SECURITY NOW!

Transcript of Episode #477

Payment Tokenization

Description: After catching up with another interesting week of security events, including the rumor of a pending SSLv3 flaw and a new Windows zero-day exploit, Steve and Leo examine the next evolution in online payment technology which replaces traditional credit card numbers with "Payment Tokens."

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-477.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-477-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here, and he's going to analyze how Apple Pay, or at least how he thinks, what we've deduced about how Apple Pay works, how that's going to change payments in general, and how secure is it. It's all coming up right now on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 477, recorded October 14th, 2014: Payment Tokenization.

It's time for Security Now!, the show that covers your security and privacy online with that guy right there. His name, Steve Gibson, a name you must remember because he is your best bet on the Internet. I just made a slogan for you. Hi, Steve.

Steve Gibson: The best bet on the Internet. Well, okay.

Leo: He's protecting us, my friends, protecting us.

Steve: I'm the only bet you have, and I'm here, and I'm keeping an eye on things.

Leo: Yes, sir.

Steve: So my original working title for today's podcast, I was going to have a little fun and call it "Apple Pray," P-R-A-Y, because they would be praying for it to happen, as opposed to P-R-E-Y, where they would be preyed upon, or other people would be prey. I thought, okay, Apple Pray, that's kind of fun.

Leo: I like it.

Steve: And I was also initially sort of skeptical, like we commented when the 5 came out, the iPhone 5, that it was missing NFC, and that seemed like a mistake. And then when they did the second round of 5 with the "s," it's like, wait, still no NFC? Boy, this seems like a bigger mistake. Now, finally, with the 6, we have NFC. But it seems like a heavy lift to where your requirement for Apple Pay is a new iPhone and adoption at the other end of NFC terminals.

But the more I looked into this, I mean, I've spent my spare time this last week digging into what, like the whole back story here. And I realized this is actually much less about Apple than Apple would like, much less about Apple than all of the technical press seems to think. I mean, that's, you know, Apple's been like the focus. Actually it's called Apple Pay, so that makes sense. But as I looked into it, I realized the good news for, for example, all Android owners, of which there are many more than Apple, and all of whom already have NFC-enabled phones, is that the role Apple has played, and, I mean, and has, like last month and this month, because we're expecting maybe that this is going to get activated in three days, right, on Thursday, maybe with the iPad announcement, and that may be when this actually happens, that Apple ends up being the prime mover.

I mean, there have been efforts at this that have sort of limped along and not quite achieved critical mass. There are players who thought they could go their own way and have, like, created pieces of this that we'll be talking about in this podcast today, that never achieved critical mass. And so we all ultimately, I think, owe Apple our thanks for finally getting around to deciding it's time for this.

Leo: Kind of nudging it along.

Steve: Well, yeah. And it's the...

Leo: I think it was, you know what, it gave it critical mass.

Steve: Yes.

Leo: And all in position, it was ready, and Apple just kind of pushed it over the edge.

Steve: Also, as I looked around, I have a better sense for how much Apple did and what they didn't. And they did a lot. But they didn't by any means create this. Their specification is based on something that began 12 years ago. And anyway, so I renamed the podcast. The podcast's name ended being "Payment Tokenization."

Leo: Well, that's catchy.

Steve: It's, well, it's also fitting for this podcast.

Leo: Trust Steve not to worry about SEO or any of that, you know. Apple Pray, now, that'd catch some downloads. Payment Tokenization, that's Security Now!. And if you listen to this show, that's what you're looking for, frankly.

Steve: Well, and the full working title is "Payment Tokenization, or Why Indirection Is a Good Thing."

Leo: Ah, all right. I get where you're coming from, dude. And I think this is a good topic. You know, Apple will certainly talk about it day after tomorrow. There was a leaked training memo from Walgreens saying they were going to implement Saturday. I think the other thing to point out is that Apple timed it because they knew that Congress had mandated and Visa and MasterCard had agreed that we would move to new payment systems next year, chip-and-pin. And so the old swipe and signature cards...

Steve: Swipe and pray.

Leo: Swipe and pray. That's the real praying. Those would become obsolete next year, and most merchants would have to, would be incented to, strongly...

Steve: Upgrade terminals anyway.

Leo: ...upgrade to touch-and-pay terminals to go along with their new chip-and-pin terminals. So, you know, it's all in one. So I think Apple is really benefiting as much from that as anything else. But hey, who cares? This is going to be good, I think. We'll find out. In just a second Steve will explain the security around it and whether you can trust it. All right, Steve, let's launch into it. You've got security news, I imagine.

Steve: Oh, we always do that. Okay. This is Patch Tuesday. And this is sort of a special Patch Tuesday because a coordinated release of information about a major zero-day exploit, which is patched with today's Tuesday patches, has all been put together. A security research firm, iSight Partners, identified some time ago something that they're calling the "Sandworm Team." There are five teams of attackers. This is my new attempt to use the word "attacker" rather than "hacker."

Leo: I like that. That's good.

Steve: Yeah, it sits well with me. I think that's the right thing to do. They've identified, in Russia, five different Russian groups. And they see them in teams because they seem to have, aside from teamwork, sort of collective styles by team. This group uses references to "Dune," the classic sci-fi series.

Leo: That's sandworms, that's where sandworms come from.

Steve: Exactly, in their command-and-control URLs. Yeah, in their command-and-control URLs and in the various malware samples that have been found. So it turns out that they've been involved, the Sandworm group, and thus the Sandworm zero-day exploit, has been attacks, targeted phishing attacks, against NATO, Ukrainian government organizations; Western European government organizations; energy sector firms, specifically in Poland; several Western, I'm sorry, European telecommunication firms; and an unnamed U.S. academic organization. So explicit deliberate attacks have been traced back to this group.

