SECURITY NOW!

Transcript of Episode #564

Listener Feedback #235

Description: Leo and I catch up with a busy week of security happenings including Symantec's worrisome purchase of Blue Coat Systems, a bad bug in Chrome, more news from the hacker Peace, Let's Encrypt's email glitch, more Microsoft telemetry concerns, some sci-fi updates, and questions and comments from our terrific listeners.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-564.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-564-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. It's another Patch Tuesday, and Steve has the deets. Plus a look at more database hacks and in fact why Peace is in the business of selling passwords. And then we'll answer questions, 10 of them, from you our audience. It's going to be a jam-packed Security Now! coming up next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 564, recorded June 14th, 2016: Your questions, Steve's answers, #235.

It's time for Security Now!, the show where we talk about security, now. Right now. Like exactly now. Steve Gibson is here. He is the man behind GRC.com, the security guru for all of us, the man who discovered spyware, coined the term, and wrote the first antispyware program. And of course he's well known for a lot of different programs, including SpinRite, the world's best hard drive recovery and maintenance utility. And he joins us every week to talk about stuff in the security world. How are you doing, Steve?

Steve Gibson: So, hey.

Leo: Hey.

Steve: Great to be with you again, as always, my friend. Lots of news to talk about, although after two weeks of the IoT Infancy we're going to do a Q&A.

Leo: Woohoo.

Steve: Oh, boy, there's some fun stuff in here, including an amazing utility for Windows users that I was turned on to by one of our listeners that I just - I didn't believe it until I checked it out this morning. So it's like, oh, my god, this works.

Anyway, we've got the news of Symantec's purchase of Blue Coat Systems, which has got a lot of people concerned because these are the people that do the man-in-the-middle SSL/TLS decryption on the fly. There was a bad Chrome bug that we never knew we had. Some more news from the hacker who calls himself Peace that's kind of fun.

Let's Encrypt had a little bit of a mistake with their mass mailing. It's a bug that I just love that we're going to talk about. Another little concern that arose over Microsoft and telemetry in a surprising place. And then I want to briefly talk about some science fiction and sort of amend some snarkiness that I've been self-conscious about, and then get into our Q&A. So a great podcast.

Leo: The snarkiest amendment. The snarkiness amendment. That sounds like fun.

Steve: So two things happened while I was assembling this. And the first is that this is the 14th of the month, which is a Tuesday, meaning that it's the furthest into the month that it's possible to have the second Tuesday. Which of course means patches for Microsoft and Adobe, nominally, because it just dropped. And in fact my Windows 7 machine has been on for an hour, still checking for updates. So I don't know. Anyway, I haven't had a chance to dig into them, but I do know there's a bunch because Brian Krebs at least dropped the news that there were more than 36 security vulnerabilities addressed today. So anyone using Windows...

Leo: Kind of amazing.

Steve: You're going to want to update yourself.

Leo: I'm opening my Windows device right now.

Steve: Now, Adobe normally drops on the second Tuesday. But apparently, when they were getting ready to release their update, they became aware of a critical zero-day that was in use, being exploited actively. So they deferred their June update until later this week. I mean, again, I hope that a problem with Flash no longer actively affects any of our users because, if nothing else, everyone should be using a browser which requires you to touch your nose with your middle finger of one hand while spinning around counterclockwise and then clicking the "Yes, I'm sure I want to launch Adobe" when the X'd window comes up and says, "You really don't want to launch Adobe."

In other words, our listeners, it's hard to imagine anybody at this point could still be caught out by an Adobe Flash problem because we're on the way of saying goodbye to it. HTML5 is a replacement, and it's only legacy at this point.

Leo: Yeah. I still haven't installed it on my Linux box. Probably will never do so.

Steve: Don't need it, yeah. It doesn't run on any of my iOS devices, and I seem to be fine.

Leo: Yeah, right. We've survived somehow for it's now nine years without it, yeah.

Steve: Yeah, there's one really neat site, NutritionData.Self.com. And I'm so bugged because they have these cool little graphics which are Adobe applets, which show the amino acid spread of whatever you're looking up. So it's a big database site, NutritionData.Self.com. Really neat. But unless you've got Flash, you get a much smaller view and much less useful view. Oh, the other thing is a cool triangle where it's got protein, carbohydrate, and fat on the three corners in different colors. And so there's like a spot showing you what the ratio of fat and carbohydrate and protein is for anything you look up, like a Carl's Jr. hamburger or something, I mean, they've got all kinds of stuff like that. But again, without Flash, it's - and I keep waiting for them to get their act together and redo this without Flash.

The other piece of news that just happened this morning that I knew you, Leo, would find interesting, and our listeners, too, is that this morning the court ruled upholding the Net Neutrality rule.

Leo: Yeah, I saw that, yeah. It's good news.

Steve: Yes. And this was a federal appeals court that upheld the effort to make the Internet service providers treat all web traffic, and more broadly Internet traffic, equally. Which of course delivers a major defeat to the cable and telephone companies.

Leo: Who will appeal it, of course, to the Supreme Court. So it's going to be a Supreme Court case.

Steve: Yes, in fact, AT&T immediately announced that it would appeal the ruling, saying it's always expected the issue to be decided by the Supreme Court. So it's like, eh, okay, fine.

Leo: We knew we'd lose.

Steve: Yeah, right.

Leo: We knew this. It's okay.

Steve: Which of course, at this point, with the Court sort of 4-4, missing the ninth deciding vote, at this point in the year it probably wouldn't happen. So essentially it...

Leo: Defers it, yeah.

Steve: ...may end up coming down to who wins the presidential election.

Leo: Yeah. Well, you know what it's going to really come down to, and of course AT&T and everybody, Comcast, are spending a lot of money at this point on Congress members because ultimately, even if it fails in the courts, they expect Congress to - because FCC can only do what it does at the behest of Congress...

Steve: Correct.

Leo: ...to rein in the FCC. And that's the thing we've got to watch out for. I'm glad the courts agree, though.

Steve: So, okay. This is interesting. The news came out, when was it? It was weird, because it was like Sunday. And I thought, what news happens on Sunday? But Symantec purchased Blue Coat Systems for $4.65 billion. Now, Blue Coat are the people that make the carrier-grade - which is to say very high-speed, high-performance, like NSA/FBI kind of stuff, I mean, like those are their customers - SSL/TLS man-in-the-middle gear, which is used by corporations and governments in order to intercept and decrypt otherwise encrypted connection traffic on the Internet.

And what's interesting is that Bain Capital - we all remember Bain Capital from the previous election. That's Mitt Romney's venture capital firm. They had purchased last year for 2.4 billion. So it's nearly doubled in value somehow in the last year. And so I guess Bain is happy. Unfortunately, I was going to say, it'll be the bane of our existence.

So here's the problem. And we talked about Blue Coat and Symantec a couple weeks ago because it was discovered that Blue Coat had a Symantec intermediate certificate which allowed that equipment to mint, on the fly, certs for any other sites that you were visiting. I mean, and that's how these encrypted-communication man-in-the-middle systems work is, in order to function, they have to be able - you have to trust their certificates.

Now, what's completely different is that normally this is on the border of a corporate Intranet. And all the systems running within that corporation have added a certificate so that they trust the certificates being created by that corporate box on the Internet. So it's sort of like, okay, you're using - and we've talked about this often. You're using corporate equipment, the corporate network. You, as an employee of the corporation, you implicitly acknowledge, and maybe explicitly, that your communications will be scanned for security purposes. Maybe that allows them to do content filtering, deep packet inspection as it's often called, and protect the infrastructure of the corporation.

So now what we have, though, is Symantec, which of course previously purchased Verizon - no, I don't mean Verizon. The people I used to buy my certs from before I...

Leo: VeriSign.

Steve: VeriSign.

Leo: It's like Verizon, only just as bad.

Steve: Only different.

Leo: Only different.

Steve: Same location in the alphabet. So, of course, so now we have a certificate authority whom we all trust. I mean, every system on the planet trusts these certificates which Symantec/VeriSign are minting. VeriSign is one of the oldest original certificate authorities. So this is sort of - so this is why this purchase generated some concern is that the manufacturer of the leading connection-decrypting appliance is now owned by the leading certificate authority.

