Listener Feedback #196

Description: Leo and I discuss the week's major security events and discuss questions and comments from listeners of previous episodes. We 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-472.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-472-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. He's going to have a look at, of course, the latest security news, including a very calm Patch Tuesday. And we'll talk about why it may not be a bad idea to use XP in a virtual machine. Just make sure you update it. All that and your questions and Steve's answers coming up next - it's a Q&A episode - on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 472, recorded September 9th, 2014: Your questions, Steve's answers, #196.

It's time for Security Now!, the show that protects you, your loved ones, your privacy, everything that you hold dear and sacred. Steve Gibson is here, the Protector in Chief. He's waving.

Steve Gibson: Hi, Mom.

Leo: He's waving at us from his Fortress of Securitude...

Steve: Ah, yes.

Leo: ...in beautiful Irvine, California. It's good to see you, Steve.

Leo: Having watched your stream all morning long with the Apple announcement

and all of that excitement. As soon as we get technical documentation on exactly how all this new stuff works from a security standpoint, things like Apple Pay, I'm sure we'll do a podcast to describe and explain, again, once we have the specs, how they have achieved what they apparently have.

Leo: Or if they have achieved it; right.

Steve: Exactly.

Leo: You know, now that whitepaper makes sense that they put out that you spent - we spent a couple episodes talking about Apple security.

Steve: Three episodes on that.

Leo: Yeah.

Steve: Yup. And they were laying the foundation.

Leo: Now, in the context of that, they were - exactly, yeah, yeah. It's pretty clear. And they talk about a secure store for your data and all of that stuff, stuff we've already looked at.

Steve: Well, and from what little we know of Apple Pay, they anonymize the transaction so that they are able to assure the merchant that the funds have been transferred, but the merchant knows nothing about you. And unlike current credit card transactions, in fact we'll be talking about the massive Home Depot breach here in a minute today, where you're trusting the merchant with your data every single time you use your credit card, this is potentially a much more secure model. And the idea being it's sort of more like the PayPal model. You need to be big enough, like someone like Apple, to be able to say, look, we're going to do this.

And, I mean, I saw that my bank, Chase, was there, that does my credit cards, and the major ones. I think they said they covered like 83% of the transactions based on the collection of banks that they've worked out a relationship with. So we have a ways to go before these little pay points appear everywhere. But we now have NFC in the phone.

Leo: Yeah, in a way there's no, I mean, Apple can put this through because they have the clout in the U.S., maybe not in the rest of the world, but certainly in the U.S. But what's great is this makes it possible then for non-Apple NFC payments. I mean, once the infrastructure is in, it doesn't have to be Apple.

Steve: Correct.

Leo: And they can lead the way, and everybody can start doing this. I think this is the beginning of the end of cash, and even of credit cards. I hope it is.

Steve: It's also nice, too, that it didn't come before now because there's no way, I mean, as you were saying, we spent three episodes looking at the iOS 7 security model and the architecture of the hardware for security that they specifically put into the iPhone 5. Now that of course moves into the 6. And so my point is, had they tried to do this 10 years ago, we wouldn't have had as mature an understanding of, I mean, the over-the-top crazy need for security in order for this not to come out of the gate and stumble.

And clearly Apple is mature enough now, from their understanding of security - [clearing throat] despite the fact that we have a lot of nude photos wandering around the Internet that apparently came out of iOS backups. But I think that we're at a point where we know how to do this right. And from what little we know, it looks like the architecture is right. So again, as soon as there's a whitepaper out, we'll certainly be talking about it in detail.

As it is, we're the Second Tuesday of the Month. We know what that means. We have news about the Home Depot breach, which is massive. A little note about Comcast. And Google declaring war on SHA-1, which, well, even before I knew that we were going to have to have a relatively short and high-speed podcast today, I decided not to try to tackle that as just a news blurb, but to make it the topic of next week. And I will explain why when we get to it. So a good podcast. And this is a Q&A, #196 today.

Leo: I love it. I've got the questions; Steve's got the answers. All right, Steve. Let's start with the news.

Steve: So we had a tame Second Tuesday of the Month, comparatively. Actually, the tamest one we've had in a long time. There was only one critical vulnerability in - this is a publicly disclosed remote code execution, but not apparently in the wild yet. So it doesn't get the zero-day mark. And this is based on IE, so all versions of IE need to get updated. This fixes that one critical - and it's a remote code execution, meaning that, as Microsoft says in their boilerplate, someone needs to trick you into visiting a malicious website with an unpatched version of IE, and then you are potentially in trouble. And that set of patches also deals with 36 privately reported less critical vulnerabilities. So they fixed a bunch of IE things.

Then there was - but only that one vulnerability that was critical. One important .NET vulnerability, they called it a "denial of service," which doesn't mean the way we're used to talking about it, like a flood of traffic. That's sort of the generic term where, like, someone can make something crash. And it's like, oh, well, look, it crashed, so it doesn't work. And this is weird. Again, not anything anyone really needs to worry about.

If you have, for some reason, installed and registered the active server pages .NET system, ASP.NET, under IIS on your machine, then apparently then that makes you vulnerable if there's some way for the bad guys to access that. So I presume that means you're exposing your server to an Intranet or, I guess, if you're really crazy, to the Internet. So but again, all it means is I guess they can crash something. So that's - Microsoft marked that one "Important."

And then there's an elevation of privilege problem that only affects Windows 8, and it

was found in the Windows Task Scheduler. And apparently, if somebody is able to logon locally to a vulnerable machine with valid logon credentials, and then do something to Windows Task Scheduler, there's a way for them to get elevated privileges for their session. And this closes that down. So time to update, as usual, but not anything breathtaking.

However, we learned, it was a week ago, there was a rumor about a credit card breach. And I think I might have actually first learned of it from a buddy of mine who is a home improvement guy and spends a lot of time going back and forth between Home Depot and his home, because they alerted their customers to something they were investigating, yet they were warning everyone that there was a problem. Well, it turns out that it is a massive credit card breach using, believe it or not, a version, an updated version of the same malware that got Target in that famous huge Target breach last December. So we're coming up on a year that it's been. Home Depot is also using XP Embedded, as was Target. So Home Depot didn't make any changes, apparently, to their system, despite being a large retailer with the same point-of-purchase systems that Target was using. So this BlackPOS malware variant got in.

Now, the damage is extreme because a ton of data was exfiltrated. Brian Krebs has been on this from the beginning. And we've talked about Brian's work. He spends a lot of time down there in the dark underbelly of the Internet. And so he reported that, despite Home Depot's quick claim that no credit card PIN data was stolen, and that actually appears to be correct, multiple financial institutions were reporting to him, at his request, that they're seeing a steep upswing over the past few days in fraudulent ATM withdrawals.

So after some more research, we know what's going on. Enough data was stolen to give the bad guys the ability to convince the banks that they're the legitimate card owner, and then reset the PIN. So people got their card numbers, the full name and the city, the state, and the zip was lost from Home Depot. And again, what that allows with some research is for people then to use often the bank's automated system. So they don't have to even talk to a human being. Banks have automated call handlers where you just use your touchtone keypad to enter the requested data. And there's enough information that they're able to essentially reset the PIN to something they know, and then go to ATMs and pull cash out.

