SECURITY NOW!

Transcript of Episode #375

Listener Feedback #153

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

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-375.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-375-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson's here, and we have 10 questions and answers. We'll also talk about, yes, yet another flaw in Java. Oracle says, eh, we're not going to fix that till next year. Steve has the details, next on Security Now!.

Leo Laporte: It's time for Security Now! with Steve Gibson. This is Episode 375, recorded October 24th, 2012: Your questions, Steve's answers, #153.

It's time for Security Now!, the show for adepts, for geeks who know their stuff or want to be adepts. Our Explainer in Chief is here, Steve Gibson. I say that, Steve, because - hello, first of all.

Steve Gibson: Hey. It's great to be with you again, as always, my friend.

Leo: I say that because I was talking to Shannon Morse, who is one of our newest hires here. She is going to be a producer and host on Before You Buy. People know her from Hak5. And she and Hak5 host Darren Kitchen are doing a new show on the YouTube Channel called TechFeed, which is all about tech. And they're doing a security show. And I thought, uh-oh. But then Shannon explained it's security for real people. I thought, oh, good. Because we don't do security for real people. We don't do anything nice and easy. No, we do everything nice and nerdy. And this show is for people who really want to understand deeply. And I think that's not my mom or my grandma or my uncle or my son. It's people who are really nerdy. Right, Steve? I mean, I don't think that's unfair to characterize this. We try and make it accessible, but...

Steve: I have no problem with that. I think that what's valuable is that there is currently nowhere else to get what we offer. We got a lot of great feedback, for example, from last week's serious propeller-winding episode on how elliptic curve crypto works. I mean, like, really a lot of tweets and neat feedback from people who were saying, hey, I actually got that. And some people were using it to put themselves to sleep. But mostly it was people who enjoyed having the stuff.

Leo: Yeah. So that's really the point. And I think it's actually the point of TWiT is I don't want to do dumbed-down content. I want to do content for people who really want to know the actual details behind this stuff. And I think we've done all right by that.

Steve: Well, and that's what interests both of us. I mean, it really is driven...

Leo: Yeah, well, that's mostly why.

Steve: It's driven by who we are. And it's like, okay, I can't produce endlessly a show about security for mere mortals.

Leo: Mere mortals? Who are they? Who be they? Oh, what fools these mortals be.

Steve: Ephemeral mere mortals. Not really sure that person exists.

Leo: But I think that if you listen to the show, there's plenty of detail here. You make it as easy as possible with your transcriptions for people to go back and look at, study, and chew. And I think this is a college-level course is what this is. And high school somebody needs to do that. Somebody needs to do elementary school. And that's fine.

By the way, this is a Q&A episode. So that's the other option is, if you don't understand something, literally every other episode you get to ask Steve questions, and he can further explain. And I think that's really...

Steve: Well, and as a reminder for people who are joining us recently, I am sort of assuming that our listeners have been with us for the last seven-and-a-half years. We're at Episode 375. And 374 previous episodes are all available online for people to get. And so there is some assumption, I mean, I don't rely heavily on content we've covered. I'll often say, I'll mention it, okay, now, we've already talked about public key crypto and how it's based typically on factoring in an episode devoted to that. So we're not going to go over the whole thing again. But anyway, the point is there is some assumption that people have been following along. And for those who haven't been, because they're just joining us recently, that is all available online at both of our websites.

Leo: Yes, at GRC.com, Steve's home; and TWiT.tv, our home. I'll not say "my home."

Steve: Your domain.

Leo: Our home. All right, Steve Gibson. Security news.

Steve: So, okay. We're now on - you've heard of suicide watch; right?

Leo: No. Well, yeah, for people who are suicidal, yeah.

Steve: Well, Oracle is.

Leo: Oh, dear. Oh, no.

Steve: We are on Oracle exploit watch, or maybe it's Java exploit watch, courtesy of Oracle.

Leo: Oh, Java I believe, yeah.

Steve: Yeah. We've spoken about the Polish security researcher, Adam Gowdiak, before, whose company is Security...

Leo: Sounds like a joke.

Steve: Huh?

Leo: A Polish security researcher. There's a joke in there, but I'm not going to do it.

Steve: No, no, no, no. Seriously. He's a good guy with Security Explorations. And he reported to Oracle, and we have discussed this already a couple weeks ago, he reported discovering a very bad zero-day vulnerability that affects all versions of Java - v5, v6, v7. He provided them with demonstration exploit and an explanation of the problem. And they've essentially blown him off.

Leo: Oh, come on.

Steve: No. I'm not kidding.

Leo: We're going to have to get a little new musical sounder: It's the Java Exploit of the Week Week Week Week. They blew him off?

Steve: Yes. They said, well, we were unable to get this into the most recent update. Remember we just had a massive Java update, 109 different problems fixed, in their October refresh. Well, now they're saying they're going to fix this in February of 2013. Kaspersky wrote in their Threatpost blog, they said: "The vulnerability and exploit were announced in late September. Gowdiak's exploit successfully beat a fully patched Windows 7 computer running Firefox 15.0.1, Chrome 21, Internet Explorer 9, Opera 12, and Safari 5.1.7. The exploit relies on a user landing on a site hosting the exploit. An attacker would use a malicious Java applet or banner ad to drop the malware and ultimately take over full control of a user's then compromised machine."

So this is as bad as a Java exploit gets. So being a little bit annoyed, Adam has now explained that he's fixed the problem, which took him, he says, under 30 minutes. It actually took him 26 minutes to fix the known glaring Java zero-day vulnerability. And what Oracle is saying is that, oh, they're already in the works testing the February update against all of their different platforms, and it's too late to put this into the release cycle, which is, what, four months away.

And so Adam wrote: "Code logic is not changed at all. Minor changes are applied to the code. None of them influence what could be described as an externally visible scope affecting third-party applications." So all indications are that this is pure bureaucratic mumbo-jumbo, to be polite, from Oracle. So the question is, this is not in the wild yet. This is privately disclosed. Adam and his group at Security Explorations know what this is. It's been made public. We know that there is this problem. And so now the question is, does Oracle find themselves more or less deliberately compromising users' machines through their reticence to fix this now, or to add this to their - to respond quickly.

Leo: But they also have other things they're patching. I mean, it's not like, I mean, they...

Steve: I know.

Leo: Okay.

Steve: Yeah. So they're saying they're holding to their February 2013 release. So we are...

Leo: 2013?

Steve: February 2013. That's the next time they're going to update Java.

Leo: Just don't use Java. Don't be foolish. Just stop using Java.

Steve: Exactly. This is across all browsers, all platforms, all versions of Java. This is bad. And this is a remote, I mean, this is as bad as it gets, a critical vulnerability that allows anyone who discovers it to take over your machine. So...

Leo: God. Now, it has to come in through the browser, or you have to run a Java program directly on your computer; right?

Steve: Nope. Kaspersky, describing it, said: "The exploit relies on a user landing on a site hosting the exploit."

Leo: But that's what I mean. It comes in through your browser.

Steve: Correct.

Leo: Your browser has to execute Java. Presumably they could manage an exploit by writing a program with code like that in it that you would download and run separately.

Steve: Yeah, but, I mean, for example, this could be in injected in an ad. So you could have a banner ad served up maliciously somewhere. No, I mean, this is the way people are now getting their machines primarily infected is visiting websites that are taking advantage, I mean, even the fiction that we're reading, Mark Russinovich's approaches to how people get compromised is this because this is how it happens now. And so here's Oracle, knowing that there's a problem and saying, eh, it's not public, so we're going to wait till February of 2013.

Leo: Or until it becomes public, which it presumably will.

Steve: We just had this huge disaster with the Macs all being compromised, hundreds of thousands of Macintosh machines. So it's like, oh, okay. So we are thus on Oracle exploit watch. We'll see what happens.

Leo: Oh, geez.

Steve: Now, many people tweeted - and thank you, everybody who tweeted - an annoying announcement. And you probably saw this, Leo. The headline was covered in many different locations, claiming a tenfold bandwidth improvement through using some algebra. And, I mean, like a typical headline was "Algebra creates tenfold bandwidth improvement."

Leo: What?

Steve: And it's like, I don't think so.

Leo: What?

Steve: But here's the deal. It's completely bogus. Researchers at MIT, the University of Porto in Portugal, Harvard, Caltech, and the Technical University of Munich have come up with a solution which can improve the Internet-using experience of a class of users. So quoting from, I think this might have been, it wasn't Wired, it was maybe Techworld. Anyway, all of the stories pretty much used the same boilerplate from the press release that the group that are commercializing this put out. It says:

"The technology transforms the way packets of data are sent. Instead of sending packets, it sends algebraic equations that describe series of packets. So if a packet goes missing, instead of asking the network to resend it, the receiving device can solve for the missing one itself. Since the equations involved are simple and linear, the processing load on a phone, router, or base station is negligible."

