QR Codes

Description: After catching up with the week's news, Steve and Leo take a deep dive into the technology of the ever-more-ubiquitous "QR Codes" which are popping up everywhere and are increasingly being used, not only for good, but with malicious intent.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-382.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-382-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with the latest security news, including an update on Microsoft's updates and a look at QR Codes - how they work, what they mean, and what the implications are for security. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 382, recorded December 12, 2012: QR Codes.

It's time for Security Now!, the show that protects you and your loved ones online. It's starting to sound like you're some, you know, Commander Gibson is here from the United States Mounted Police with a tip about online hoaxes. No.

Steve Gibson: Actually, I think GRC is an acronym for, like, the Canadian...

Leo: Is it really? Oh, that's funny.

Steve: ...Mounted Police. Yeah, there's a GRC acronym for Canada that's something about police and horses or something. I'm not sure. I used to get that a lot.

Leo: Well, he's not mounted on a horse. He's not wearing a red wool uniform. He's not wearing a Smokey the Bear hat. But in his Atari T-shirt, he does represent our last best hope for safety online, Mr. Steve Gibson.

Steve: If that's the case, you might as well get under your desk.

Leo: We're in trouble. It's getting worse. It's getting bad out there, Steve.

Steve: Well, we're going to talk this week about something that's been on my list of things to talk about for quite a while, and that's QR codes. They are increasingly ubiquitous. Everyone probably even knows what they are. They're those little square barcodes containing we don't know what. And therein lies the problem. If you Google "QR," as in quick response - that's what QR stands for, quick response - "codes danger" or "malware" or anything like that, you get pages of people beginning to recognize that this is the latest route of exploit. And in fact I was reading one article when I was just getting myself full of backgrounder stuff. This reporter saw a little QR code sticker on the seat in a subway train. And without any, didn't say what it was, it was just there was that little grid, matrix, and it's like, ooh, I really want to scan that.

Leo: Oh, oh, oh.

Steve: Find out what it is.

Leo: Danger, danger.

Steve: So I want to talk about, sort of cover the whole spectrum. One of the cool things that I have learned, and it'll be fun to share, is there's much more obvious structure to it. I mean, obviously it's structured. We can look at it, and our brain is very good about it. But there are a couple details which, after I tell you about them, you will never not be able to see them, and you do not see them now. And as with anything, if you don't know anything about something, like imagine you couldn't read, and you looked at a page, it would just look like nothing. I mean, you wouldn't see any meaning there. But once you understand the shape of alphabet characters, then you look at it, and it's like, oh. Suddenly you will lock onto it. And similarly, by explaining the structure of the QR code, just because it's fun, everyone will be changed.

Leo: Wow.

Steve: They will then - you'll be familiar with it. You'll look at it, and it'll mean some - people won't be able to read it, but you'll understand much more about it than you do now. And so that's sort of one part of the fun. And then, of course, we'll talk about what's happening because it is becoming a problem. There is this - it's the typical tension that exists between convenience and security. And it's a little bit related to the problem that we discussed over on the near field communications side with NFC tags because, although these won't jump out and bite you the way radio can, we can't read them. Our machines read them. And often, by the time they do, it's too late. So there are some things that can be done to protect ourselves. So we'll have fun catching up on this week's news and then looking into these wacky little square tags. I mean, I've got one, I just looked, here on the back of my Starbucks espresso...

Leo: They're everywhere. You can't get away from them.

Steve: Oh, yeah, exactly. They're cool. And they were invented by a car company in Japan, like 18 years ago. So they've been around for a long time. But, boy, have they, well, now, and of course the enabling technology was cameras, camera phones.

Leo: Right, camera phones, yeah.

Steve: Because the famous comment is "The best camera is the one you have with you." And now with cameras and phones - and has anyone figured out why Philippe Kahn thinks he invented that, by the way?

Leo: He did, by the way.

Steve: That thing just sticks in my brain.

Leo: He did.

Steve: How did he invent the camera phone?

Leo: He was, many years ago, he had his baby. His baby was born probably 10, 15 years ago. I actually talked to Philippe about this. And he said he was taking pictures, thought, boy, really be nice if there was some way to quickly send this to family and friends

Steve: Ah.

Leo: And he invented - Philippe, now, Steve and I are of a vintage where we think of Philippe Kahn as the founder of Borland International, which was a great company which made a Pascal called Turbo Pascal that kind of transformed high-level programming.

Steve: Oh, and he really upset Bill Gates by destroying Microsoft's language model.

Leo: Yeah.

Steve: Remember Microsoft was charging $800 for a C compiler. And Philippe said...

Leo: And what was - Turbo Pascal was, like, 50 bucks; right?

Steve: It was $49. And it just blew the socks off the...

Leo: Oh, it was super fast.

Steve: And Bill never forgave Philippe. It just completely uprooted his whole plan.

Leo: He's also a musician, as I think people know. Plays, I think, the saxophone.

Steve: Yup, and a pilot.

Leo: And a pilot. And a very, these days, avid sailor. So he invented this technology called LightSurf in 1998. So this gives you an idea of how he could be there. In fact, this is an image that he took of his daughter's birth in 1997. I remember we interviewed him on I think it was probably The Screen Savers about this. And this was one of the first images taken and sent by camera phone.

Steve: Very cool.

Leo: What's neat is LightSurf Technologies, I think, had its bit of code in every single camera phone for many, many years.

Steve: No kidding. Wow.

Leo: Everybody licensed it from him. VeriSign bought them in 2005, according to Wikipedia. He then started something called Fullpower. Philippe is actually, you know what, we've got to get him on the show.

Steve: He's a great guy. He and I used to have breakfast every year at COMDEX. That's when I was doing the column for InfoWorld, and so there was always a "have breakfast with the reporter and tell him all the wonderful new things your company is doing."