So, and in fact Brian reported that he'd heard from the manager of a large unnamed West Coast bank that had lost more than $300,000 in two hours yesterday, meaning Monday of this week. We're recording this on Tuesday. So on September 8th, yesterday, $300,000 was transferred through fraudulent withdrawals in two hours due to PIN fraud on multiple debit cards that had all been used recently at Home Depot. So another big bad breach. And once again, un-updated Windows XP Embedded.

Not like this was a zero-day flaw. These are systems which had not had updates for a year. So this malware is using a well-known exploit, but people don't think of these things as being PCs. They think of them as being credit card terminals, despite the fact that it's got a full-function, very powerful OS embedded in it, which was their first mistake, rather than using something obscure, like a traditional real-time operating system rather than a commercial Windows desktop system. But it's less easy to develop for those. You can't program those in Visual Basic the way you can Windows.

Also on the news I just saw this sort of - people were upset that Comcast's XFINITY WiFi public hotspots were performing an on-the-fly JavaScript injection of random sites that people were using. So a tech reporter was using Comcast's XFINITY WiFi public hotspot and noticed a banner sort of run across the bottom of his screen a couple times. And he was savvy enough to say, okay, wait a minute. And he captured the source of the page

and took a look at it and found that Comcast was injecting into this page their own JavaScript. And so while you could argue, I'm sure in the terms of service that the person probably clicked through, Comcast said, oh, yeah, we reserve the right to embellish the pages...

Leo: Well, far be it from me to defend Comcast. But they're doing it every seven minutes, and it's not an ad. It says, "You're on Comcast WiFi." It's to let you know where you're getting this Internet access from. And that's what Comcast said. What we're doing is, because people are using this nationwide Open WiFi network we've made, we think that they need to be notified that they're on it, and that's what we're doing. They're not putting an ad in it yet. I mean, if they do, then that would be bad. But I don't think this is so bad.

Steve: Well, okay. So I don't disagree with you. But from a technology standpoint, first of all, many people do disagree with you.

Leo: Oh, I know.

Steve: But those are the kneejerk security guys who believe that it's - like there's an ethical breach if the user receives something that the site they're visiting didn't send. So, I mean, this is an injection of executing script. And of course this is blockable if you're over HTTPS. And so what I was getting from our listeners about the story was yet another reason why you want to be HTTPS and not just HTTP, because Comcast would not be able to inject this...

Leo: Well, wait a minute. They're the Internet service provider.

Steve: Right.

Leo: They could break HTTPS. They could break SSL and do it.

Steve: In order to do that transparently, you would have to accept a certificate from them so that they were then decrypting your security.

Leo: Right. Which would be even worse, I understand, and they're not doing that.

Steve: Yeah.

Leo: But other ISPs do this. This is not an unusual thing to do. And remember, you're using - they're your ISP during this time.

Steve: Right, right.

Leo: So if they wish to, they could.

Steve: I'm sure in the terms of service it says we reserve the right to do this. It's...

Leo: Well, as soon as they serve ads, I'll get just as up in arms as anybody. And I understand the security issues, as well. But I can understand why, I mean, this is, you know, they're popping it in just to say, hey, don't forget you're on Comcast right now.

Steve: Right. Okay. And so we didn't have a lot of news this week. However, Google really upset certificate authorities. We've discussed how last year Microsoft announced that in 2017 - and this was in 2013, late 2013 - Microsoft said we're no longer going to honor SHA-1 certificates four years from now. Google has taken the position, and publicly, and amid a huge amount of controversy, that they don't want to see the same foot-dragging that we saw with the MD5 hash signatures occur with SHA-1.

So Google has said that two months from now, in November of 2014, they're going to start putting pressure on SHA-1 certificates by noting in the user interface, by changing the UI - and in fact what you get is a little yellow triangle, if you're using a certificate that has certain characteristics. And I want to, I mean, this is a really interesting controversy because it's possible to see their side. It's also - but the certificate authorities have really been vocal and, I think, made some great points.

So that's going to be our topic for next week. I don't want to - I want to do this interesting issue some justice by giving it full coverage. And so that's our topic for next week unless the sky falls between now and then.

Leo: Which can always happen in this business.

Steve: It does.

Leo: Yeah.

Steve: Also I wanted to mention I made a mistake in stating that USB was not differential communications. I think it may have been in the context of Ethernet, which is differential, and we have at least one question about that in today's Q&A. USB is four conductors. And I know better because I remember, I mean, I've built USB peripherals before. My little KetoFlute used a USB interface that I designed. The four conductors are 5-volt power and ground, and then it's basically a bidirectional time-shared USB that runs in a master/slave mode where the master is able to use both conductors in an outgoing direction and then release them for use in the other direction.

So that's the way the technology works, not, as I said, one going in one direction and one going the other direction. So thank you to the people who brought that to my attention. I know better and just misspoke.

And then just an interesting - in the context of a Q&A I found a question from Cedric

McGraw in Moncton...

Leo: Moncton.

Steve: Moncton. What's NB, Canada?

Leo: New Brunswick.

Steve: New Brunswick, Canada. He said to both of us: "Hello, Steve and Leo." He says, "I'm a longtime listener and think the show is fantastic. I'm a high school IT teacher..."

Leo: Oh, neat.

Steve: "...and I incorporate your show into my classroom every week. The students love talking about security issues. I've got a quick question about SpinRite. Is there any benefit to running SpinRite on an empty drive? For example, if I use DBAN on an older drive, is there any advantage to running SpinRite before installing the OS? Thanks for the great education you provide your listeners week after week."

And so to answer Cedric's question, yes, a substantial advantage. And many people use SpinRite, not only on older drives that they've used DBAN on, or perhaps wiped the drive, but even on new drives. It's probably not as necessary as it once was because drives are now smart and are able to deal, as we've spoken of in the last couple weeks, with surprise encounters of defects on the fly. It used to be, before drives were smart, you would run SpinRite because it was the brains that would handle dealing with defects that were encountered. Then the drive was sort of freed up from being smart and didn't have to be.

Still, it makes sense because we talked about how SpinRite - the drive doesn't just know that there are problems out on its surface. It's not psychic. The only way it knows there's a problem is when it encounters one trying to properly and correctly read the data back from the drive. So SpinRite works with the drive in order to show it problems. And that's probably more useful on an older drive that you're recycling than on a brand new drive.

But many of SpinRite's users routinely give a brand new drive, even a brand new drive, a pass through SpinRite, knowing that it's a much better way to sort of bring the drive current. Notice that almost nobody actually formats drives anymore. The so-called "quick format" isn't a format. All it does is put the file system header, essentially, on the front of the drive. And then you begin to fill up the drive.