So it's like, okay. Well, listeners to this podcast have enough understanding of technology to probably get or guess what's going on. First of all, there's no way that you get a tenfold bandwidth improvement. And in the press release they talk about how they were demonstrating this by playing a YouTube video on a train somewhere, and theirs was just playing smoothly...

Leo: Oh, please.

Steve: ...without problems, while other people around them were having glitch-y, stop-y, unworkable experiences. So, and I don't mean to downplay what they've done. This is nice. But this is not a tenfold bandwidth improvement thanks to some algebra. What they've done is they've added error correction code. It's like RAID for WiFi. And something like this has been done recently. I want to say that the Steam distribution system does this also because I know that Mark Thompson and I, a couple years ago, he was implementing his own content distribution technology for a group that he was working with. And so we were talking about this, and he was looking into the code and so forth. And I've often talked about error correction in the context, of course, of hard drives.

And so it's clear that what they're doing is they're actually adding some overhead to each packet sent, such that, if some are missing, using typical error correction technology, which is not rocket science by any means, they can fill in the missing data. So, I mean, it's cool, and it's clever, but it certainly doesn't get you a tenfold bandwidth improvement. And they talk about how, if a packet were lost, what you would normally have to do is ask for it again. You'd have to send back, oops, we don't have this packet, and get it again. Now, the streaming protocols already get around doing that. They'll just skip the packet. You and I are talking back and forth on a streaming protocol designed to be tolerant of lost packets.

So anyway, so I just wanted to respond to the many people who tweeted saying, hey, can you tell us about this? First of all, calm down, it's not a tenfold bandwidth improvement. That just isn't available. It is, in fact, if you had perfect packet delivery, it would be a slight bandwidth reduction. Yet in the presence of a certain level of lost packets, then that extra overhead ends up benefiting you, if you're in a situation where your particular use of the Internet is intolerant of a roundtrip delay for dealing with a lost packet, because the overhead you've added to every single packet allows you to compute the contents of lost packets.

So anyway, there's a company, Code-on.org, which is the licensing LLC that's been set up among these parties. It's an MIT/Caltech startup called Code-On Technologies, where they're making this available. And I think it's a good thing. Maybe it'll catch on for certain uses. But a tenfold bandwidth improvement, no. I mean, the only way - you'd be hard pressed, actually, to make the system do that. If you had really high packet loss, then at some point even this system won't help you.

I mean, I guess it could be dynamically adaptive so that, as your packet loss goes up, it starts putting more redundancy into the packets in order to increase the amount of correction, to increase a tolerance for packet loss. It doesn't, I mean, these problems have been solved all over the place in all kinds of ways. The dynamics are different here than, for example, on a hard disk drive because here you do have the ability to ask for data again if you are unable to correct it. On a hard disk drive, once the data has been written, then you lose that data from the write buffer, and you now are required to recover it and correct it when you are trying to read it later. So things are a little bit different than in a real-time data flow.

But anyway, it's an interesting thing. But it'd be interesting to see if you could actually demonstrate a tenfold bandwidth improvement. It would have to be an extremely lossy environment, where you were performing heavy replacement of lost packets. I don't even know. To me that just seems like you're really stretching it.

Another little piece of interesting news that a number of our listeners picked up on and made sure I knew about: A mathematician, Zach Harris, received a spoofed email from Google. That is, it looked like an authentic piece of email from Google, from some sort of a headhunter, like offering him a job. And he wasn't interested in the job, but he was interested in the fact that this was clearly spam, but it should have been impossible to spoof it because it was protected with Google's DKIM, the Domain Keys Identified Mail. We've talked about the technology of DKIM before. The idea is that the mail is signed with Google's private key, and DNS is used for publishing their public key. So this is a very nice, simple, sort of straightforward demonstration and application of the use of public key crypto. That is, you use DNS to publish the public signing key, which is used to verify the signature done with the private key.

The problem is, when Zach, who is a mathematician, took a look at the email headers, he realized that it was a 512-bit public key. And we all know that's not enough anymore. And Zach said, he was quoted in an article saying, "I like factoring." So he factored from the email the crypto that Google was using because he was able to look up, using DNS, look up their public key. He cracked it. And then he spoofed his own email back to Google, and it was like between Brin and - is it Serge?

Leo: Sergey.

Steve: Sergey.

Leo: Larry Page and Sergey Brin. He faked, he pretended it was him?

Steve: He faked an email...

Leo: Might as well go to the top.

Steve: ...from one of them to the other, in both directions, an email that was clearly spoofed.

Leo: That got their attention.

Steve: Exactly. And what happened was the next day Google's DKIM went to 2048 bits.

Leo: Wow. What was his email? Yo ho ho, dude? Arrrgh.

Steve: So then he looks around, and it turns out that the same weak DKIM keys are currently in use by PayPal, Yahoo!, Amazon, eBay, Apple, Dell, LinkedIn, Twitter, SBCGlobal, U.S. Bank, HP, Match.com, and HSBC. So what this is - and again, this was - the headlines were ridiculous. The headlines were "Massive Internet Security Vulnerability." Okay, no, no. This is for spoofing email. And maybe people thought, well, we'll use a 512-bit key because who's going to bother factoring that, just to spoof.

Leo: It's not a high-value target.

Steve: No.

Leo: Although it is. I mean, it's not insignificant, either.

Steve: The DKIM standard calls for a 1024-bit minimum key.

Leo: Oh, interesting.

Steve: And that makes sense because, I mean, 1024 bits is fine today. We already know that 512 isn't. 768, which is splitting the difference between 512 and 1024, eh, it hasn't been hacked yet. There's been no, I mean, that's like where the contests are. The factoring contests are sitting at - looking for factoring a 768-bit key. 512, as we've already seen, and as Zach re-proved, is no longer strong enough. We need 1024 bits. So I imagine everybody will pretty quickly strengthen their antispoofing public key crypto to 1024 bits.

It must be, again, remember these are relatively expensive operations. Public key crypto using RSA style is expensive. This is another reason why switching to ECC, elliptic curve crypto, for this sort of application would make sense. Because mail is being processed a lot, obviously. There's huge amounts of mail going back and forth. And this requires a relatively expensive crypto process in order to process email headers. So that's probably why they were sort of hoping they could get away with a shorter 512-bit key because it would be lower processing overhead. Doesn't look like that's going to be possible any longer.

Leo: Interesting.

Steve: Because we've got lots of people who like factoring.

Leo: Yeah, and know how to do it.

Steve: And then lastly, I wanted to point out a story that has been out for the last few days about a surprising compromise at Barnes & Noble. Sixty-three stores spread all over the country - California, Connecticut, Florida, Illinois, Massachusetts, New Jersey, New York, Pennsylvania, and Rhode Island - were all compromised, having somehow the PIN pads on the checkout registers were compromised. And we've talked about PIN pad compromises in the past where somebody would maliciously sneak a little radio into the design of the pad. This surprised people. This is 63 stores, geographically spread.

When Barnes & Noble informed the FBI, the FBI asked Barnes & Noble not to go public with the information because it wasn't clear whether this was an inside job, whether this was organized crime, what was going on. And the barrier of secrecy has since been lifted on this so that Barnes & Noble was able to inform the world.

And the takeaway is, if you are a listener, or know somebody who has used the PIN pads at Barnes & Noble stores in California, Connecticut, Florida, Illinois, Massachusetts, New Jersey, New York, Pennsylvania, or Rhode Island, then you should consider seriously - and this is the advice from the FBI and Barnes & Noble - change your PIN. Because many customers of Barnes & Noble during a period of time, apparently around through the summer, were subject to - and, I mean, this has been exploited. This was found because people were getting their debit cards, ATM cards compromised, and fraudulent charges were showing up on them. And they tracked down the common factor among all these people were they had been at Barnes & Noble. And so this problem was located.

Leo: This was a hardware hack. They were able to get some hardware.

Steve: Well, and then there's still not full information available. I tried to dig down, and they're now saying this has happened, but it's not clear whether it's hardware or software. It would be, I mean, I just don't know. But they are saying it is the PIN pad. The stories show a picture of one, as if it's that physical thing. But it could be, for example, PIN pad firmware, which is in the gray zone. Is that hardware, or is that software? Well, it's firmware. So specifically the word means it's somewhere between soft and hard.

Leo: It's firm.

Steve: Yeah, it's firm. So last week I mentioned Michael McCollum's newest novel, which I had just finished reading; and, as a consequence, we were able to have our regularly scheduled podcast on elliptic curve crypto. I read right up to the finish line Tuesday night. And it is now published.

