SECURITY NOW!

Transcript of Episode #445

Listener Feedback #184

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-445.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-445-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We've got questions. We've got answers. And Steve makes a shocking admission: He's not worried about upgrading Windows XP. He thinks he's safe. Find out why, next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 445, recorded March 4th, 2014: Your questions, Steve's answers, #184.

It's time for Security Now!, the show that covers you and your security online, your loved ones, your privacy, and all of that, too, with this guy here, the Explainer in Chief, Mr. Steven "Tiberius" Gibson of the Gibson Research Corporation.

Steve Gibson: [Laughing] You know that "Tiberius" has made it onto my Wikipedia page?

Leo: I know. I'm thrilled. I'm thrilled. Hey, Steve, it's good to see you once again.

Steve: Likewise, Leo. Great to be with you again, as always.

Leo: Q&A episode this week.

Steve: We do, despite the fact that Apple has given us the document we've been dreaming of, and our listeners have been dreaming of, I've been dreaming of, we've all been, but I just can't - I wanted to give it its own podcast, and we haven't done a Q&A

for so long that we're going to cover that all the entire next week's podcast. And I'm praying that nothing happens. Let's ask the hackers, please...

Leo: Leave off. Give us a week, please.

Steve: Don't do anything dramatic. We have a relatively light news week this week, which is good. I want another one for the forthcoming week, so that we won't - so we'll have time to do this fabulous Apple document justice, which will be the topic for next week.

Leo: And we talked about it on MacBreak Weekly just last hour, and I was really hoping that you would do this because Apple, first of all, I'm not qualified to judge the contents of it, and I'm really curious if what Apple says makes sense. And then they used some terms that were new to me, like "tangled."

Steve: Well, and Apple is less forthcoming, unfortunately, about this stuff. I mean, for example, there's a huge grumble in the industry that there's still no information from Apple about exactly how that extra goto fail line got stuck into the code. Certainly they have auditing tools. They know what session added the code. I mean, they have to know, with modern source code management technology, they know how that happened. Yet nothing. So...

Leo: Yeah, I'd really like to hear more from Apple on that.

Steve: And wasn't there just some shareholder deal that happened where they voted not to adopt industry best practices for protecting their users' data?

Leo: Oh, I didn't see that.

Steve: I saw it go by, but I didn't have a chance to track it down. Apparently Woz is onboard, but something was done with proxies, and they voted not to adopt what the EFF and others have recommended for protecting their consumers. So it's like, okay. And then, in fact, we're going to lead the show with Bruce Schneier's commentary on the Apple SSL.

Leo: Oh, I can't wait to hear what Bruce says.

Steve: Bruce weighed in. We've got another major certificate mistake was found and fixed, but it's widespread. We've lost another bitcoin exchange that just declared itself bankrupt.

Leo: Really, another one.

Steve: Another one. We've got a Peeping Tom problem. This is old enough, this was late last week that it was the - you probably know about it, the Yahoo! webcam catastrophe from the U.K. A mistake was found in Threema's UI. And I've got some more news about SQRL. We're going to internationalize it and make it multilingual.

Leo: I want to thank you for TextSecure because that was Moxie Marlinspike's secure SMS replacement for Android.

Steve: Right.

Leo: I've been using it all week, and it really has just replaced my standard text tool. So it's a very good SMS tool, and it has this additional encryption availability.

Steve: There was some feedback I heard about it having problems in international contexts.

Leo: That might be, yeah.

Steve: But if it doesn't affect you, then...

Leo: All of that is fixable. I mean, that's just code. And apparently he's working hard on iOS and Windows Phone versions and so forth. I think that has the best chance of being a great universal secure SMS system, SMS over Internet. So Bruce Schneier, what does he say about goto fail, hmm?

Steve: Oh, well. So this is last Thursday Bruce blogged, and then also added an update on the same day, also on Thursday, February 27th. So the title was "Was the iOS SSL Flaw Deliberate? Last October," writes Bruce...

Leo: Wow.

Steve: I know, "...I speculated on the best ways to go about designing and implementing a software backdoor. I suggested three characteristics of a good backdoor: low chance of discovery, high deniability if discovered, and minimal conspiracy to implement.

