy coming up next. Thank you, Steve.

Steve: Thanks, my friend. Talk to you next week. And we will analyze the industry's current password policies, and it's not good news.

Leo: Yeah. I've actually often wondered this. Why don't you just time out after three bad requests?

Steve: Not good news.

Leo: Yeah. Although when that happens to me, I hate it.

Steve: Exactly.

Leo: Because I've got big thumbs, and I make a mistake, and then I've got to wait.

Steve: Exactly.

Leo: Or I can't remember the password, and I try all my regular ones, and then I've got to wait. All right, Steve, see you next time.

Steve: Thanks, buddy.