SECURITY NOW!

Transcript of Episode #288

Listener Feedback #111

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

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-288.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-288-lq.mp3

TOM MERRITT: This is Security Now! with Steve Gibson, Episode 288, recorded February 16, 2011: Your questions, Steve's answers, #111.

It's time for Security Now!, the show you need to watch or listen to if you want to stay as safe as you can on the Internet. Joining us, the man behind GRC.com, ShieldsUP!, and SpinRite, Mr. Steve Gibson. Hey, Steve.

Steve Gibson: Hey, Tom. Great to be with you again for our third and final show together.

TOM: Our triptych of Security Now! episodes together. Leo, of course, as we've mentioned, is on vacation. He will be back next week, though. And I can't say I haven't enjoyed doing the show with you. But it will good to have him back.

Steve: Has been a pleasure, absolutely.

TOM: Absolutely. Well, let's get into - we've got some security updates, some new stuff around Windows, some clarifications. We've got some password stuff coming from Google. And of course questions and answers today.

Steve: This is our Q&A episode.

TOM: Yeah, a lot of Q&A about Bitcoin coming up.

Steve: Yeah, in fact, we could have done all of the questions about it, it's generated so much interest and curiosity from people. And actually I got some nice comments from people saying, hey, it was sort of a nice change of pace to talk about something crypto related, but not - some guy said "not just another Adobe security flaw" sort of thing.

TOM: Right. And something positive, too. Something that hopefully is working to better the world, rather than you just trying to defend yourself against somebody who's trying to take you down.

Steve: Right. Well, so I noted sort of in just the overall running news that Chrome, Google's browser, had crept up to major version 9 and then quickly had a couple things fixed. Google was mum, as they always are, about the details. Just that's their way. And so they fixed multiple, as always, unspecified vulnerabilities, which they rated as high. They repaired a flaw in an unspecified race condition in audio handling, an unspecified crash when printing PDFs, and an unspecified use-after-free condition related to image loading. So we don't really know what those are. But we're glad that they keep fixing these problems for us and moving Chrome along. And it's funny, when I noted that it was at v9.something, it's like, wait a minute, when did that happen?

TOM: How'd it get there?

Steve: When did that happen?

TOM: Yeah, I know. I do like that Chrome updates itself in the background. It's less disruptive and all of that. But I also miss being able to keep up on what's actually being done to my browser.

Steve: Yeah, it's a different model. My sense is they probably got it right. They came into the game later, so they were able to look at what Microsoft was already doing and had been doing, what Adobe had been doing. And they probably made a decision, it's like, well, okay, there's a certain class of person who wants knowledge-base index numbers and wants to go in. And certainly with the huge corporate load that Microsoft is dragging along with all of this, you can imagine that IT personnel historically are the reason we're doing Patch Tuesday. That is, Microsoft has consolidated these changes to Tuesday. Once upon a time these things were just being released all over the place, whenever it happened. And it caused IT people a huge amount of trouble because they were constantly getting patches from Microsoft. And in some cases those things broke specific IT infrastructure configurations that Microsoft had not been able to test for. So by relegating it only to second Tuesday of the month, IT sort of is able to calendar that and arrange time. They also really need details about what's happening.

So maybe one of the benefits Google has is that it is only a browser and not an OS that they are changing. You might imagine that if it had the decades of history that Microsoft's OSes are still legacy-wise dragging forward, that they probably couldn't get away with this. But as a browser it's like, yeah, yeah. And I agree with you. It's interesting to sort of see this approach as compared to the other approaches which are, it's like, okay, so we have all this detail. Do we really care? We just want it to be fixed. So that's what Chrome does, it just fixes itself constantly.

TOM: And anecdotally, I can say Flash has been performing better for me since this dropped. I don't know if that's just luck. But I haven't had as many of those Chrome crashes. So that's good news.

Steve: Well, and I did a reboot of my system yesterday. I don't do it often. But all my icons went away. So it's like, okay, well, I guess it's time to reboot Windows. And I got a new Adobe, I got a new Flash, and a couple things. But this version of Chrome did also include an updated version of Flash. Remember that they're now taking more responsibility for their plug-ins and going to be updating those, as well, moving forward, something we talked about some time ago. So we have seen that.

I wanted to give a shout-out and thanks to a frequent tweeter to me, who goes by the handle @Captn_Caveman on Twitter. He sends news often of things, like he's the first person to bring things to my attention. He did so just this morning. I got an early heads-up on a new zero-day exploit that's been found in the Windows SMB system. That's the Server Message Blocks, is what SMB stands for. We commonly refer to it as file and printer sharing. SMB is the protocol that it uses for file and printer sharing.

Now, it's caused some ruffles on the Internet because it's a remote code execution vulnerability. Secunia confirmed this. It had been posted by some guy calling himself Cupidon-3005, who posted this on the Full Disclosure mailing list, talking about - and that was why it was a zero-day exploit was the first anyone knew about it was when this, you might argue, irresponsible disclosure was made publicly. Secunia confirmed that this problem exists. It's a buffer overflow that can be triggered by sending a too-long server name string in a malformed browser election request packet.

Now, it's not browser in terms of web browser. This is the - unfortunately we have a name collision in Windows. The browser is also the name given to, like, the file browser that we sort of see in the form of Windows Explorer on our desktops. So you have, in any version of Windows, you've got a file server and a file browser which are individual services that actually you are able to start and stop, although all kinds of horrible things break if you do that. Now, this is not such a big deal, though, because this is, fundamentally, file and printer sharing ports. Which, of course, it's the reason I created ShieldsUP! a decade ago was to alert everyone to the problems of that. That's one of the main drivers for personal firewalls being created almost a decade ago. And many ISPs have taken it upon themselves to block those ports, that's ports 137, 138, and 139, and also 445, which are the ports that SMB services are running on. And now with Windows Firewall installed in Windows machines and enabled and running by default, you really are not very vulnerable to this.

The only place I could see a problem would be local vulnerabilities. For example, we have seen instances where malware will use fileshares in order to propagate within a corporate network or within a home network, once it gets into one machine through some other means. So it would be, I think, almost impossible for this to be a big problem out on the Internet because there's no way that malformed packets can reach your computers' ports 137, 138, 139, and 445 behind multiple layers of protection. And NAT routers, of course, NAT routers are providing us with a good hardware firewall, essentially.

TOM: If you're doing anything to protect yourself, you're probably going to be okay, it sounds like.

Steve: Exactly. The first thing you would do, anything you did would solve this problem. So it has been confirmed. I wouldn't be surprised if we see malware maybe adding this to its bag of tricks for spreading itself around within an organization or home network, once it gets a foothold inside. But it doesn't look like it's going to be - it's not going to be some major worm that's going to spread across the Internet like we used to have in the old Code Red and Nimda days and so forth.

TOM: Well, that's at least a good side of the news from them. But there is another critical vulnerability, though, being reported from Microsoft.