Now, we talked about this a couple weeks ago, and I said, you know, okay, yeah, that seems not good. On the other hand, remember that our browsers are now trusting upwards - I don't remember the number. Was it like 800? I mean, it's like everybody in the world. And certificate authorities that are trusted exist in all major nation states. And it's nave to imagine that none of those, for example, random foreign certificate authorities that we trust because we want to be able to go to websites that are using certificates that they signed, it's nave to imagine that someone in the government can't require them to produce a certificate to install in equipment on their borders to allow them to perform deep packet inspection.

And stepping back from all of this, I sort of found myself thinking, okay, well, what does this mean? What this really means is that connection encryption had a brief, shining moment - and then it died. That is...

Leo: Sigh.

Steve: It's over. For a while we would bring up a connection just to give our username and password in a way that someone passively eavesdropping couldn't see. And even then it wasn't really secure. I mean, that would prevent them from seeing our username and password. But then the site was in a big hurry to switch us back to HTTP with a cookie, which is what kept us then logged in. But the cookie over HTTP could be sniffed. And that's what Firesheep did, which we talked about years ago, was it sort of automated the process of allowing somebody who would just be running Firesheep as an add-on to Firefox to grab all of the logged-on authentication cookies of everyone around them and just click on one to log in as them, to impersonate them.

So then, of course, we had this migration to encryption, where oh, you know, can't have that happening. And processors got faster. Servers got faster. We got SSL accelerators. And it stopped being the case that encryption was a performance and a computational burden on the server. So it started to become ubiquitous. Well, the powers that be don't want ubiquitous encryption. So now there are appliances which will be minting certificates on the fly that will almost certainly not require our clients to have a certificate from the appliance. They will be doing this, and we won't know.

And so the way to prevent this, or at least to be aware of it, is something we've also talked about, and that's certificate pinning, because what cannot be spoofed, no matter how much technology you have, is the fingerprint of the certificate. And this, of course, is the way Google has been catching any instances of their certificates being misused like this, somebody else signing a Google.com cert, because the second Chrome sniffs a Google.com cert that didn't actually come from Google, all hell breaks loose because Chrome has embedded in it the valid fingerprints of the certificates from Google. And if anyone ever tries to go to a Google property and receive a non-authentic Google-signed certificate, the browser immediately alerts Google.

So this is sort of, I mean, that's nice. And of course this is why I built the GRC's fingerprinting pages because GRC offers this service where my connection, which is sitting right on the Level 3 backbone and definitely has no third-party appliance between it, my connection looks at the certificate and shows it to the user, and they're able to compare that with the certificate they have received. And of course they should be the same.

So anyway, what I think this means - and everyone listening to this podcast has sort of watched my evolution over a relatively short span of time, going from thinking that this is a horrible idea that a corporation should be doing this, to then - I guess this is probably some stages of grief that I've gone through because at some point there's resignation, and then somewhere there's acceptance. And then, you know, it's like...

Leo: Did you get to anger and denial, or did we just skip right through that part?

Steve: And at this point I'm at the give up stage.

Leo: Acceptance, yeah, that's it.

Steve: Because, yeah. What this means is we need to plan. And so, again, the listeners of this podcast, it's like, okay, wait, then what do we do? Now what? So things that we've been talking about for years, like TNO, Trust No One, it means that you yourself encrypt something, and you absolutely put no trust in the encrypted channel. And actually, later on in the Q&A is an interesting thing that I've failed to mention about SQRL is that its technology is so strong, you don't need to protect it with encryption. In fact, we had until recently in the spec you could use SQRL://, or QRL://, meaning not secure, and it doesn't care because it's completely safe against a man-in-the-middle eavesdropping attack. It's that good.

Leo: That's cool. That's really cool.

Steve: Yeah, it's very cool. So anyway, Symantec owns Blue Coat. They say it's beefing up their corporate enterprise offerings. And I'm thinking, uh-huh, yeah. And so it'll make it much easier to deploy this technology in a corporation because I'll be very surprised, I mean, there's no law that I'm aware of that says Symantec - well, no, there probably is. There's probably the CA Browser Forum would look down big-time on Symantec simply supplying certificates for that hardware which was going to be implicitly trusted because what that would - or, no, I guess it would, well, it would be Mozilla and Google and so forth. It would be the manufacturers of the clients that implicitly maintain the cache of public signatures of the CAs which are used to verify the certificates. Symantec would be endangering itself and the trust that our clients have in certificates signed by

Symantec/VeriSign.

So anyway, bottom line is I don't think connection encryption is long for this world. I mean, it'll be there, and it'll mean that we're no longer having in-the-clear connections. But there will be little way points throughout the Internet where that traffic is decrypted, somebody snoops on it, and then it's reencrypted, and off it goes again. So it sort of raises the bar, but it's no more like you have any clear reason to firmly believe that end-to-end encryption that is provided by standard off-the-shelf TLS with the public key crypto system, that that's going to be safe. I just - there's no reason to believe that anymore, and lots of reason to say, eh, don't think so. So we'll need to do something for ourselves.

Without any of us knowing it, Google's Chrome browser fixed a frightening bug which existed in its integrated PDF viewer. The guys at the Talos Group of Cisco identified a problem with JPEG 2000 image rendering, which is the image renderer built into Chrome's integrated PDF viewer. They notified Google, and it got fixed. The open JPEG library doesn't have this problem - that's where this code came from - just because the build process prevented this from happening. Google's somewhat different build process is what exposed this problem. And what this meant was, while this existed, if it was known to anyone - and it's not clear that it was ever exploited. But a malicious site could create a PDF which, if it were viewed by Chrome, would execute code in the context of the browser in the user's machine.

And this highlights, I mean, the fact that this did exist, it got found, it was reported, it was patched, and everyone got it, this is now the way this ecosystem has to work. We've acknowledged that, in these very complex systems, no matter whether the code is open source or closed source, no matter how many people have looked at it, there are going to be problems. What we need is exactly this kind of system. We need a system where a researcher finds it, privately reports it, it gets patched, and that patch is automatically pushed out so that it's removed from actual use in the field. And at that point the researcher is able to say, "Look what we found. We know what we're talking about." And users are not put in danger, and the problems are solved. That's the model. That's what everyone has to do who is producing complex systems with this level of connectivity.

And it was controversial in the beginning. I remember when Microsoft first began to do this with Windows. It was like, whoa, you're not patching my system without me deliberately knowing what's going on and inspecting it. And it's like, well, okay, now it's, hey, it's June 14th; 32 bad things have just been found and fixed, so make sure you update your system soon. It's the world we're in now.

Leo: Well, I've just finished updating Windows. It didn't take all that long, what, about 20 minutes.

Steve: Oh, and as a matter of fact, you mentioned it, and I'm looking at - now it did say 11 important updates are available, and one optional, 77MB.

Leo: You know, that optional one you're going to love. It's this thing called Windows 10. You're just going to adore it. But don't worry, you don't have to do anything. We'll take care of it from here.

Steve: Just give it a try. You'll love it.

Leo: You're going to love it.

Steve: Speaking of which, we're at 1,150,000 downloads.

Leo: That's amazing.

Steve: It's slowed down now to only 20,000 new ones a day.

Leo: Oh, my god, that's amazing.

Steve: I meant of Never10, of course.

Leo: Yes.

Steve: Okay. So we talked in the last, actually the last couple weeks about this hacker named Peace, apparently short for Peace of Mind. I'm not sure. Maybe that's tongue-in-cheek.

Leo: It's P-E-A-C-E, so no puns about piece of anything else.

Steve: That's correct.

Leo: It's Peace, baby.

Steve: And of course he came in the news because he was the person offering the additional 117 million LinkedIn account credentials from 2012, which is what we believe tied into Mark Zuckerberg's breached Twitter account and "dadadada." Somebody tweeted me my face during the podcast when you switched over to the dadadada. I was like...

Leo: It was very funny. By the way, I after the show went to HaveIBeenPwned, and yes, my email address anyway showed up on all four. The Adobe breach is a long time ago, the LinkedIn, MySpace, and Tumblr breaches. But this was the thing that blew me away: 359 million MySpace accounts breached. Only 164 million LinkedIn accounts. So, whew.