And when the attacks were analyzed some time ago, the iSight Partners group, the security firm, saw the use of a then-unknown flaw which was - I got a kick out of this because it says all supported versions of Windows. Well, it didn't exist in XP. XP has always been safe. It was introduced in Service Pack 2 of Vista and has been in Windows ever since. So from Vista Service Pack 2 on is this zero-day, which they analyzed and informed Microsoft of immediately, saying, you guys may need to fix this pretty quickly because this is being exploited right now by Russians against various groups around the world. So today's Patch Tuesday fixes that, among the regular gaggle of - there's a bunch of .NET vulnerabilities, a collection of remote code execution things.

So, again, standard advice. This is Patch Tuesday; update your Windows. Though I would argue that, now that this is known, it's become known, there will be a heightened interest in using it before it gets patched, which means that patching is all the more important. And I think I remember seeing that they were explicitly using PowerPoint presentations as the delivery vehicle. So they were sending PowerPoint presentations to people with human factors, social engineering exploits wrapped around them to get people to open these. And that's the way this stuff was getting in. It was a vulnerability in the OLE, the Object Linking and Embedding package manager in Microsoft Windows and Server, as I mentioned, introduced with Vista Service Pack 2. So it affects all versions of Windows up through 8.1 and Windows Server versions 2008 and 2012.

It is an arbitrary code execution vulnerability in something called the "packager.dll." That's the file that has the problem, which allows the execution of INF files. So PowerPoint was the entry vehicle that leveraged this packager, as we used to call it, OLE, Object Linking and Embedding, that was part of the evolution of Win32 as Microsoft was moving forward before .NET and all that. So just I would say don't - either be very careful with anything that might get sent to you, and/or get Windows patched sooner rather than later.

Leo: Any exploit, well, it's a zero-day, so there's exploits in the wild already; right?

Steve: Yeah. Oh, it's been happening. And so targeted attacks. And they've been keeping it relatively quiet and using it while it wasn't known. So the calculation, the calculus about how to use this changes today with - now that it's become known, there's no reason for those who know of it to keep it quiet. And analysis of this vulnerability will allow other people to figure it out, to reverse-engineer the patch and make it more widespread. So this goes from being hush-hush targeted, you know, we know of a vulnerability nobody else knows about, so we're going to be careful not to expose it. Now the logic is let's use it as fast as we can before people get patched. So during this Window from disclosure to patch, what we've seen in the past is a flood of attempted exploit. So the more casual user is now going to be exposed to this, where before it was being used in much more undercover...

Leo: Spearphishing.

Steve: Yes, exactly. Exactly. They were using it to penetrate specific organizations. Nice hat, Leo.

Leo: Thank you.

Steve: [Laughing]

Leo: There's a reason why I'm wearing this panda hat.

Steve: I'm going to just keep focused on my notes here.

Leo: Someone will explain it to you on Twitter, I'm sure.

Steve: Okay. Speaking of foxes.

Leo: Yes.

Steve: We have a new Firefox v33. And something happened with 33 that I thought was really nice, which is that we've touched on this off and on, and I know you've talked about it on the network. And that is the problem with arguably the best compression technology, which is H.264, being encumbered by patents. And patents have long dogged the high-quality compression because they've always been present. And so people have sort of had to use less, a little sort of maybe patent-murky alternatives and so forth. Cisco has finally agreed to essentially open source and license their open H.264 codec. And with Firefox v33 that just happened hours ago, if you go, if you've got Firefox running and go into Help > About, you'll see that you've got probably 32.02, and it'll immediately start spinning its little wheel and downloading 33.

What's interesting about this is Firefox cannot themselves bundle this open H.264 codec. But when you load Firefox 33, it will, it itself, your instance of the Firefox browser will go get a verified build. They figured out how to verifiably download from Cisco a codec built from the source which Cisco has put up on GitHub. So all of this fancy glue was put together that ends up allowing Firefox to now finally support H.264, you know, arguably the codec everyone wants to support, not, for example, I think it was forced to use VP8, which was the one that sort of had - it kind of came through, what, Adobe and Flash and...

Leo: VP8 was a Google - wasn't it Google that...

Steve: Yeah, that's right.

Leo: They wanted an unencumbered video codec they were looking for, yeah.

Steve: Right, right.

Leo: And then of course MP4 and the others, owned by somebody.

Steve: Yeah. And now we have H.264, which is available in Firefox.

Leo: But that's owned by the MPEG Motion Picture Group, that's the problem, you know. And they've said we'll never charge for that, except we don't know because Microsoft and Apple and all the others are big players in there. So that's why Google has to do this. It's a big, you know, you're stepping into a big political fracas that's been going on for some time.

Steve: So the good news is Firefox has it now.

Leo: Yes.

Steve: For free.

Leo: Yes.

Steve: And when you load Firefox, it'll go to Cisco and get the codec.

Leo: Right.

Steve: So also...

Leo: That should be a hint right there that there's a problem, but go ahead.

Steve: Yeah. So faster, snappier searches in Firefox 33. They did something with JavaScript strings that made a huge difference. They were recognizing that large text apps like Gmail was taking an awful lot of space when you loaded a whole bunch of JavaScript strings. And they also noticed that most of the strings, even though they were taking up two bytes because they were unicode, they were strings that would fit in a byte. And so what this change did, and this is first reflected in Firefox 33, is they're now using 8-bit characters where they can, rather than 16 bits. And in some testing, for example, they loaded Gmail in Firefox, and in Firefox 32 it took 11MB on this one browser tab to contain Gmail; in Firefox 33, 5.5. So it cut it in half. Which you'd expect from a character representation going from 16-bit characters to 8-bit characters. So this is neat for users of heavy JavaScript string apps like Gmail. So a nice improvement in Firefox 33.