Leo: Oh, good.

Steve: In order to tweet about it as I did yesterday, I wrote a review on Amazon. The title of my review was "A Great Adventure Which Unfolds Between Continents, Rather Than Between Stars." I said: "I'm a huge fan of Michael McCollum's always terrific space operas. They rank among my most favorite sci-fi, which I recommend routinely and without hesitation. They're so much fun that I've read most of them more than once; and, if you haven't yet discovered his Gibraltar and Antares trilogies, don't stop shopping after you have grabbed this latest work, 'Euclid's Wall.' In 'Euclid's Wall,' Michael brings us down to Earth, or back to Earth, or stuck on Earth after an innocent mistake made by a pair of PhDs in the near future, in 2087..."

Leo: You know, those PhDs again. I'll tell you.

Steve: "...pretty much ends the world as we know it. Whoops. Thus the sailing ship on the book's cover, 100 years post-catastrophe, as humanity struggles to recreate and rebuild pieces of what has been lost. Lost was the technology required to fly, along with pretty much everything else we take for granted these days. The whole idea is really quite thought-provoking, and more than a little chilling.

"Into this hauntingly plausible future, Michael introduces relatable and engaging characters, then weaves the sort of clever plot puzzle I've always enjoyed about his novels. Though I finished the book several weeks ago, it has remained with me, a bit haunting, like a recent vivid dream.

"If you have an imagination that's usually satisfied by phasers and photon torpedoes - though you won't find those here - I believe you'll enjoy this voyage on Michael's high seas every bit as much as I did. I rated this book five stars because I think it deserves every one of them. I bet you think so, too."

So for anyone who has read Gibraltar or Antares or all of the other things that Michael does, he of course is at SciFi-AZ.com. And he's got his books - he publishes them himself. He bought all of the equipment a long time ago to print and bind softcover novels, just because he wanted to be vertically integrated. But also it's available at Amazon on the Kindle store. So it's a really fun book.

And while I was going through the mailbag for today's Q&A, I ran across - there was a subject line that I said, huh? What? It was "Honor Harrington One Year Later." Chris Schwanekamp in Columbus, Ohio, wrote. He said, "In Episode 318, on 9/15/2011, you mentioned the Honor Harrington series; and, based on your enthusiastic endorsement, I started," he says in quotes, "'reading it,' unquote. I actually listen to the audio books in the car to and from work. I can't thank you enough, and the others who mentioned it to you as well, for bringing this series to my attention. It was my very first sci-fi series, and it was simply incredible. Once I started reading, I didn't look back. I dropped Security Now! like a bad habit. I read all 13 books, then broke off and went through almost all the short stories, then back to re-read 'On Basilisk Station' again. But my favorite book by far was 'Echoes of Honor.' Book No. 14, 'Shadow of Freedom,' can't get here soon enough.

"I've now started to catch up on all the Security Now! episodes I missed. But I've got to be honest, you and Leo are a bit of a letdown compared to the Honor Harrington series.

Still, I admit, I've missed you guys." So I'm glad to have you back. "This series was life-changing for me, and I'm sure I'll be reading and re-reading these books the rest of my life. For that I can't thank you enough. Now, with regard to Security Now!, 'Let's be about it.'" Which actually is one of Honor's favorite phrases, so it's very much like Picard's "Make it so, No. 1."

Leo: What is it again?

Steve: "Let's be about it."

Leo: "Let's be about it."

Steve: Generally the way she ends her meetings, she'll...

Leo: "Let's be about it." "Make it so" is a little more dramatic than "Be about it," but okay.

Steve: Exactly.

Leo: "Make it so" was taken, yeah.

Steve: And I've been flooded with really neat health-related low-carb feedback from our listeners, whom I asked to send me their results and experiences of any sort. And so we will verify that the calendar is free this coming Sunday, October 28th, 2:00 p.m. Pacific time, in which case we will, if it is free, record Episode No. 3 of "Over the Sugar Hill," essentially looking, at six months later, what our listeners have discovered and what you and I have discovered.

Leo: Now, Dr. Mom fortunately still has some painkillers left over, so...

Steve: She can medicate herself.

Leo: Premedicate. Giving her a good warning here.

Steve: That's good, yeah.

Leo: So, yeah. Well, I don't - Chad's not here yet, but I don't anticipate any difficulty doing that this Sunday. So plan on it.

Steve: Okay. So anyone who wants to listen live, we're targeting at this coming Sunday, October 28th, 2:00 p.m. Pacific time. It'll of course be a TWiT Special, so you'll be able to grab the podcast any time after that.

Leo: Yes.

Steve: As we have the other things. And I just did want to mention briefly, since I'm an iPad lover, that yesterday was the big keynote and introduction of the, I mean, anticlimactic iPad mini announcement, as well as a refresh of a number of Apple's other things. And I think the thing that I like most about the mini, Leo, is that they managed to get very slim margins on two sides, where they don't have the camera or the home button, holding it in portrait mode as opposed to landscape orientation. In portrait mode it would be the left and right edges. And so that gives it an overall sort of svelte feeling, which I like. And it was 9.7 diagonal measure, which is easy to remember because the regular size pad is - I'm sorry, 7.9, because the regular pad is 9.7.

Leo: Oh, I didn't even think of that. It's flip-flopped.

Steve: Yeah. 7.9 and 9.7. And it looks like a nice device. I guess the criticism has been that it's on the pricey side. It'll be interesting to see how the market judges that because we do have things like the Nexus 7, which at $199 is just a very nice, obviously non-Apple solution. And as I was saying to you before we began recording, the one thing that chafes a bit is that Apple still bumps the price by $100 as you increase memory size from 16GB to 32 to 64, because EEROM, this nonvolatile memory, just doesn't cost that much. And for them to gouge us $100 per step does seem much. And, for example, other companies, high-volume producers like Amazon, have a much lower price increment as you increase the amount of nonvolatile memory on their devices.

Leo: Well, and Google has an announcement on Monday, in which I think they're going to update the Nexus 7. Yeah, I think they're doing a - I think, the rumor is, they're going to do a Nexus 10 that will be higher DPI than the iPad 3 or 4.

Steve: [Strangling noise]

Leo: So hold on. I don't think the price, I mean, 70 bucks more, remember, you're quoting $200 for an 8GB Nexus 7. It's 250 for a 16GB. So compare 250 to 329. That's not - a 70 bucks premium for access to the Apple iOS Store and all of that, I think that that is about as good a price as one would expect from Apple.

Steve: Yeah, when you put it that way, I think you're right.

Leo: By the way, you don't have to order one. Orders start Friday at midnight, Thursday-Friday at midnight. And then they'll arrive the following week. And of course we order one because we have to. So I'll be getting the WiFi one.

Steve: Yeah. I think what I'll probably do, I won't own one until I make the mistake of walking into an Apple store.

Leo: Well, that's why - I want to hold it because some people have said - and of course I didn't go to the event, and people who did held it. And it looked like it was maybe a bit of a stretch for the hand. I don't know if you can hold that in one hand, even though that's what they're promoting.

Steve: Yeah. And frankly, I'm a little wondering about - I like the thin margins, the thin frame.

Leo: Thin bezel, yeah.

Steve: The bezel. On the other hand, it's useful that it's a holdable dead zone on the current size pad. And I find myself, sometimes my hands will wander on the screen and, like, trigger something that I don't mean to do. So you won't be able to do that on this pad.

Leo: Well, apparently they've patched iOS to reject accidental touches on the side. Again, no one knows till they try it. Somehow it knows when you want to touch it and don't. Remember, that was a problem with the first Kindle was you'd hold it, and it'd turn the page by accident.

Steve: Oh, my god, yes. It was so annoying because everyone would want to take it from me, but it was all page-turn button. And I'd be like, uh, okay, hold it right over here in order not to lose...

Leo: They fixed that. Amazon did fix that. So, you know, I'm withholding judgment until I actually hold one in my hands. I can't tell if it's going to be worth it. I certainly, I love my Nexus 7. I like the idea of a 7" tablet. I really liked the Galaxy 7 when it came out. So we'll just withhold.

Steve: Yeah. And for me, I guess we have the iPad 1 resolution that's been reduced in size. So the pixels per inch increases, yet the overall resolution is still 1024x768. So it's the same quarter of the resolution of the third-generation and fourth-generation, the newly announced fourth-generation full-size iPad. For me, I think that's still the sweet spot. I like having a screen that's that size. Because, I mean, I use it more than I use any other device. I just love my iPad.

Leo: Have you got your Paperwhite yet?

Steve: Oh, I was just going to take us there.

Leo: Okay. Take us there, baby.