Leo: So get ready for this. So, okay. Starfish. Remember it was Turbo Pascal.

Steve: Remember Starfish, yup.

Leo: Then Starfish, which I think was an early synchronization. I mean, this guy's been always thinking ahead and has been disruptive every time. He created TrueSync, which was the first over-the-air synchronization system, in 1994, sold it to Motorola for 325 million in '98. Now there's LightSurf, which he sold to VeriSign for an unknown amount of money.

But I would say he's probably on his second billion now because his new company, his current company, created something called MotionX. And if you use a Nike+, if

you use a FuelBand from Nike, if you use the UP band from Jawbone, if you use a Fitbit, they all contain MotionX, which is the hardware that senses your motion and ties it to GPS. And also he has software, the best GPS software on the iPhone and iPad, which I recommend all the time, which is MotionX GPS. That's Philippe Kahn. So this guy, he's got to be on his second billion by now. I mean, this guy is brilliant. And it couldn't happen to a nicer guy.

Steve: No, it couldn't. And it's nice also to see that he wasn't just a one-shot deal. I mean, he...

Leo: Oh, boy. He should get more credit, frankly. I don't know why he doesn't get more credit. He's had four companies, each of whom have disrupted, transformed technology. He, by the way, is an avid sailor. The reason I know that, we had a company sail a few years ago on a beautiful boat, Humphrey Bogart's old yacht.

Steve: Oh, he's a competitive sailor, too.

Leo: Yeah. So this team goes by, vroom, and they're all hanging out on the side, and they're wearing rubber suits. And the skipper says, "Yeah, there goes Philippe Kahn's boat." And he says, "You know, he's really serious. He's out here every day with his paid team of sailors. He's very serious." And I'm not talking America's Cup sailing. It's like really smaller boats and hardcore like you are. This is, like, they were hanging out. Their heads were hitting the water. And they're hanging out, and the boat is at least 30 degrees. Unbelievable.

Steve: Well, and Philippe was always a big guy. So if he leans off the side of the boat, he's going to be some serious ballast.

Leo: I didn't see Philippe, but anyway. Go ahead, Steve.

Steve: Your team nudged me gently last week, saying, you know, Steve, we need some content from you for the Security Now! Christmas Special podcast.

Leo: Yeah, 'cause I ain't working on Boxing Day.

Steve: Uh-huh. And I thought that I had something that I thought would be really fun.

Leo: Cool.

Steve: Back in 1990, 22 years ago, SpinWhite - SpinWhite.

Leo: SpinWhite, ha ha ha ha ha.

Steve: SpinRite was relatively new. We had a relationship with the largest software distributor in the world. This is before the Internet, and so software was sold across the counter in boxes by Egghead and Fry's and...

Leo: Yeah. I wonder what happened to Ingram and all of those big developers, distributors.

Steve: Well, and my relationship was with the largest distributor, called SofSel.

Leo: SofSel, yeah.

Steve: And they bought Ingram Micro and - or, wait, Micro-D. Anyway, there was a period of coalescing. Anyway, SofSel did a multicity tour that they called "The SofSel SofTeach," where they took some of their selected vendors. And, I mean, it wasn't free. We paid for the privilege. But it was a privilege to go to, essentially, little mini tech conferences and explain our products to all of the retailers in the area. And I don't know what moved us to do this. But we taped them. And...

Leo: Oh.

Steve: And we taped me explaining how and why hard drives die and how SpinRite interacts with that. And I had all of the tapes from Chicago's, it was a Saturday and Sunday. I think they were labeled Sunday No. 7 and Sunday No. 9. And I remember I was exhausted by the end of the day. Anyone who sees them will know why I was exhausted because I'm quite animated. Anyway, I found the tapes. The camcorder that recorded them would no longer play them because I had it, too. It was Hi8 format, back predigital. This is truly 1990. I had a Hi8 deck. It refused to load the tape. But good old Amazon came to the rescue. I found a used Hi8 camcorder for $199.

Leo: Wow. I want to borrow that from you because I have a bunch of Hi8 tapes.

Steve: I got it and ran it through my analog-to-DV converter, trimmed it, fixed it up, compressed it, and shot it up to you guys Monday.

Leo: That's awesome, Steve.

Steve: So what I want to do is maybe get with you Saturday or Sunday after your Tech Guy, and we'll just do a little 10-minute intro.

Leo: Good.

Steve: And then we've got something very fun.

Leo: We could do it after SpinRite this week or next week, too, would be another thing. Or before SpinRite. That might be better because I think we're doing some Giz Wiz stuff on Saturday. So...

Steve: Okay, perfect.

Leo: Yeah, we'll just do it during the regular.

Steve: And the only thing I want to say is that it really - normally I really try to focus, as our listeners know, on conveying information that works just with audio because I recognize that we're still hugely oriented toward people who are, just as you were saying with the Audible technology, who are listening to the podcast. Unfortunately, I have a blackboard, and I am using my body a lot, like when I explain about the lifecycle of a screw which is trying to hold the stepper motor in place, or explaining how tracks drift. My arms are flying around. So this particular holiday special really needs to be seen to be appreciated. You might listen to it first and then, like, okay, I've got to see what he was doing because there's also a lot of laughter from the crowd, and you may not know why they were laughing at me if you weren't seeing it.

Leo: That's great. Oh, I'm so thrilled, Steve, thank you.

Steve: So I think it'll be fun. And, my god, my hair was black. Oh. I had as much as I wanted of it. Just incredible.

Leo: Ah, yes.

Steve: So it was a little blast from the past, quite fun. Oh. And no matter how techie anyone is, how much they think they know about hard drives, I'd forgotten some of the detail that I went into that explains things that I'll bet you there isn't a single person listening to this who doesn't actually learn something that they didn't know. So it's got something there, too, in addition to being me making, well, not an idiot of myself, but...