And then they've beefed up session restore. So I guess there were some ways that Firefox could collapse or crash or your Windows could die or various things could happen where Session Restore would not be there for you. And it looks like, I mean, they're very bullish about that now in 33. It looks like they've got that nailed. And then a bunch of other little things like CSS3 stuff. Custom counter styles are now supported. There's some experimental work. A 4x4 matrix is represented natively, which speeds up 2D and 3D operations for matrix management and multiplication and manipulation and so forth.

So just, you know, it's funny because, as I wrote down 33, I thought, boy, you know, it wasn't long ago we were at v3, and then we moved after a couple years to 4. And then of course they went on to this whole, okay, we're going to accelerate our versioning and not just move so slowly, which has been a nice change. And, you know, Firefox continues to stay current. It and Chrome I think are probably the two browsers of choice now.

Now, all we have is a rumor of something wrong with SSLv3, which is not going to be announced until around noon tomorrow. So this is Tuesday the 14th of October. Around noon on Wednesday October 15th the release of information which is known to those who are apparently busily patching SSLv3 right now. We just don't know what this is. I picked up something in the grapevine about a version regression problem, like maybe there's a way to cause it to use security suites that you'd rather it didn't use, that kind of thing. Or maybe it's a TLS problem where there's a way to get TLS to fall back to 3, and then you lose some of the features of TLS that v3 of SSL didn't support. And it might be, you know, not a big deal. We just don't know.

I wanted to mention it because I'll certainly be on the lookout for this tomorrow. I'll tweet what is known, if it's important. I'm happy to jump on with Sarah for the second round of Tech News in the afternoon if you guys want me to figure out what this means, maybe just say, okay, it's nothing.

Leo: I'll pass it along, yeah.

Steve: Yeah, or maybe it's something. So at this point we're sort of in this odd place where the news is out that something is going to be announced tomorrow that those who are responsible are right now patching in advance of everyone finding out. And again, it may be a small thing. My sense is it's - the phrase that I like is "tempest in a teapot," that it's like not that big a deal. But, you know, we'll have to wait and see. So I just wanted to give people a heads-up that there maybe some news. Oftentimes it seems that things happen on Wednesday, right after the podcast. It's like Shellshock was a Wednesday event. It's like, oh, no, we've got to wait seven days.

Leo: Steve, you just call, we'll get you on this. We'll figure it out. Just let me know, yeah.

Steve: I also did pick up on another - it's funny, having spent our time talking about supercapacitors some years ago, people now think of this podcast as - I guess their sense is they can send me news of, okay, what does this mean about energy storage, and I'm interested enough in it that I'll figure out what it means, and then I'll tell everybody.

So what's interesting about this is this is a news item that was posted on Physics.org site about sort of where we are is are we going to stay with batteries, or are we going to do something bizarre and new like supercapacitors. And batteries have the advantage of a huge supply chain in place; lots of electrochemistry understanding. There's already like a big industry in place. And so if we do something completely new, there's a big tooling time, you know, for like tooling up to implement something. If we're able to make what is a relatively small change to an existing electrochemistry, then that has the potential to actually happen, which is good because that's when - when the battery gets into our phone is when we care.

So what we have here is that, a relatively small change from the guy that did lithium ion 34 years ago, who's still at it. And this is changing the traditional lithium ion graphite electrode, the anode in the lithium ion cell, to a gel suspension of a titanium dioxide nanotube. And nano stuff happens in batteries now because what you need is surface area. The key is, in order to prevent batteries from dying because the surface area gets corroded and corrupted, and the limitation on charging rate is also a service area deal, the more area you have, the more current you can push into the battery without causing damage to the battery. And so nano stuff seems to be where we are. So these nanotubes are, as you can imagine, being a tube, high surface area things, in this case made of titanium dioxide.

The result of this is they have batteries in the lab which are fundamentally lithium ion chemistry, that is, lithium ion electrochemistry, which is well understood, and for which, like, all of this existing manufacturing system already exists, which can be charged to 70% of full capacity in two minutes as opposed to many hours; lasts 10,000 recharge cycles, which is to say has a 20-year life; and is basically an electrochemistry already in place. So patents are being granted. Licensees are signing up. And this could be - this is something, it feels like something that we could see a year from now where the next generation of our portable devices, for example, iPhones with nonremovable batteries, could be charging in minutes rather than hours, and where battery lifetime of a nonremovable battery is no longer an issue.

And of course the electric vehicle market is panting for this sort of thing. You need the capacity to deliver a tremendous amount of current in a short time. But this could start to make the equivalent of electric gas stations feasible. If you can get 70% capacity in two minutes, well, we spend two minutes putting liquid gas, you know, gasoline in our tanks right now. So the problem is, you know, the typical home doesn't have the ability to deliver that kind of current in that short a time. So but something like an electric filling station could. So anyway, we'll see. But so thank you for those listeners who sent this to me. I got it from a bunch of people because they said, oh, you know, what does this look like? And I said it looks good.

Finally, there's been some odd news about Dropbox over the last week, about did they get hacked, was Dropbox hacked. Pastebin has been the recipient of some several hundreds of login credentials which have been verified. Before Dropbox neutered them, people were downloading them from Pastebin and logging in successfully to other people's Dropbox accounts. Which reminds me. Before I forget about it, I ran across something that I guess started off as a title in a recent security conference that Ars Technica picked up on, and then I saw it there and tweeted it. I just loved it. And it's a perfect meme for the Security Now! podcast. Rather than the Internet of Things, which is the terminology we typically use, for the Security Now! podcast it's the Internet of Other People's Things.

Leo: [Laughing] I like it.

Steve: Which I think is perfect.

Leo: I don't know if Microsoft would like it, but I like it.

Steve: So the Pastebin poster, the guy who's been putting these hundreds of login credentials up on Pastebin, claims that 6,937,081 account credentials have been leaked. And the use of a number like that is a little convincing. It's like, oh, you know. You'd expect an actual number, rather than, oh, almost seven million. Okay, well, no, 6,937,081. That many. So maybe this is a come-on. We're not really sure what this is because they've sort of posted a few hundred. Then the attackers are asking for more money. They're asking for money in return for the release of additional credential data.