If you ever do a long format, that's a mistake you don't make twice because on drives these days it takes, I mean, hours. I mean, it makes SpinRite look fast. The full format is so slow that no one does it. So running SpinRite on the drive sort of is really your first real contact with the drive's surface that allows the drive to quickly remap and hide and deal with any defects that may be there before you begin filling it up with data. So absolutely a good thing to do.

Leo: All right. I have questions. Steve has answers. He should. He prepared the questions. I'm not going to say anything.

Steve: Yeah, I did it last night because I knew we were going to have a busy morning watching Apple do all of their stuff.

Leo: Well, you know, one thing I really admire about Steve, most of these shows and most of us will answer anything off the cuff, without any research, best of our ability. I've made a living doing that for 30 years. Steve likes to actually research them, come up with the right answer, that kind of thing. He's crazy. Question 1, Tom Rhodes, Houston, Texas. He says, and this is going to be a battle that'll rage for a while, "No, ground loops are not a problem with Ethernet."

So this goes back to the question that we were talking about, somebody wanted to bring Internet access to a shed 100 feet behind the house. We said just put in conduit and Ethernet. And then somebody said, oh, no, the difference between the ground voltage potential on each building, because it's inevitably going to be different, is going to cause a problem, and one of them is going to draw power from the other one, the whole thing is going to fry, and you're going to go to hell.

Steve: Right. So...

Leo: And he says, no, no, no. He says if something caused the voltage potential between two endpoints to exceed, like, 1,500 volts, like a lightning strike, okay, it would jump the isolation gap and cause bad things to happen. But there is a trans- the 10BaseT is a transformer which is coupled with it which provides isolation up to 1,500 volts. Nothing to worry about. A lot of other people said you can have separate ground for the Ethernet. You don't have to worry about that. There's ways and ways.

Steve: Yeah. So I just wanted to mention, first of all, a very good engineer friend of mine immediately responded that Ethernet is transformer-coupled. And so the idea is, what that really means is, if each end is transformer-coupled, then the wires which are running between the ends each go to one side of a transformer. They're sort of - that means that the wires themselves are floating without any reference to anything, without ground reference. And that is not - without being relative to ground. And this is what I was talking about when I was talking about a differential signal where you have this notion of a differential versus a common mode voltage.

And so the concept is one end on the other side, on its side of the transformer, induces through magnetism a current in the outgoing side of the transformer, which creates a flow of current through the Ethernet to the receiving side transformer, which then induces a magnetic field that causes a current in the inner side, sort of the receiving side of the other endpoint. And so the idea is that the actual instantaneous relative voltage of the two sides doesn't matter. And now that's true up to some point, the so-called breakdown voltage, or isolation voltage of the transformers. If lightning struck one end, you could imagine that the lightning would get across the insulation of the transformer. Then you'd have a problem.

So anyway, I did want to come back to this to sort of clarify that, as long as Ethernet is reliably transformer-coupled, then you really should be able to have a relative offset of ground between the two ends without there ever being a problem, and even have them moving relative to each other because it's the current flow through each end, both transformer-coupled, that that's what carries the signal, not their instantaneous voltage. So thank you, Tom. And it's difficult to understand, then, given this, why people are having and reporting problems with ground differentials between endpoints with Ethernet because transformer coupling should eliminate that completely, even if it's changing a lot.

Leo: Why do I feel like we probably haven't heard the last of this?

Steve: And just use fiber optics, and then you don't have to worry about it.

Leo: Yeah.

Steve: Use plastic.

Leo: Something that doesn't conduct.

Steve: Yes.

Leo: I just know there's going to be another - this is going to go on for a while. There's something about these things because they don't really matter, they're very small potatoes, but it's easy to have an opinion about it. And it just - this is the kind of thing that can rage back and forth for years.

Steve: Oh, and believe me, we're looking at the summary of a mailbag...

Leo: Oh, okay.

Steve: ...full of people saying, oh, you have to have transformer coupling because even the ground differentials between the third floor and the fourth floor...

Leo: Even there.

Steve: ...is enough, you know, blah blah blah. I'm like, okay.

Leo: Yeah, that makes sense, yeah. It's going to be a problem everywhere.

Steve: The wind blows across the top of the tower, and the electrostatic charge created -

okay, yeah, so...

Leo: Is it powered, this transformer? Or is it just some sort of passive thing?

Steve: Yeah, no, it's passive. It is literally - it's two completely separate coils of wire, wrapped around each other.

Leo: Got it. That's what they do.

Steve: And so you drive the coil on one end, and that induces a current flow on the other side. And, I mean, and so really, specifically, so that you have this DC isolation, yet you're coupling the AC signal across this insulating gap. And so it's an isolation transformer.

Leo: I should know this because of course this is all part of antenna construction and...

Steve: You passed your license.

Leo: Every ham probably is screaming at me right now. I'm a bad ham. I told you that. No memory at all.

Steve: Hey, you got the job done. You got your license.

Leo: I got the license. That's all it is.

Steve: And if you remember your call letters, then you're fine.

Leo: Yeah. And if I'm ever going to put up an antenna, I ain't gonna do it. I'm going to have somebody who knows what they're doing do it.

Steve: That's right. Little more to the left.

Leo: That's my job.

Steve: That's right.

Leo: Perfect.

Steve: It needs to be wife-compatible.

Leo: Actually, we have three acres. I could put it anywhere. I got, in fact, I'm looking at these eucalyptuses.

Steve: You actually do have a south 40. You actually do a have a south 40.

Leo: I totally could put up an antenna, yeah, at this point. Barry in Central Georgia, he's wondering about something called ATA Secure Erase. Steve and Leo, longtime listener, big fan, yada yada. I work in the defense industry as an IT and network security guy. When disposing of old hardware - laptops, et cetera - one of the things I do is erase the hard drive of the surplus equipment, for obvious reasons. I used to use a block-wiping program, DBAN or the like, on a DoD-short cycle, which is three passes of pseudorandom changing data followed by a pass of all zeroes.

However, I've noticed that many newer drives have the option of a drive firmware-based Secure Erase and Enhanced Secure Erase functions, SE and ESE respectively. When I've tried them, the erase times are typically half that of using a block-wiping program. However, documentation of SE and ESE is hard to find. Can you shed any light on these functions?

Steve: So I always, whenever I see Enhanced Secure Erase, I get a chuckle out of it because it's like, wait a minute. Was Secure Erase secure or not?

Leo: Apparently not. It's sort of secure.

Steve: Because it's like, no, no, this one is the really secure one. It's enhanced, as opposed to the one that we were selling you before. It's like Apple. Only when they have NFC, did you see?

Leo: Then, yeah.

Steve: Now it's the best thing ever.

Leo: Now everybody ought to have it, yeah.

Steve: But yesterday, eh, not so much. Okay. So there is no documentation which the drive manufacturers offer. The only thing we know is that the secure erase functions, and that's plural, are NIST-approved. So I don't know what that means. The difference between Secure Erase and Enhanced Secure Erase is that there are - it's possible, using the ATA specification, to create some hidden regions on drives where, if you ask double-pretty-please, then you're able to get access behind this barrier. But otherwise, the drive reports a size smaller than it actually is. Secure Erase doesn't go past the barrier. It only erases the obvious visible part. Enhanced Secure Erase blasts right through those sort of hidden special areas.