Leo: I can't wait. I can't wait. This is exciting. What a great idea, Steve.

Steve: So we are just past - this is Wednesday - the second Tuesday of the month, which of course we know is Microsoft's day to drop their monthly packet of updates on us for Christmas. They had seven updates fixing 10 security holes, most rated critical, in Windows and Word and Exchange Server. And there was even a crazy one. There was a file system folder naming parsing vulnerability that affected XP, Vista, and Windows 7 so

that, if you browsed in Windows Explorer to a specially named folder, you could have your machine taken over. So you don't see those often.

And there was something interesting that fits my theory of new is not always better, which was a critical patch for IE9 and 10, but which is not a problem in any earlier versions of IE. So Microsoft added some things. Some of the new stuff they added in IE9, which carried over into 10, has a problem, and so they're fixing that. I mean, that's just, as we've said many times, code that has been pounded on more has been more pounded on. And that's a good thing.

At the same time, Adobe is updating both Flash and Air, their essentially synchronized technologies for rendering Flash content either in a browser or in a standalone app. And you can check - it's across the board. Windows, Mac, Linux, and Android all need to be updated, although the Flash players in IE10 and in Chrome should be auto-updating. So those you don't need to worry about. And typically Flash is notifying you, or Firefox is telling you you've got an obsolete version. And so in general we're getting this to happen more often. But Adobe.com/software/flash/about will verify, if you go there, that the Flash Player version you're using in your browser is current. And then for Air it's Adobe.com/air will do the same thing.

And I noted something, and this is the "If it can be done, it will be done" department, which is we now have a relatively sophisticated botnet whose command-and-control servers cannot be found because they're using TOR. So it turns out that the TOR Project implemented reverse anonymity. They implemented something called TOR Hidden Services, which provides for servers the same kind of protection that TOR has traditionally provided for clients. That is, normally it's a client who uses the TOR system to hide themselves from their activities out on the public Internet.

And at some point it occurred to the people who now manage the TOR network that, hey, how about people who want to offer services that they want to hide? So the reverse process. So that exists now in the TOR system. And I made a note to myself that, ooh, that sounds like a fun podcast, to look into how it does that, because we understand how the client gets hidden, but it's a whole different problem to hide a service, yet still be able to find it. I mean, when you think about it.

So there is a botnet called Skynet which has been found. It's about 15MB of stuff. But hidden among the files that lead people to believe it's not bad are a conventional Zeus bot. And even the fact that we're now using phrases like "conventional Zeus bot" is a little troublesome. Oh, yeah, just you run with the conventional Zeus bot. There's also a TOR client for Windows in there. And then the CGMiner bitcoin mining tool with the OpenCL.dll which the CGMiner uses to interact with the CPU and GPU for high-performance bitcoin hashing. So apparently this is a botnet which, among other things, is a distributed bitcoin mining technology so that people are stealing cycles from unwitting users' CPUs and GPUs to use them in the background for minting bitcoins, all being hidden and unable to find thanks to this new service, TOR Hidden Services. So I thought that was interesting.

And we talked about, several times, the iOS universal, I'm sorry, Unique Device ID, the UDID, which it became clear, first to Apple and then to the industry, was being abused. It was a per-device tag which all iOS apps were able to get. And they just used it. Even though Apple said you really shouldn't, the apps said, yeah, but there it is, and you're giving it to me. So as we mentioned before, back in March, Apple began, after saying don't use it, they began refusing submissions to the store for apps that were using it, and they made it very clear it was going to go away at the end of iOS 5.

Well, it did. And so app authors had plenty of notice and time to stop using it. And the problem with it was not only that it did tag to your device, but what happened was all the apps running in a given device received that same UDID. It was the device ID. So there was a problem with apps and tracking systems comparing notes and being able to cross-associate apps to a single device. So in iOS 6, Apple formally removed the API, the Application Programming Interface that allowed access by the app running on iOS to that device's ID. They replaced it with something called the IDFA, which stands for Identifier For Advertisers. So very explicit, IDFA, the Identifier For Advertisers. And what's different about this is that each app now receives a unique per-device IDFA, but every app running in the same device gets a different one, and there is no way to associate them. So the apps still get tracking information, inasmuch as it's a unique ID for the device. But the problem of apps recognizing that they're on the same device from the tag is solved because each app gets a unique one, and there's no way to interconnect them.

Now, this can be turned off, for anyone who wants to. But once again you've got to dig down a little bit. And it's not obvious where it is. I found it after realizing that it was there. And in fact it was some listener of ours who kindly tweeted me, "Hey, Steve, look what I found," and he sent me two pictures. It's like, whoa, where is that? And so I found it. You go to, of course, in the Settings icon in any iOS device, phone or pad, under Settings in the General tab or the General category. Then you've got to go to About, which has a whole bunch of stuff. Scroll all the way down to About, and near the bottom you'll see a new word: "Advertising." It's like, huh?

So if you select Advertising, then that takes you to an otherwise blank page that just has one thing on it. And I was looking at it, thinking, well, Apple, normally you're kind of helpful. You put some text in there to say this is what this does if it's on or if it's off or what it means and so forth. There's nothing there. All it says is "Limit ad tracking," and it's normally off by default. Meaning not that ad tracking is off, but the limiting of the ad tracking is off. So in other words, you want to turn it on if you want ad tracking to be limited.

And so what this cleanly and globally does is it denies this IDFA new feature to the apps running on your device for the purpose of tracking. And so there is no longer any way for them to lock onto, for apps in your device to lock onto your device directly. And I thought that was a cool feature, so I wanted everyone to know about that. Otherwise it's not easy to find.