Steve: Well, yeah. Microsoft has a DLL that does all of the HTML and CSS parsing. We seem to be talking about CSS parsing problems now weekly. It's almost like some new module of Windows gets focus of the bad guys, who then start pounding on it. And it's like, oh, look, apparently Microsoft hasn't given much security attention to this. And so we see a stream of problems coming from one particular avenue. The only real explanation for that is that people are all looking in that direction at the moment and finding problems wherever they look.

So this is another MSHTML.dll problem. Reportedly, there's a dangling pointer than can be exploited. That would mean a pointer which is created and is not destroyed, still pointing to some memory that it's somehow possible to exploit through causing someone to visit a maliciously formed web page. IE6, 7, and 8 have been confirmed as affected. And I would be surprised if 9 wasn't because this is going to be - this MSHTML.dll would be a core component of what Microsoft is doing for rendering web pages. So this is a critical vulnerability. It just happened on Tuesday so Microsoft has had no chance to deal with it. Just wanted to sort of alert people that it exists. So with any luck they'll be fixing it and we'll be talking about it for a Patch Tuesday in March.

TOM: All right, we'll keep an ear out for that. Let's move on to some security news. Google added some increased security for logging into Gmail and Apps. They were calling it two-factor authentication when I read about it. But it's not - is it exactly that?

Steve: Well, okay. So we've talked a lot about multifactor authentication, the idea being that you don't want to rely just on something you know because it's possible for someone else to find out something you know. You could disclose your password. That's something you'd know. You know what your password is. You can write it down. A keystroke logger could be in your machine watching you enter it, so suddenly whoever is dumping the output from the keystroke logger, now they know what you know. So knowledge itself is a bit hard to keep hold of often. So the notion of multifactor authentication is that you - and again, we've done podcasts ad infinitum about this, the idea being that you need something else, for example, something you have in addition to something you know. Well, if it's a something in the physical world, then much as a bad guy in some other country may wish they had what you have, they don't have it, and they can't get it because it's something physical which has a physical property which can be tested.

TOM: Yeah, a lot of times people have little key chains that just run on an algorithm, that tells them what number to type in. Kind of works on the same principle as your car key does. You just don't see it in a car key.

Steve: Right. Famously, the eBay or PayPal football that we talk about is a little LCD dongle. You press the button, then you get a six-digit code, which is running a well-known public algorithm, but with an unknown secret key. So the serial number of that device associates it with its key by some agency like VeriSign, their VIP program for identity protection, which now Symantec owns. They're the service that you register your token with. And they know at any given time what will be displayed on that LCD.

So I wish - the one thing that Google has done, well, the one thing that they failed to do was they did not support VeriSign's VIP system. As we'll see, it's really gaining traction in the world. And it would be nice if they just said, okay, in addition to our own stuff you could also use your existing PayPal/eBay football. Or there's also an eInk version, same six digits, but it works on sequence rather than on time in order to minimize its power consumption. Yet Google is going their own way, which is fine.

They have offered, or now announced and created, a rather feature-complete, multifactor authentication system. So you're able to, under the User Settings tab, for example, in Gmail, you're able to - and I believe this is probably fully rolled out by now. This was a few days ago, and they said they would be rolling it out across Gmail users over the coming couple days. So if it's not there now, I imagine it will be there soon. So you're able to enable this and, for example, give them a phone number where, when you log into your Gmail account, after authenticating yourself with your username and password, a Gmail robot will phone the number you've registered. So, like, your cell phone right next to you will ring. When you answer it, this bot speaks the six-digit code, which you then enter into another field in addition to username and password to confirm to Google that you're in possession of your phone. So something you have is this second factor of authentication.

TOM: And that was the part that was a little hard for me to wrap my head around because it's actually being sent to you. So it's not really something you have. But I guess there's really no other way someone else could get it, so it acts like something you have. Because it can come as an app; it can come as an SMS; it can come as a voice speaking to you; right?

Steve: Well, yeah. So the good news is they've done a comprehensive job. So one thing it'll do is you can give it a phone number. You can also have it send you an SMS text message of the same content, if you would prefer that. Or you are able to download, and I did for my iPhone, you can download an app, either for Android, Blackberry, or iPhone, which essentially contains an algorithm that will generate the code for you in order to authenticate yourself with Google. So again, I'm a little annoyed because we already have all of this with, as I was saying, the VeriSign-Symantec VIP system. All they had to do was make that an option. And then now, for example, I have the Symantec VIP system on my Blackberry. Well, now if I want to use Google authentication the same way, I need to add another app. So we've talked often about the notion of not having a keychain full of separate dongles.

TOM: Right. I got my Blizzard, and my PayPal, and my Google, and, yeah.

Steve: Exactly. So I'm annoyed on that front. Maybe they'll add that. It would be great if they did. I don't think there's any cost associated with authentication from the person who's requesting authentication. Maybe I'm wrong, so maybe that was the problem is that there was some cost associated with it, and so Google said, well, we're not paying Symantec for the privilege of authenticating...

TOM: Especially if they want to put it out in this many different forms, if VeriSign was charging for each different delivery method or something. That could run it up, too. There's also an interesting thing about being able to carry some backup codes for when you're offline.

Steve: Yes. They did something that's very nice, which is you can print a number, I think up to 10, static one-time codes, which you carry in your wallet, or wherever. But the idea being if you, for whatever reason, you loaned your phone to a friend, I guess they're going to wonder why some robot is calling and spitting out numbers. But so you do have backup, one-time-use codes. The other thing that they did to lessen the burden of this somewhat is you can choose to remember the verification on a given machine for a month, for 30 days. Which is nice. So that means that, if you were to authenticate on a new machine, that is, you were at a friend's house, and you said, oh, I want to check my Gmail, well, there you'd need to have either your phone or your app running or your wallet with the 10 static passcodes in it, definitely, to authenticate on that new machine. But for the laptop you're always using, your machine at home, it might be annoying if you were constantly having to receive phone calls from Google in order to login to Gmail in the privacy of your own home. So here this allows you to back off a little bit and say, look, only every month I will reauthenticate myself on a given machine.

TOM: Does that work with something like Eudora, Thunderbird, MailApp? Does that allow you to use those for 30 days without having to constantly enter in that second bit?

Steve: Well, okay. So the other thing that they did, this is the final piece of this, is that they recognized that there are non-web browser-based third-party apps which, because they have existed for some time, will prompt you for a username and a password. But they don't know yet about this new authentication system, so they don't have a third field where you enter this code. So the one additional thing Google did to really make this system complete is you're able to use one-time passwords instead of the password you've associated with your username. They're able to give you one-time passwords that allow you to still log in through non-web browser-based apps. So they've really covered the bases. Again, I wish they had followed through and supported existing authentication technology rather than inventing their own. But from what I could see, you could argue it's everything that the Symantec/VeriSign approach offers, plus more because you're getting SMS, and you're getting an audio loop through the phone, and then these extra features. So I think it's very cool.