Steve: I really like it. It's interesting because, if you turn the - there was a lot of hype about it. And I was interested to see, for example, if they actually increased the resolution, if they actually increased the contrast. Well, they didn't. They carefully designed the fonts, which absolutely makes total sense to have done that. But if you turn the light off and hold the two right next to each other, showing exactly the same image, I've done A/B comparisons, there's no difference whatsoever.

So I think - and I remember when they went to, like, the DX versus the earlier one. They were saying, oh, yeah, we've done much more contrast. No. Not much more. They keep making the frame around it darker so that the background of the eInk screen looks lighter by comparison. So now the Paperwhite, I mean, don't get me wrong, I love it, but I'm annoyed by bogus claims. And so with no light on, they are identical gray on gray, dark gray on light gray.

But it's spooky. As you bring the illumination up, it has the effect of only lightening the background and not apparently lightening the dark print. And so with that illumination turned up, even just like a quarter of the way or a third of the way, not so that it's a flashlight, but just so that it really does increase the contrast. And so it's super effective.

And in fact I was only talking about it from a lighting standpoint. And I gave Jenny hers the day after it came, and she wrote the next morning, and she said the light is not my favorite part. It's all the other features, the what is it they call it, the "bones" of the book? I think that's the...

Leo: The bones?

Steve: It's something. I can't remember the term. There's, like, they've done something that gives you much more visibility into the structure of the book itself.

Leo: Oh, yeah, yeah, yeah. No. X-Ray [laughing].

Steve: That's right.

Leo: Yeah, I can see where your mind was going with that. X-Ray. Look into the book. They do that with movies and TV, and it's kind of interesting. On the Kindle Fire HD, as you're watching a movie, it will pop up in the controls the name of a cast member on the screen, and you can click it. It will go to Wikipedia. It's an interesting idea.

Steve: Yeah, a nice way of leveraging the medium.

Leo: Knowledge, yeah, exactly.

Steve: Well, again, while going through the mailbag, I ran across a note from Mike Woods in Cheshire, U.K., who was wondering about SpinRite's "double bit flip," which I talked about, I think it was last week. And I thought, since we're doing a Q&A this week, in the spirit of Q&A, I would share this. He said, "Hi, Steve. Thanks for the look under the hood of SpinRite. Your explanation of the way it flips all the bits twice to test the surface was fascinating. I have a question, though. As a satisfied user, I know that, when SpinRite gets to a problem area on a drive, it slows down and can take hours before moving on. How does that work? Is it just flipping the bits continuously until they come back the same as they are sent? Mike."

And so the answer is no. The bit-flipping that I talked about is what SpinRite does to test the surface and, in modern drives, essentially assist the drive itself in recognizing there's a problem. Remember that once upon a time drives were dumb. And so SpinRite had all of the technology in it for relocating defective sectors somewhere else. It understood the file system. It knew how to mark the sector bad in the low-level format to prevent it from ever being used again, even if you reformatted the drive. It understood how to parse everything in the file allocation table and directories and everything so that it was able to essentially dynamically relocate data that had been recovered to somewhere safe and then knit the file system back together with the data moved.

None of that is necessary today because drives have taken over that responsibility. The problem is, having responsibility and executing on that responsibility are two different things. As I have said before, it's only when the drive is asked to read the data that it's able to detect that the data can't be or is difficult to read. And so there's like a gray zone, if you can think of it that way. If it reads with no trouble at all, then the drive's happy. If it's unable to correct the data, then the drive is not happy, and that's when it returns an error saying I can't read the sector. It's only when you're in between that, when there was a problem that required correction, and that the correction was severe enough that the drive starts to get worried that it may not be able to correct it next time, that then the drive is stimulated to relocate the sector to somewhere safe.

So SpinRite's arduous recovery process happens first. And only after it's able to recover the data from the sector does it then invert that twice in order to see whether there's actually a problem on the physical sector, or whether, for example, today, the track densities are so high that, if you bump the drive while you're writing to it, the head will be jerked off center, and so it could write that track a little bit away from center. So there's nothing wrong with that location, it's just that some vibration hit it at the wrong time. So once SpinRite recovers the data, it then does its double bit flip to show the drive whether there's a problem or not. And if there's not a problem, it'll put it back down where it got it. If there is a problem, then the drive will say, ooh, it's not safe to put the data here, and it'll handle the relocation on our behalf. So it's a little complicated, but it all works.

Leo: Pretty impressive. Anyway, enough.

Steve: Okay.

Leo: Enough. Let's get to questions. 10 of them.

Steve: Where were we?

Leo: Where was we? Starting with Question No. 1 from Patrik Petroff in Gothenburg, Sweden. I, by the way, just love our international audience. I just think...

Steve: We have a strong international audience, yes.

Leo: Well, this show especially, I think. Because, as you said, there's nothing like this anywhere in the world.

Steve: Well, and I guess the feedback allows us to get a sense for where our listeners are because otherwise you're just sort of broadcasting to the Internet, and you don't know where everybody is.

Leo: Exactly.

Steve: But we do know.

Leo: You know, I kind of know, too, because during TWiT and many of the other shows, but TWiT especially, we have live audiences. And it has now become my stock question. Okay, who's traveled the farthest? How many of you are outside the U.S.? And there's at least two or three international listeners always in the audience, often from Scandinavia. There was somebody in from Norway yesterday. Patrik Petroff says: Hi, Steve. A small side - by the way, he sounds Russian, doesn't he. A small side note that you might want to share on Security Now! is the fact that DNSCrypt uses - ready for it - ECC, that thing we were talking about last week.

Steve: Elliptic curve crypto.

Leo: That's what I was trying to remember. I keep wanting to say "error correction."

Steve: Me, too.

Leo: From the OpenDNS FAQ, Question 4: "Is this using SSL? What's the crypto and what's the design?" OpenDNS is a service, free and paid service that we talk about a lot, OpenDNS.com. "We are not using SSL. While we make the analogy that DNSCrypt is like SSL in that it wraps all DNS traffic with encryption the same way SSL wraps all HTTP traffic, it's not the crypto library being used. We are using elliptical-curve crypto, in particular the Curve 25519 elliptical curve. The design goals are similar to those described in the DNSCurve forwarder design." Wow. I have to say, I love OpenDNS.

Steve: Well, and this was in my notes last week. And since I don't literally read my notes, I just forgot to mention this. But I saw Patrik's note, and I said, hey, I want to make sure that I remind our listeners that, yes, everything we heard and learned about elliptic curve crypto and the crazy way it works applies to DNSCrypt, which I know we have a huge number of listeners who are OpenDNS fans and users and have been experimenting with DNSCrypt. ECC is perfect for that. And the whole point of DNS is that, as we have often discussed, it uses the UDP Internet protocol rather than TCP. TCP makes sense for establishing a relatively persistent connection. UDP makes sense for a query and response, which is much lower overhead.

With TCP, as we've discussed when we talked about it, there is first the three-way handshake of a SYN, a SYN ACK, and an ACK. Then you've established your low-level connection. Then you negotiate the SSL handshake, which again is multiple packets, while everybody agrees on what they're going to do and so forth. And only after all that can you begin to actually send data. Well, that would be a huge amount of overhead to put into DNS.

So DNSCrypt uses ECC because you're able to use DNS to get the public key of the DNS domain that you're wanting to get authoritative information from, and then make a single query which is encrypted, and receive a single reply packet which is also encrypted. So you get authentication and privacy and super-low packet-level overhead; but also, thanks to the efficiency of elliptic curve crypto, it's low-level processing overhead. So it's just a perfect use of elliptic curve crypto. And I was glad that Patrik reminded me that I forgot to bring that up last week. And I knew that our listeners who are OpenDNS users would find that interesting.

Leo: As are most of us. Certainly I am. I love OpenDNS. It's nice, though, you put your trust in a company, based on whatever research you do, but it's nice to get that reaffirmed from time to time - oh, yeah, they really are state of the art.

Steve: They're doing it right, yes.

Leo: Yeah, I just love that. And of course we did talk abut the fact of using DNSCrypt some time ago. David in Durham, North Carolina wonders about practical fingerprint reader insecurity: Steve, on your last feedback episode you mentioned how UPEK fingerprint-reading software has very weak encryption in the registry for your credentials. I have a Dell Vostro - I always say Vostro.

Steve: Vostro.

Leo: [Italian accent] I don't know why, but it seems like it should be Vostro. I have a Dell Vostro 3550 that has a fingerprint reader on it. I don't use the laptop in a public place very often; but, when I do, or when I take it to work - once in a while I work from home - I've learned how to use the Windows key L sequence - Windows L - when I leave it alone. I'm also doing these things needed to make sure I don't get any virus infection on the computer. With that in mind, how accessible is my registry to those that do not have direct access to my computer, physical access, or without a Trojan or virus? If it's hard to get access to the registry, does it matter that it's not super well encrypted? P.S.: I've just checked with Windows 7. Regedit does require administrative permission to run, even if you're already logged in as the administrator. You have to escalate. I guess I'd better change my settings so that I have to at least swipe my finger to run it. And if that's true about regedit, what, if any, security measures are present for a program to read the registry? I'm thinking none. What do you say, Steve?