Also in the news this week, a group calling themselves "Team Ghostshell" did what they sort of called "end of year housecleaning." They had all these pesky accounts and record details that they had accumulated from a breathtaking range of networks - NASA, the European Space Agency, Japan's Aerospace Exploration Agency, Interpol, the Pentagon, the Federal Reserve in the U.S., the FBI, a defense contractor known as L-3 Communications, and more. There's a Pastebin link which I've got in my notes here. You can probably find it if you just Google "Team Ghostshell WhiteFox." They are calling their project Project WhiteFox. The Pastebin page is rather stunning. It scrolls on and on and on with an enumeration of the networks that they have penetrated and pulled account and record details from, totaling 1.6 million accounts and records, which they have then sprayed all over the Internet, and they've got links to them all with mirror sites and backups and things. So it's just becoming sort of sad that our networks are as porous as they are.

And I saw a little bit of news that I wouldn't have normally brought up, but because it's something that we talked about years ago, Leo - and I was stunned this thing is still going on. Remember, I think she was 24 at the time, there was a young mom with a couple kids, her name was Jammie Thomas-Rasset. And she was that woman who

somehow got Kazaa - remember Kazaa? This is how old this is.

Leo: Oh, yeah, yeah, yeah, I remember.

Steve: She got Kazaa on her laptop or her computer, whatever, and there were 24 songs that were there that she didn't download, she wasn't intending to pirate, I mean, she was, like, completely innocent. And the RIAA...

Leo: Well, I'm not sure she was completely innocent.

Steve: Okay. Well, they stomped on her and...

Leo: Yeah. There's no question about that. But we don't know if she was...

Steve: ...[indiscernible] of $222,000 is what they want as compensation for her having these 24 songs. The point is that the Supreme Court of the United States has been asked now to review the jury's decision that she needs to pay the RIAA $222,000.

Leo: I wouldn't expect any change, but go ahead.

Steve: No, no. And the Supreme Court has...

Leo: Not from them.

Steve: Yeah, and the Supreme - not the current court.

Leo: No.

Steve: And the current court has declined to hear two other RIAA-related filesharing cases in the past which they've been asked to review. And the petition that her attorneys have provided to the court seems a little odd. It takes the argument that the damages are unconstitutional.

Leo: Yeah. Well, no, I think that's the only argument they can make with the Supreme Court.

Steve: Yeah, okay.

Leo: And I don't think she's - I think that they stopped contesting her guilt. I think

it's more like the damages are out of control here, Your Honor.

Steve: Yeah, how is this fair?

Leo: But I don't suspect that'll be changed.

Steve: No. So I saw this, and again sort of smiled. I put the headline as "IPv6... Anyone???"

Leo: Hello?

Steve: There was sort of a helper body, for lack of a better term, set up in the U.K. The BBC reported this. The body was set up to get the U.K. moving over to the 'Net's new addressing system. That group has been shut down in protest at the complete indifference to its work by the U.K. The group was called 6UK, which was set up to advise ISPs and firms about the move from v4 of our IP addressing scheme over to v6. But it's been wound down after its board realized that its work was futile without official backing which was never forthcoming. And the director, a Mr. Sheldrake, was quoted as saying, well, the fact that no government website sits on an IPv6 address might be part of the problem. So the U.K. doesn't have what we have had over here in the U.S., which is our government famously mandating that people that work with them must support IPv6. And so that's one way to get this to move, I suppose.

Under the category of "Looney Tunes," we have the continuing adventures of John McAfee, who, since we last spoke of him last week, has been arrested. You'll remember that last week some pictures that were taken of him by Vice magazine - who are now being referred to as his "ex-friends" at Vice magazine - they contained metadata which had the GPS coordinates of his location at the time the photo was taken. He's now accusing them of orchestrating his arrest so that they could be there to report on it. Who knows. I'm not taking a position on that one way or the other. I don't know. So he was arrested. And the first judgment from the judge was that he had entered Guatemala illegally and was going to be expelled as a consequence back to Belize, here John was concerned that he would be "found having accidentally hung himself," unquote.

Leo: Oh, geez.

Steve: But he did actually say that. But it turns out that John suddenly started having small heart attacks.

Leo: Oh, wow.

Steve: Or so we are led to believe. There's been some question as to whether they may have been asymptomatic heart attacks. But the claim was that it was over the stress of all of this. The heart attacks which may have occurred were at least enough to get him hospitalized, which were also long enough to get his attorney to find another judge to

overturn the original declaration because it turns out that entering Guatemala as John did is not illegal. And so he's now been released. He's apparently also feeling a lot better, and he's on his way home to the U.S. So his blog is a continuing adventure, for anyone who has some time to kill.

Leo: He says he's flying to Miami. So if you're in Miami, watch out, here he comes. It's always something.

Steve: And a quick little update on what I am calling "The Quiet Canine Project." I just didn't like Hush Puppy. I mean, I loved Hush Puppy, but it was just too generic.

Leo: It's like, what is it, yeah.

Steve: Yeah. And so the Quiet Canine sort of talks...

Leo: That's good. It says what it is.

Steve: I like it. It talks about the benefits.

Leo: Yeah.

Steve: Yeah. Many people are building them. I wanted to let people know that, even internationally, I've heard from people in the U.K. who ordered the so-called "LaunchPad," which is the little development platform that I'm using, for $4.30, with free shipping. It comes the next day, or maybe in two days, even if you're in the U.K. Someone in Australia got it overnight.

Leo: Wow.

Steve: So even internationally it's still $4.30, and they pay the shipping. So a super bargain. The Portable Sound Blaster Group, as it is still called over on Google, is going strong. I've got source code posted. People are assembling the source. They've got the chips programmed and generating square waves to drive the amplifier. They're building the amplifier. Some people are using Arduinos to generate the signal, some just using CMOS. So there's a lot happening.

I've heard from many people who have dog barking problems, either their own dogs, they would like a training device that is more humane than whatever they're using currently, or maybe more effective, and people with neighbors. And I'm now working on the pages to get them up on GRC so that there'll be a permanent place to have the documentation for the project. And once I get a sense for the size of the beta test group, I plan to build a bunch of these myself and provide them so that I can get some feedback about how they work. So we're moving forward on that.