Now, the other thing that a really secure erase in the firmware can do potentially is, for example, erase the spared-out sectors. One of the controversial things, we talk about this all the time when we talk about how can you really erase a drive or even a thumb drive, because we all know that sectors are being removed from use when they're no longer reliable. And those could contain some sensitive data which has sort of been taken away from our access. The address that we were using, that we used to be using to access that sector, now accesses a different sector. And it's sort of like it's in hyperspace. You can't get to it from here.

So what the firmware-based erases could do is be smart and also go wipe those. But again, we're just sort of having to take everyone's word for it. The manufacturers say, oh, yeah, we used to offer secure erase. Now we've enhanced that. And the NIST says it's good enough, so take our word for it. So this is one of those places where they're just saying it works. And, oh, it can be half the time, that is, twice the speed, because it doesn't have to transfer any data. You just sort of say "erase yourself" and come back in a while.

And actually you are able to query and ask how far along you are. That allows operating systems and secure erase tools to give you a progress bar. But that allows the drive, for example, never to skip a beat. Even with SpinRite 6.1, where I'll be doing 32MB transfers, which is the largest transfer available through the ATA interface, even the SATA interface, you just can't - there's no way to give it a larger sector count than 32MB worth. That's 64K sectors, 16 bits' worth of sector count is the largest available.

Even there, at the end of one of those transfers, we'll miss a rev or two asking for the next block, let alone processing the data. So this just sort of says, "Drive, go offline and come back empty." And so it can really scream, if we trust that it's really done the job. So that's sort of a mixed blessing. Maybe do both. Do a DBAN; or when I offer that feature, it'll be way faster than DBAN. Probably not much slower than the drive by itself. I think I'll be able to get up to about the Secure Erase speed. Although I would argue that it's certainly possible that the firmware could do a better job than any application-side software because the firmware could get into those dark areas that have been essentially taken out of the normal drive space.

Leo: Corby, Reno, Nevada is going to solve the email security problem.

Steve: Whew.

Leo: Oh, Corby. Thank you.

Steve: Yeah, thank you.

Leo: Steve, I just finished listening to the episode about email security. I think this problem might already be solved. Sometime in the past you talked about Threema. I use it now, and I love it. It seems like Threema, or an approach like it, could be the core of a secure email system. It already has the end-to-end security and store-and-forward approach. It's a secure text-messaging program, we might parenthetically add here. Doesn't seem like it would be that difficult to turn it into a full-blown email

system. Your thoughts?

Steve: Well, so he's right. The problem is, as you note, Leo, it's not email. And the other problem is, it puts the burden - it's not a big burden, but it's more than no burden - on the users to exchange keys. And remember, this is the one that's got the three levels. It's got three spheres. And I think you can either be yellow, orange, or green. I don't remember what the colors were, actually. And it's not until the Threema clients physically see each other displaying their QR codes, into which is encoded the certificate, that you merit the three green dots, meaning we're absolutely certain about the identity of each other.

And so this is really the problem. It's not that we don't have the crypto technology. It's the plumbing of dealing with authentication, making sure someone didn't intercept your identity and substitute their own, in which case you'd be encrypting data for the interceptor and not for the person you were expecting to. So, and it's not that we don't have solutions. It's that, as we talked about last week, it's really looking like email kind of ought to just be left alone. Let email be insecure.

And we know people are working on really terrific, wholly new designed systems to give us an email-like solution that really is secure. Because, for example, as we also said last week, there's still the metadata. It's often interesting to people who are nosy who you are exchanging email with, even if they can't see into it. And the older systems, like PGP, never attempted, never claimed to protect you from the leakage of who you were talking to. They were only creating an envelope into which no one could see. But they could still see the "to" and "from" address on the envelope.

Leo: And necessarily because it's like, you've got to know where you're going.

Steve: Yeah. And really when you think about it, protecting the metadata is tricky because we've even seen the Tor system that is all about anonymizing. But if you have - if you could see all of what Tor is doing, and you saw things going into it and coming out of it, then you might be able to match those two events up and essentially use that form of metadata to deanonymize who was communicating with whom. So the Internet wasn't designed to hide that level of communication. And so it takes being clever, like randomly changing packet sizes and deliberately introducing delays through the network so that people are unable to see into what's actually going on.

Leo: I kind of like your point, which is let email be email. There are plenty of use cases where you don't care who sees it or who knows what you're doing.

Steve: Exactly.

Leo: And the advantage of an open system like that is it's open. It's easy. It interoperates. The other thing, I guess, Threema is not open source; is it?

Steve: I don't think so.

Leo: I don't think so.

Steve: No.

Leo: So I'm not going to trust it. So here's - this is the other thing. You were great on TWiET yesterday talking about all of this stuff. And I spouted my Steve Gibson lesson, which is that, if you want stuff to be private in the Internet, there's only one way to do it. That's encrypt it for it hits the Internet with a key only you have access to, end-to-end encryption and transport, encryption on the cloud server, and encryption on the way back. And it's Trust No One. That's the phrase you came up with which means only you have the keys.

Steve: Right.

Leo: Not a good solution for email at all.

Steve: No.

Leo: But I guess we could have some sort of symmetric key technology or some sort of key exchange. But public key works.

Steve: Well, I think what we're probably going to see is we're going to see a raft of these instant messaging systems which really are secure, things like Off The Record protocol and Threema protocol.

Leo: And by the way, OTR is open source.

Steve: Yes.

Leo: And so I like that.

Steve: Yes. And incorporates things like Perfect Forward Secrecy that were never part of the original PGP design. And then I think, as you were saying, email will just be email, and we'll end up with a new system to solve the need for something else.

Leo: There's lots of stuff that we send that doesn't matter if the government or anybody else saw it. And email works really well for that. In fact, most of the stuff I use email for is public, might as well be.

Steve: Yup.

Leo: Nothing wrong with that.

Steve: Yup.

Leo: Question No. 4 from Eric. He's wondering and worrying about a new Windows XP installation. Hey, Steve, I have to look at some old data that's going to be replaced with a newer system. But the application I need only runs on Windows XP. Since I no longer have a current running version of XP, I'm going to have to create a virtual machine and install XP in there. Now, usually when I install a new operating system, the first thing I do is go out and get the Windows updates. But since XP is no longer supported, and there are theoretically unpatched exploits out there, I'm thinking it's not even worth the time. I'll just stay offline with that virtual machine and only use it to review the old program I need to upgrade. Your thoughts?

Steve: Really bad idea.

Leo: Oh.

Steve: Really bad.

Leo: I was going to say go ahead. Oh.