Leo: [Muttering]

Steve: I know. Meanwhile, Dropbox has reset the passwords of all the credentials that have been posted. And in the latest news, Dropbox responds: "Dropbox has not been hacked." They said, quote: "These usernames and passwords were unfortunately stolen from other services and used in attempts to log on to Dropbox accounts. We had previously detected these attacks, and the vast majority of the passwords posted have been expired for some time now. All other remaining passwords have been expired, as well."

So I don't know what the story is. To me, this maybe looks like a come-on by the people who posted these, saying, look, here are some credentials that work. So imagine that they in fact got a much smaller number - or say that they took an existing large breach, because there have been large breaches on sort of an ongoing basis. They attempted to use those to log into Dropbox; found a bunch of people who unfortunately reused their password on the leaked accounts, their leaked credentials, and on Dropbox; gathered those together, posted them on Pastebin, saying not only do we have these hundred, or hundreds, few hundreds, which is probably all they actually have, but we have 6,937,081 more. Pay us some money, and we'll release those. Which in fact they may not have. To me it sounds like they probably don't.

So I think that's the actual, you know, piecing together from what fragmentary information we have, that sort of fits the facts. So we'll see if anything more comes of this. But if we're to believe Dropbox, and I think we have to because they made a formal statement, I just don't think there's a major breach of seven million Dropbox accounts. They're saying no, I mean, they're saying we know what happened. And that's all. Which to me makes sense.

And I did want to share a fun note that I ran across. This was just, okay, this was sort of a blast from the past - you'll get a kick out of this, Leo - and it also incorporates a bit of a SpinRite testimonial from Matthew Power, who is a SpinRite customer. And he wrote to our support email, saying, he said: "I'm a very satisfied owner of the SpinRite product. Since my purchase, SpinRite has brought back THREE [all caps] unmountable drives," exclamation point. He writes: "I firmly believe in this product. Because of SpinRite, people bring me their PCs when the other guys can't get them to boot."

Then he says: "I see that it is distributed in FreeDOS, which is awesome. I love SpinRite's DOS shell look." He says: "How did you do that?"

Leo: It's not a look. It's real.

Steve: Uh, exactly. He says: "I would love to write a small game with that DOS shell look and feel."

Leo: Um, okay.

Steve: "Maybe run in FreeDOS, as well. But I have never been able to figure out how to get that look. Are there any of your free examples I should be looking at?"

Leo: Aw, cute.

Steve: "Or maybe you know of a site I should check out?"

Leo: He's talking about your ASCII graphics; right?

Steve: Well, yeah, he's talking about the fact that it is running in DOS. It's a DOS app.

Leo: And you have the DynaStat stuff, but that's all ASCII.

Steve: Yes, like my...

Leo: You're just drawing in ASCII.

Steve: Yeah, like, you know, graphs and charts and all that. And he says, he says: "Remember when MS" - and then he has a link to Edit.com - "had the orange-block mouse cursor? That's the look I want. I have never read anywhere where someone knows how to do that. And yet SpinRite has it, minus the mouse. Any help? Thank you. Matt." So I just got a kick out of that.

Leo: One way to do it, use DOS.

Steve: For what it's worth, yes. He says successfully recovered unbootable systems for people, three unmountable drives, and other data recovery work. So thanks for sharing that, and a reminder to our listeners that SpinRite works. And Matthew, if you're listening, yeah, I mean, this is the real deal. This is SpinRite running in DOS. Now, 6.1 will remain there; 6.2, where I'm going to see how much more I can do for USB, will stay the same. But I'm planning to finally leave that environment for SpinRite 7. I'm going to - I'm not going to put myself under a time constraint for SpinRite 7.

Basically I'm writing 6.1 and 6.2 to buy myself time so that, I mean, basically I don't think 7 will ever be, nothing will ever be able to be faster than the next version of SpinRite. I'm going all out for performance to meet the needs of contemporary drives that just keep getting larger and larger, so that you'll be able to use SpinRite overnight on a three or four terabyte drive and have it done in the morning. That's the goal. 7.0 will be all kinds of features that people now want, like simultaneous operation across all the drives they have. There's no reason I can't do that except none of the existing infrastructure in SpinRite supports that. Or drive cloning and file system awareness and all these, you know, like the next generation of things that it would be fun to do. I enjoy developing SpinRite.

So the 6 series, these next things are there to tide me over while I start from scratch. I'm going to start over. I found the kit that I want to use. I'm still going to be natively booting my own system, that is, not hosted on an OS, but doing my own, but with a GUI interface and a mouse and a contemporary UI and a sort of a new foundation that will then, in the same way that a multi-windowed interface does, allow me to grow the product in the future; whereas I'm sort of stuck with, I mean, nice as the DOS shell look is, the user interface is limited. And functioning in a DOS-based environment, I've sort of reached the limit of what I can do. So that's where I'm headed in the future.

Leo: Well, for those who wish to give that great graphical user interface look with DOS and ASCII prompts, there is a library for Linux called Curses - you could write to it with Python or Ruby or anything else, or Bash shell - that'll let you do exactly that. And it's a library. I bet you you wrote a library with graphics primitives in assembler. Steve does his own libraries. He doesn't use anybody else's. But for those who don't have access to Steve's assembly language libraries for x86, Curses, C-U-R-S-E-S.