Leo: You mentioned last week that you were not crazy about Google Groups, that it changed a bit. Do you know that Google+ has now added Communities? Which is probably why Google Groups was kind of left on the side of the road.

Steve: Ah. And does it give us - I looked at Communities, but I couldn't see any sort of structure to it. I guess I like being able to have threads and subjects and threaded discussions and...

Leo: Yeah, it's got all of that. So when you create a Community - you can have multiple moderators, by the way, which I'm sure will be nice for you. We have one for TWiT and The Tech Guy. It's called This Is TWiT and Team Tech Guy, if you want to look at them. And you can create - it's more like a forum. You can create subheadings. So, like, I have on Team Tech Guy "Help Me." And so you can create something within that category. But when you post, it also can be post-wide. Threading is only by question; right? So you say something, and then people respond to that; say something, people respond to that. Which is kind of like a forum, basically.

Steve: Yeah. Okay, good.

Leo: So I think they've done a good job with Communities. And anyway, I know it's too late for this one, but just for future reference.

Steve: I'll go check it out, yeah. And in totally randomness, I just got a note from 23andMe.com, noting that their price has dropped to $99. So, yes. They're dropping the price down as quantity increases. They recognize that you and I paid a lot more than that. And to make us feel okay...

Leo: How are they going to do that?

Steve: ...they're explaining that the reason they're doing this is, to get the benefit of this for everyone, they need to increase the sample size. They need the largest database of DNA they can get. And I will say that the people I've turned on to it have just had a ball. You get a little test tube that you drool into for a few minutes, and then you snap the cap closed, which releases a suspension agent. You shake it up and stick it in a postage-paid box, and it goes away, and a few weeks later you have access to a really compelling, interesting breakdown of what's known about your DNA based on all of the feedback that they've received. So anyway, it's only $99 now, so making it increasingly affordable.

Leo: Cool.

Steve: Speaking of increasingly affordable.

Leo: Yes.

Steve: I have a little SpinRite story from a Dan Spengeman, I think, S-p-e-n-g-e-m-a-n, Spengeman. He's in Shrewsbury, New Jersey. I'm sure he knows who he is. And the subject line he used caught my attention, said "SpinRite Equals Cookies." And I was like, what?

Leo: Uh-oh.

Steve: "Hey, Steve and Leo." You're Leo. "Longtime listener, LastPass user," and now we have the increasingly common term "blah blah blah" to fill in for all the other how much they love us.

Leo: Ditto. Ditto, ditto, ditto, yeah.

Steve: Yeah. "I work for an IT consulting firm in Central New Jersey, dealing almost exclusively with business computers on a contractual basis. We are not a break/fix shop, but as is sometimes the case, we agreed to take in a laptop that was outside of our agreement contract. The laptop was for a good client that had several critical crucial pictures, the only copy, I might add, of damage left by Hurricane Sandy to their home that needed" - so you can imagine why they absolutely had to have them - "that needed to be sent to the appropriate insurance company. If these could not be recovered, there'd be a lot more tears shed over this horrific natural disaster.

"Upon receiving the personal laptop, we attempted to boot, only to have the Vista progress bar continually scroll across without ever loading the Windows environment. Having seen this before, I promptly loaded a copy of SpinRite and ran a Level 2 scan. It came across one unrecoverable sector, recovered others, and then got stuck at about 4.22 percent. I waited a while, then rebooted the laptop and left the room for a bit. I came back to see the computer running through a Windows Chkdsk, encountering several orphan files, and finding some unrecoverable data. However, it continued to run, rebooted on its own, ran another Chkdsk on its own; and, much to my surprise and delight, the laptop is back up and running perfectly."

Leo: Wow.

Steve: "I ran another Chkdsk as a precaution, which came back totally clean. But I plan on rerunning SpinRite to ensure it gets the GRC seal of approval. I called the client, and they were elated, so much so that we're getting some homemade baked goods as a token of their appreciation." Oh, that's the cookies. "So when I get fat off of sugar-coated goodies, I have you to thank and blame." And he has a little tongue-sticking-out :P. He says, "Thanks for a great product and a great podcast. Dan."

Leo: Awesome. Awesome, awesome.

Steve: So thank you, Dan.

Leo: QR codes, Mr. G.

Steve: So, okay. I'm very impressed with the design of these. Not everything I can say that of. And I'm impressed that this was done 18 years ago. I mean, that's a long time ago to have the kind of foresight that the designers of these codes had. They were originally done, they were created for rapid tracking of automotive parts during automobile assembly in Japan. And they're just - they were done right. So if people listening have access to one, if there's one around you, maybe your phone, like for example Starbucks, the Starbucks app for scanning their loyalty program uses a QR code barcode, so you could look at that. If there's one around that you can look at when I'm discussing them, you'll have a better time visualizing what I'm talking about. If not, you can remember some of this stuff and then check one out when you see it.

The codes are designed - they're square. They always have an island, what they refer to as a "quiet zone" around them. So they're designed to stand by themselves. If they're a sticker, then they're in the middle of a sticker that's got some white margin so that there's a so-called "quiet zone." The cool thing about them is that they are orientation neutral so that you can - you don't have to be exactly face on. You don't have to have them exactly upright. They provide all of the information necessary from their image to allow software to essentially spin them around and orient them and flatten them, even if you take a picture of them at an angle.

So the most prominent features on the QR code are three large targets that are out at the very - at three out of the four far corners of the QR code. And of course it's not two targets or four because you want to get a quick rotational orientation. So they chose three, to use three rather than one because three gives you an immediate sense of size and angular orientation at the same time. There's a smaller target in the remaining fourth corner. In the way QR codes are normally oriented, if the three big targets are both at the top and then also over on the left, then it's the lower right corner that is missing the big one, which is an instant cue for rotational orientation. There is a target there, however. It's four bits in to the left and four bits up from the bottom is the smaller target.