Steve: Funny, the number I had is 360. And I'm about to get to that.

Leo: Right there, yup. Anyway, just pointing out. Of course I changed my passwords. But just because you're in this database doesn't mean anything; right? Those are old, probably old records, I would think.

Steve: Yes. And so he was back in the news because he is now offering 51 million account credentials obtained from iMesh, which was a peer-to-peer filesharing service started in '09 that just shut down last month. And so these contain email addresses, usernames/passwords, IP addresses, location information, and other information on users. Now, the good news is they were all hashed and salted. The bad news is, unfortunately, they used MD5.

Leo: Oh, well.

Steve: Which produces a short hash and is highly accelerated on many GPU platforms, so it just doesn't pose much of a barrier for anyone wishing to reverse those one-way hashes. All of that data, the 51 million accounts, are currently on sale on the "dark web" for half a bitcoin. And when I saw that half a bitcoin was $335, I went, what? And I went over, and bitcoins are now, like, yeah, 650 bucks. So Mark Thompson is a...

Leo: Happy camper.

Steve: Happily cranking away.

Leo: And no longer has to pay for heat.

Steve: So as a consequence of all of this - well, it is getting into the summertime, so I don't know what that does to this.

Leo: Now he has to pay for cooling. There goes all that bitcoin riches.

Steve: So as a consequence of Peace of Mind being in the news, Wired reached out, the editors at Wired reached out and posted a message in the dark web forum - it's called TheRealDeal marketplace - and said, "Hey, we're Wired magazine. Would you be willing to talk to us?" And they arranged some secure IMing and decided that - oh, and this Peace person said yeah. So we know a lot more. First of all, there is a seller rating system on this RealDeal marketplace, where the seller Peace has a 100% satisfaction rating.

Leo: He does good work, that Peace.

Steve: He does good work. A+++ feedback, and comments like "Follows up with questions and delivers promptly." So when you're looking for 167 million user accounts from LinkedIn, this is the guy you want to go to. So he does have a growing inventory - currently 167 million user accounts from LinkedIn, 360 million from MySpace, 68 million from Tumblr, the 51 we just talked about from iMesh, also 71 million from Twitter. And then also there's a Russian social media site called VK.com. He's got 100 million of those. So we learned a few things during this conversation with Wired.

Leo: All right.

Steve: Okay. So during this conversation, the IM conversation with Wired, the Russian hacker...

Leo: Peace of Mind, piece of whatever.

Steve: Peace of Mind said, well - and so they were asking, like, okay, what's the whole deal here with selling these blocks of stolen, hacked, whatever?

Leo: Are they asking him why he does it?

Steve: Well, sort of like what's the background. And he said: "Well, these breaches were shared between the team and used for our own purposes. During this time, some of the members started selling to other people. The people who we sold to were selective, not random or in public forums and such, but only to people who would use the data for their own purposes and not resell or trade. However, after enough time went by, certain individuals who had obtained the data started to sell it in bulk, $100 for 100,000 accounts, et cetera, to the public. After noticing this, I," writes Peace, "decided to start making a little extra cash to start selling publicly, as well."

So then Wired responded and said, well, "Why didn't the crew want to sell the whole collection earlier?" And Peace explained, "It is not of value if data is made public. We had our own use for it; and other buyers did, as well. In addition, buyers expect this type of data to remain private for as long as possible. There are many databases not made public for that reason and in use for many years to come." So Wired says...

Leo: That's interesting.

Steve: Yeah, isn't that? Wired says, well, "What was your own use for it? How were able to make more by selling the data privately?" And he said: "Well, the main use is for spamming. There is a lot of money to be made there."

Leo: So it's just the email address they're selling.

Steve: Yeah. Well, no, using. They're...

Leo: Oh, using their accounts for spamming.

Steve: Correct.

Leo: Yeah, okay.

Steve: He says: "...as well as in selling to private buyers looking for specific targets. As well," he says, "password reuse, as seen in recent headlines of account takeovers of high-profile people. Many simply don't care," as he writes, "to use different passwords, which allows you to compile lists of Netflix, PayPal, Amazon, et cetera, to sell in bulk." And then he finally said that they generally get somewhere between $15,000 to $10,000 and sometimes down to as few as a couple thousand dollars for a list.

But so really what was interesting was that there's more going on than is revealed publicly. We're notified of major breaches. Obviously one means is when the company recognizes itself that there was an APT, an advanced persistent threat of some sort. The other way is when the evidence demonstrates somehow the data got away from them. But then there's another classification, and that is databases that have been exfiltrated where, as he writes, they're more valuable if that fact is kept secret. Meaning that in targeted attacks people are getting accounts breached, and that's because no one has notified them of a breach. The site may not realize that it's been breached and had all of its passwords taken. So I just thought that was some interesting - and yay for Wired for taking the initiative to do that.

Leo: Yeah. Very interesting. I wondered myself, how does this get used?

Steve: Yeah. So I just love this bug. And I like the way Let's Encrypt handled it. They were completely upfront and dealt with it immediately. They wrote: "On June 11," so that's three days ago, "we started sending an email to all active subscribers who provided an email address" - and I should mention that that's 383,000 subscribers - "informing them of an update to our subscriber agreement. This was done via an automated system which contained a bug that mistakenly prepended" - and the way they first explain it is not the cool part. They say "mistakenly prepended between zero and 7,618 other email addresses to the body of the email. The result was that recipients could see the email addresses of other recipients. The problem was noticed and the system was stopped after 7,618 out of approximately" - and here's the number - "383,000 emails." That is to say 1.9% were sent.

They said: "Each email mistakenly contained" - here it is. I love this. "Each email mistakenly contained the email addresses from the emails sent prior to it, so earlier emails contained fewer addresses than the later ones." So I just love this. So there was a bug where some buffer of email addresses wasn't being zeroed. And the recipient for this email was appended to the existing one in some fashion so that the longer you waited, the more email addresses accumulated in the email that was going out, up until some person, the 7,618th person, received all previous 7,617 email addresses.

So they finished, saying, "We take our relationship with our users very seriously and apologize for the error. We will be doing a thorough postmortem to determine exactly how this happened and how we can prevent something like this from happening again." Oh, that's good. "We will update this incident report with our conclusions. If you received one of these emails, we ask that you not post lists of email addresses publicly."

And, you know, I think that was - they handled it as well as they could. As they say, stuff happens; and they 'fessed up and shut it down as quickly as they could. But I just got a kick out of the idea that, as this thing was rolling along, doing their mass mailing, it was just adding email addresses to the existing ones. Or they said it was in the body of the email, so maybe it was somehow transferring them in. Who knows. But anyway, whoops.

Microsoft raised some concern. It actually first appeared on Slashdot and then got some coverage because a developer using Visual Studio 2015 discovered stuff in his binary that he didn't put there. And it was unfortunately named "telemetry_main_invoke_trigger." So it was like, wait. I wrote some code, and I compiled it with Visual Studio 2015, and in my code is "telemetry_main_invoke_trigger." So everyone of course is a little touchy about Microsoft these days, with all of the telemetry being not only in Windows 10, but also then being ported back into earlier versions of Windows just because Microsoft wants to know how we're doing. So Steve Carroll, who is one of the high-ranking managers of the Visual Studio team, answered everyone's questions regarding what they're calling an "undocumented feature."

He said: "Our intent was benign. Our desire was to build a framework that will help investigate performance problems and improve the quality of our code optimizer should we ever get reports of slowdowns or endemic performance problems in the field. We apologize for raising the suspicion levels even further by not including the source in the Common Runtime for this." And they said: "This was just an oversight on our part."

Okay. Well, I dislike the idea that telemetry code is being embedded in everything that Visual Studio 2015 compiles. But I'm not using Visual Studio 2015, and won't. So for what it's worth, anyone who is, it would be nice if there was an option, if it was opt-in rather than just Microsoft assuming that we all want it and embedding it in binaries. No, thanks.

And I did want to mention that next week's topic is something extremely cool. Intel has a next-generation technology which they're hoping will go a long way to thwarting one of the major weaknesses in the bare metal-style exploitation. We've talked about so-called "return-oriented programming," where return-oriented programming is a technique whereby malicious code can get around the problem that it can't execute its own data by cleverly finding existing code at known locations and using that to achieve its ends.