Steve: So this is a great point. I mean, this is the difference between - and, well, it's the difference between a theoretical problem and a practical problem, which is why I used the word "practical" fingerprint reader insecurity. The reason this was a concern in the security industry and is potentially a concern for users, or if nothing else you just need to be informed, is Microsoft did the right thing by never storing a decryptable anything, information, in Windows that would allow the reverse-engineering of the username and password. Microsoft doesn't do it.

So the security event that was a concern was the discovery that this UPEK software was not very difficult to get that from, that is, they were storing the username and password, and you could reverse that to acquire them. So except for people using the UPEK fingerprint system, that would not be possible. So it's a different thing to say, okay, so what does that mean? Well, it just sort of says there's a problem. But David's right in questioning, well, okay, but practically, how worried should I be?

And I would say probably not very because the danger would be that something malicious gets into your system and then looks to see whether you are using the UPEK software, goes into the registry, gets the data, and is able to decrypt it. So then the problem is that this malware which you've gotten already gets to know your username and password. Well, it already is running in your computer, so it's achieved most of what it would want to do with your username and password. Yes, there are all kinds of privilege-escalation attacks that it could employ if it wasn't empowered to do that. But hopefully the registry is protected.

He asks about needing privilege to access regedit itself. It turns out that there's extreme granularity of security in the registry. Individual items in the registry can have full-on Windows ACL (Access Control Lists) applied to them. So security is very granular. So it's not just regedit that you need to have access to. It's your logon credentials allow you to see and read and alter that information with a high level of granularity. So it's very well designed at that level.

So anyway, I liked this question because it allowed us to look at, okay, what's the theoretical problem versus the practical problem? Theoretically, the concern was they were storing data that was your username and password that could be extracted and returned to plaintext. Microsoft never does that. So this represented a chink in the armor, essentially. But in order for that to be valuable, something has to get that data. And you have to be in the machine already. So your machine's already compromised, typically, in order for that to be a problem. And that's probably not a big deal.

Leo: I always - that's me, by the way. You're channeling me because I always ask that question. But how much should I worry about this?

Steve: Right.

Leo: That's always my question. Taylor in the Greater Seattle Area - must be his phrase. I can't imagine you saying that.

Steve: That's how he described where he was.

Leo: "I am in Greater Seattle." He's probably in Redmond, Washington; right? How can we stay safe? Hi, I've been listening to your podcasts. I love them. They're a great source of information. They cover a lot of little topics very, very well. But I'm wondering, if you could set up an ideal install, say of, I don't know, Windows 7, what would you do - what would you do, Steve Gibson - to keep it safe?

Personally I'm using Windows 7 64-bit. My daily use account is not an admin account. I have a secondary account for when I need admin privileges with a separate password from my main account. I use Avast! with scanning set to high sensitivity and its behavior shield set to ask me if it should allow a system change instead of deciding on its own. I use Firefox, the 64-bit Waterfox variant, and run it within a sandbox - Avast! offers a sandbox that you can use on demand to paying users - with Adblock Plus and NoScript installed. This guy has double-barred the door.

I do scans every few months with Spybot Search & Destroy, and I make sure not to visit any shady websites. I've also used ShieldsUP! - that's Steve's online service - to make sure I had a 100 percent "True Stealth" rating. And if that were not enough, I also use KeyScrambler to prevent keylogging should something somehow get onto my computer. This guy's worried about something. I don't know what.

These are all free services, minus the manual sandbox in Avast!. While it does have a sandbox in the free version, you can't manually tell it to run in a sandbox. It'll decide based on whether it thinks that thing can be a threat. So he decided to pay for Avast! so he has the ability to tell it, run everything in a sandbox. All this has a very, very minimal impact on system resources and responsiveness, but I'm always searching for ways to harden my security. Do you have any advice for me, or am I good? Oh, wow.

Steve: Well, yes. Taylor, I think you are...

Leo: Think you're good.

Steve: You are wrapped up so tight it's amazing you can get anything done with that machine.

Leo: He may have overdone it, even.

Steve: No, I respect what he's done. I know that we have a lot of listeners who are running similarly. And there are people who could use more security like he's employed, who are unfortunately getting themselves infected all the time because they're not. The only thing that occurs to me in listening to all that is that you might want to crank up User Account Control to one notch above its default in Windows 7. It's easy to find it. You just put "UAC" in the little search term after you click the Start button, where you find stuff. Put "UAC," and there's one item it'll find, and it'll bring you up that slider which is a four-step slider. It's normally set, the default is to the next to the highest. You can knock it up to the highest, if you'd like a little more protection from things altering your system. And that's the only thing I can really - a simple, easy thing to do if you really want to, if a belt and suspenders are not enough, you'd also like to suspend gravity so that your pants can't fall down by themselves. That ought to do it.

Leo: Yeah. It's going a little farther than I'd go.

Steve: Yeah.

Leo: And I should probably be doing all this because I am probably a target of hackers as a public figure, as are you, in ways that somebody who's a private individual might not be.

Steve: Well, there are the active attacks and the passive attacks. And most people are being affected, I mean, yes, spearphishing is a problem. Someone sends you email deliberately designed for you to look like somebody you know, look like a service you're using. I mean, I've gotten some PayPal spam, and I'm a PayPal user. And it's like, okay, wait a minute. And then I look, and I realize, oh, they sent this to the email address in my WHOIS registration instead of what PayPal knows me as. So, yeah, you do need to just be on the lookout.

Leo: Yeah. All right. I have another question for you, Steve, as one might expect. Michael in Europe, and I mentioned this, the Catch-22 of the friends logon authentication on Facebook: When I first heard Scott's suggestion on how to handle the OAUTH authentication spoofing problem that I initially raised a few weeks ago, I really loved the idea of showing the user photos of his Facebook friends.

Incidentally, I just wanted to mention this, Leo Laporte speaking now, they do. They actually do this. It happened to me the other day. It said we have three different ways you can authenticate. You can have a message texted to you. I can't remember what the three were. But one of them was identify friends. And I remembered our conversation. I said, I can't believe this. So I tried it. It's hard because you may not recognize everybody. And you have to recognize a certain number of them to get in. Eventually I just gave up. It's a lot more cumbersome, as you might imagine, to do this. Anyway, just so you know, Facebook does in fact do this.

After thinking about it for a while, I started wondering: To show photos of your Facebook friends, doesn't Facebook first have to know who you are? And if you are currently not logged in, how could Facebook do that safely before you enter your login credentials? Whoops. Well, actually no because, when you try to log - well, I'll explain what's going on with Facebook.

Considering that this is supposed to prevent you from entering your credentials on a malicious site, I think this could only work if Facebook or other OAUTH-supporting services would heavily rely on the tracking of logged-out users. Or am I missing something here? You, I'm sure, are going to say something, Steve. But I should point out that you are giving it your email address and then saying I would like to authenticate as this person.

Steve: Well, and that's just the point, is that who is authenticating whom here? What Michael was talking about is different than what you're talking about. Your experience is you proving to Facebook that you are who you say you are.

Leo: Right. Oh, he was talking about an OAUTH situation. I get it.

Steve: Yes. And remember that we realized there would be a big potential problem, which I really do think we're going to have, with people getting used to the convenience of OAUTH, which is what is the underlying technology that allows that "Log in using your Facebook account" which we're seeing more and more frequently because it's so effective. It's a much lower friction means of authenticating for so many people who are also Facebook users when they're going to some other site where they haven't been before. They don't want to create an account just for that visit. So they authenticate using their Facebook credentials.

And the problem, of course, is that that site takes you to Facebook. It would be very easy for it to take you to a Facebook spoof site, where you then authenticate to Facebook, or you think you are, but you're not. So the idea was, oh, make Facebook show you your friends so that you know it's really Facebook that you're authenticating to, which is the reverse model of what you were explaining, Leo, where you're proving to Facebook you're you by selecting your friends from among a grid of imposters. So, and the point is, Michael's exactly right. It doesn't work to have Facebook show you your friends because we've already covered exploits of this sort, or this "ilk," if I may use that word, where the malicious site would go to Facebook, pick up pictures of your friends, if Facebook was showing you your friends, and then present them to you, so it'd be able to get around using known friends and having Facebook prove that it is Facebook you're authenticating to.