The coolest feature that you would not notice until you know about it, and you're about to know about it, and you can check for it on every QR code you ever see, and it will always be there, and you'll know something no one else knows because it's not obvious, but it's just, I love this, is that linking the inner corners of the three big targets is a clock track, essentially. If you look between the upper left and the upper right inner corners, you will see on/off, on/off, on/off, on/off, on/off, on/off. That is, it is a 50 percent duty cycle. It always looks exactly like that. And the same thing from the upper left to the lower left target has the same on/off, on/off, on/off, on/off. So that provides the referenced clocking information for the size and additional position orientation of the code.

The actual code itself has a format number and a version number which are stored in the bits immediately surrounding the three large targets. So sort of the closest surrounding bits of the three targets contains - which is also obviously always easy to find and at a known location, given that you know where the targets are - tells the software what the version number and format of this particular QR code is. The densities range from one to 40, so you have a huge range of storage. And you can store an amazing number of characters, on the order of I think it's 2,000 characters. They're not quite eight bits

because of the way they encode them. They use a strange, like a 45-character set. So it's all uppercase and then a bunch of special characters. Enough, though, that you are able to encode non-case-sensitive URLs, and then there are - there's a binary mode where you do store full eight-bit bytes.

So the actual encoding starts at the lower right corner, in the very far corner. And the codes are stored in blocks of 2x4. The reason they're done in 2x4 is that they would like them to be as square as possible so that the bits in a byte occupy a smaller surface area. A lot of attention has been given to error correction because it's recognized that, for these to be robust in the environment, they might get torn, they might have a black mark through them, or they could even have a chunk missing. And in fact the error correction technology can also have - it can range, like there's low, medium, and then the third one they call "quartile," or high. So low, medium, quartile, and high.

At the highest level of error correction, fully two thirds of the surface area of the QR code is involved with error correction. But that means that, in return for committing to having that much redundancy, in return for that, you're able to lose as much as a third of the QR code tag. And, in fact, Wikipedia has a sample that I ran across where they've written the word "Wikipedia" right across the center body of the QR code, yet you can still read Wikipedia's URL, despite the fact that you just obliterated a chunk of the QR code, because error correction makes up for what's missing.

And this is the same Reed-Solomon style error correction code that hard drives use, and for very much the same reason, in the same way that you might have a little defect on the surface that causes you to miss a physical region on the drive, you similarly could have a physical defect in the QR code tag printing or, as I said, a smudge or something dropped on top of it, or a part of it got torn off. So this technology allows for robust recovery of any missing data. As a consequence, they don't want to string the bits out in a long string because that doesn't model as well the kind of defects that they expect that the tag might suffer. Well, so they'd like them to be like a 3x3 square, but that doesn't work for eight-bit bytes. So they compromised to a 2x4 tile.

So essentially there are 2x4 tiles that run along one edge and then turn - it's not in a raster scan, where it jumps back to where it began and moves over. Instead it turns around and heads back the way it came, two bits over. And then it turns around again and heads back. And if you find yourself at all curious about this, the way they handled the details are a little bizarre because the tiles will interact with these targets and these other fixed aspects of the code which represent barriers. Sometimes they skip over them, or sometimes they slide around them. Yet they've come up with a disciplined, clear way of handling every instance, which ends up in some cases creating these weird sort of pentomino-shaped things that all fit together. But the spec handles it, and the software knows how to deal with it.

Now, one problem that they encountered - again, I'm impressed by how clever they were - was that it would be possible, when you think about it, for the data to emulate the fixed features of the QR code. And so you would call that "in-band signaling." There would be an in-band signaling problem which is the information theory term for the problem that you're trying to have formatting data that tells us about, not the actual content, what you want to have separate from the content, yet you're just printing this all with ink. So how do we know what's data and what's formatting?

So the designers of this said, okay, well, we're going to lay out this information, but what if the information we lay out looks like a target, for example? For one thing, what if there's a big white space in the middle? That would be a problem because all of these technologies, even back to hard disk drives, the hard disk drives are what's known as

self-clocking technologies, meaning that rather than wasting space by having a clock signal or clock track in addition to the data, they arrange for the data to be self-clocking, for the data to provide its own timing information.

The way you do that is you deliberately make sure there are not, for example, in the case of a hard drive, a whole bunch of zeroes. If you had a whole bunch of zeroes where zeroes meant nothing is happening, then the problem is, when something finally does happen, you need to know exactly how many things didn't happen. And that can be dicey if the hard drive is not spinning at a constant pace. Similarly, here, if there was, for example, some stretching or a wrinkle in the QR code, that could cause a local change in the frequency of the pattern that's laid out in the visual field. So you wouldn't want to have a big, solid, black blob because you might have a problem knowing exactly how long it was or what size it was, and that's crucial, especially on a high-density code taken from far away, crucial for knowing how many bits are there.

So what these guys do is one of the formatting controls actually lays down a mask which is used to XOR the entire dataset. And we've talked about XORing before. We discuss it often in crypto. We know that, when you XOR something, you are selectively inverting bits. And if you re-XOR it and selectively reinvert the same bits, you get back to the original thing you had. So the XOR process is perfect because it's very simple to do. And depending upon the nature of the data which the QR code contains, they have a library of, I think it's eight different XOR patterns which are mathematically derived from the X and Y coordinate of a little 8x8 patch.

So it's an 8x8 pattern that, for example, one of them is a checkerboard. One of them is vertical stripes. Another one is horizontal stripes, separated by two areas of white, and so forth. So what the QR encoder does is first it lays out a non-masked, non-XORed, that is, QR code. And then it's got criteria for things that would be a problem, like long runs of ones or zeroes, or just a coincidental formation of a blob somewhere because of the way the zig-zagging pattern happened to have the bytes fall.