So, for example, the stack is often where short-term buffers are allocated. And one of the tricks of old was that you would - say like this JPEG 2000 problem. The system would fill a buffer with data from the JPEG image, and then that buffer would be on the stack of the system, and then a mistake in the JPEG interpreter would allow the instruction pointer of the processor to jump into the stack and execute the data as instructions. Well, that got fixed by marking these various segments of memory as executable or not. That was data execution prevention (DEP) that we talked about years ago, where the stack, which is meant to be data, the processor hardware would disallow executing any memory marked as data.

So then what the bad guys did was they said, okay. They looked around in the kernel and found little snippets of code, like at the end of a larger subroutine, just it might only be a few instructions, that would do something useful. And because it was already in the system, and it was actually code, that couldn't be protected with data execution prevention. It had to be executable. So they would jump to that location toward the end of a subroutine and get a little bit of work done. And then the return instruction at the end of the subroutine would jump back to them. Then they would jump somewhere else into some code, probably in the kernel somewhere, and get a little work done.

Okay. So then the way to thwart that was address layout randomization (ASLR), address space layout randomization, where we started scrambling up where things were in memory so they weren't always at the same known location. Problem is, turns out there's a lack, for architectural reasons, there's a lack of granularity in where things can be put. They just can't be put on arbitrary boundaries. They're generally pretty well fixed. And it turns out there have been lots of ways that people have come up with for figuring out where things are in a way that doesn't set off any alarms.

Intel has a response now that solves this problem, potentially. They call it CET, Control-Flow Enforcement Technology. And it is very cool. It will be next week's - get your propeller-head ready, wound up. And I think it'll probably be fully unwound by the time we're done with that podcast next week.

I did want to mention my talking about SQRL, as I'm starting to more, that we're getting near release point. I got an interesting note that evidenced some confusion. Chris M. in Tennessee asks, he says: "Hi, Steve. Love the podcast and proud owner of SpinRite. I have a question about SQRL compatibility." He says: "Will all websites work with it, or will each individual site have to turn something on for it to work? Does it work with applications, or is it solely a website thing?" He said: "For example, will it work with the Dropbox or Google Drive applications on Windows?" And he says: "I got excited when I heard you talk about it, but then I started thinking that maybe I misunderstood what it was."

So, Chris, and anyone else who's wondering, it does require that SQRL be supported by the website that has chosen to offer it for login. The good news is that the credentials it is storing in order to do its job are small, and all of the work, essentially, is done in the client. The server-side just performs a single cryptographic operation, just the signature verification. So the implementation on the server side - and this was all deliberate. We wanted to keep the complexity in the client because we only have to make a few of those. We need my client for Windows that also runs under Wine. And so Linux and Macs can use it. And I'm sure we will get native clients for those platforms. Jeff's had his iOS client running for some time. Ralph had the one under Android, and there's some guys working on their own Android clients.

The point is we need very few clients. But those few clients can run on many, many more servers. So we wanted the server side to be very simple and essentially minimal. And so the idea is that it needs to store only a little bit of data per user and just perform a signature verification is the entire burden on the server. So unlike a password manager, for example, where the password manager manages passwords, and either you remember your password or you use the password manager, but there was no server-side implication, SQRL does have a minimal server-side impact, so servers will need to decide that they want to support SQRL.

The flipside of that is they don't have to protect their databases. We've spent a bunch of time so far talking about Peace and all of these credentials that he's exfiltrated from systems and the vulnerabilities. We're constantly running around having to change our passwords when we find out that some major site has had a breach. All of that goes away. Websites no longer even need to keep those credentials secret as a consequence of the way SQRL works. So it has the potential of solving, not only the problem from the user standpoint, but it completely removes the burden of having to keep these secrets secret over on the server side. And that's huge, another reason that I think we'll probably see some adoption driven.

I wanted to mention, we were talking about Let's Encrypt a minute ago, and the email screw-up that they had. I had an amazing experience Saturday evening, 7:30. I'm sorry, Sunday evening, 7:30. I was, as I mentioned to Leo before, I think before the podcast, I was bringing up a new server for GRC and got to the point where it was all working under HTTP, and I needed to bring up encryption. And I thought, you know, I need a certificate. And I use, as everyone knows, DigiCert's certificates and services. And I like the green bar to show up on iOS devices and in browsers and to get the extra treatment. Let's Encrypt, of course, has been a huge success only providing domain validation, which is the only level of a certificate they offer - not organization validation, that is, OV, nor extended validation, EV certificates.

Anyway, the point is that about maybe two months ago DigiCert contacted GRC and said, hey, it's time for us to reverify your corporation information. And we weren't getting a certificate at the time, and I remember thinking, wow, that's, okay, proactive. And I didn't really get it. Except I was able Sunday night at 7:30, thanks to DigiCert having been proactive, having recent current information about GRC, company ownership, status and all that, everything that you need to do for the extended validation certificate.

As a consequence, I was able to mint my own DV certificate on a Sunday evening and have it in about 10 minutes. I mean, from start to finish, 10 minutes. And I just thought, okay, this is too cool. And so I put a note here in the show notes just to say, yes, Let's Encrypt will give - oh, there's Fred. I forgot to mute this iPad next to me, so it's been making noises during the podcast. I apologize.

So, yes, Let's Encrypt does domain validation. But for anyone who needs and wants a higher level of certification, organization validation or EV, extended validation certs, I've never had - all of my experiences with DigiCert are amazing. And I get feedback from people who have followed me into DigiCert and said their experiences have been similar. So I was just sort of sitting here two nights ago at 7:30 thinking that was amazing. Now this new server that I'm bringing up has an EV cert 10 minutes after I set about making it happen. Wow. How the world has changed for the better.

Maybe six weeks ago or so I feel like I was a little bit snarky about something, and I wanted to correct the record. And this was with regard to the books by Richard Philips, whom we've talked about before. He has a trilogy called the Rho Agenda, R-H-O Agenda trilogy. "Second Ship" is the first one. And in fact he has a website, SecondShip.com, just like a web page placeholder there. And then "Immune" is the second book, and "Wormhole" is the third. Anyway, he then wrote a second trilogy which is actually a prequel to the Rho Agenda trilogy, which explains where Jack and Janet came from. They're two cool, super-secret spy people who we're introduced to in the Rho Agenda trilogy. And so he goes back in time, and he wrote three books there. And now he's started on the sequel to the original trilogy.

And the point is, I just reread all seven of them. There's the original trilogy. I guess I started by reading the Jack and Janet "Once Dead," "Dead Wrong," and "Dead Shift." Then I reread "Second Ship," "Immune," and "Wormhole." Then I read "The Kasari Nexus," which is the first of the next trilogy. And I just had a ball. Leo and I were talking before the podcast about Peter Hamilton. And there's sort of a different style that they have. And what I felt I was a little snarky about was just being, like, as if to say, well, Richard's aren't as good. And I retract that completely. I just had...

Leo: They're different.

Steve: ...so much fun. Yes. They're, for one thing, Peter has never written a short book in his life. And so on the Kindle it shows little dots for where you are in the book. And the Hamilton novels, if you load them in the Kindle, it rescales all of the dotting for all the other books so that they shrink down because Peter's are just all so long. So these are just - they're fun. If you are a Kindle Unlimited subscriber, as I am, they are also all free. So if you're just looking for - if you haven't read them, then I can recommend them without reservation.

I would start with the second trilogy, which is the prequel trilogy, starting with "Once Dead," then "Dead Wrong," then "Dead Shift," just because then you're reading them in proper temporal sequence. And then the Rho Agenda, and then into the next trilogy. The bad news is he isn't finished with the next trilogy. I think we get the second of the final trilogy later this year. So for what it's worth, without reservation, I know that people have read the Rho Agenda series before. But I just found all of them. I just reread them all and really, really enjoyed them.

Leo: Cool.

Steve: And, finally, a note from Dan Hankins in Scottsdale, Arizona, who asked an interesting question, I thought, in the spirit of this being a Q&A episode. He asks: "Does SpinRite mask impending catastrophic failure?" And it's like, oh. Now, remember that we talked a couple weeks ago, someone else, another listener introduced the notion of zombie drives, where he kept saying, you know, drives keep failing, and SpinRite brings them back to life, so they're zombies. They're like life after death.