TOM: They do have a challenge question if you lose your second factor, if you lose the thing you have. Let's say you only had it set up on an app. You didn't set up a backup phone. That's one thing that's different from these other systems is, well, usually they say, hey, you lose that football, you're going to have to call us, and you're pretty much out of luck. Does that make it less secure, that they have that challenge system at the end? Or is that challenge system set up pretty good?

Steve: Certainly I think it makes it less secure. I know that, for example, even when I'm using my football and authenticating myself with PayPal, underneath the little field where they're at, they're prompting me for the six-digit code, they say "Click here if you don't have your football with you." And it's like, okay. And then you answer a couple questions which are, unfortunately, what city were you born in and what's your nickname or something, I mean, the kind of thing...

TOM: Really? PayPal does that.

Steve: Oh, yeah.

TOM: Oh, wow. Because Blizzard doesn't. Blizzard for World of Warcraft, you lose your dongle, you've got to call them and plead because your account is toast.

Steve: Frankly, I wish there were a way to even, like, disable that in PayPal, where you could say, look, I'm really very careful about what I do with my football. I'm not going to lose it. Please, for me, I don't want my security softened by falling back so easily to these security questions. Make me jump through some bigger hoops. But no. So, yes, it's certainly the case. And again, it's a tradeoff. In some of the articles that I was reading about the announcement of this, the author said, well, you can imagine that there will be some Gmail users who are going to get themselves tied up in knots. They'll somehow type in the wrong phone number. Or they'll one way or another cause themselves some grief as a consequence of this additional security. But that's always the, hey, that's the nature of additional security. That's going to happen.

TOM: We also have some additional security being added to the Sandy Bridge chipsets; right?

Steve: It's so cool. Actually it was a listener of ours, a Steve Fintel of Intel, has been working on this, he says, for the past year. I caught a note, he went to GRC.com/feedback and left a note for me that I saw, to bring to my attention the fact that the soon-to-be-released, like next month, the next generation of chips from Intel, as you said, the Sandy Bridge chipset - get this - will be incorporating exactly this Symantec/VeriSign-compatible, one-time password technology in the hardware.

TOM: At the chip level, wow.

Steve: Yes, at the chip level. So what it means is that your desktop machine, your laptop, will incorporate this technology to mean that you don't have to carry the football. You register yourself with whatever authentication provider that they're supporting, and they currently support both the Symantec/VeriSign VIP system and Vasco, which is a major supplier. In fact, I think Vasco was the actual hardware producer for the original PayPal football. That actually was one of their tokens. I remember at the RSA Conference a couple years ago seeing all these very familiar-looking tokens in the Vasco site, or at the Vasco booth at the RSA Conference.

So this is that familiar six-digit code that changes every 30 seconds, which we're used to seeing on the PayPal football. The Intel chipset will build that in, I mean, it's a trivial algorithm, so it's cool that they've done it. And not a big hardship for them, but very nice because it does provide - while it doesn't mean that you have the portability of the football, meaning the football allows you, or any of these portable tokens, or even for that matter Google's approach using a cell phone, to authenticate on any random machine. What this does is it authenticates that machine. So that's very nice because from a standpoint of bad guys logging in from overseas, where they don't have access to your machine, they have no way of knowing what six-digit code your particular machine would be generating at this point in time.

So what it does, of course, is it lowers the cost of this level of authentication. One can imagine that a couple years from now all the Intel chipsets - and this is a public protocol, by the way, which is why Intel has been able to incorporate it, why there are several different authentication back ends, and that means that AMD will be able to do it. And essentially, not long from now, it will just be built in. All of our machines will have this multifactor authentication. Our software will know how to query it, and we'll be able to surface that and use it to authenticate ourselves online. So this is - I think it represents a major step. I mean, Intel's move to putting this in the hardware is a very nice major step forward for authentication.

TOM: The thing you have is your computer.

Steve: Yeah, exactly.

TOM: Now, is there a danger that malware could get on your machine and be able to exploit that somehow by reading it out of your chip and sending it out?

Steve: I'm assuming Intel did this correctly, which means the vulnerability would exist if this weren't in hardware. If this were anywhere in the software running on your machine, then malware has pretty much equal access to anything else running in your machine. So, for example, underlying this sequence of six-digit numbers is a secret key, is a cryptographic key which, based on time of day, drives a cipher which produces the six-digit code.

So the beauty of Intel doing this in hardware is exactly like having the football, which is a freestanding separate piece of hardware. Which is to say that I'm sure Intel will have done this correctly, so that the key itself, which is the only means for knowing what this six-digit code sequence is, is absolutely unreadable by any software, period. That is, it's unique. It's burned into the chip. It's got to be printed somewhere, like maybe on the outside of the chip. There has to be some way for you to register your laptop, your desktop, with the authentication provider so that they're able to determine what this is. So somehow that has to have been handled. But the idea would be that there's absolutely no software interface that allows that key to be read, period. Which means that there just isn't a way for malware on your machine to get it and then leak that out. It has to be the case that they've done that right, or they haven't accomplished anything.

TOM: I hope they've done it right.

Steve: They certainly have the ability to do it right. Oh, and I did want to mention, for anyone who's interested, a short URL, blissfully short, Intel calls this Identity Protection Technology, IPT. So you can go to IPT.Intel.com, and they've got there two lists, one of hardware that is available, as it becomes available, and of websites that are currently using the Symantec VIP technology, and that list is growing. So, and interestingly, the hardware available list has no entries in it at the moment because I think it's, like, March 11 is when they will begin rolling this out. So the Sandy Bridge chipset, as it becomes available, will incorporate this. So IPT.Intel.com, as this happens, you could use that to help select your next computer because I think this is great. I mean, I don't see a downside to this at all, the idea of your hardware building in a universal, easily authenticated, six-digit code that's changing constantly. It's just another factor of authentication that all of our systems will have before long.

TOM: All right. Well, I hope you're right. I hope it works that well. And it is exciting to think about not having to carry around another thing on your keychain, for sure. Federal Trade Commission is finally catching up with Security Now! and making some good recommendations.

Steve: And what I like about this, I caught it, I think maybe a couple people tweeted it to me. So I was clued into it there. The headline was "FTC Warns About Public Wi-Fi Hotspot Dangers." Well, of course that's no news to any of us. The good news is that, if any of our listeners have ever wished there was a friendly, easy-to-use, really well put together website that they could point their less savvy friends and relatives to, I've got to say now there is. The FTC site is called OnguardOnline.gov, just all run together, OnguardOnline.gov. And it's very nicely designed. In fact, I was impressed by it. Of course, when I went there the first time, NoScript on Firefox did not allow JavaScript to run. And in black against a dark blue background, so actually it wasn't very legible, but it still said, "To view this site's flashiest features" - pun intended, I guess - "please ensure that JavaScript is enabled and use the latest Flash Player. If you don't enable JavaScript or install Flash, you will still find the site useful.

TOM: Oh, nice.