Steve: What Eric is suggesting is that he takes an operating system which is more than, what, 12 years old - was it in 2001 that XP happened? - and use it without changing it. Even though Microsoft has stopped patching it, they had been patching it for 13 years. By all means, take advantage of that. I turned one of my own Windows XP machines on, sort of a tablet installation that I use for some purposes, that I hadn't had on for, like, six months. It had 113 updates, which it received a couple days ago. So even though Microsoft isn't moving it forward from last April, they were for 12 years.

And we absolutely know that the Internet is crawling with stuff that would love to get at an original install of Windows XP. So definitely worthwhile installing, from scratch, use Windows XP SP3, or install Windows XP and then install SP3, if you don't have one that's already got the SP3 rolled into it. And then run through just one or, well, however long it takes or how many iterations till it says there are no more updates available. That one I think you can safely use because then you're using something that is current as of when Microsoft stopped supporting it. And as we know, the XP world hasn't collapsed. There have been no cataclysmic...

Leo: No, it's true.

Steve: ...predictions about XP, and we're now six months along.

Leo: A couple things. So your point is that they are, even though they're not making new patches, all the existing patches will be applied if you do Windows Update to a new install of XP.

Steve: Yes, yes.

Leo: That's good to know. I did not know that. But secondarily, he says, "I'm not going to go online with XP. Why would I need to patch it?"

Steve: Now, you're right. If you're really not online, if you set up the virtual machine so that it doesn't have access to the Internet, and as Eric said, it's just for running some application that is XP only, then I agree. You could save yourself the time. But by all means make sure you remember that that thing hasn't ever been patched.

Leo: He's got a time bomb there, yeah.

Steve: Because it is just - it's a magnet for all this stuff, I mean, remember all the problems we were having before Service Pack 2, when they finally turned the firewall on? Oh, my lord. I mean, Code Red and Nimda and MSBlast and, I mean, it was just - it was crazy back then.

Leo: That actually was a watershed moment in security. The day that Service Pack 2 came out, and they turned on the firewall on XP, that was kind of the turning point in some ways.

Steve: Yes. It made a big difference.

Leo: Made a huge difference. So, yeah, you'd have to make sure, and you can in virtual machines, you can disable the network interface.

Steve: Right.

Leo: So you can do that. You're right, though, it makes me nervous because you're on a machine that probably does have Internet access. It makes me a little nervous to have that there.

Steve: You know what I would do is I've had machines like that. I set the wallpaper or the background to bright red.

Leo: Yeah, poison.

Steve: So it's like giving me an - I'm always having a reminder that this is - because it's so easy to just, like, oh, click IE or, well, my god, can you imagine using IE6 on unpatched Windows XP?

Leo: That, okay, that's crazy talk. That's a good point. You'd be using - I don't even know if you'd have IE6 on there. You'd have whatever the Internet Explorer was that came out when XP came out. That's like, what, IE3? I don't even know what that is.

Steve: Might have been 3. I remember 3.

Leo: I don't even know. Or air gap the hardware, too. Run it in a VM and air gap the hardware. Then you're safe. But you're right, it's just - it's a time bomb.

Steve: I'd just bring it current. And then you have an XP you can use for other things, too, and not just have to be completely afraid of it.

Leo: Bill in Michigan says it was IE5 originally.

Steve: Ah, yes.

Leo: Ah, good old IE5. Make sure you turn on Active Desktop for lots of fun. Stuart Ward in Reading, United Kingdom, one of the designers of cellular 3G security, wow...

Steve: Yeah.

Leo: ...shares some security information with us. Steve and Leo, you were discussing the false base-station attacks on mobiles in the last show. And actually since the last show there's been a lot of news about there being lots of fake cell towers out there. For all - I don't know why, but apparently they're not uncommon. He says: I was involved in the design of the security for 3G networks. You can actually see the spec at 3gpp.org. And yes, the original 2G SIM authentication protocol is one-way. The network authenticates the Mobile Station, but the Mobile Station cannot verify the network. But I should tell you it was updated with the work on the 3G spec and the transition to the USIM. So if you're using a USIM on a 3G network or later, the phone will authenticate the network. But as phones have to work seamlessly on 2G networks, unless - oh, here's the...

Steve: Uh-huh.

Leo: Here's the unless: Unless you have specifically set your phone to only use 3G networks, and you have a USIM from your network operator, it's still subject to these attacks. And that's, I bet you, that's rare.

Steve: Yeah, yup.

Leo: It's certainly not the default. One of the things in the spec, it says that the phone must show an indicator when the secure connection is not enabled. This is known as the Authentication and Key Agreement, or AKA, protocol. This is universally ignored [laughter], and there are hardly any phones that show this, despite the fact that the spec says they're supposed to. If this were enabled, it would be a good indicator that you were not on the network you think you are on. The reason this is ignored is that, well, operators don't want the support calls they'd get if there were problems and have to switch encryption off, which they do surprisingly often.

Although it's not possible to tell if your phone is connected to such a rogue base-station, the presence of these shows up to the network operator because of a spike in handover fails as nearby phones try to hand over established calls to the false base-station. Keep up the great podcast. Stuart, Reading, U.K.

Steve: Fabulous, fabulous information, Stuart.

Leo: Wow, love that.

Steve: And this is the classic protocol version downgrade attack, where the attacker arranges to cause you to basically use an older protocol by making the newer secure one unavailable, and so the device falls back. We've talked about this in SSL all the time. That was a classic attack. In fact, there's even an SSL that does no encryption. And there were early clients that, if the server said, no, I don't support encryption, the client said, oh, too bad. Well, let's talk anyway. And so you thought you were using SSL, but it had a null cipher, as it was called.

In this case, the base stations only advertise 2D, yeah, I'm sorry, 2G support because, if they were to do 3G, there's a chance that the phone would try to authenticate them, and they would fail that authentication because they're not actually AT&T or Verizon or whomever. So they just say, "No, no, we have 2G here." And the phone says, "Oh, well, okay, I guess that's all I can get." And since they're broadcasting a stronger signal, and you're closer to them than you are to the authentic tower, your phone chooses to go 2G.

Leo: You wouldn't be able to make phone calls, either; right? I mean, or are these rogue towers...

Steve: No, you are. You are because they knit your connection.

Leo: Ah.

Steve: They're able to make a 3G connection out to the cell tower, but you connect to them, and that allows them to intercept and have complete access to your decrypted conversation and texts.

Leo: So then you wouldn't - oh, I guess you would get a failed handover, handoff, because there isn't a real handoff from the network's point of view.

Steve: Exactly, because normally you're going from one AT&T tower to the next. And so they're able to negotiate you. Here you leave there and go outside of its range, and suddenly your phone's saying, "Hey, I'm taking this conversation over to here." And the here says, "Huh? I don't know what you're talking about."

Leo: There's nowhere there. There's no here there.

Steve: Right. There's no here there.

Leo: Right. Ed Killian, Cold Brook, New York. He's going to ask us a little bit about Apple's SSD insecure erase. This is another one of those topics like ground differentials and Ethernet that just goes on and on and on.

Steve: But people care about it because they want to know if they're...

Leo: Well, as we should.

Steve: Yeah.