And so Daniel says: "I run SpinRite in maintenance mode" - or I guess he means for maintenance purpose. There really isn't a maintenance mode. It just fixes whatever it finds. He says: "...in maintenance mode on all my drives a few times a year. Then I had a thought: How am I to know when a drive needs replacing? If SpinRite keeps fixing up a drive that's becoming increasingly marginal, the first warning I'm likely to get is when the frequency and density of errors overwhelms SpinRite's ability to correct them. At what point" - I'm sorry. "At that point, wouldn't I be left with a used-up drive that even SpinRite can't fix? If I'm using SpinRite to keep my drives running smoothly, what should I look for to tell me when it's time to retire a drive?"

I thought that was a great question. So there is a screen that is worth looking at. It's the SMART screen, S-M-A-R-T, Self-Monitoring - I always forget that acronym, S-M-A-R-T.

Leo: Because it's not what you think it is.

Steve: Self-Monitoring something Reporting Technology.

Leo: I always want to say "and recovering," but it stands for...

Steve: Yeah, it might be "and recovering." No, it's not, no. And reporting.

Leo: It's not what you think it is, yeah. It's something.

Steve: It's reporting. I can't remember what the "A" is, though. Analysis. Self-Monitoring Analysis and Reporting Technology, SMART.

Leo: That's it, that's it, yeah.

Steve: Anyway, that's the drive's own publishing of its view of its internal state. And so that's like metadata. That's drive metadata, completely separate from here's a sector, read it, give me that sector back. Or, no, here's a sector. Write it. Give me that sector back. Read it. This is metadata saying how hard it was to do that. Did it have to work? How is its spare sector pool holding out? Those things show, and SpinRite will reveal them to you on that SMART screen.

So, Dan, while you're running SpinRite, you can just rotate through the UI and look at the different screens. Just go look at the SMART screen. I have a page on GRC that completely explains, with little highlights and callouts and bullets and everything, all of the different aspects of that SMART screen so you can figure out exactly what it's showing you and how. And the point is that you're using it in maintenance mode. You're not having any problems with your drives. I would say, I mean, that's the ideal situation. If at some point you do have a problem, then SpinRite will probably fix it.

But at that point then you need to decide, okay, am I keeping this backed up? Would I be inconvenienced if it did suddenly die in a way that SpinRite couldn't recover? So there is sort of a gray zone where you need to think about how much life support do I want to have SpinRite continuing to provide? Because at some point, you're right, if the drive is absolutely positively determined to terminate itself, it will win that battle because ultimately it's the drive's responsibility to return data. And if it just refuses, nothing we can make it do will get it to work.

Leo: And that's just smart. Steve Gibson, it is time for a Q&A. Are you feeling smart today?

Steve: Ready to go, yup.

Leo: Raring to go? Well, you chose the questions. You probably have had time to think about the answers. So I'm going to assume the best. This comes to us, first question, on the Twitter from Scott Norris. He's @scottnorris2012. He says - I love this. Can a six-digit PIN, numeric, ever be safe for online banking, even with other mitigations? What if the database were published, for instance, as we just described?

Steve: I thought it was sort of an interesting question. That is, our first reaction might be to say, oh, no, no. There's not enough entropy in a six-digit PIN. On the other hand, that's one in a million. And so the point that I wanted to make is that the question really comes down to how that PIN entry is managed. That is, there's a one in a million chance for - if we assume the PIN was assigned randomly.

So one concern is, if the user is assigning their own PIN, then there's a problem because people are not good with random stuff. I mean, it's almost - it would be very surprising if the six digits itself were high entropy, if it wasn't something that the user chose thinking, oh, well, this really doesn't matter; right? So it was something easier than six completely chosen at random digits. So one concern is that the actual entropy in user-chosen six-digit PINs is way lower than one in a million. So that's a reason that our multifactor authenticators, like the one-time six-digit authenticators, as we know, that's not really -

they are six digits, but we all remember that the fifth digit, well, the sixth digit, I'm always numbering from zero, so technically.

So one of those digits is incrementing to show, to sort of perform better time synchronization so we don't get a high level of misfires when the clocks are out of sequence. But if six digits are, for example, in an SMS message, where it's not time-based, but it's just "We've texted you this PIN to your phone, what is it," those are chosen at random. Then you've got literally full entropy, one in a million chances of guessing. So the point is that one in a million chances of guessing, if the system also has good lockout, is probably still secure. That is, if it just lets you guess constantly without limit, then we could say, okay, there's just not enough entropy there.

It would be feasible for a million guesses to be produced, just exhausting the whole search space. But if, after one or two mistakes, the system says we're sorry, you're locked out until you then provide some additional level of authentication, yeah. I mean, just saying "six digits" isn't enough. It absolutely depends upon how the six digits are chosen and how the system that authenticates them behaves itself. But I could see something - I wouldn't feel comfortable doing it, but I could argue that, yeah, it could be made safe.

Leo: Fair enough. But of course most of time, why bother? I mean, why not have a stronger password?

Steve: Right.

Leo: Matthew N. Dudek, who is @mndudek on the Twitter: Steve, I have a request for a podcast topic, or at least a Q/A session. Can you review BitTorrent Sync at GetSync.com? I'm interested on how secure it is, but also how it works as far as how it uses the keys and identities and ports and network connections. Is it possible for the same key to be generated? Is the data encrypted as it's synced? And how does the user's identity factor into it? Thanks again for a great show.

Steve: So Matthew, you have asked a question that I have been asked so many times.

Leo: And we've answered.

Steve: And, well, I would dearly love to answer.

Leo: To the best of our ability, yeah, right.

Steve: Right. And they even released a whitepaper about a month ago which says nothing. Since the very beginning, I was in touch with them. I made the mistake of starting up a relationship with their PR flack, and unfortunately that's the best way to describe the guy. Then I started getting spammed with promotional nonsense about how wonderful this is. And I said, you know, tell me how it works. Get somebody who can produce a specification.

Now, apparently the protocol has been reverse-engineered because we did talk about it being available, there being a compatible clone on GitHub somewhere. But what was beautiful about working with Joe Siegrist at LastPass was that Joe said, yeah, here's how the entire system works. I mean, he even produced a web page that had readable JavaScript that demonstrated how the encryption/decryption system worked, answered all the questions, published and produced all the information, which is what allowed me to say, okay, these guys nailed it. This is what I'm using. And I did.

Unfortunately, the BitTorrent guys, for whatever reason, don't feel they need to do that. Clearly, people are using it without any idea how it exactly works in an official document for them. And so my feeling is, sorry, I'm not going to help you guys to promote this if you won't tell me how a secure system functions. Maybe somebody they'll get around to doing that, in which case I'll definitely do a deep dive into it. But at this point they won't tell me.

Leo: It underscores my position all along, which is you want to use, if you're going to rely on something like that, you want to use open source because you want to be able to validate that it does what it says it does, know what technologies it uses, et cetera, et cetera. That's kind of the whole idea.

Steve: LastPass was not open.

Leo: Right.

Steve: But it's open technology.

Leo: Well, you looked at the source; right?

Steve: It's open protocol.

Leo: Right.

Steve: Yeah. So, I mean, open source means that you could, you have the ability to check it. However, we see as many problems at OpenSSL as in anything else.

Leo: Right, right. But it got caught, and it can get fixed, because it's open source so people can review it.

Steve: Right.

Leo: And the problem is, in closed source software, you're just kind of saying, well, I trust you. I use BitTorrent Sync. And I use it to back up my Minecraft server. So it's, like, not a high-security issue.

Steve: Mission-critical, right.

Leo: Mission critical. It's a really convenient way to back - so I take those three folders that are the Minecraft servers, and you get a unique number, as you know, you get a unique number. You share that number with BitTorrent Sync running on another system. And it'll automatically synchronize. And I can do it on multiple systems. So I can have, I mean, in that sense it's very convenient. And so you're not saying don't use it. Just don't assume that it's safe somehow.

Steve: Correct, correct. And relative to open source, while we're here, what I would say is that I think it's the future. That is, in terms of the arc of the industry, we began in a complete, I mean, there was no notion of open source. It was proprietary software with ridiculous license agreements that left the users no recourse despite any problems the systems would have.