So what it does is it then applies each one of the eight different masks to get eight different candidate QR codes. And then, using the algorithms for its criteria for judging the quality of them, it picks the one which it likes the best. Which is which best meets whatever criteria it has. Basically, there aren't any confusing structures. It's broken all the data up, there are no blobs, and so forth. So it ends up choosing one of those masks, and that's part of the encoding information which goes into it.

And then, finally, starting as I said at the lower right corner, there is an encoding type which is four bits, then a length which is eight bits, followed by that much data. And at the end of that, there can be another encoding and another length and more data. So you're able to have multiple formats of data in a single QR code. You're about to have variable density. You're able to have a variable amount of error correction. And understand that error correction means redundancy. So the more error correction you have, the larger percentage of the whole area is not data. So there's a natural tension between how much area are we going to commit for error correction versus how much data are we trying to store.

However, all of the QR tags are square, and the original designers recognized that there might be a situation where they wanted a lot of data, but the square aspect ratio was a problem. So rather than confuse the code by allowing non-square QR codes, they instead created a way of appending codes. So it's possible, for example, to have eight little ones in a row to essentially give you a very rectangular, stretched-out, single QR code. A properly equipped reader that scans all eight of the little guys will recognize essentially where they are and how to concatenate the data in the individual codes into one single

large one.

So anyway, it's an old, yet well-designed technology. It's not clear how you would improve it, even today, 18 years after it was created. It was patented at the time. Of course by now the patents have expired. But it was formally placed in - it was released to the public. The patents were expressly there to keep control of it from a standardization standpoint. Even the word "QR Code" is trademarked. But anyone is free to use it. So this was done in a very open intellectual property fashion, just like you would expect. And of course that's one of the reasons that it has been such a success, that and the fact that because, very much like the Internet, because it was really done right. It was designed with an amazing amount of foresight. It's a technology that we can use today for quickly transmitting data. And of course that's also the problem because it's something that our smartphones can read and interpret that looks just like dust to us, just like, okay, I mean, we all know what they are now.

As I was saying at the beginning of the show, I was looking around. I bought some Starbucks espresso roast, and there's a little QR code on the bag that I don't know if my smartphone is supposed to read it, but it's got all of the characteristics that I mentioned. I see the little alternating pattern between the inner corners of the three big targets and the little target four bits up and four bits over from the lower right and so forth. I mean, now it's very possible to sit down, if I needed to, and figure out what this thing says.

Leo: The real question is why anybody would take the time to take a picture of it when you could just, I mean, probably just sends you to Starbucks.com; right?

Steve: Yeah, I mean, it might be used for their own manufacturing process. It might be the...

Leo: Is it better than a barcode from that point of view? It's more data?

Steve: Yeah. It is more data. And that's really the only difference. Essentially you can think of it as a two-dimensional barcode. And barcodes, one-dimensional barcodes that we still see on the backs of books and things are stripes. And so they're meant to be read by a one-dimensional reader. Normally that's a laser beam spun by a mirror. And we always see that at the checkout stand at Barnes & Noble or in...

Leo: Grocery stores.

Steve: ...supermarkets, yeah. So there's a beam that sort of opportunistically scans across it. And there's certainly a protocol there also. But it's only able to represent something like a UPC, like a Universal Product Code, and nothing much more complex.

Leo: As I remembered, there are a number of barcode protocols.

Steve: Yes, yes, exactly. Here these things can be used to represent anything you want. And what has happened, the reason, if you were to Google "QR Code Danger," for example, I mean, it is all over the place. The bad guys have figured out that people like

these, and people are taking pictures of them. I mean, and the advertisers are leveraging it. You'll see a poster with one, and it's like, here, take a picture of this, go to our website. Well, what's happening now is that popular QR codes are being covered up with stickers to take people places other than where the poster which you would tend to trust would take you. And as I mentioned before, I read some articles where people just, like, see a barcode in a bathroom, a QR code...

Leo: For a good time...

Steve: It's like - I know, for a good time.

Leo: QR this.

Steve: Exactly.

Leo: But, now, that's an implementation issue because the camera can take a picture and say, hey, I'm about to send you to...

Steve: Yes.

Leo: Do they automatically go to that site currently?

Steve: There are two problems. One is that we're depending upon the implementation to be, well, the implementation of the interpreter to be correct. Notice that the implementation of the Flash interpreter has been a problem since it was born. As a consequence, we have all these problems called "buffer overflows." Well, it's entirely possible to have a buffer overflow in the QR code interpreter. I don't know that any exist. I hope that one never does because what that would do is essentially override the best intentions of the software that was interpreting the QR code. In the same way, for example, that a buffer overflow down in our TCP/IP stack used to cause problems before the packet even got to the firewall.

So even a firewall wouldn't protect us if just the act of receiving the packet allowed a bad guy to take over the operating system. If the act of viewing, it would be literally the act of seeing this particular pattern could cause data to be overwritten on the stack and take over your smartphone. So there's that problem. And then, exactly as you say, Leo, the question is, what does the phone do? And what is unfortunately the case now is that there are hundreds of QR code readers, and they're also being built-in natively into the platforms. And we have the tension between convenience and security because there's something a little extra sexy about just taking a picture of it and, whoop, you're automatically there at the website.

This has been enough of a problem that, for example, Symantec has a free product now in beta which I would recommend to anybody who thinks QR codes are interesting. It's called "Norton Snap." And it's available for free for iOS and Android. And it does just what you said, Leo, but one more thing. It also does reputation checking. So you take a picture of a QR code that you believe leads you to a website. And Norton Snap will

intercept it, check the reputation of that site, give you a big okay green ball or warn you off with a red, who knows what, skull and crossbones. And it also de-obfuscates the QR code.