Steve: It was because then what I got was a large sort of scrolling window of topics. And the site was still usable. Now, when I saw that I thought, well, that's okay, good job. And nice that they really went to some effort to make it still function even without scripting. So when I did tell NoScript to trust the site temporarily, and I generally say temporarily for sites that I don't intend to go back to all the time, just because I don't want that list of sites I trust to grow forever, it seems unnecessary. So per session, I just said, yes, trust it. The page refreshed. Of course that little notice disappeared. And what I got was a very nice Flash and script-enabled experience, which is unfortunately what most people will get right off the bat because they'll be flying in there with script and Flash flying. So it works both ways. I was very impressed.

And under the WiFi Hotspots topic, which is highlighted there, they say "Wi-Fi hotspots are convenient, but they're often not secure. Learn how encryption protects your personal information, and get other tips for using public wireless networks." And if you click that, or anyone you tell clicks it, what you're taken to is a page I couldn't have written any better. I mean, it's really nice. It's got a few points up at the top which, I mean, are all correct and all perfectly phrased. And then nice explanations. Even as far as at the bottom, going as far as to suggest and recommend and mentioning Force-TLS and HTTPS-Everywhere, those Firefox add-ons which tend to encourage sites that are able to to be over SSL all the time.

So anyway, again, nothing there is anything that our listeners don't already know. But, oh, and there was another link to another even nicer sort of touchy-feely site that I had not been aware before by an outfit called the Internet Education Foundation, called Getnetwise.org, which is even friendlier, and very nice. So I wanted to bring it to our listeners' attention as something that they could refer friends and relatives to, to give them a nice hand-holding, if you want to learn more about security, this is a great place to start sort of site, Getnetwise.org. So I really recommend them both. I was very impressed.

TOM: Good stuff, yeah, good information. Coming from the government, too, so...

Steve: Better late than never.

TOM: Exactly, exactly. It's nice to have a couple of easy URLs that you can tell people, like, hey, if you're worried about this. And I do get people asking me, yeah, those public WiFi hotspots. I'm like, well, the ones that say "Free Public WiFi," you can very safely ignore those. Don't even go to them. But if you're worried about the other ones, the ones that seem legitimate, like a Starbucks Coffee, go to OnguardOnline.gov. That's great. It's good to have something like that.

Steve: And I should also mention, I mean, I hope our listeners will check those sites out because the WiFi hotspot was just one category of about 20. So, I mean, they talk about phishing, they talk about dangers of gaming online, they talk about email security. So, I mean, it's very comprehensive. The WiFi was just one of maybe 20 different topics. So I'm really, really impressed that that exists now and wanted our listeners to know.

TOM: Right, let's finish up the news section with Symantec releasing a little bit of new information about the Stuxnet virus.

Steve: Well, it's a little bit of new information on top of an amazingly comprehensive report. In fact, I think we'll do an entire Security Now! podcast about Stuxnet finally. We've been talking about it of course since it first reared its head. But there was, for the longest time, not much known. Symantec has updated, I think it's at v1.4 at the time of this podcast, what they call their "Stuxnet Dossier." And in fact, if you want to find it, you can just Google, that's probably the easiest way to find it is to Google "Stuxnet Dossier." I did tweet about it yesterday when it came to my attention. So you could also just check my tweet feed. I'm @SGgrc. And you'll find a link there that I tweeted the link to the PDF.

It's a 69-page report. That's why I don't intend at all to cover it in detail now. But there's enough meat in it, really, really interesting stuff in terms of, like, I mean, they've been able to backtrack this thing to the original 10 machines which were first infected and what happened after that, all the way out at the other end to the five specific sites that it was targeted to and did ultimately infiltrate. So it's going to make a great podcast. I'm going to read all 69 pages, and I'll share what I learn with my listeners.

TOM: Yeah, this is really detailed, down to compile time and infection times of different attack waves and clusters of infection graphs, little cluster graphs. This is incredible what they've done here. You do need to have Symantec.com approved for scripting because it's a PDF.

Steve: Ah, right.

TOM: Thing I found just now. It's like, why am I getting a black page? Oh, right.

Steve: And I did want to correct something that I said last week when we were talking about the Windows Update for the USB thumb drive autoplay. It turns out Microsoft put that in the optional category.

TOM: Oh, they did, okay.

Steve: So what was different was it will not install itself without your knowledge, but it will be visible for the first time in Windows Update. So instead of having - so what the difference was, instead of people having to go to some knowledge base article and track it down, as was necessary before, now it's in optional updates. And presumably it's just going to sit there in optional updates until users either say I don't want to see this anymore, as you're able to do with Windows Update, or ignore it, or install it. So it won't do it to you. But it is available much more easily now through Windows Update.

TOM: All right. We've got 10 good questions to get to in our Q&A session today. But I know we've got another SpinRite testimonial to talk about first.

Steve: Just a little quickie. I got a kick out of this because our listener referred to it as a recursive, the first recursive testimonial. His name is Mike Woods, and he wrote saying, "SpinRite helps me watch Security Now!. And he's in England. And he said, "Hi, Steve. I'm a regular Security Now! watcher. A few days ago" - hi, Mike. He says, "A few days ago I switched on my media PC to watch the latest episode, and I got the dreaded Windows Recovery screen. Two or three unsuccessful boot/repair/fail cycles later, I gave up. After hearing many testimonials for SpinRite on Security Now!, I decided to give it a try. One purchase and 10 hours of disk activity later, I was able to reboot the media PC and am currently watching the latest version of Security Now! on it. I thought you might find this story pleasantly recursive. Thanks for a great product. Mike."

TOM: That's great.

Steve: So thanks very much, Mike.

TOM: I'm glad you're able to keep watching, Mike. All right, let's move into Listener Feedback #111, start it off with Spencer in Fayetteville, Arkansas. And like we mentioned at the top of the show, a lot of Bitcoin questions in here because people are excited about this. But when you start using it, you immediately start having questions. When I was launching it during the show last week I had questions about, well, wait a minute, it doesn't seem to be connecting, how long does it take to connect, all that kind of thing. So here's Spencer's question.

He says: I'm a longtime fan of Security Now!, great stuff, and was very intrigued by last week's topic, Bitcoin. I perused the trade page that lists eCommerce sites that accept bitcoins as payment. As one of the most visible crypto geeks on the web, would you, Steve, ever consider supporting Bitcoin payments in exchange for GRC products? And then he says: Errata: A very large number of bitcoin transactions occurred on February 9th. Just a coincidence that Security Now! aired that day? Take care.

Steve: Well, I had a number of listeners write in with that question. And the answer is yes. I think - I like the idea. I think it's cool. I did write my own eCommerce system. So my feeling is, I mean, why not? What the heck? I'm not sure what it would mean to do that, but I think it would be fun. So next time I'm in my eCommerce system making some changes - I probably will be at some point in the future. I haven't touched it, knock on wood, actually since I wrote it. It never had any bugs. So it's been a while since I've been in there. But I know that our legislature in Washington keeps looking at all of the tax revenue which is being lost because Internet transactions are nontaxable under most circumstances, and I don't know how long that moratorium is going to continue. So I keep watching that nervously, thinking, okay, well, I'm probably going to have to go back in and deal with tax at some point. So when I'm in there, I think I probably will. I think that would be cool. Basically what it would mean is that I would be publishing that long alphanumeric string token, which would be our Bitcoin account. And people who had bitcoinage could move money over to us that way, which it would be authenticable, and it would be irreversible and, I think, fun. So I plan to do that.