Steve: You know what I did, I prototyped - there was a DOS-based tool that Dan Bricklin created. It was called Demo, Dan Bricklin's Demo. And basically you moved the cursor around, and you could draw line art and mark blocks and move things around. And so I - and this has really, this has always been my development style. I did the same thing with SQRL, for example. People will remember I completely designed the user interface, and then I wired up behind it. And I did the same thing with SpinRite. I designed the UI because I'm UI-centric. That's what the user sees. So I think of the product from a standpoint of how it's going to be used. And oftentimes that gets me in trouble because I'll be overly aggressive. I'll say, oh, this is what I want it to do. And then when it comes time to doing it, it's like, oh, this is going to be hard. But now I'm committed because I know how I want it to look. Now I just have to make it go. So, yeah, that's the approach I take. And it was with that beautiful screen editor that Dan Bricklin wrote.

Leo: Yeah, and then it probably just output arrays of ASCII numbers that you would then just blast onto the screen, I would imagine.

Steve: I wish I could say I did. Actually I wrote raw native code to implement each page. But I did write a...

Leo: Now move over once and put ASCII 374. Now move over again once and put ASCII 374. Like that?

Steve: Well, I had blocks of text. I had strings. And there was a lot of...

Leo: And you could make - I've done this, I mean, I used to write stuff like this all the time. You just make an array that contains all - and then you iterate over the array, and it's very quick.

Steve: Yes, correct.

Leo: Very easy. I'm sure you did that.

Steve: I did that.

Leo: Yeah. [Whispering]

Steve: Okay. So as I mentioned at the top of the show, I began digging into the back story, essentially, behind Apple's Pay technology, wanting to bring an understanding of it to our listeners for the podcast. And what I quickly ran into is something called the EMVCo, E-M-V-C-O, as in company, which is an LLC, a limited liability corporation, formed back in 2002, originally by Europay, MasterCard, and Visa, thus EMV, EMVCo. Then JCB, which was originally the Japan Credit Bureau, joined two years later in '04. Then American Express joined five years later, in '09. And then most recently China's UnionPay joined last year. And so now there are six main entities, each sharing a one-sixth interest in this EMVCo: China UnionPay, Visa, MasterCard, American Express, Discover, and JCB.

And the charter for EMVCo has been to agree upon standards for moving credit processing forward. And the major concept is tokenization. It's the buzzword we heard during the Apple Pay presentation. But as I guess Apple tends to do, they gave us sort of the feeling that this was all theirs. Apple, without question, did a lot of work because the spec which exists really should be called a "meta specification." It describes the players in the dance, but not the details of, at the low level, of how they communicate.

So, and even today that's still somewhat obscure. There's this notion of - the term is, shoot, I'm blanking on the - oh, the term is "cryptogram." And it's like, okay, cryptogram. And so I'm seeing the word used, but nowhere am I seeing a clear description of what it is. So at least in the meta specification, it's just sort of like, okay, a man behind the curtain, it's there, but we don't talk about what that is in detail here because we're too busy talking about the committee meetings that we had and who does what.

But here's the concept. The problem with breaches of security is credit cards escape. Credit cards are being given to the merchant. Credit cards are being given to, you know, when you pay for your meal at a restaurant. Credit cards are being put into websites when you purchase things. And the term is PAN, Primary Account Number. And so a PAN, P-A-N, is the official term for the user's actual account number, what is currently the only number we have, the account number, which is our credit card. And of course it's accompanied with expiration and the CVV, the card verification code, the little three- or four-digit thing on the back that you often have to use which is explicitly not on the mag stripe so that it can't be lost if the mag stripe is captured. So this payment token, what the payment token is, is a stand-in for the PAN, the P-A-N, what we're used to as typically a 16-digit; but it turns out it can vary from 13 to 19 digits, so it could be longer than 16. I've never encountered a 19-digit one, but they can be.

So this is - so the first thing to know is that this payment token looks like a credit card number. It is processed like a credit card number. And in fact it even obeys the so-called Luhn, L-U-H-N code. That's that deal that helps catch digit transposition, where all credit card numbers, to be valid, there's a sum of nines. That is, you add the digits up, and essentially you always end up, I remember what the algorithm was, I implemented it for GRC's eCommerce system myself, but anyway, you can look it up. The idea is that it sort of has - credit card numbers have a built-in checksum.

So the key takeaway here is that what we're being given with a payment token is a pseudo credit card number. And that's the key. It's not some base64 thing. It's not some crypto string. It's not ASCII. It's not, you know, it is a credit card number. And it was done this way because, as it moves through most of the system, there's no difference. No one can tell the difference. So, for example - oh, and as best I can tell, I saw one reference to the possibility that it was a per-use token, but largely I think that's not the case.

Leo: I think that's what Apple said it was.

Steve: And I don't think it is. In fact, I don't even think it is in the Apple case. I know, for example, that there is an instance, there's another use where they call them "card-on-file merchants." And, for example, Amazon is a card-on-file merchant where - meaning that they have their customers' cards on file and so that we're able to log in, reauthenticate, and click the Buy It Now button and, bang, our credit card is charged.

Leo: So Apple, in other words, Apple hasn't documented this, really.

Steve: Correct.

Leo: Okay.

Steve: Correct. And the documentation that I have seen, and I should say that I've watched, I've sat through really boring webinars of executives who were, I mean, like First Data has one, and there's links in the show notes to the First Data - there's a page there, Leo. You scroll to the bottom. There's a 60-minute webinar where the slides don't change nearly often enough. And people who don't actually seem to know what they're talking about are talking about things nevertheless.

Leo: No.

Steve: And so their terminology, I kind of cringe because it's like, oh, gosh, that's not - that can't be right. But I'm trying to, like, get the truth out of this. And but so my belief is - and so there may be a one-time something that is apparently...

Leo: What Apple said, because I was there...

Steve: Okay.

Leo: ...listening carefully.

Steve: Yup, I was listening, too, but go ahead.

Leo: I believe, I'll go back and check, is that there was a one-time-use token or a one-time-use number, not the credit card number.

Steve: Correct.