So Michael's right. The suggestion that we entertained a couple weeks ago has a Catch-22 problem. It's funny, too, because I had this itch when we were talking about it. It's like, something doesn't feel right about that. But this is exactly the problem. So, good one, Michael.

Leo: Yeah, that's a good catch, of course. And so that's the different situation, as an OAUTH provider it wouldn't be a good choice.

Steve: Right. In OAUTH you're...

Leo: You know nothing.

Steve: The potential exploit is you authenticate to a spoofed site. So the idea was how to prevent the site being spoofed. In what you were describing, Leo, it's...

Leo: I know it's Facebook because I went there. Unless somebody is in the middle.

Steve: Right. And you're then being asked to prove that you are who you are to Facebook.

Leo: It was a little onerous, I've got to say, after about the eighth picture. And about three of them I go, I have no idea who that - now, maybe that's because I have more friends than I ought to. But it takes pictures, at least the way Facebook's doing it, it says, okay, this person's your friend, and it takes a random picture from their collection, which could be a baby picture. It's not that easy. They probably should take the profile picture would be - but even that, many of my friends don't have themselves in their profile picture.

Steve: Right, right.

Leo: So it's, eh, nice, it was a clever thought.

Steve: Yeah, the problem with OAUTH, I mean in this mode, is very much like the problem we fall into where the advice for preventing it has been don't ever install software you didn't ask for, that is, you didn't go looking for. I love that advice because oftentimes, one way or the other, we're being asked to install something. You go to a site, oh, you know, click this to install this control to get the full benefit of the site or whatever. Or you're somewhere else, it's like, oh, you need to update your Flash viewer in order to watch the movies here. Well, that's the way people get themselves zapped all the time. They didn't go seeking that software.

Similarly, you just made the point, you knew you were at Facebook because you went to Facebook. OAUTH offers the convenience of taking you there. Well, being taken there by a third-party site is very similar to being offered software which you didn't go seeking. So the better way to make OAUTH safe would be to log into Facebook on one page of your browser persistently, stay logged in, then go log in here. You should be able to bounce over to Facebook, where you're already logged in, and then bounce right back. In which case you've short-circuited the possibility of having to be asked to authenticate to Facebook. But again, that sort of blows the whole benefit and the convenience of OAUTH. So that's not going to work, either. These problems don't have easy solutions.

Leo: No, that's why...

Steve: Or maybe any solutions.

Leo: Right, right. It's thorny. Kevin Daudt in The Netherlands - see, another international listener - has been thinking about software whitelisting: I listened to the podcast about whitelisting of software. I thought it in principle a good idea; but I wondered, what's your opinion about who defines the whitelist? Comparing software whitelisting like Apple does in its iOS to firewalls wouldn't be the same as, let's say, Microsoft defining what ports are open on your firewall. Would you lay this power to one company known for excluding software on iOS for other reasons than security? Wouldn't it be better if the user himself could decide which whitelist to use?

Steve: Yeah. I thought this was interesting because it sort of pulls us forward to the next part of the problem. It's easy to say, oh, whitelisting is the future. But then it's like, okay, where does the whitelist come from? How do we decide? If the user is in control, then you might as well not have it because it's like...

Leo: They're not going to do it.

Steve: ...oh, I want to run this software. What do you mean I can't run this software?

Leo: Yeah, in effect you are in control right now. I mean, you decide what sites you go to.

Steve: Yeah, I just...

Leo: How's that working out for you?

Steve: Exactly. I think the way we're going to solve this ultimately is we're going to have systems which are more bulletproof, where, I mean, there are many times I'm seeing things in email on my Windows machine, and I say to myself, oh, I'll have my iPad in a few hours when I'm out. I'm going to open this and poke around then. I mean, I'm saying to myself, I don't trust doing anything on a Windows platform. The iPad is safer, as I am using it. I mean, the fact that it doesn't have Flash or Java, and it's just a little bit of a - it's like having a sandbox. It's a little bit of containment.

Leo: Moving along to Chris Ward in Houston, Texas. He's a little worried about Amazon's eBook ownership policy. I don't know if you saw this story, but it was kind of a shocker to me, too, and I actually tweeted this a couple of days ago. So did Cory Doctorow. I know you've an avid Kindle fan, as am I, Steve, and you read a lot of eBooks. I'm really disturbed about a recent article about Amazon mysteriously deleting a user's account and, more importantly, all the books on their Kindle without warning or reason. He points to an article at BoingBoing, where Cory, I'm sure, posted the details. I'll go there in a second.

Amazon often charges more for eBooks than paperbacks. I don't know if that's relevant anyway. But now it appears you don't even buy them. You are only renting at Amazon's whim. This is extremely disturbing to me, even though I have backups, and I supposed I could strip out the DRM if it came to that, like many other Security Now! listeners. But it seems like this move is a huge step backwards for eBooks. I'm curious about your thoughts. And, from a security point of view, what is the best way for legal eBook owners to protect those eBooks they have rightfully purchased?

Steve: Well, okay, a couple things.

Leo: Let me - can I summarize the story so that people know?

Steve: Yeah, yeah.

Leo: It was a user, I think in the U.K., I'm trying to remember where, who just all of a sudden got her Amazon library deleted without explanation. Actually, no, I'm sorry, she's Norwegian. Her name was Linn. Her access was revoked without warning. Her account was closed. Her Kindle was wiped remotely. And then I read this on a Norwegian blog, as did apparently Cory and others. Now, again, because this Martin Bekkelund and his blog doesn't give us the last name of Linn, we can't verify this. But the emails coming from Amazon said - now, remember, she's Norwegian. "Your Amazon U.K. account has been closed, as it has come to our attention this account is related to a previously blocked account." And we can't tell you any more than that. She asks for more information. They say we can't tell you any more. She asks again. They said no, this is it, we're not going to tell you any more. Don't try to create another account. Bye bye. And she's out of luck.

Now, apparently, according to BoingBoing, there's an update on this story. Her account was mysteriously reactivated after this article was published. But it does raise the question, who owns those books? And if you somehow piss off Amazon, or iTunes, or Audible, and they decide to eliminate your account - now, in the case of Audible they can't erase the Audible tracks. But you can't play them unless you put them on an iPod or something like that. So what do you think?

Steve: Well, I can speak to Amazon and Kindle because I've looked in years past closely at the way it works. And there was something that happened to me where I felt I was being unjustly restricted. I think I had - there was a book that I was reading that had a very low download count. And I hit the download limit, and I had read it over on a PC that I use, as it happens, with the stair climber. And I had reset up the machine, and it still thought that other instance of Kindle on the PC existed, so it had - I've never had much success backing down that download limit when you hit it. Most books I've never had a problem with. I've got, I don't know, 15 Kindles or something. And I don't put things on every Kindle. But anyway, the point is that, out of curiosity, I experimented with Calibre. And, oh, boy, is that effective.

Leo: What is it?

Steve: It just strips the DRM right off, all of it.

Leo: How do you spell it?

Steve: C-a-l-i-b-r-e. And it is a general purpose eBook...

Leo: Oh, Calibre. Interesting pronunciation. Yeah, Calibre.

Steve: Oh, I thought it was Calibre.

Leo: Well, it might be. It's probably a play on "libre," or free.

Steve: That's what I thought.

Leo: But I know about this, and we've always called it Calibre.

Steve: Oh, okay.

Leo: Calibre, I like that, too. You know what, who knows who's right.

Steve: Anyway, so, I mean, it works.

Leo: It's free.

Steve: And I was surprised. And it's a nice eBook archiving system. Now, I ended up removing it from my system because I don't use it. I was just sort of curious, does this really happen, does it work and so forth. In the case of Kindle, it is also a portable storage device. You can plug it into a PC or presumably a Mac, although I never have, and there is a Documents folder, and there's all your eBooks. So it is possible for a user to back up his Kindle or her Kindle onto any other device - the books are relatively small, they're .mobi format - and then restore them. The books are locked to that device unless you go and remove the DRM, which there are tools for doing, and they're effective.

There was also, I think it was a format conversion I needed, where I needed to remove the DRM for some purpose. Anyway, something that I owned, I had purchased, and I was like, okay, I feel like I have a legitimate, you know, I'm not stealing this from anyone, a legitimate reason for doing this. So that technology exists. It is very disturbing to think that it would be possible for someone's previously purchased product ever to be taken from them. This is a problem when everything goes "e," and we've got connected devices, and at the publisher's whim these things can be removed. I mean, we're seeing stories drift out about this happening on iOS devices or Windows 8 and so forth. So at least in the case of the Kindle, you can back up your library and maintain your own copies.

Leo: And should, apparently.

Steve: Yes. Exactly. Very good point. If you are concerned about this, it makes sense to do it, and it's pretty simple.