TOM: How much would it be in bitcoins? Would you just keep it at parity? We drove up the value of bitcoins, I think, talking about it last week.

Steve: Actually, I think that also happened, yes. We drove it above a dollar per bitcoin. I know that it generated a lot of traffic. There was a comment on the Bitcoin.org page saying that - no, it was the Bitcoin status account on Twitter said that, due to unusual level of activity, they were having problems with their server. So I think we did tax them. Our listeners taxed them.

TOM: We did some stress testing.

Steve: Yeah. So I guess I would somehow use the current currency trading rate versus the U.S. dollar for bitcoinage, which you could probably determine online, to determine at any given time what fraction of bitcoins was equal to the price of the software. So, interesting problems.

TOM: Makes sense. Question No. 2 from Anthony Headley in Mississauga, Ontario says: So isn't this just a great way for funds to be laundered? And a lot of people in the chatroom had this question last week when we were talking about this. Since you can create multiple personae, couldn't the following be attempted? Badguy1 wants to send 10,000 BTC to Badguy2 without regulatory scrutiny. So Badguy1.1 sends 5,000 BTC to Badguy2.1. And then Badguy1.1 sends 5,000 BTC to Badguy2.2. Badguy2.1 and 2.2 are the same physical person, but there would be no way of knowing this. Worse yet, huge amounts of funds could be transferred via sneakernet, TrueCrypted databases, between borders without being able to be traced. It is interesting being able to see BTC transactions in real time, though.

Steve: Okay. So, first of all, yes. What I got a kick out of was of course it seems to me that what Anthony is referring to here when he talks about 10,000 BTC is the fact that, at least in the U.S., and I guess in Canada, where he is, transactions of up to, I think including, $10,000 do not have to be reported to the government. But transactions with your bank greater than $10,000, even $10,001, require a bunch of paperwork on your bank's part and end up needing to be reported. So he was talking about breaking up a $10,000 transaction into two $5,000 transactions and noting that you could easily create, as you can, multiple Bitcoin accounts, essentially, and transfer these numbers to different ones, although they end up all going to the same guy. And in fact presumably what one of those accounts would do, then, would be to transfer $5,000 of bitcoins, 5,000 bitcoins to the other account, which can be done for no cost, so there's no reason not to do so.

Okay. So the point is, these are all anonymous. So you could transfer a million dollars, I mean a million bitcoins, between two different Bitcoin accounts, and no one would be the wiser. The system is anonymous. It's distributed across the Internet. It's based on account numbers which have no identities associated with them. The only way you own the coinage that you do is that you have the private key that matches the public key. And the only thing ever visible out on the network are the public keys.

So this is like God's gift to money laundering. Which is, I mean, it's true. But this is always the dilemma. I mean, this, for example, is the crypto dilemma. It's like crypto is also God's gift to people who want to hide things. And unfortunately, it can be good people who just want privacy, to which they're legally entitled; it can also be bad people who are using cryptography to communicate bad deeds which they don't want to communicate under the public scrutiny because they could be caught.

So we have the same dilemma here. What we have is we have a fabulously powerful technology, enabled by crypto, which is absolutely bulletproof, just like crypto is. And just like crypto, this application of crypto creates a dilemma. And that is, yes, if this currency takes hold, as it is acquiring traction rapidly, so that you can move real world currency into bitcoins, and then you can transact bitcoins anonymously, and you can move bitcoins back into even some other different real world currency, then yeah, I mean, this has all the potential for being used both for all kinds of, you can imagine, like, humanitarian purposes where there are good reasons that you want, for example, an oppressive government not to be able to monitor these transactions, but also for, unfortunately, bad guys to move money around. I mean, that's what happens when you end up with a technology like this which is that powerful. It's just it's part of the bargain.

TOM: Glowdime (sp) in the chatroom points out that the $10,000 reporting rule in the U.S. was changed effective January 1, 2003 to a "suspicious activity rule," along with the enaction of the Patriot Act. So now I guess just putting money into Bitcoin could flag you because it's suspicious activity.

Steve: Well, and it's funny because, as I was talking about this, when I was just now talking about moving real world money into Bitcoin and back, it occurred to me that the way you would do that was through some of these trading companies, and they would probably have these reporting criteria. So maybe what Anthony was talking about was doing it 5,000 bitcoins, or actually probably $5,000 in this case, at a time, in order to avoid those reporting criteria. That is, yeah, you don't want to look suspicious, so you do 10 $1,000 transactions rather than one $10,000 transaction. And that's why he was talking about Badguy2.1 and 2.2. You could also be sending them, those $10,000 pieces, to 10 different account numbers which are in fact all owned by the same person.

So I get a better sense now for what he was saying. And absolutely, this system without limit allows you to create these accounts. And as he says, when he talked about moving data across borders, the idea is that your Bitcoin wallet is just your private key collection. And it's up to you to protect that, to back it up, to preserve it, to keep it private. So if you were to bundle that up into an encrypted container of some sort, then, yeah, I mean, no one knows what's in there. You can carry it wherever you want. And you are carrying your Bitcoin bank account with you.

TOM: Put your Bitcoin in your hidden TrueCrypt volume.

Steve: Right.

TOM: Question #3, John O. in Argyle, Texas follows up on the Windows Autorun episode, says: You said you were pretty sure it would be possible to reenable the autoplay function in the registry after the patch was applied. You were right, and Microsoft has a

Knowledge Base article on how to do this and has a Fixit solution also.

Steve: Yeah, I just wanted to post this for people who might have gotten themselves in trouble, who might in the future get themselves in trouble if, for example, they disable this autorun by installing it on a system where it turns out that they wish they still had it. It's support.microsoft.com/kb/967715. So again, it's support.microsoft.com/kb/967715. And that deals exactly with that problem, that is, you've installed this optional Windows Update on whatever systems, but on a given system you want it back on again. So Microsoft, their little Fixit solution is the one-button click in order to reenable. So that does exist.

TOM: All right. Thank you, John O. Question No. 4 of 10, Danial Bulloch in Concord, North Carolina, USA, North America, Sol3, Milky Way, Universe - wow, he's really covering all the bases here - wonders about lost change in Bitcoin. He says: Listening to Episode 287 left me with many questions as to how it could functionally be considered. But the one issue that stood out to me

most is the thought of "lost change." Since the coins appear to be stored locally, would it not be entirely possible to "drop" a coin if it wasn't properly backed up during hardware failure or another problem?