Leo: And I have been lately saying, if you're using SSD, turn on encryption now. We've got more questions. Steve's got more answers. We'll go to Ed Killian's question next from Cold Brook, New York: Love the show, even though I've only been listening for two years. Well, we are in our ninth year, so where have you been all my life, huh? Hi to Leo, whom I loved on Screen Savers and does a fantastic job now. I had to read that. Sorry, Steve.

Steve: That's why I put it there. That's what he wrote.

Leo: Thank you. I - yada yada. I have three Mac Air books, and I'm putting them into storage. Therefore I want to erase the hard drives. I've booted from an external Mac OS X 10.9 install disk. Actually it's a USB disk, USB drive. I run Disk Utility to erase the drive. I select the internal Apple SSD, the TS128B hard drive. Then I hover the mouse over the Security Options button, and it's grayed out. It says: "Secure erase not available on this type of drive." Makes you wonder how are they erasing it and how secure it is. Well, you know what, I've got to give a little credit to Apple for saying that. I didn't know they did that.

Steve: That's nice. And the answer is, it's not actually what we think of as an SSD. It's enough of the ATA specification that it looks like a drive. But we know that it's not a boxed drive that Apple got from Seagate or something. It's a couple little black chips on

the motherboard. It's highly integrated. Their costs are nothing. And so what they did is they created enough of a drive to have it work. But so it doesn't - it's not a full rendition of the ATA spec because Apple just doesn't need one for their own purposes.

So it looks, you know, it obeys the commands. SpinRite will be able to run on it. Apple runs on it. The various virtual machines, you can even run Windows, as we know, on a MacBook Air. It makes a great hardware platform. So it's enough of the drive to do the job, but it's missing the Secure Erase feature. It's just not part of what Apple implemented for their own embedded drive technology.

Leo: So it doesn't have anything to do with the fact that, and we've talked about this in the past, that really there isn't a guaranteed secure erase for SSD.

Steve: No. It's just that somebody said, hey, you know, they were writing the firmware for their own implementation of what we call an SSD. It is a solid-state drive, but Apple just made it up. It's not commercially available like the boxes we buy and plug in. It's just a couple little chips on the board.

Leo: See, I thought this was more about the issue which we've talked about before, that it isn't really possible to 100% securely erase solid-state disks.

Steve: What we really need, and it's not part of the spec because drives are still pretending to be perfect, and they're pretending to manage themselves, we need an interface to the...

Leo: Low level.

Steve: ...the EPROM, to the controller itself.

Leo: Yeah, yeah.

Steve: Right. Where, like, we can look at the mapping table, or we could zero the mapping table and say bring it all back, or swap these in so we can erase them. Or have the ability to know when we say "erase everything," it's erasing everything. And instead, they're just saying, eh, we don't have that feature.

Leo: It's because wear-leveling intervenes, right, between your...

Steve: Correct.

Leo: ...view of the drive and what's actually happening.

Steve: Right. And what we need is the ability to programmatically penetrate the wear-

leveling abstraction because that's an abstraction. We're looking at a subset of the actual drive, and we need to say, uh, no, give us access to the whole thing just for the moment.

Leo: Having said that, if you do an erase on an SSD, you're mostly getting the data; right?

Steve: You really are, yes.

Leo: There might be slack space that's not erased, or some individual sectors that get overlooked because we can't really see which sectors are used. But for the most part you're zeroing it out.

Steve: Well, and this is why the really good advice is what you referred to earlier, which is turn on encryption before you put anything of yours on the drive.

Leo: Right.

Steve: Because then it's noise which is recorded. Even if that noise gets swapped off because of a wear-leveling need, it's like it doesn't matter. It's just pseudorandom junk. And then when you wipe your keys from the drive, nobody can ever access it.

Leo: It's a small leak, but it's still a leak. And this is, by the way, true on your phone because that's using solid-state memory. So when I get a new phone, I'm going to get the new iPhone, the first thing I do, before I put any personal data on it, is I turn on encryption.

Steve: You and I are both going to be up at midnight on Friday night, my friend.

Leo: We are.

Steve: Or, I guess...

Leo: No.

Steve: ...12:01 Saturday morning.

Leo: Yeah.

Steve: Wait, is it Friday morning?

Leo: California time.

Steve: Friday morning or Saturday morning?

Leo: Friday morning. It's Thursday-Friday.

Steve: So Thursday night. Okay.

Leo: Yeah, don't miss it because you'll be screwed.

Steve: Oh, ho. I'm not missing it.

Leo: And especially because you want the - and I both, we both want the 6 Plus, the big one.

Steve: Everybody. Everybody. That's the one. I don't think they're going to sell any of the 4.7s.

Leo: Isn't that interesting? That will be telling.

Steve: What was the - it was a hundred dollars cheaper for the smaller screens?

Leo: Well, with the contracts. They didn't tell us what the off-contract prices were. So it's 200, the usual off-contract price for an iPhone 6, 299 for the iPhone 6 Plus with 64GB of RAM, which, you know what, people are buying too much - not RAM, storage. People are buying too much storage.

Steve: I agree with you. It's hard to imagine what you could do with 128GB. It's like, 128GB.

Leo: That's so you never have to ever erase a picture or a video ever again. Please...

Steve: And if you think your iCloud storage is full now...

Leo: Oh, lord.

Steve: ...just wait till you start having multiple 128GB blobs tearing around.

Leo: Well, to their credit, they did drop the cost of iCloud storage to $10 a month for a terabyte. I'm sorry, $20 a month for a terabyte.

Steve: I don't want that.

Leo: And I don't like, no, I'm [mumbling].

Steve: No, no.

Leo: Chris White, Charleston, South Carolina. He's puzzled - as am I, so I'm glad he asked this question - by Perfect Forward Secrecy. In your most recent episode you describe Perfect Forward Secrecy as being unbreakable should the keys ever be compromised. I've been wracking my brain, trying to imagine any encryption that could accomplish this feat. The only thing I can come up with is some sort of pre-shared nonce that must be combined with the encrypted one-time key to decrypt successfully. But I don't know. Even that could be compromised. The only way to be completely sure in the case of compromised keys would be to add a time-based component, and that doesn't sound trivial for email.

As you may guess, I'm pretty much a novice at this stuff - you and me both - so there's probably something simple I'm missing. Can you fill in the gap? And thanks for all the great work you do with Security Now!. Because of you, whenever the conversation turns to digital security - quite often lately - I'm informed and stay up to date. How do it work?

Steve: Well, I loved the question because clearly Chris cares enough to have really thought about this. I mean, he sat there, and he's like, how can you arrange...

Leo: How could they do it?

Steve: Yes. And so the secret is something known as a "key negotiation," where the two endpoints are able to, in real time, exchange some data with each other. And the trick is, because this could be happening where there's an eavesdropper, they need to be able to do it in plain sight, meaning that - so imagine a protocol where the two ends are going to send each other some information such that, when they each receive it, they're able to both arrive at the same secret from what they exchanged. Yet the person watching that, even if they, like, capture the data going by that they each sent to each other, that person in the middle can't figure out the key that they have now agreed upon through this key agreement protocol.