Leo: I think the issue is also this kind of - first of all, we're used to books. Paper books.

Steve: Yes.

Leo: And nobody who sells me a book can come into my house and take the book.

Steve: Right.

Leo: And say, no, no, we don't like you anymore.

Steve: Look behind me on the video, Leo. You will see walls of books.

Leo: Right. But I should point out that Amazon is not selling you a book. They're giving you a revocable license to read the book.

Steve: Correct.

Leo: So they're not doing anything that they haven't already said to you that they could do if they choose.

Steve: Yup.

Leo: And that's part of the deal. And so there's this disconnect because we think about books, and Amazon's thinking about DRM data that they rented. So I'm sure you're violating the terms of service by removing DRM and backing them up anyway.

Steve: Well, wait, wait. Let's be clear. You can back them up and leave the DRM in place.

Leo: But then you're still screwed because, if you don't have an Amazon account, you know it phones home and validates.

Steve: I don't think it does, Leo.

Leo: Really?

Steve: I think it's locked to the device.

Leo: Oh, okay.

Steve: So if you turned off the radio, then you could restore the book any time you wanted to, to that Kindle, and it would read it just fine.

Leo: That's good. And the problem is, of course, there is a doctrine, which is a defense, by the way, not an offense, of fair use that you have the right to back up data. But it's not a law. And the DMCA does not say that, and et cetera, et cetera. But so there is this disconnect between the old way of doing things and the new way of doing things. I'm not defending it. I think it's something we've got to be very aware of, and I think you're right, people have got to back up. But nobody backs up their Kindle.

Steve: And you mentioned the DMCA. And it just sends a chill down my spine when I think about how security researchers have been blocked from researching crypto by the DMCA. And what condition would we be in if people, for example, were unable to notify Oracle that there were known problems they had found in Java? You know Oracle doesn't like the fact that this has all been made public and that their dirty laundry is flapping in the breeze. They would wish that were not the case. And I just - I hope that we don't legally screw this down any tighter than we have now because we will all suffer, much as we actually do, arguably, from the DMCA.

Leo: And Calibre will convert it from the .mobi format with the DRM to an unprotected EPUB. But I think you have to use a special plugin.

Steve: Yeah. And the format conversation is never really very good. It does what it can. But, yeah.

Leo: Most people are just going to say, hey, I've got this. But I think it's important that they understand that they don't have it. And if they're worried about this - you may not care if you lose all your Honor Harrington novels. You've read them all. I love books. You love books, obviously. And I love having books.

Steve: Yup.

Leo: Maybe, I don't know, it's a really interesting issue.

Steve: I love having eBooks. And at some point a while ago I did dump all of my Kindles over to my system's hard drive, as I'm legally allowed to do. I mean, you just plug the Kindle in, and there they are. Just drag and drop them; and, bang, now they're somewhere out of the Kindle, safe. And at any point in the future I can move them back into that Kindle. Or if some cataclysm occurs that prevents me from doing so, we know the tools are available for removing the DRM under terms that we feel are ethical for us, and we're able to do that.

Leo: I should also point out that there's some question about, if you have a library of records or books, you can give them to your heirs. And apparently that's not legal if you have iTunes music or Kindle books. So there's this whole issue. If you really want to keep something and hand it down, buy the physical media, I guess. Pretty old-fashioned, though.

Steve: Well, and technically, isn't loaning a book to a friend, isn't that a violation of the publisher's rights?

Leo: Well, yeah. And they have this specific little feature that lets you do that in a very limited fashion.

Steve: No, no, I mean a physical book.

Leo: Oh, physical book? Is it?

Steve: Yeah.

Leo: I don't think so. Really? I can't just give you the physical copy? Sure I can.

Steve: I think, you know...

Leo: Otherwise libraries would be out of business. Of course you can. Of course you can. You think that's illegal?

Steve: Remember, libraries have been in trouble before. They've had to fight for those rights.

Leo: I do believe that you are allowed to give a book to somebody that you've read and let them read it. I do believe that is legal.

Steve: You can't read mine. That would be...

Leo: The chatroom is saying that's first-sale doctrine, and of course protected. But the problem is now in this digital world you can't, unless the company lets you, and there are very restrictive means of doing that. It's ridiculous. It's ridiculous. No, but the music industry has tried to kill - remember record resale, used records. They tried and failed. But they don't have to worry about it anymore. They've got DRM. Or not. I don't know.

Christopher Friday in San Diego, California is an unwitting "joiner." Oh, dear, that sounds painful: I opened my Documents folder and noticed something called "Join Me" in the lower left-hand section of the window - it's like "Eat Me," "Drink Me" - where all the externally mounted drives are listed. I have not authorized any app downloads, nor knowingly accepted any meeting software. The most recent thing I can remember doing is going to YouTube and creating a personal Google account so I can look at YouTube videos. I'm very concerned that my computer is now compromised with software designed to let others see what's on my screen. I never knowingly accepted or used "Join Me." Has anyone ever heard of "Join Me" being loaded maliciously on a target computer via social networking in order to gain access to personal information? Thank you. Christopher Friday. Does this ring a bell?

Steve: Yeah. Join.Me is like a very lightweight screen-sharing app from the LogMeIn people.

Leo: Ah.

Steve: And it's Join.Me, so they've used the .me top-level domain. So it's Join.Me. You can go there and get a token, essentially, then email that or tell it over the phone to a friend, who goes there, puts their token in, and they're able to view your screen. So the good news is it's not malicious. Christopher somehow, who knows, maybe he was installing something, and it was one of those, oh, uncheck this box if you don't want Join.Me added to your system. Which it just annoys me that Java still has that checked for loading all of the Google toolbars or whatever it is they're promoting. But I would imagine you can go to Add/Remove Programs and cleanly remove the Join.Me agent from your machine, Christopher. So that's what that's about. And you can get rid of it.

Leo: I'm wondering, it could even be, I suppose, a bookmark he might have dragged to the desktop or something like that. Might not even be anything to worry about. Glenn Hyatt in Philadelphia, PA wonders about "ECC Keys, How Many Numbers?" Steve, thank you for your rundown on elliptic curve cryptography: fascinating, useful. Seems to me the keys must involve more numbers than you describe. The public key is a point on a curve in two dimensions; right? That would be a pair of numbers: x, y. Does the NIST standard offer a convention whereby the public key is a single binary string that is broken into a pair of coordinates?

The private key, well, that's the number of times to add the point of origin to itself, as I understood from your explanation. But isn't the point of origin also a secret? Is that part of the private key in some way? I'm glad he's asking these questions. I don't even understand the question. Thank you for your laudable work teaching all of us about security over the years. Glenn Hyatt. Huh? What?

Steve: Well, okay. I deliberately simplified this so that we had sort of a visual, conceptual, okay, I kind of know how this thing works approach. The math gets immediately deep, and it involves things known to number theorists as "Galois fields" or "finite fields" of prime number size raised to an integer number. And, I mean, it just goes crazy. But so I didn't want to go any further. And I have not looked into it at the level required to implement ECC myself because I haven't needed to implement ECC myself. There's lots of open source software. Anybody who is really curious, just Google "elliptic curve cryptography," and there are more resources available. But I figured, okay, I achieved the goal of sort of giving people a taste for it, which is really all I was trying to do, rather than...

Leo: Thank you.

Steve: ...nail it down to, okay, let's go write one during the half-hour podcast.

Leo: Bless you, Steve.

Steve: Or hour-and-a-half podcast.

Leo: Bless you, bless you, bless you. Terry Richard, Toronto, Ontario, Canada writes about Windows 8: Mayday, Mayday, Mayday! Warning, Will Robinson. I've been a Security Now! listener since Episode 1. Great stuff. Like you, I want to stick with Windows XP until the wheels fall off - maybe they have - XP no longer supported. I have a computer running Windows XP that's running fine, and I really don't need a new machine at this point in time. But I read in yesterday's news an article about Windows 8, and the reader comments that went along with it. The general consensus seems to be this new OS is a disaster. The recommendation is get a Windows 7 machine while you still can. Some say they'll wait for Windows 9, but who knows what this will look like. Actually, I don't even know if there'll be a Windows 9, myself. But that's a conversation for Paul and Mary Jo.

Even though I don't need a new machine now, is it a good idea to get a new one anyway, on the chance that probable Windows OSes in future will be unsuitable for desktop work? I ordered two new computers, but I can cancel if you should say something else.

I also have a comment: As per your suggestion, I read last year the excellent book "Zero Day," re-read it this past week. The situation described in the book, unfathomable to me. That much of our infrastructure is connected to the web, that's incomprehensible.