Leo: Which would be the token, presumably, and another PIN that's associated with the Touch ID or something else. But it was a two - it was two factors, or two things. And I'm pretty sure Apple said one-time use. But maybe they didn't mean on both. Maybe the token is not one-time.

Steve: So, yes. So before I forget, because I've been meaning to say...

Leo: Because maybe Tim Cook doesn't know.

Steve: He may have been part of the webinar. And, like, no, I mean, he wasn't. This was all First Data people. But they were in the lab. One of the guys has used the Apple Pay system 20 times. He's giddy over the user experience. And so the way this works is when - okay. I'm sort of out of sequence here. Way back in the chain of action, so we have a - say that we're in a retail establishment, and there's a touch pay terminal.

So you've got the purchaser with their iPhone. You've got the merchant whom you're visiting. They have, essentially, a provider that they have a relationship with. And this is a company like First Data. They're like a credit clearinghouse. Nova is one. And for example, that's the relationship I have with GRC's eCommerce system because, as we know, I wrote my own, did all my own eCommerce system from scratch back before I did SpinRite 6. And so I have a relationship with one of these clearinghouses, direct. And I don't remember if it's Nova, but like First Data is one and so forth. So it's not directly with Visa or MasterCard or Discover, but it's one stage removed. It's with a credit processing, like merchant services company. And then they tie into the credit card processing network where Visa and MasterCard and American Express and so forth reside.

So there's this pipeline that the credit card goes through. And what's happened is there's been one more player stuck in this chain on the other side of that payment clearinghouse, in front of Visa, MasterCard, American Express. And the terminology is a Token Service Provider, TSP, a token service provider. The token service provider is responsible for maintaining something called the Token Vault. And the token vault is where the relationship between the actual PAN, the Primary Account Number, which is the credit card number, and the token are maintained. The token is mostly a random number. So that's key. It is not cryptographically derived from the credit card because, if that key ever got loose, then all of the relationships would be vulnerable.

Leo: You've read this document from Apple, right, that says: "When you add a credit or debit card with Apple Pay, the actual card numbers are not stored on the device nor on Apple servers. Instead a unique device account number is assigned" - is this the token? - "encrypted and securely stored in a secure element on your phone. Each transaction is authorized with a one-time unique number using your device account number." That doesn't change, obviously. And instead of using the security code from the back of your card, Apple Pay creates a dynamic security code. So it sounds like, not only is the credit card number not involved, but it sounds like it's using a one-time number each time.

Steve: No.

Leo: It's not?

Steve: No. What's also in your phone is one of these tokens. Because as I understand it - and again, unfortunately what we don't yet have is real clarity. But from the people I painfully listened to, it was clear that the phone actually contains a 16-digit static token.

Leo: That's the identifier for the phone.

Steve: No.

Leo: No?

Steve: That is the - it's the token. And so here's the process. And again, I'm subject to the fact that there's still not complete clarity. When this association is created - so, for example, a user, an iPhone user scans a new card, right, because we know that they're able to do that. They take a picture of their credit card.

Leo: Yeah.

Steve: The card number is extracted. The card number is - and there's something called a BIN, B-I-N, which is like the ranges of card numbers. It turns out that within the 16 digits - for example, people may have noticed, and I don't remember now, but like all cards beginning with 5 are MasterCard. All cards beginning with 4 are Visa. All cards beginning with, like, 6 or something are American Express. Or there's some - so there's a system like that.

Leo: You can validate these card numbers, obviously; it happens all the time on the web.

Steve: But the point is that's actually - the card numbers contain routing information. So there's this notion of routing. There's a routing database. And so the user scans the card. From this routing information, that number goes to the actual issuing entity, Visa, MasterCard, or whomever. Now, they are token service providers. I know that. So they offer this new service. But there's also third-party token service providers.

Leo: Could Apple be one?

Steve: I don't think so. I was looking carefully to see about that.

Leo: So this is, look it, this is - Apple's unequivocal. When you add a card to Passbook, its number is never stored or shared on your device or Apple servers. I mean, that's unequivocal.

Steve: I think that's true. And so here's the point, is that what happens is the credit card number helps the system find the issuer. Then they request from a token service provider a token.

Leo: Ah.

Steve: And that's where this mapping gets made.

Leo: Because they have to know whose card it is.

Steve: Correct.

Leo: There has to be a mapping.

Steve: And so what looks like a credit card number, but is not, the token is associated with that user's actual - that PAN, which is called the Primary Account Number, which is the credit card number, and that 16-digit number comes back. The bank also provides the artwork, which is why what we see looks gorgeous. And the last four digits of the PAN, which is actually the credit card number, that is kept. So what the user sees is this beautiful-looking credit card art which came from the issuer, from Chase or from MasterCard or whomever. And they see the last four digits of their actual credit card number, and that's what's used on receipts and so forth, so that they can figure out which card they used to buy something and that. But that's all that's there. So their credit card number is not present. But a stand-in for their credit card number is present.

And so that's what this whole payment tokenization is.

Now, Apple could very well be doing more than that; that is, there could be some additional hocus pocus. I've heard mention of encryption. And so what I've said so far, nothing is about encryption. So the beauty of the system I've just described is it was a tiny change to the existing infrastructure. Only in this multiparty system, which involves the movement of what looks like, what is credit card numbers.

The idea is that, because we have a routing system that knows how to route based on a credit card number, then we're able to add another entity. We're able to add this thing called a "token service provider" that provides indirection. That's where I talk about indirection. They maintain the so-called "token vault," which remembers the association between the randomly assigned, what looks like a credit card number but is now just a token, and the actual credit card number. And they own a block of these credit card numbers so that they're routed, so that these pseudo credit card numbers are routed through them, and that's where the mapping occurs.