If it is possible, would that not result in a slow decline in the currency in circulation, with a slow increase in the value per coin? Sure, today this wouldn't be an issue; but if the currency was being used for potentially hundreds of years, at some point the only remaining currency in the system would be fractions of coins, adding up to a very little total. I guess, given enough decimal accuracy, a house costing one bitcoin wouldn't be so bad. But people wouldn't like the concept of a week's worth pay being worth 0.0003 of a bitcoin. This is an interesting question, which is...

Steve: It's brilliant.

TOM: ...like you said just now, we have to back it up. What happens if we don't?

Steve: Well, exactly. And he's completely right. The idea is that your sole proof of ownership of coinage is what you have in your wallet. Now, it occurred to me that there is a transaction trail from the beginning of time, that is, literally - which in this case is 2009, when that first block was created, the genesis block in the Bitcoin system. And that chain of blocks is a transaction log of every single transaction that occurs. So that would allow you to see where the original 50-coin owners were. But those inherently only contain their public keys, not their private keys. And we know that the whole point of the public key/private key pair is that you can't get one from the other.

So it is the case that, if you lose your wallet, if your hard drive crashes and your wallet's not backed up, I mean, essentially the instructions that are part of using Bitcoin talk about making a backup immediately after transactions. And you can understand why. I mean, this doesn't have any of the benefits of real world physical currency that just can't spontaneously evaporate. This stuff really can. And exactly as Danial suggests, once it's gone, once you lose your private key, which is your only claim of ownership to coinage, there's no way to get it back. You can't get it back from your public key. You can never prove that a given public key was yours unless you have the matching private key. So he is exactly right that, to the degree that people lost coins, once lost, they're gone forever. And there's no way to prove that they're gone forever to the system. So it would always enforce that 21 million coin absolute limit that it will be approaching, and ultimately there will be fewer than that number of coins in the system.

TOM: Maybe there could be an unclaimed bitcoins office, so to speak. Yeah, because you'd have to put in some kind of time limit to say, if you don't refresh and verify your ownership of the bitcoins, we'll put them back in circulation. But that's not the way it's set up right now. Right now they're yours forever. You don't have to prove that you own them. But at some point they might have to deal with that. That's a really interesting question.

Steve: Yeah. I don't see that there's a solution to this. I mean, one of the things that is cool about a system with this level of integrity is there are problems without solutions. And this is a problem without a solution. I mean, you can't have, oh, I'm sorry, I lost my coins, can you replace them, and anonymity both. Because if you're going to have absolute anonymity, then it has to be the case that you are unable to prove that you have lost your coins. Those two things go hand in hand. And I would argue that one of the cool things that this system offers is, by virtue of being a peer-to-peer network, it's absolutely anonymous.

TOM: John J. Jobst in Columbia, Illinois, wonders about faking out Bitcoin. All right, let's see if he's found another weakness built into the system. He says: Your Bitcoin broadcast was very interesting and a welcome diversion from hearing about the latest Adobe exploit. I was wondering what would prevent me from running Bitcoin on my 486 processor so that I wouldn't significantly increase the difficulty factor for everyone else, then offloading the problem to my GPU farm, then returning the answer to my 486 to upload? You said it was open source, so it should be possible to reprogram. Could he fool it that way?

Steve: Okay. This is another one of the things that is so cool about the system. I think that this bitcoin concept and the way it's been implemented is academically interesting just because the designer of it, in a sense, thought of everything. I mean, the system holds together, and it works. And it is unspoofable. So here's why what John suggests can't work. It is the network as a whole, operating in this vast peer-to-peer interconnected mesh, it determines the rate at which the puzzles are being solved by counting the rate, by looking at the rate at which these puzzles are being solved. By "puzzle," of course, I mean finding the tweak that is added to a block in order to get its hash to have the required number of leading zeroes.

So all the machines that are in the network which you have told, please mint coins for me, they're taking blocks and trying to solve the puzzle. If they're the first one to do so, which they announce on the network, they get the 50 bitcoins in the block. So all the machines have the incentive to announce their solution to a block as fast as possible. Which means all machines are going to be announcing the solution to the block as soon as they can because it's by doing so that they win 50 bitcoins at this point. So the network as a whole can see the rate of those announcements, which will always be made as quickly as possible. And it's the network as a whole which is balancing the difficulty to keep that rate at six announcements per hour, six blocks created per hour, which brings 300 coins at this point, new currency, into the network per hour.

So as more machines join the network, or as more powerful machines join the network, statistically, because of the overall increased processing power, there will be an increase in the rate at which the puzzles get solved. And so the network scales, by agreement, scales up the difficulty to slow down the rate at which the puzzles are being solved. So John's example of having sort of a slow processor front end and a screaming GPU farm back end doesn't get him anything because, if he's got the screaming GPU farm on the back end, it'll be providing solutions to his pokey 486 front end much more quickly than somebody else's, and so the network will recognize that, wow, this guy's minting coins over there somehow. But what happens is, if that continues, then that minting rate, the puzzle solving rate goes up. And so the entire network will agree, whoops, we've got to add another zero to the puzzle, make everybody work harder in order to bring the rate of block puzzle solution solving back down. It's just conceptually such a fantastic system.

TOM: Now, this begs another question, which is actually our next email from Efrain in Miami, Florida, who's, like, wait a minute, how can I compete, then? I'm a little confused about how this works. Does my old recycled machine have a chance of generating bitcoins against the massive bitcoin miner machines? And since I have multiple old machines, could I have them all run as one account? I guess my old machines combined cannot compete with a bitcoin miner machine. There's a sense, he says, that only the rich get richer.

Steve: Well, okay. That's the other cool thing about the nature of this. I also have had three machines running full-time for a week now. My little Mac Mini that we're doing the podcast on, actually I stopped Bitcoin just so that it wouldn't interfere with our podcast.

TOM: It does really use up a lot of cycles, I've noticed.

Steve: Oh, boy.

TOM: My fan runs constantly when I've got it on.

Steve: Yes. And in fact I've got one machine that's got - in fact, it's the one I built for the video experimenting, which is an i7875. And it's putting out heat at a much higher pace, because I've got Bitcoin running on it, than it did before. And it's generating 5,418 hashes per second. So that's the rate at which it's able to attempt to solve these problems. My little Mac Mini was doing 1,400 hashes per second. But here's the idea. The idea is that it's chance. It's statistics. All the machines in the Bitcoin network are guessing, essentially. They're guessing a 256-bit number which, when hashed along with the block, will produce a result that solves the problem, meaning some number of leading zero bits in the resulting hash. There is no way, I mean, the beauty of a cryptographic hash, the reason we use hashes, the thing that they provide for us, is it is statistically - I've lost my vocabulary.

TOM: Statistically unlikely, maybe?

Steve: Maybe the word "statistic." It's computationally - "impossible" is too strong.

TOM: Improbable?

Steve: Improbable. There is no way to brute force the hash. You cannot put something in and deliberately design what comes out. So you have to guess. You have to hash your guess along with the block and see what happens. Now, the only advantage that the faster machines or the GPU-based machines have is they can do many more guesses per second. So statistically on average they are more likely to stumble upon the solution before somebody with a slower machine. But that's not to say that somebody with a slower machine might not get lucky.