The good news is the open systems are continuing to mature and increase their breadth and capability. I just brought up a complete front-to-back open source server with FreeBSD as the underlying Unix OS, Apache 2.4, the MariaDB instead of the MySQL database, and PHP 7, 100%. And, I mean, it's fully functional. And I'm completely happy that I can understand anything that I want to about it. Nothing is hidden. And it was a great experience. So I really do believe, I think the future is in that direction.

Leo: In more ways even than operating systems or servers. I think even on desktops. We're already doing it to a certain degree on Chrome OS. And even Macintosh has - its underlying components are Darwin, which is an open source system. History is on our side, on the side of open and free software, I think.

Steve: Yeah.

Leo: And if you're talking about quality, I think you could say it's every bit as good. In many cases. Not in every case.

Steve: Yeah. I would say polish, it lacks a little of the polish. But it's getting better.

Leo: As people who - as accomplished programmers continue to start contributing to these things, you're going to get better and better stuff.

Steve: Right, right.

Leo: And what's nice is you might have a thousand people working on something. And that's why keeping stuff up to date is in some ways easier. But it also has negatives. Software that's not popular doesn't do very well in the long run because you need a certain number of people supporting it.

Johannes Dankel in Chicago, Illinois wonders if Steve's running a double standard:

Steve, last week you went after Microsoft's claims that Windows 10 is more secure, saying only history can prove a system's security. So what about SQRL? Aren't you doing exactly the same thing by asserting SQRL's security as you have been doing?

Steve: I thought this was an interesting question. And my response is that SQRL is unbelievably simple compared to Windows. There is no - I was going to say there's no person that understands Windows. There's no group, there's no team, there's no building of people up in Redmond that understand all of what Windows is doing. And this is part of the problem is, as we know, complexity is the enemy of security. The more complex these systems are, and the more unknowable they are, you know, how could anyone make an assertion about something they don't understand, that they don't know?

And so SQRL, by comparison, is drop-dead simple. It is a few cryptographic operations. And what's very cool is there's maybe 20 or 30 other people who each individually understand the entire thing. So SQRL is so tiny that an individual can look at it and say, I understand the whole thing. Well, okay, except for maybe the equations of the Identity Lock Protocol that I don't understand, except that I'm able to demonstrate that they're simple equations, and they hold, and it does what it says. But really that's the difference. And you'll also hear me say "as far as we know." As far as we know, blah blah blah. Because that's what any responsible crypto person says is you're always couching your assertions in "To the best of our knowledge, this is what we think." Which is how I always try to address this.

But the beauty of SQRL is that it is so tiny. It is completely knowable and understandable, comprehensible by just one person. And there are many people who understand it completely and as deeply as I do. And that makes it completely separate from Windows, which nobody understands. There's nobody who could possibly encompass, whose knowledge could encompass Windows. It's just too huge.

Leo: Mike in Taiwan shares a chilling story from when he was in Greenville, South Carolina: Steve, I enjoyed your last segment on the insecure baby monitors. It brought to mind something I encountered in the early '90s. I had just purchased a radio scanner. I was scanning through the 49 MHz range. This was around 6:00 p.m. on Friday before Memorial Day Weekend. The scanner locked onto a carrier with the audio of a husband and wife arguing. At first I thought, well, this is just a TV show. But then I recognized the voice of my neighbors about seven doors down.

These neighbors owned - oh, this is amazing - a high-end custom diamond jewelry design store. It was a family-owned, second-generation business, employing other family members, as well. It seemed a sister had left the store earlier in the afternoon to head for the coast for the holiday weekend. My neighbors were getting ready to leave for the coast around 6:30 to join her. But shortly after the sister had left, they received a shipment of $100,000 worth of unmounted diamonds to the store. The problem was the sister had left with the keys to the safe, hence the argument.

The husband was mad because they were going to have to cancel the trip and stay home to babysit the diamonds. The wife said, "No, we can just put the package under the bed and stuff it up against the headboard, and no one will ever know. Even if the house were broken into, no one would look there for anything to steal." So they put the diamonds there and headed off for the coast for the long weekend.

If an unscrupulous person had overheard this, it would have been easy pickings. The house backed up onto some woods. All that person would have had to do in the wee hours of the morning, enter the property through the woods, break in through the back door. They'd be in and out in 30 seconds, gone with $100,000 worth of diamonds, without anyone seeing them.

The range of these 49 MHz monitors was about 700 feet. Flash forward to 2016 and the current crop of Internet-connected baby monitors. Can you imagine the potential consequences if this conversation were broadcast worldwide? So the warning is be aware of what you say and do around these monitors. Also be aware that audio range is not just confined to one room. It could pick up sounds from other rooms, especially if you're a noisy family. Mike, currently living in Taiwan, formerly from Greenville, South Carolina - and, no, not $100,000 richer. That's a funny story. That's great.

Steve: I thought that was great. And I've often spoken on this podcast about the analog cell phone days, when I was playing around with a scanner and listening to people's cellular conversations that they assumed were private and personal. And it was like, oh, goodness.

Leo: Woz admits to doing that. Woz says it was great fun.

Steve: Oh, yeah. I mean, it was - it's like, well, sorry, you're using a radio.

Leo: And I can listen.

Steve: Yeah. Anyway, so of course the flipside, the information had context because Mike knew the voices of these people.

Leo: True. He knew where they lived.

Steve: Exactly. And so as I was thinking about this, I thought, well, okay. If you had the IP address of this - you knew that you heard this conversation, had the IP address, then I was trying to think if there's any way to get the MAC address from that. Maybe if you could localize it closely enough, if you could get the MAC address, then you might be able to drive down the street until you found where the WiFi was, assuming that they had WiFi in the house, and so forth. But anyway, I thought that was just too fun not to share with our listeners, that, yes, you do need to be careful about what gets broadcast out of your home.

Leo: Hysterical. Nicklas Keijser in Stockholm, Sweden reminds us about the Shodan search engine: To put a bit of a fine point on your excellent scary baby monitor coverage last week, if you have a paid account for Shodan, you can search for specific ports which require no authentication. Images.shodan.io, paid version, search for port:554, that's the RTSP port, and you'll be even more frightened. Wow.

If you don't have the paid version, just type "port:554 has_screenshot:true." You'll still find them. Yes, some baby monitors are there, but a lot of other stuff, as well. No one seeing what's there right now would consider putting devices with known security problems online.

Steve: So we haven't talked - we've talked about Shodan a few times, and I've heard you mention it on other TWiT podcasts. I went there; and, boy, has it matured. It is nice-looking. And I don't remember that it used to say "The Search Engine for the Internet of Things." But it says that now. That and refrigerators - it literally says refrigerators and buildings and power plants. They're clearly having fun frightening people with what their machine finds.

And so what Shodan is, think of it as a port scanner for the Internet. It's not just scanning port 22 or 23 or 25 for specific services. It's scanning them all across the 4.3 billion IPv4 space and indexing it. And so it's like, yeah, want to see what's on the real-time streaming protocol, port 554? Sign up, and we'll show you what we found.

Leo: Do a lot of things use RTSP?

Steve: Yeah, apparently a lot of cameras are using them to stream.

Leo: Interesting. Well, well, well.

Steve: Yeah.

Leo: Sean Schwegman in Huntsville, Alabama wonders about password managers: Mr. G, Mr. G, would you offer your opinion on what to look for when choosing an app for storing passwords? I currently use Keeper (KeeperSecurity.com). However, like most people, I have no idea what to look for regarding effective encryption methods and best practices. I can only trust what the developers publish on their websites. Would you please take a moment and talk about what security methods to avoid and/or trust?

Steve: So I thought this was interesting because apparently, I don't know how long Sean has been listening to the podcast, but our longtime podcast listeners all know, as I was mentioning before, that when I looked at password managers years ago, I was searching for information. I want to know how they work. And it was the founder and designer of LastPass whom I was able to get the best answers from. And so I chose them.