So the beauty is that the merchant thinks they're taking - and even the merchant terminal thinks it's receiving a credit card number. And from some people it is. From some people, they'll still be swiping their credit card. But Apple Pay users will be presenting a pseudo credit card number which runs through the system through the same process, except because it's a different range of numbers it gets switched over to this token service provider, where it's turned into, at the last stage, it's turned into the actual credit card number as it goes to the final routing stage. And the beauty of this whole system is your actual credit card number is never exposed.

So, for example, card-on-file organizations like Amazon - oh, and that's the other thing. There could be and probably will be multiple mappings between pseudo credit card numbers and your one real credit card. So, and those mappings could be and probably would be offered by, maintained by different token service providers because they would just be happening, you know, like Amazon could choose to use a different token service provider. And Amazon would decide, we no longer want the responsibility of maintaining actual credit card numbers. So we're going to swap them for tokens. And so Amazon says, using a backend API, works with their token service provider and exchanges all of their users' credit cards, actual credit card numbers, with tokens. And then the whole system continues to work.

Amazon then no longer - actually, one of the things that happens is they no longer need to worry about the full level of compliance because they're not actually - they're no longer storing and needing to encrypt users' financial data. They have this level of indirection. And what this means is, because you have an N-way to one mapping, is if breaches and losses of information occur, it's instantaneous to cancel a given organization's mapping. So if, for example, RSA got hacked, if RSA was tokenized, that is, they were only maintaining tokens rather than actual credit card information on file, they could immediately cancel the tokens, and they would no longer be usable, except there's one other piece of this. And that is there's this notion, a new notion of domains.

Right now we are in a domainless environment with credit cards, which means that a credit card number is honored no matter where it comes from. Now, of course credit card companies' security tries to help us out. We've talked about how like I can't buy gas with one of my cards because it just raises the alarms whenever I do. And there's also been situations where someone has flown across the country, and then tried to rent a car, and it was denied because they used their card that morning to put gas in their car or to park at the airport, and then suddenly they appear across the country. And again, security systems go crazy because it's like, wait a minute, there's 3,000 miles' separation between two uses of the same card in six hours. And so that seems questionable.

Now what happens is Amazon's use of tokens would be tied to Amazon. So no one could use the token that Amazon received from their token service provider except Amazon. So this is a whole 'nother layer of security we've never had before. Similarly, the token that would be issued to Apple, or actually issued to an iPhone user, but it would be known to be an Apple iPhone token, it could never be used as a credit card number by anyone else. So this is another aspect of this. And again, this is not Apple's invention. All of this existed. And Apple will probably - probably deserves credit as being the entity that finally got this off the ground.

I originally encountered, when I was looking at this, sort of this mythology which has numbers that I think makes it real, that Apple has been aiming at this for some time, that they began work in earnest at the beginning of this year, in January of 2014; that Visa had a thousand people on the project; JPMorgan Chase had more than 300; that it was done under a cloak of super secrecy. No bank knew of any other's involvement explicitly, but they had to know that Apple was going to be involving as many players as possible. So it is absolutely the case that Apple will probably end up earning and deserving the credit for being enough of a big player to finally get the system off the ground. Lord knows it has lots of buzz.

But the good news is that there are an awful lot of Android phones out there that have had NFC capability a lot longer than the Apple iPhone 6. And given the proper technology at the handset, all of this is available. So one of the things that became very clear was that, while Apple Pay - the trademarked Apple Pay specifics is Apple and proprietary, and they have some patents. The system in general is not of Apple's creation. Everything I described with token service providers and the system of routing and credit card numbers standing in for others, that's all existed before this. It just it never really happened. And exactly as you were saying, I think before we began recording, Leo, since chip-and-pin has been mandated to happen next year, and existing terminals will have to be upgraded, now they'll be upgraded with NFC receivers and this touch pay technology.

Leo: And we all win then; right?

Steve: Yes, yes, we all do.

Leo: Does the Android, the Wallet, Google Wallet, touch and pay, work kind of the same way? I mean, are they all similar?

Steve: Well, the question is the hardware support. One of the other aspects of this, I mean, there are many cool things. It's very cool that there's this notion of a domain, the domain validity, where the token was issued, who the token was issued to, and where it is then valid. Because that means you're not actually using your credit card, you're using your token within a restricted domain. So that token isn't useful, for example, outside the iPhone.

Well, one of the other new features is, it's a 00 to 99 scale of how confident we are in what they call the "user to token binding," that is, what's the security at the user end? And one of the things that Apple did - again, I don't always think they do this deliberately. I think they sort of luck into some of these things. But I've heard you mention for the last several weeks, Leo, many times on the network you've referred to how good the Touch ID system is on the iPhone.

Leo: It works very reliably, yeah.

Steve: And, boy, I wanted to jump in there and say yes. I no longer - I used to have mine covered up on my iPhone 5s. Of course I famously had the typo keyboard, and so I had no access to it. Now I'm back with the 6. It has never failed me once. I did chuckle a little bit watching it training more than the iPhone 5s did.

Leo: Yeah, yeah.

Steve: They learned their lesson. They needed to give it more samples upfront to get it trained up. But it just works perfectly. And so I think with that and with this notion of the Secure Enclave, sometimes referred to as the Secure Element, the idea that you have a coprocessor that is read-only, that is, you can ask it to do crypto work for you, you can say, like, encrypt something for me, or hash something for me. But it uses a key that it generates internally which it never exports. There is no API, no way to get that exported. You can only have it do the work on your behalf and then provide the results, which is really good security. And that's - it's intimately involved in processing the data from the Touch ID processor in the phone.

And so the point is that Apple is able to assert and substantiate in this rating system in a very high degree of confidence, that is, of security. And what that lets them do is negotiate a lower cost for transaction through the payment system. Because, for example, traditionally, there was just a binary. It was called CNP, Card Not Present, or Card Present. And card-not-present transactions had a higher cost to them, higher transaction processing fees, because they were regarded as less secure because who knows why the card's not present? But, you know, it could have been stolen, and so it's not present; whereas - as opposed to a card-present transaction, where you're doing a card swipe. A physical card swipe means that it's just likely - it's much more likely that it is the user, no, that the user is actually holding the card than some sort of electronic hack happened over the Internet, and that we're doing a card-non-present transaction.