TOM: Right. It really is like a lottery; right? The rich people can buy more lottery tickets, but that doesn't mean they'll win.

Steve: Precisely. That's exactly it. If you bought more lottery tickets, you would be increasing your likelihood of winning the lottery. But how many examples do we see of some guy who only ever bought one in his life, and now he's a multimillionaire. So if you've got old machines, and you're curious, until you get tired of it, or until you get tired of the house warming up too quickly, they could be running Bitcoin. I've got several that are doing it, I mean, just sort of to see.

Now, it is the case that a typical PC today, running for about a year, would solve one puzzle before everybody else. So that would earn you at this point 50 bitcoins. I don't think I'm probably going to do this for a year. But I'll do it for a couple months before it's like, okay, fine. And there's a chance, I mean, a chance that I will look one day, and my bitcoin balance will be 50 rather than zero. And that would just be fun if that happened. I think that would be cool.

And to answer Efrain's question, each of the instances of Bitcoin that you install would have a different private key and a different public key. That is, they each have their own account. But remember that there is zero cost, zero transaction cost in this system. So if multiple machines each scored some money, you just transfer the money from one to the other. Each Bitcoin instance running, you're able to see and enumerate all of the keys that you've generated, all of the little accounts. And so all you would do is just send the coinage from one machine to the other. And in fact, the act of doing that is what generates transactions out on the peer-to-peer network, which creates the puzzles everyone is trying to solve, to validate and lock up those transactions. And then the funds go and are confirmed. That's just such a - it's an incredibly cool technology.

TOM: Yeah, it is. I'm really excited about it. I've got to put it on my gaming machine and let that thing crunch at it for a while if I'm going to have my best chance. Although you can't pay your electric bill in bitcoins yet. That's the only negative to it.

Steve: You're using up more power, yes, than the coinage you're generating.

TOM: All right. We'll finish up with some non-Bitcoin questions. Seven out of 10 is Eric Stearns in Denver, Colorado, who wonders about transparent open source versus opaque closed source. He says: You, Steve, and lots of other people, make a critical distinction between closed and open source software. The source code is available for the open source software, making it fairly easy for a knowledgeable person to evaluate how the software operates. I presume that the key benefit is that you can look at the instructions in the program before they are compiled into something useful for the computer to process.

But I've never understood the technical differences between the two, and what makes a closed source program so much more difficult to understand than an open source program. It must be possible to deconstruct how a closed source program operates by looking at the compiled code. Is the principle benefit of open source software the programmer's comments that are omitted by the compiler? Is there some reason that compiled code couldn't be deconstructed? Is it just that closed source programs can't be evaluated in a reasonable period of time? If so, how much longer would it take to understand the operation of a closed source program? Again, that's Eric, SpinRite owner for several years, he says.

Steve: Well, this is, I thought, a great question. And we happen to have a perfect model in something happening right now, which is Stuxnet. Stuxnet, the code, has been available on the Internet. It's been passed around freely for many, many months. And only now are the people who have been spending full-time trying to understand it reaching real definitive conclusions. If instead they had the source code for the Stuxnet package, they could have answered any questions in an afternoon. So that really is the difference.

I'm a little bit of a dinosaur, as we know, because I write all of my stuff in assembly language. That is, I write in the code that C compilers and C++ compilers and JavaScript and so forth, that are compiling and not interpreting, compile down to. So people through the years, with all the free stuff that I've put out, some people sometimes come into the newsgroup and grumble that why don't we have the source code for the DNS Benchmark? And normally somebody else will type in and say, wait a minute, Steve writes his code in assembly language. So all you have to do is disassemble it, and you do have the source code. Whereas compiling does create - okay. So the point there is that what I write is exactly what the computer executes, and you are able to disassemble that back to essentially the code that I wrote.

However, as Eric suggests, the act of assembling and compiling, you lose a whole lot of context. You lose normally the long and friendly names of variables, so you can't see, like, what's being added to something else. You see a couple of memory addresses being added to each other. But it takes a long time then, by looking at the code, to track the entire history of references to those memory addresses, to then begin to say, oh, you know, I think this area of code is doing this. And that area of code, which is doing, like, there's some I/O operations over there, it's doing that. And so, if you don't have the source - even if it's just disassembled assembly language, but even more so if it's disassembled compiled language, because there's a much greater distance between what the author wrote and what the machine executed due to this compilation. That's what makes writing in a higher level language easier - higher level because there's more of an abstraction between what you write and what the machine does - than lower level languages.

But the point is that there is a huge amount of valuable information lost. The names of subroutines, you can see that code jumps to somewhere, but you don't know what that somewhere is supposed to do until you analyze in detail what it appears to be doing. But the source code gives a name to the subroutine that, in well-written code, tells you all you ever want to know about it, so much so that you can sort of ignore what's in the subroutine. The subroutine says "print the string I've been passed." And so it's like, okay, if you jump there, that's what it's going to do.

So none of that information survives the compilation and assembly process down into machine language. All you're left with is sort of just this dense nugget of things jumping around and memory instructions and memory locations being added and subtracted and moved around. And it takes a huge amount of effort to sit there and basically, like, reverse engineer the intent and the purpose and the actual function from what all that does. It's so detailed that you want to step back further from it.

But the only way to do so is to literally understand it at that level first and then sort of work your way back to a higher level understanding, assigning these locations names and assigning subroutines names as you begin to figure out what they mean. So it really is a whole different ballgame to have just the result of something like the Stuxnet worm, which only now, months after sitting there, I mean, there have been guys at Symantec whose lives are Stuxnet for the last six months. And they're now beginning to understand it because all that was left was this little dense nest of instructions.

TOM: Those guys are salivating at the idea of having had an open source Stuxnet or an assembly-written Stuxnet, aren't they.

Steve: Yeah, somewhere is the source code. And it would have just, I mean, it's all laid out in the source code. They have recreated the source code, essentially; but, boy, that was a lot of work.

TOM: Three more questions. John Webb is next, from Mechanicsville, Virginia. And he's got a challenge. He says: Hey, Steve Gibson, I've been a fan of GRC, Security Now!, and a user of SpinRite 6 for some years now. I greatly appreciate Security Now! and regularly check the GRC website for the latest transcripts. However, in the recent Episode #286 with Tom Merritt, I was surprised that Intel's plan to implement an improvement to chip architecture to block zero-day malicious attacks was described as a "looney tunes announcement of the week."

As most zero-day attacks typically involve a buffer overrun in the stack to hijack the transfer of execution control, I have to say that I do not so lightly dismiss a chance to improve system security by implementing improvements in the processor design. Rather than dismissing this claim, I'm inclined to wonder how it might be possible for the processor to block attempts to overrun the buffer. I do not work for Intel, nor do I have any inside information; but that need not prevent us from using what we know and what we can imagine.

Steve: Okay. So I think John misunderstood, and there were a couple of other people who asked a similar question. So I wanted to address it. What you and I were chuckling about was Intel's claim that this solved zero-day exploits.