I don't know anything about Keeper Security. But, for example, I just went to KeeperSecurity.com. They have business and personal versions, or offerings. And under the Keeper Free, as they call it, they offer the features of local password storage, single-device, and email support. And then, if you pay $30 per year, you get unlimited password storage. So I guess that means that there's a limit to your password storage for the free version. Unlimited devices and sync, and so we know that the free one is a single device without sync. Unlimited secure cloud backup, so we know that the free one doesn't do cloud backup. And unlimited secure record sharing, so we know the free one doesn't have that; and fingerprint login, so the free one doesn't have that; web app and 24/7 support.

So just in terms of - so I know nothing about the crypto level because none of that is available. None of that is shown. But what I do know is that LastPass, my chosen password manager, gives you all of that stuff that you're paying them $30 a year for in the free version. So, and a complete disclosure of the technology, cloud syncing, multiple devices, runs on all the platforms, yada yada yada. So there's been some concern that LastPass was purchased by - do you remember who, Leo? I'm blanking on...

Leo: Yeah, LogMeIn.

Steve: That's right, by LogMeIn. We were a little concerned that big fish was buying little fish.

Leo: Nothing bad has happened yet.

Steve: Yeah. And I don't expect it. I mean, I think Joe, you know, Joe is still at the helm. And the acquisition gives him more resources. So LastPass is my choice. And Sean, the problem is, unless companies completely open their kimono and show us how their stuff is working, like BitTorrent, for example, with BitTorrent Sync, if they want to keep it proprietary, they can. But people who care about the details are then unable to audit what they're doing and recommend it. So I can't recommend Keeper Security. And frankly, just what they're offering is not comparable to what LastPass offers.

Leo: There is an open source solution called KeePass that a lot of people like. And you could at least verify those details for yourself on that. The interesting, I mean, if you're asking what attributes to look for, one of the things you and I have gone back and forth over is the tradeoff between convenience and security. LastPass stores your password database on their servers, which is of course inherently risky. Not a big risk; but if their servers were compromised, and somebody got all the databases, they'd be able to try to crack them, brute-force them at their leisure because they'd have them. And that's a problem with a password vault. It's a single point of failure, all your passwords in one blob.

And so there are other password programs. I think Keeper might be one that offers this, that allow you to deal with the details of syncing yourself. You don't have to put your blob on their servers. I know 1Password does it that way. And so for some people that's more desirable. 1Password has other issues, which we've talked about.

Steve: Yeah.

Leo: So it's hard, I think, for an end-user to know what to do. I mean, you could look at - anybody could say AES-256. But that's only a small part of the overall equation.

Larry Hamid in Ottawa, Canada wonders about trusting biometrics: Steve, I was wondering if you had any thoughts or would consider doing a small segment on biometrics. In particular, we've seen many biometric products and capabilities getting a lot of attention lately. These include behavioral biometrics, like the way you hold your phone or the way you swipe, et cetera. Others are wearables that monitor your EKG, your heartbeat.

Some years ago I was in the biometrics industry. Back then we were bombarded with legitimate concerns and challenges regarding the accuracy of the biometric. Have things improved? To me, it seems strange that this sort of dialog has been lost, as if it really isn't that important anymore. I'd like to know how the accuracy of a fingerprint system, for example, compares with some of the smartphone-based behavioral schemes that are being proposed. That's a good question.

Steve: So we've talked about all kinds of biometrics in the past. I've mentioned how getting into the datacenter at Level 3 required me to put my hand on a reader or a scanner in order to verify the physical properties of my hand, and then also enter a short PIN. So there was two-factor. And of course inside I'm under camera surveillance, and all of the individual facilities are locked, so there's a lot of security there. And of course we were talking about fingerprint readers and the Apple products, for example, and now being made available over on Android devices also.

However, what we often see is that these things can be spoofed. There's been some concern, for example, about this new facial recognition which has become popular. But until that technology is actually able to do a 3D recognition to see that it's actually a curved face, what's been demonstrated is you just hold up a picture of the person and it unlocks because it can't tell the difference between a picture and the real person. There just isn't the sophistication there.

Similarly, we keep running across, no matter what kind of fingerprint technology it is, people invariably create gummi bear fingerprint copies of various sorts and fool the fingerprint sensors. So something that's even weaker than that, like how am I holding my phone, or the specific timing of the keystrokes as I enter my password, those we would call "heuristics," sort of seat-of-the-pants rules of thumb or weaker signals.

Stepping back from all of this, I just keep coming back to something you know. To me, something you know seems like the ultimate better way to secure. But the tradeoff, for example, that Apple has made, where you have a fingerprint that you're able to use under certain circumstances - for example, you can't use it if you haven't logged in for two days because they're wanting to protect the user. So there's a set of tradeoffs to lessen the vulnerability presented by the weaker signal that I think all biometrics probably is. And if that fails, then you fall back to something you know in order to say, yes, this really is me.

So, and I don't see this as a technology problem. That is, I don't see it getting better in the future, except maybe the more parameters of the person you took, for example, a fingerprint and a retina scan, you sort of up the ante in order to create a composite signal from more biometrics. And then, if you really care about security, also something you know, I think, is probably the way to go. But biometrics, I mean, this is the classic tradeoff of convenience and security. Biometrics are, almost across the board, implemented for convenience, not for security.

Leo: There you go. And it's always that tradeoff.

Steve: Yeah.

Leo: Let's see. Moving on. Steve in Hong Kong, the one with the famous post office - I take it he wrote that - voices a SQRL worry: Steve, in SN-563 you noted that SQRL allows an identity on a site to be reset without authentication, and access only then granted again by the use of a master key. Isn't this a potential attack mechanism on SQRL itself, whereby users could be deauthenticated en masse? Further, a man-in-the-middle attack could trigger a reset when it spots SQRL being used for authentication, and then simply sit and wait for the master key to be entered. Just thinking aloud. Thanks for all the great podcasts over the years.

Steve: So two things. First of all, I wanted to make sure that I corrected the record. It's not possible for just anyone to go and disable the SQRL identity. He's using the word "reset," but he's talking about what I was talking about where you're able to disable the identity if you believe that your SQRL identity may have been compromised. But you use your SQRL client to perform the disable. So not anybody can just disable anyone else's SQRL identity. You have to still use SQRL to tell the site, "Hi, it's me, but I want to now disable future recognition of me." And so that prevents any sort of a denial of service of this sort on the system.

And then at that point is when you need the higher level of authentication provided by the so-called "rescue code" in order to say, okay, now I've got my rescue code. I want to reenable access, and I'm also going to rekey the identity to solve this problem of being concerned that the identity may have escaped from me.

And then the other thing he brought up is this man-in-the-middle attack. And I will say again that one of the coolest things about SQRL is it is so safe to use that you can do it over a nonencrypted connection. Until only maybe a few months ago, this was an argument over in the newsgroup where all of this has been debated, is whether or not we ought to formally allow nonsecure use of SQRL. And I finally sort of unilaterally said no. Yes, it is secure to do that, but why? We're in a world with Let's Encrypt, where we can get domain validation certs for free. Encryption is the future.

I was worried that people wouldn't understand that SQRL is so strong that it could be exposed to nonencrypted communications without in any way reducing the integrity of the protocol. And so I thought, there's just no - I understood the argument that there might be places for nonencrypted use. But it's 100% encrypted on top of the existing security of the protocol. So a little bit of belt and suspenders.

Leo: Nothing wrong with that. John in Montreal wonders whether a human brain really is better: Perhaps you, as I, have experienced the horrible feeling that can overcome you when stopped at a traffic light and a car in your peripheral vision moves backwards. Admittedly, this does not happen often; but when it does, your instinctive reaction is to assume that your vehicle is creeping forward. You step on the brake with enough force to break it and nothing happens. Panic sets in momentarily until you realize what is going on. The trouble is that, during this period of confusion, really bad things can happen. It is what cost Kennedy his life while piloting his small plane. Who?

Steve: I wasn't sure. I thought maybe you as the great repository of knowledge would know what Kennedy...

Leo: I don't know. I don't think - I don't know.

Steve: ...mispiloted his plane. But this was interesting. First of all, I loved John mentioning this because this has happened. I assume it's happened to all of us, where for some inexplicable reason the car next to you goes backwards.

Leo: John Jr. It is actually what happened to John John. That's right. John Kennedy Jr. I forgot, yeah, that's right.