So, for example, all of the SpinRite purchases that I process through our clearinghouse are card-non-present transactions. And so I pay a higher fee to have those transactions performed than Apple is paying for theirs because they've established a higher level of security, verifiable security for their transactions.

Leo: Well, I'm looking forward, we'll find out more Thursday, presumably. I hope Apple will publish more details. They did put out a security document, didn't they, today? Somebody said they did.

Steve: Oh, okay. I've not seen it. And so to sort of explain this, there is something more than I have not found documentation on. There is something called - it's based on the Visa spec called 3-D Secure. And that's this cryptogram. And it's not clear whether it's a hash or whether it's encryption or what. But one of the things we need is we need the existing system to be able to function with something that is not the user's credit card number. And that's where this whole token service provider that has issued blocks of tokens from which it randomly creates these associations to the user's actual credit card number. And then this thing looks like a credit card number and moves through.

So it could be that there is a challenge response mechanism. There could be additional crypto. There's something called a cryptogram that we'll get clarification on, and I'll certainly explain what that is as soon as we know for sure what it is. But in any event, Apple invented some of the glue. They did not invent this underlying technology, which is good news because it means that - and there was another piece of news that I meant to include, that Samsung, like within the last few weeks, Samsung announced something like this. And so they may be riding on the coattails and the fact that now that Apple's sort of got these specs up and running, other people are going to be able to do this, too.

So the good news is, I mean, I was a little worried. If it was only iPhone 6 and people who had compatible terminals, this was going to be a little rough getting off the ground, I thought. In fact, I saw in one of these webinars they were saying that there's a - that Apple says iPhones have a three-year, typical three-year cycle life, that is, the typical iPhone user is updating every three years. So they were seeing and preparing for a three-year curve, adoption curve, which in Visa and MasterCard life is probably acceptable. It's like, okay, yeah, three years, that's fine. Everyone will be up to speed in three years. Everyone, within three years, everyone will have an iPhone 6 or 7 or 8, and those will all be NFC-capable and Apple Pay-capable, and off we go.

So the good news is I think that Android users very soon will end up having essentially a touch pay compatible solution, subject to the technology in the phone offering the security that is needed. And that is scalable. The APIs now have the notion of what is the security environment at the transaction point. And that is reflected throughout the entire transaction.

Leo: So you mentioned at the beginning of the show that perhaps tomorrow there'd be some information about OpenSSL.

Steve: Yes.

Leo: And about 1:00 o'clock, right when we started the show, this was posted on the InfoSec Diary. There is, there has been an OpenBSD patch which perhaps triggers or gives you some information about what that SSL bug is. The bug affects SSL3, is critical. So far there hasn't been any release. We're waiting for that from OpenSSL.org. But this is the OpenBSD patch. And according to InfoSec, Johannes Ullrich writing...

Steve: Yup, he's good.

Leo: This looks like memory corruption/use after free vulnerabilities being patched.

Steve: Ooh, that's not good.

Leo: So give you something, a little homework for tonight, Steve.

Steve: Cool.

Leo: Hey, Steve Gibson, he does this every week, can you believe it? Unbelievable. Every Wednesday, or I'm sorry, Tuesday, around about 1:00 p.m. Pacific, 4:00 p.m. Eastern time, 2000 UTC, we start talking about security. And if you don't hear it, you might miss something important. So make sure you listen every week. You can watch live, listen live. But if you want on-demand versions after the fact, we have a whole panoply of choices. Steve has 16Kb audio at his website, GRC.com. Also very nicely written transcripts that you can peruse, listen along, or just read them by themselves. That's probably the smallest form. Not as much fun as watching; but hey, you know, to each his own. That's GRC.com.

You'll also find SpinRite there, Steve's bread and butter, the world's finest hard drive maintenance and recovery utility. Many freebies. It's also where you should go if you have a question or a comment that you'd like to get to Steve. He has a feedback form there, GRC.com/feedback. Doesn't accept email. You could tweet him. Sometimes we'll use tweets. But that's the best place to go, GRC.com/feedback. You'll also find lots of great freebies, information about SQRL, Vitamin D, carbless diets, everything. It's all there. It's just it's like Dr. Bronner's soap bottle. It's just everything you'd want in one place, GRC.com. You can also get higher quality audio and video versions of the show at our website, TWiT.tv/sn. And it's, you know, it's like one of the oldest podcasts in the world. So - it is. It is.

Steve: Yeah.

Leo: Eight or nine years we've been in the making.

Steve: Yeah. Other podcasts have come and gone, and we're still going strong.

Leo: A lot of them.

Steve: Yeah.

Leo: I kind of, you know, I remember when I was mad at Kevin Rose because he did Diggnation because it competed with TWiT. And, well, that's gone. And then I was mad at John C. Dvorak because he did Cranky Geeks, and it competed with TWiT. And, well, that's gone. So basically we've just outlasted everyone.

Steve: Well, and our listeners have.

Leo: And you have, too, at home, yes.

Steve: Yeah.

Leo: So anyway, you can get it. There's a TWiT app on every platform, including Roku. You could watch live. You can watch after the fact. There's all sorts of ways to do it. I don't think it's too tough. Just Google "Security Now!" and you'll find it. Thanks, Steve. We will be back next week. And we'll probably be doing questions and answers.

Steve: Yup.

Leo: But who knows.

Steve: Oh, I think we probably will.

Leo: Depending on the news.

Steve: Yeah.

Leo: All right. We'll talk to you next week.

Steve: Okay, my friend. Thanks.