I mean, that's the least you would want, I think, is to have the QR code interpreted for you by the QR code scanner in your phone, be able to just eyeball the domain. I mean, if it's like weregoingtogetyou.cn or .ru or something, it's like, okay, I don't think I'm going to click that link. So it'd be really nice if it was a two-step process. The code was converted to a URL. You then had to manually click, like acknowledge that this is where you want to go, giving you a chance to see what's there. At the moment, many of these don't. They like the whole magic carpet ride, just wave your phone and off you go.

It's important for our listeners to know, not surprisingly, I mean, once again, if it can be done, it will. The bad guys have figured out that this is a way to get people to execute URLs under a false pretext, like by putting stickers over existing codes and replacing them. You might think, oh, that's probably - there's a new code from what they had before. No, someone with malicious intent came by and covered up the real one.

Leo: Another use for QR codes is both LastPass and Google use a QR code to add to Google Authenticator. I guess it's probably Google Authenticator that's doing it, and LastPass is supporting it. But when you want to set up Google Authenticator on your smartphone to use with LastPass or Google's two-step authentication, they put a QR code on the screen, which you click.

Steve: Nice. It's a beautiful way to send information from...

Leo: And then it links to them, yeah.

Steve: Yes.

Leo: No data entry. You don't have to enter a magic code or anything like that. And it is very instantaneous. Google, of course, has Google Goggles, which is a QR code reader. I would imagine they keep an eye on that.

Steve: I'm sure.

Leo: And then there's another one that's very widely used on Android, that almost everybody has. I think it's just called Barcode Reader or something like that. It's kind of a generic name.

Steve: And it does bring up a URL so that you're able to look at...

Leo: You know, I was trying to find a QR code in here so I could take a picture of it. I guess I may just go to Wikipedia and do that. Because I'm curious. I don't know. So let me - I'll tell you what. Let's go to the Wikipedia QR code.

Steve: There's even, by the way, a QR code website from the original innovators, the Japanese guys. It is sort of a rough translation to English, but it's much better than my Japanese.

Leo: So you can just take a picture of that, or show that. How do you get a QR code to it? You upload it?

Steve: Oh, that one is just a site for, like, where the spec lives and so forth.

Leo: Ah, I see.

Steve: But, yeah, Wikipedia, or just put in - just Google "QR code," and you get all kinds of stuff.

Leo: There's plenty, yeah. I don't know why I was having such a hard time.

Steve: There's images for QR code. And there is a - I ran across a site, too, I think it was that one...

Leo: Let me just try. I have Google Goggles on here. Let me try that one. Okay. And we'll just scan a QR code. And it's good, actually, it's done exactly what you would want it to do. It's interpreted what it saw in the camera. It says "Wikipedia, the free encyclopedia," and it gives you the URL. And then I can tap a button...

Steve: Nice.

Leo: ...if I would like to do something with it, including, I presume, go to the web page, yeah. So that's the Google Goggles. So it interprets and then gives you a chance to back out if you don't like what you see.

Steve: Yeah. The one thing, I mean, so you certainly want that feature. And I would say don't go crazy with 50 different ones of these.

Leo: Yeah. Pick some well-known ones. You don't want...

Steve: Because, yes, exactly.

Leo: ...exploits.

Steve: Exactly.

Leo: I'll try the other ones. I don't have it on here, I think. I just use Google Goggles when I need a barcode reader. There's also RedLaser, which was one of the very first barcode scanner apps. They've been around for a long time. I'm sure they're safe. Yeah, the ZXing team does one called Barcode Scanner. I have that also on my phone. Let me - oh. Yeah, same thing. Actually, it's quite good. In fact, this gives you a lot more - this is another good one. It gives - this is Barcode Scanner from ZXing. It gives you the URL, it tells you what it is, and it also even gives you some information about the format of the QR code, an image of the QR code, the data, some metadata.

Steve: Cool.

Leo: You can open it in a browser, share via email, or share via SMS. So I would say these two, which are the most commonly used on Android, are probably...

Steve: Yup, stick with them.

Leo: Yeah, yeah. All right, Steve. Good. I'm glad we covered that. That's a great subject. I'm fascinated by all that.

Steve: I've wanted to dig down and understand them for some time. And now everybody who has listened to the podcast knows everything that you and I do.

Leo: So now you know the rest of the story. We do this show every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time, 1900 UTC on TWiT.tv. Please watch live. I love it because we can interact with the chatroom, and I can get that into the show. But if you can't, you can always listen after the fact. There's audio and video available on demand. Steve has two unique on-demand features. Of course, we always have the audio and video on the TWiT site. But Steve has two unique formats. One is a 16Kb audio for the really small audio file. And if you're a small audiophile you'll enjoy it. And he has the text transcript, which is an even smaller version. But I think most people get the show and read the text transcript. But it's a nice feature because it's also Google searchable, so it's a good way to find the contents of those shows.

All of the shows are available on his site, GRC.com, as well as our site, TWiT.tv, and wherever better podcasts are aggregated, like iTunes and all the other ones. You'll find other great things like SpinRite, the world's best hard drive maintenance and recovery utility, at GRC.com - Gibson Research Corporation. Lots of freebies, as well. And Steve's on Twitter as @SGgrc, and he tweets there regularly.

Steve: Yup.

Leo: Steve, thanks so much. I look forward to doing a special. Steve Gibson from the '90s.

Steve: Yes. Let's record our opener at the beginning of next week.

Leo: I'll come in early, just for you.

Steve: Cool.

Leo: Yeah. Or if I don't, we'll just do it at the beginning of next week because I'm having a hard time getting here on time, let alone early.

Steve: Yeah, no problem.

Leo: Thank you, Steve. I appreciate it.

Steve: Thanks, Leo.

Leo: We'll see you all next time on Security Now!.