Steve: Whoops. Anyway, so I've had that happen to me, where a car next to me, in my peripheral vision, moves backwards. And your assumption is you're moving forwards. And so it's a bit of a jolting moment. Anyway, I liked this because I wanted to follow up. John was of course talking about following up on the self-driving car discussion we had last week. And I've been thinking about it. And what occurred to me is that the advantage the car has - and I can't believe I'm saying this because I can't believe cars are driving themselves. It just seems too soon to me.

Leo: Too soon.

Steve: But cars have the advantage of 100% vigilance. They're not going to get distracted. They're not going to be sleepy. They're not going to be intoxicated. They're not going to be arguing with somebody else in the car, or turning around halfway to make sure that the kids in the backseat are okay. And I just think, okay, that's a substantial advantage, that the car is 100% irrevocably, permanently focused on its job, and it allows the people inside not to have to be so concerned. So, yeah. Again, it just seems too soon. But, wow, that 100% vigilance factor I think is a significant benefit over people that are just inherently not 100% vigilant.

Leo: One last question before we wrap this up. And it comes from...

Steve: Oh, and this is the Incredible Tip of the Year.

Leo: Of the year. It's from Jon Borgen. He's local. He's in Vacaville, California with an incredible tip for Windows administrators: Steve, I just finished the podcast from yesterday. You had mentioned you wanted to perform a clean install of Windows 7 on your new Carbon X1 from Toshiba - or Lenovo. Says that you didn't have the drivers as Toshiba, but I think he means Lenovo.

Steve: Oh, yeah, he does say that, yeah. He meant Lenovo.

Leo: There's a terrific piece of freeware called Double Driver that I've been using for years. It's made things like a refresh install so easy. You can find the software at BooZet - not an auspicious name, B-O-O-Z-E-T -BooZet.org/dd.htm. With it, you can view, backup, oh, backup and restore any and all drivers on your machine.

Steve: It's unbelievable. But keep going.

Leo: When you first run it and do a scan, it'll show you all drivers it detected - even printer drivers - and will automatically check the box next to everything that's not from Microsoft, although you can quickly "check all boxes" if you wish. The default settings have always grabbed everything I needed, and I have never had to back up any of Microsoft's drivers. After that, you can back up to either a single compressed zip file or a folder hierarchy containing all of your drivers. I've experienced some issues in the past using the built-in compressed zip function, so I just usually back up the drivers to a folder and then manually compress the main folder if needed.

In addition to daily backups of my machine, I also keep an offline compressed zip containing all my drivers, just in case. With the drivers backed up, you could either then manually install them one by one in Device Manager or use Double Driver again to batch-install them all for you, which is what I usually do. The only issue I've run into is getting the drivers back onto the freshly formatted disk, as I've often been without USB drivers, so I can't use a thumb drive. If the machine has one, the CD/DVD drive will usually work; but then you need to burn a disc with the drivers on it, and you also have to include the Double Driver utility. I've forgotten to do that once or twice.

I believe the X1 Carbon, though, doesn't have a CD drive anyway. In this case, before reformatting the disk, I would create a new partition from within Windows and put both the driver backup folder and the Double Driver utility on it, as well as any other files or folders you want to keep, and then reinstall the OS in the original C partition, so you can then access the driver backup natively from within Windows once you're back up and going. I don't know if you were still looking to reformat that X1 Carbon or not; but, if so, I hope this finds you well. Thanks for everything you do. And you've got a link.

Steve: Well, okay, so here's the deal. First of all, when I started reading this, I'm thinking, eh, what's it going to do? Like copy the sys file or the something somewhere. This utility is amazing. I don't even know how it does what it does. But it recreates the original set of files - the INF, all of the whole file set. Because a Windows driver is not just one file. There's a whole bunch of other registry gobbledygook and settings, and as I said, INF files. There's a .CAT file and other stuff. It finds it all and pulls it all back together. So it's like what the manufacturer originally had, which allows Windows to install it.

Now, the bad news is, not only is BooZet.org a somewhat worrisome domain, but neither of the download links work. The first link goes to Dropbox, with Dropbox complaining that there have been too many uses of the link, and so the account is shut down. And the other link went somewhere else. I can't remember now where. It went to some strange page that didn't seem related. So I thought, okay. I have to find this. So I went looking, and I found three copies of it. And I was very careful because I'm not wanting to download malware into my machine, certainly not to recommend or cause anyone else to do so. I found it on three different download sites. And I took hashes of the three zips, and they're identical.

And one of the sites is one of Mark Thompson's very good friends, SnapFiles.com, which is a website I know through Mark, the guy who runs it. He scans everything. It's 100% aboveboard, and I would trust it completely. So what I did for the podcast for all of the people listening to this that this sounds interesting, is I grabbed the file from SnapFiles. You could certainly go there and get it. But I put it on GRC.com only because - and I read the license agreement. I'm not prohibited from doing that. It is freeware, and it's freely available and copyable. And I made it as easy as possible: GRC.com/doubledriver.zip. GRC.com/doubledriver.zip.

And again, I was just - I'm astounded by this thing. This is - I wish I knew about this years ago. But anybody who juggles Windows systems around who doesn't know about this, check it out. It's very nicely written, brings up a clean list box of all the drivers in your system, as he said, with the ones that are not Microsoft checked. There are options for "uncheck all," "check all," "check only Microsoft." You can back the whole thing up into a directory hierarchy, which you then zip. And then it adds its own little INI, which is an information file for it, which is what it uses if you want it to do the driver reinstall.

Anyway, I was so impressed: GRC.com/doubledriver.zip. I inspected the files, looked at them carefully. And as I said, multiple download sites have the exact identical zip. So I'm as comfortable as I could be without having downloaded it from the site of origin, because you can't, that this thing is legitimate and is free of any problems. And I know that the place I got it from, SnapFiles, aside from it being absolute bit-identical to several other locations on the 'Net, this guy really checks the stuff over as well as - better than I would. So there you go.

Leo: I'd have to guess it's abandoned, where the program hasn't been updated since August 2010. And the two things he links to, as you mentioned, one is a Dropbox, the other one is Ubuntu One, which is a service that was discontinued some years ago.

Steve: That's what it was, right.

Leo: So I think that he's probably either passed or just completely lost interest.

Steve: And the site, when I saw - he uses jpegs, and it came up blurry and then slowly got clear. And so it was doing incremental rendering on a low-bandwidth connection. And I thought, wow, I haven't seen that for a long time.

Leo: I think this has been in the closet, this server's been in the closet for a while.

Steve: Uh-huh.

Leo: And maybe he got a job, I don't know.

Steve: I know that there are people who manage Windows, I know that Simon Zerafa, for example, will just be jumping up and down. This thing is just - it's astonishing.

Leo: Yeah, pretty cool. Pretty cool. Steve, we've come to the end of the show without error. At least as far as we know. But it there were errors, we'll let you know next week. That's why you want to be here every week. No, not for that. You want to be here every week because there's always great stuff. You can find Steve's SpinRite, the world's best hard drive maintenance and recovery utility, along with SQRL and all the other cool things he does, at his site, GRC.com. He also has the audio version of the show and human-written, readable transcripts so you can read as you listen, or just read. And search, great for search, too. All of that, GRC.com.

We have audio and video of every show, of course, at our website, TWiT.tv/sn. And let's see, what else? You can find it everywhere because it's been around for a few years. Just subscribe to the show on your favorite podcatcher, or use one of those great TWiT apps on every platform. And make sure you come back each and every week. You wouldn't want to miss it. We do the show Tuesdays, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. And you can join us in the chatroom, too, at IRC.TWiT.tv.

Steve: And next week have your propeller beanies handy.

Leo: Oh, what are we doing?

Steve: Because we're going to do that cool look at Intel's solution to the problem of hackers being able to execute code on your system with this CET technology that further enforces and locks down what can be done when something malicious gets into your system and is trying to execute code that was never intended to be run that way. That's Intel upping the bar in another very cool fashion. So I guarantee everybody will have fun getting down at the bare metal level.

Leo: Very cool. Thank you, Steve. We'll see you next week. Bare metal level...

Steve: Thanks, my friend.

Leo: ...on Security Now!.

Steve: Bye.