It's diabolically clever. It uses the fact that, if you take a number, and you raise it, say we'll take, well, in crypto parlance it's called the "generator," so we'll call it "G." And you raise that to the power of "A," and you raise that to the power of "B." You get the same answer, the same result as if you take the same generator and raise that to the power of "B," and then raise that to the power of "A." In other words, the order in which you exponentiate something doesn't matter. It's commutative.

So the secret here is - and then we add one more twist, and that is we do this in what's called, again, the fancy term is a "finite field." And what that really means is that we're not just going to - we're not going to take "G" to the power of "A" to the power of "B," which could be like this ridiculously huge number. Instead, we take that answer "mod" something, as in "modulus," which really just means we divide that by another number, and all we keep is the remainder. And it turns out that even adding the modulus - and this is really where this is the trick - adding the modulus, where all we keep is a remainder, all the math still works. Yet in sending just the remainder to the other end, we're not giving away the answer. We're sort of - we're only disclosing sort of a piece of it.

And it turns out then that what each end is sending is - so side "A" takes this generator, which can be publicly known and is normally just standardized, and the first side comes up with a random number. We'll call that "A." So they take this generator to the power of "A," and then do modulus to, again, another agreed-upon number that can be publicly known. Then that gives them this remainder. They send that to side "B." And at the same time, side B has come up with its own random number, done the same thing, raised "G" to the power of "B," and then take that mod whatever, and they send that remainder to the first side. Then each side raises their value to the remainder the other side sent, and they arrive at the same number. Yet all the person in the middle sees is these two remainders going back and forth from the mod operation, which tells them nothing.

So it's just - it's genius. Diffie, Whit Diffie and Martin Hellman, who I actually knew and met at the AI Lab at Stanford back in '73, they invented this, the so-called "Diffie-Hellman key exchange protocol." And if you want to see it, there's some nice pictures that Wikipedia has that describes this further, Chris. But I think you've probably got the hang of it. And just it's brilliant. And this is the way we do ephemeral key negotiation on the fly such that it doesn't matter if someone catches our encryption that was used to authenticate the ends. They don't get the encryption key because that was negotiated on the fly, even in plain sight.

Leo: Question 8 from Richard Warriner in Bedford. We have a lot of English folks calling us and writing us.

Steve: Yeah.

Leo: He worries that he might be wearing out his iPad's flash: A thought came into my head whilst out running this evening. I have a 64GB iPad which has about 6GB free what with all the apps and stuff. Most of this is static, never changes. Now, I listen to several podcasts every day, and most are videos. So my 6GB free is constantly being written to when I download in the morning and then deleted from during the day. This obviously repeats every day.

I know flash RAM is fine to read from repeatedly, but not write to. It's actually not flash RAM, it's NAND. The question is, when I'm using the 6GB of flash, am I going to wear it out prematurely because it's my only usable storage? If so, what can I do? Should I be worried? Regards, Richard.

Steve: Okay. So this is sort of an interesting question.

Leo: It is a good question.

Steve: I liked the way he thought about it because he's saying I've got 64GB, but 58 of it is full and never changes. Well, except that apps are being updated with frightening regularity. But so that leaves 6GB free. And he's listening to podcasts. And so that little free space is being written and written and written and written and written all the time, compared to the bulk of it that's not being written. So is he going to burn that out? And the answer is, well, maybe your great-great-great-grandchildren would have a problem if they're still listening to podcasts and recycling that little corner of the flash memory. It's a matter of scale. Because generally we're talking about on the order of 10,000 writes. And if you're doing one write per day, that's 10,000 days.

Leo: Other stuff is going to wear out sooner than that.

Steve: Your body is going to wear out sooner than that.

Leo: Well, no. That's only, like, 30 years. That's not an infinite amount of time.

Steve: Oh, okay. And we don't know how old Richard is. But I take your point.

Leo: Yeah.

Steve: So, and of course then we add to that that, as the flash, because of this problem, as bits wear out, they'll be taken out of service and new fresh little regions get brought in. The so-called "wear leveling" is happening. So, and actually that's the other side of this is, even though you're seeing a small portion that appears to be written over and over and over, what wear-leveling is, is the deliberate spreading of the writes out across the entire region of the drive.

So the drive could take some of the data that isn't being written often and put that over on an often-written area, freeing up the un-often-written area, which will then get some writing to it. It levels it out so that hotspots, exactly like Richard is describing, aren't a problem. Although in general this is really only a concern if you were using that solid-state drive as, like, your main drive in your PC, where you're just thrashing on it all the time. I mean, our hard drives never stop. They're just busy doing something all the time.

Leo: It is possible to over-thrash it; right? I mean, didn't Mark Thompson use it for swap file or something?

Steve: Yes. In order to do an experiment, he used an SSD flash drive that - remember the swap factor that was...

Leo: But these were early ones.

Steve: Yes, that was a square. And he set it up as his swap file, and it just swapped it to death. I think it lasted a matter of hours, and it was toast.

Leo: But this is not that thrashing.

Steve: Yeah. And we've had a lot of progress in technology since then. And the wear leveling really does solve the problem. And the fact that 10,000 writes is actually a lot when you consider, like, a podcast is changing in that region.

Leo: Yeah. I've not, you know, I've never - I use SSD in everything now.

Steve: Right.

Leo: I don't have spinning drives. And I have yet to have a problem. They may be more robust, we'll see, but they may be more robust than spinning drives.

Steve: The only problem, and this is, of course, I have a bias because I hear about people with problems, is that they sometimes just spontaneously die.

Leo: Well, yeah.

Steve: Whereas hard drives tend to creak for a while and get cranky and slow down and then, I mean, they give you some clues and some, like, some ability to, like, kick yourself for not responding to those clues sooner. Whereas SSDs just suddenly, unh, I'm now a black piece of plastic.

Leo: Well, and I have had flash memory die for cameras, and that's exactly right. It just stops working.

Steve: Right.

Leo: One day it works; one day it doesn't.

Steve: It says, "I'm not memory anymore."

Leo: No, I'm just a piece of black plastic.

Steve: Yup.

Leo: Freddy in Stafford, Virginia. He was thinking about those security questions that we talk about all the time, and in fact that Apple says were part of the flaw that allowed the nude pictures to be stolen from iCloud. Steve and Leo, while listening to the podcast on the celebrity nude photo compromise, it struck me that wouldn't it be fairly simple to modify the standard mother's maiden name/pet questions to ones that each person decides for themselves and enters the answers? That would eliminate a lot of the pre-attack social engineering research. Some sites do that. Good sites do that.

Steve: Yup. I've seen the same thing. But when I read this, I was put in mind of a recent comment you made on one of the podcasts, might have been TWiT on Sunday, I don't remember when it was...

Leo: Several times I've said this.

Steve: But your advice was, and this is why I wanted to use this question, is absolutely never tell the truth.

Leo: Right. Just lie.