TOM: Right. It's not that it isn't more secure, it's that it eliminates the possibility. And you just should never speak in those absolutes.

Steve: Yes. So what Intel explicitly said, or this CTO, I think it was, of Intel, he was quoted as saying that this would eliminate zero-day exploits, which is exactly equivalent to saying this will eliminate all security vulnerabilities. I mean, that's what that means. The difference between a zero-day exploit and one that's not is that some researcher found it and told Microsoft about the problem, and they fixed it before the world found out about it. Zero-day exploits are ones where somebody found a vulnerability and didn't tell the producer of the software. We just saw it happening in the wild. So what the Intel CTO was quoted as saying, essentially, is there will never be another security vulnerability. Which is patently ridiculous.

TOM: And, yeah, what John is saying here is that, well, what about buffer overrun? And that's a great point; right? But what the guy said was not "we've solved buffer overrun problems." He said "we've solved all problems." And I bet that's not exactly what he meant to say. Or maybe he did, I don't know.

Steve: Well, yes. And so, for example, buffer overrun is one way that we've talked about of a bad guy getting code that they provide to run in your system. But there's also the return-based programming that we've talked about, where instead you jump to the end of existing code to get it to do a little bit of your work, and then it comes back. And then you jump to the tail of a different subroutine, like sort of into the middle of a subroutine, so that the last few things it does before it returns are what you want.

And so, for example, if the claim was once made that data execution protection, DEP, would allow you to prevent data buffers that were marked as non-executable from executing any code. So a buffer that was in one of these non-executable regions couldn't be executed. Well, somebody might have said, oh, well, that's going to prevent all security problems. It's like, whoops. It doesn't prevent jumping to the end, the tail end of existing subroutines, which by definition are executable, and having them do the work for you.

So unfortunately there is no pot of gold here in this arms race with security. No doubt Intel has some new hardware that will allow, when implemented, software to leverage it somehow to make some problems or some exploits harder to do. That's a good thing. That raises the bar. I didn't mean in any way to presume that Intel was not going to give us anything. But I was just saying the statement that was quoted was looney tunes because it can't be true.

TOM: Yeah, and I still agree with you. All right, let's finish up with a couple of Brits. Amos Kittelson in Bristol, U.K., has an interesting idea. He says: In Episode 286, Question 4, you and Tom discussed the difficulties inputting long WiFi passwords into portable devices. I started wondering if QR codes could help. QR codes are two-dimensional barcodes that can hold up to 4,296 alphanumeric characters in a variety of different formats. Most camera smartphones have apps available that allow you to scan the contents of the QR code into the phone's memory. "Barcode Scanner" in the Android market, for example, scans the contents directly into the clipboard, allowing you to paste it into your WiFi settings. May I suggest using the link below for creating a QR code, printing it out, and storing it in a safe place for use when necessary. Of course, this wouldn't help with a Kindle, but it's a start. Thanks for the great podcast and SpinRite, which has saved me from disaster more than once. So you'd have to get the different devices to play along with this, but what do you think? Is there any downside to having your password printed out on a QR code?

Steve: I think it's a clever means of getting it in. And he provided a link to a site that I think is really nifty that I wanted to share with our listeners. The site is Zxing.appspot.com/generator. So again, that's http://zxing.appspot.com/generator. And it is a very clean, nicely designed, QR code-generating web page. I'm sure you have to turn scripting on in order to use it. But it provides a lot of different formats of types of QR codes, and prompts you to input fields, and even has one for WiFi, where you're able to put in the SSID number of a hotspot and specify what type of encryption you use and put in your password, and it'll convert it into a QR code. So that's going a little further. But you can also just handle random ASCII, and it'll convert it. So I thought that was cool.

Apparently there's a good QR code scanner for Android, and there's one that I found for iPhone on the iTunes store. A bunch of them are there that are free that didn't seem worth the time downloading them. But, yeah, I thought it was cool. And it was a way to get something into your system. And they do transfer the data to your clipboard. So from there you could paste it into the password field of your WiFi. And so you can sort of carry that little gizmo around in your wallet, and no one would know what it was.

TOM: I like this. I'm trying it out. You only need, because it uses the Google API, you only need Google's scripts turned on for it to work. So if you've already done that, it'll work for you immediately. And it really works fast and clean. We just need to get Microsoft and Nintendo and Sony and Amazon to support this so we could just hold it up to our game consoles and say, there you go, there's my WiFi password.

Steve: Exactly.

TOM: Now we're in. All right, we've got one last question. And then we're done. And I have to say that I'm sad to see it go because Leo's coming back this week, and I've really enjoyed this. But let's get to the question first. This comes from Jerry in Swindon, England, about all of our talk of Firefox add-ons and track attackers and all of these different ways of stopping yourself being tracked by ad agencies. He says: Thanks for the continuing informative podcasts. Over the last few weeks I've been following your updates on site tracking. I recently was putting the Firefox add-on, Better Privacy, onto a new PC when I stumbled upon an add-on called Ghostery. I have to say it appears great. You do have to ensure you go into Options and select "All" to activate the 340 blocked sites. If any site is not blocked, the small icon in the Firefox status bar can be clicked to add the site trying to track you. You can opt for a small window to pop-up to tell you what sites are currently being blocked. And I have to say I had no idea how much "track-attack" was going on. It can also delete Flash and Silverlight cookies on exit. It probably sounds like I am the developer, given the amount of fluffing I have given this add-on, but I promise you I'm not. I have just found it to be a great tool for my PCs and those of my relatives. Just wondered if you had heard of it. Jerry.

Steve: Well, I really wanted to thank Jerry and also to share it with our listeners. I had not heard of it, and I'm really glad to know of it. I've had no chance to play with it yet. But it's on my list of things to do. So if it ends up being something that I end up using, I'll let our listeners know. I do have something I added which is blocking or deleting Flash cookies on exit. And if this does that, too, then I will probably remove that other one and add this and see how it goes. But I just wanted to give our listeners a heads up. It's Ghostery, and Jerry thinks it's great.

TOM: Yeah, if you go to Ghostery.com, they have add-ons for Firefox, Safari, Google, Chrome, and IE. So it's not just a Firefox add-on. That's great.

Steve: Very nice.

TOM: I think I now am going to steal Jerry's idea and use it as my tool on This Week in Google because it's a Chrome plug-in, and I definitely want to try this out.

Steve: Cool.

TOM: All right. Well, that is the end of this episode of Security Now!. Thanks, everybody, for watching. And Steve, thank you so much for letting me fill in for Leo while he was gone. This has been a blast. I've really enjoyed it.

Steve: Tom, you've been a great host. And Leo, we know, likes to travel around. So I'm sure this is not the last that our listeners will be hearing from you on this podcast. I look forward to your return.

TOM: I'll bet you a bitcoin you're right.

Steve: Thanks, Tom.

TOM: All right, thanks, Steve. We'll see you all next time on Security Now!.