I have some files which I would not wish anyone to tamper with. I store them on a computer never connected to the 'Net, never has been. If I could take such precautions, why is it that infrastructure computers are connected to the Internet? For the sake of convenience? As an example, nuclear power stations have been around since the '60s, and they weren't connected to the Internet then. They worked. Why connect them now? Well, because they need to download our podcast.

Thank you for the Security Now! podcasts. The knowledge I have gained from them has in all probability protected my computers from Internet risks. Regards, Terry Richard.

Steve: Okay. So clearly I feel more strongly about Windows 8 being a steaming pile of we know what I've described Windows as during an early episode of this podcast.

Leo: But is that due to security issues or just the user interface?

Steve: Oh, I just...

Leo: You don't like the user interface. But we don't know yet if it's not secure, do we?

Steve: I don't like upgrading for the sake...

Leo: You don't like new, yeah.

Steve: Exactly. I don't like new. We know what new is from a security standpoint. It's always bad, just by definition. I have to say, though, that I have softened my position on Windows 7 dramatically. I have been using it a lot recently - not myself, I'm still happily on Windows XP, which has 530 days remaining of support, that is, Service Pack 3. Windows XP SP3, 530 days remaining. So that's still good for another year and a half. But I will definitely move to 7, not Vista, and not 8. So Terry, I really think what people are suggesting is wise. I think 7 is the next place to land. And with any luck, 7 will take me into retirement. So I do not want to move to Metro and what they have done to Windows 8. It's just like, oh, my god. So, yeah, I think 7 is...

Leo: You know what's interesting, the reviews are starting to come in for the Windows RT Tablet. And while I did expect kind of a howl of pain from real users, the reviewers have been very positive. Now, that is not Windows 8. That's the Windows 8 tablets.

Steve: Right, which is...

Leo: ARM-based.

Steve: ...an ARM-based device, right. And just his second point about why things are being connected, unfortunately, it's sad. It's because they can. And as you said, Leo, although it may not be to download this podcast...

Leo: They've got to get eBay and...

Steve: Well, oh, and look, we can manage remotely. What if the alarm goes off, and I'm in bed?

Leo: Terrible idea.

Steve: Oh, god.

Leo: Do not manage nuclear plants remotely, please?

Steve: Yeah.

Leo: If you don't want to be onsite, then don't do the plant at all. Just don't. If you don't feel safe enough to sit next to the core, don't build it in the first place.

Steve: Yeah. So the answer is pure convenience and absolute sheer stupidity.

Leo: No requirement.

Steve: No. My god.

Leo: Jim Schimpf in Derry, PA - is that our last question? Did I actually get through all of them?

Steve: Yeah.

Leo: He's wondering about crypto in NFC. We did a great episode. If you're interested in NFC and how it works, go back a couple episodes. He says it was a great show, too: Just listened to it, shows how far behind I am. Well, that's not that far. That was just a few weeks.

Steve: Yeah.

Leo: Your explanation, very clear, showed me that it's sort of RFID on steroids.

Steve: Yup.

Leo: As you mentioned, encryption is part of the standard, but not used much yet. So how prone is the system to interception? 13MHz is a pretty clear frequency, at least in the daytime. Would an extremely sensitive shortwave receiver pick this up at, let's say, 10 meters? 10 meters away, not 10-meter frequency. The modulation is above the audio, but it's easy to imagine a hacked receiver could demodulate the signal. Thanks again for making my commute a complete pleasure. Jim.

Steve: First of all, I think it's 315MHz, if memory serves, or 335 or something. I don't think it was 13MHz. That seems way low. But yes. Where we need to go, and I hope we go there soon, is to implement crypto in NFC. We absolutely have the technology with public key crypto enabled by elliptic curve crypto, ECC, to perform communications between devices where they are strongly authenticated and absolutely protected against eavesdropping because that's exactly, for example, what we have with SSL. When SSL is working, the whole point is man-in-the-middle attacks cannot work. Eavesdropping attacks cannot work. It is entirely possible for two endpoints, an NFC-enabled card, a smartcard of some sort, and the reader to freely talk back and forth if they've got public key crypto. That's the key. You have to have that. And then it doesn't matter if you broadcast what you're doing or if somebody receives it. It is in the spec. As I have alluded to a couple weeks ago, I've got some things - I don't. I have some information about things that will be happening that are exciting in the near future that we'll be talking about when I'm able to that are going to solve these problems in a cool way.

Leo: Aren't you a sneaky cove. A sly cove, you. Steve Gibson is slyly hanging out at GRC.com, the Gibson Research Corporation. But he's not doing so anonymously. Oh, no. You can go there and ask questions. Go to GRC.com/feedback, and in a couple episodes we'll answer some questions. You can follow him on Twitter, @SGgrc. You can go there and get SpinRite, that would be a good idea, the world's best hard drive maintenance and recovery utility. Gotta have it if you've got a hard drive. But there's lots of free stuff there, security apps and password information. Dietary information, too. Don't forget, we're going to do a special "Up the Sugar Hill, Down the Sugar Hill" Part 3.

Steve: Over the Sugar Hill. Around the Sugar Hill. Through the Sugar Hill. To grandmother's house we go.

Leo: An update on carb-free living with Mr. Gibson, Sunday, 2:00 p.m. Pacific, 5:00 p.m. Eastern on TWiT.tv. And we'll make a Special out of that so you can read the first, listen to the first two in preparation and listen to the third after that.

Steve: And for everybody who has sent me their stories, thank you very much. For those who want to and haven't yet, you can just go to the Health page under Research from the GRC Main Menu, and there is a feedback page. By all means, send me stuff. I'm reading them all. And, boy, Leo, I'll just, I'll tell you, it's really been gratifying to see how many people have, you know, the way the information was presented clicked for them. They understood the science and the why. And, wow, results have just been phenomenal. We'll be talking about it on Sunday.

Leo: Fabulous. Can't wait for that. And then of course we'll be back here on Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern, 1800 UTC for the next edition of Security Now!.

Steve: The Halloween edition.

Leo: Oh. Are you coming in costume?

Steve: Uh, okay.

Leo: I have a costume I'll wear for that show. I just thought of it. Perfect. The perfect outfit for that show. You can always watch live. We like it if you do. But after the fact, too, on-demand versions available in a couple of versions. Now, Steve on his site, GRC.com, has 16Kb audio, which is tinesy, and the even smaller plaintext version. He does transcripts, human-written transcripts, so they're intelligible and spelled correctly and that kind of thing. That's at GRC.com. You can come to TWiT.tv for the higher bandwidth audio and even video, if you want to see Steve's smiling face. Although I don't know why anybody would download the video. People do, though. Like one in five download video of this show.

Steve: Wow. Cool.

Leo: They like to look at you.

Steve: Ah, there's not much to see here.

Leo: It is the kind of show you could just listen to and get pretty much 99.999 percent of it.

Steve: Yeah, and I do, I'm very conscious of the fact that most of our listeners are listeners. And so thank goodness, too. Because, boy, if I had to provide graphics to go along with all this, I'd just never get anything else done.

Leo: I would love it. We'd have to have a full-time illustrator. But that would be great if we did. Next time. By the way, I've just been corrected, the UTC time is wrong because we end our summertime here in the United States on Sunday. So...

Steve: Yay.

Leo: Yay. So I'm going to, instead of adding seven, I'm going to add eight, and it'll be 1900 UTC. I hope I did that right. Is that right? 1900 UTC?

Steve: So we're springing forward and falling back. So we set our clocks back.

Leo: We fall back.

Steve: Ooh, we get an extra hour; right?

Leo: We get an extra hour. But of course UTC never changes because it's enlightened. It's always the same.

Steve: UTC and Arizona. They don't change, either.

Leo: U.S. is November 4th. So, no. Okay. See, that's what I thought.

Steve: Oh. Oh.

Leo: That's what I thought. So now I'm confused. So I was right, it's 1800. We'll figure it out by next week because it's November 4th. That's right, Keith, thank you.

Steve: Okay.

Leo: And that even further confused it because they changed the date. And now I'm really - in fact, still some software does it wrong.

Steve: I know.

Leo: I don't understand.

Steve: And the clocks we have on our bathroom mirrors, Leo. No, they know.

Leo: GMT does not change, but British Standard Time does change. And Universal Coordinated Time never changes.

Steve: No. Although it does die in 2038. That's going to be a problem.

Leo: Yes, because it's on UNIX.

Steve: 32 bits. Yes, baby.

Leo: I hope I'm alive to see that. And then we'll be able to say, "I remember Y2K."

Steve: Okay. You have another podcast to do.

Leo: I'd better go do it.

Steve: I think so.

Leo: I'm just thinking. And that's always a mistake.

Steve: I'll talk to next week.

Leo: Thank you, Steve. We'll see you next time on Security Now!.