Steve: Yeah. You do need to record your lies. Maybe you could always use the same lie, but that's as dangerous as using the same password on other sites. You'd really like to do per-site lies, which means you then need per-site records of your fabrications.

Leo: As you already have with your password manager.

Steve: Yeah.

Leo: Right.

Steve: And the problem, I think the reason more sites don't allow you to make up your own question is it's going to confuse some people. And apparently they've got problems with their databases as it is. And so adding fields of arbitrary length questions that they solicit from people, that seems to be beyond them. So, yeah.

Leo: The point of these is to give you a way to authenticate if you forget your password. And sometimes they use the equally crappy method of what are the last four digits of your Social. I just recently was authenticated using my mother's birthday, another terrible. And by the way, that wasn't something I chose. That's just what they use.

Steve: Yeah, or your favorite teacher is a favorite one. I mean, I guess pet's name is so clich now that they're embarrassed to still ask that.

Leo: But is there a way that you could do this kind of backup authentication? I guess there is. That's what Google does with your smartphone. You give it a smartphone phone number in case somebody steals your stuff, and you say don't, you can't verify it using my email, call me.

Steve: Yes.

Leo: That's a good way to do it.

Steve: Yes, yeah.

Leo: You call me.

Steve: Yeah.

Leo: Frank in Munich figured out...

Steve: Well, and I was just going to say, the other beauty of that is then, if someone is trying to hack your account, you get a call.

Leo: I get a phone call.

Steve: And it's like, oh, wait a minute.

Leo: It's too expensive.

Steve: Ah.

Leo: It's too expensive. Frank in Munich figured out the free CA that was dropped from Firefox: I'm guessing, when you were originally talking about StartSSL, you were thinking about these guys, cacert.org.

Steve: The moment I saw that, it's like, oh, yes, that is what I was thinking of. And on their inclusion status page, this is just sort of a nice, well-meaning, small certificate authority that's sort of trying to get themselves going. Lots of people don't support them, that is, don't have their root certificate. FreeBSD doesn't. Safari has it under advisement or consideration or something. Firefox has it marked red, as like, uh, no. We're not going to - and that's when I had the moment.

So thank you, Frank. You solved the mystery. I kept talking about it on the podcast, it's like, I'm sure I saw that somebody, like Mozilla was going to stop supporting something.

It's like, yup, that's - and so these guys are like, just, they're well meaning, but they're not providing much authentication, and none of the browsers want to honor their certificates. So they're kind of having a tough time. But the mystery's solved, thank you.

Leo: Thank you. Last question. Kind of sad.

Steve: Well, considering that it's 4:00 in the afternoon, I think we're on track here.

Leo: Yeah. TN2 coming up, for those of you tuning in. Before You Buy will be right after Tech News 2Night. And there's a big breaking story on Tech News 2Night that is not Apple.

Steve: Ooh.

Leo: Brent in Canada - I'll just, I'll leave it at that. Brent in Canada suggests that zip codes make a lousy second factor: Steve, I live in Canada. We don't have five-digit zips. We use a six-digit postal code. Actually it alternates between letters and digits. U.K., similar. I frequently visit the U.S., and often run into gas pumps that require a five-digit zip code. When I do, I always have to go inside, and the attendant swipes my card manually to avoid the need for a zip code. Seems it's pretty easy to bypass this as a second factor. It also shows why zip codes might not be useful as a universal second factor. Perhaps a phone number?

Steve: Well, okay. So this was sort of interesting because...

Leo: We actually use this to identify Canadians who visit the United States, I just remembered.

Steve: This is, I think, the wrong way to think about this. The idea is that we're looking for something that will prevent fraud at the gas pump, and zip code does that. I mean, it's not perfect. But specifically the bad guys don't want to go in and present their card because the attendant might ask to see their ID. And so the whole point of this being at the pump is that it's a very well-known way for people who have stolen a credit card to check to see whether it's still good or not. And they get some gas in the bargain. And this just stops them cold because they typically - this is something they don't know.

And so, yes, a phone number, I guess. But a phone number seems a little more too personal to be used. I'm not sure you want to be entering your phone number, whereas a zip code - the point being your phone number identifies you uniquely. The zip code doesn't identify you, yet it's still something no thief can guess. They're just not going to know what the zip code is. So I like this as a second factor in that setting. It's certainly not universal, but it does the job.

Leo: Yeah, exactly. And it also helps us identify Canadians.

Steve: [Laughing]

Leo: And other "furriners." No, I'm kidding.

Steve: That's right.

Leo: I'm kidding. I love Canadians. You know I love Canadians. I know where Moncton, New Brunswick is.

Steve: Just think of the keyboard we'd have to have on the gas pump if we were going to just let people put in their wacky Canadian zip codes.

Leo: [Laughing] Hey, Steve. What fun, as always. Steve Gibson is the man in charge at the Gibson Research Corporation and absolutely our security guru everywhere in this building. I mean, it was great seeing you on This Week in Enterprise Tech yesterday. It's nice to have a go-to guy that we can just say, "Hey, Steve, what's the story?" and you tell us. Next week we're going to do - what did you say?

Steve: I think we're going to - I want to do a deep dive, industry willing, into the controversy surrounding Google's unilateral decision to start to put pressure on SHA-1 certificates well in advance of, like, everyone else in the industry's feeling for when we should stop honoring them. We've sort of agreed, eh, 2017. Chrome wants to do it in two months.

Leo: Google's a very take-charge kind of company.

Steve: They're definitely...

Leo: They're using their clout.

Steve: They've definitely stirred, they're stirring it up lately, yes.

Leo: If you want to get the transcripts of the show, Steve has them, really nice transcripts written by a human, Elaine Farris, at GRC.com. He also has 16Kb audio for the bandwidth-impaired. You'll find lots of other great stuff at GRC, including SpinRite, the world's best hard drive maintenance and recovery utility, and Steve's bread and butter. So buy a copy. Come on. You'll also, if you go there...

Steve: Keeps the wheels turning.

Leo: Yeah. If you want to ask questions for future feedback episodes, you could do

that at GRC.com/feedback. Please don't send email to either me or Steve. We're far too busy playing Minecraft to answer your email. And...

Steve: I'm busy writing SQRL.

Leo: Oh, yeah, that. Oh, that's what you're doing.

Steve: And then back to the next version of SpinRite. But point taken.

Leo: Something like that. Anyway.

Steve: Yeah.

Leo: We have audio and video, full quality, at our website, TWiT.tv/sn. And you can also subscribe wherever finer podcasts are aggregated, like iTunes, or use the apps, we've got lots of them, Stitcher, that kind of thing. It's easy to find us. This is the No. 1 security podcast in the world. I don't think you'll have any trouble. Now nine years in the making.

Steve: We're in our 10th year.

Leo: Yeah, 10th year.

Steve: Because we had our ninth birthday. So we're in our 10th year.

Leo: Wow.

Steve: Woohoo.

Leo: Thank you, Steverino. See you next week.

Steve: Okay, my friend. Right-o.