"The critical iOS vulnerability that Apple patched last week is an excellent example. Look at the code. What caused the vulnerability is a single line of code, a second 'goto fail;' statement. Since that statement isn't a conditional, it causes the whole procedure to terminate. The flaw is subtle and hard to spot while scanning the code. It's easy to imagine how this could have happened by error, and it would have been trivially easy for one person to add the vulnerability. Was this done on purpose? I have no idea," writes Bruce. "But if I wanted to do something like this on purpose, this is exactly how I would do it."

Leo: That's fair. That's fair.

Steve: Yeah. And he edited to add after the blog was first posted, he said: "If the Apple auditing system is any good, they would be able to trace this errant goto line, not just to the source-code check-in details, but to the specific login that made the change. And they would quickly know whether this was just an error, or a deliberate change by a bad actor. Does anyone know what's going on inside Apple?"

Leo: Yeah, and why aren't they talking to us?

Steve: Yeah. Again, that's why their publication of this substantial security disclosure document, which is the topic of next week's podcast, it's so refreshing to get that. I mean, I remember, even the iOS 7.0.6 update that fixed this flaw that we talked about last week, the goto fail flaw, and that Bruce just blogged about, even that just sort of - there was just, like, nothing in the description. It said...

Leo: Well, that's typical, though.

Steve: ...fixes an SSL problem. Oh, well, isn't that nice.

Leo: Yeah, this is completely how they are. And I don't real agree with their way of doing things, but that is kind of how they do them. They've never been very forthright.

Steve: Right. They're behaving more like a consumer product company than a computer company. And so I can understand that they're, I mean, they dropped the word "computer" from their name. They see themselves as a consumer product company. Yet they're selling computers. I mean, they're selling high-tech gadgets that have this kind of vulnerability. Your can opener is a consumer product, and it doesn't have a problem with SSL vulnerabilities. But iOS devices do. So they're sort of straddling, and it would be nice if they were more forthcoming. I think we'll see how they evolve in the future. But this document, as I said, again, does represent a much-needed disclosure.

Leo: I would like to see a similar document explaining their code testing and validation process. It begs to be explained now because that error is such a ridiculous error that it should have been caught in many, many ways. And the fact that it didn't get caught really puzzles me.

Steve: Well, and Bruce doesn't note here, but others have, I keep reading it, that it was just a month after this was introduced into the code base for Mac OS X and iOS 6 that the NSA slides that Edward Snowden disclosed indicate that Apple joined PRISM. This is like, okay. Well, again, we don't know. But, ooh, does that timing look painful.

Leo: Well, and if that's the truth, that would explain why Apple doesn't say anything. And so the longer they go without explaining this...

Steve: Oh, and Leo, can you imagine, I mean, we know what the install base of iPhones is globally. Can you imagine on some level the pressure they must be under by the NSA to make surveillance possible? I mean, we now know there is pressure. At some level there is pressure. So I just - and, for example, this is why is suspended my work years ago on CryptoLink, because it was clear this was coming, and I didn't want to be in that kind of pressure. I mean, we know what Ladar Levison went through, and we've since heard evidence of other companies being pressured to make this information available. Apple has to be a target of that, given, I mean, everybody I see is holding their iPhone.

Leo: Yeah. Yeah, yeah, yeah. Yeah.

Steve: So it turns out, however, it's not just Apple who has certificate verification problems. Just, I mean, this is hot news. This just happened. There is an alternative SSL/TLS open source package, GnuTLS. And everyone talks about OpenSSL as, like, sort of the industry standard, that's the benchmark against which you test things to make sure that you're running, and for good reason. OpenSSL is very mature. The problem with OpenSSL is that its license is not GPL compatible. So if you want to do GPL compatibility, you need to use GnuTLS, which is the alternative.

So right now on the GnuTLS.org site, on their security.html page, where they list known problems that they have encountered and fixed, fresh, top of the page says: "A vulnerability was discovered that affects the certificate verification functions" - whoops - "of all GnuTLS versions. A specially crafted certificate could bypass certificate validation checks. The vulnerability was discovered during an audit of GnuTLS for Red Hat." And Red Hat has this linked also from their site. They asked themselves the question: "Who is affected by this attack? Anyone using certificate authentication for any version of GnuTLS." And then: "How to mitigate the attack? Upgrade to the latest GnuTLS version" - which is 3.2.12 or 3.1.22 if you're still on the 3.1 track - "or apply the patch for GnuTLS 2.12.x."

Now, okay. So, for example, who's using this in the field? Well, Apache for the last three years could be configured to use GnuTLS as the means for getting TLS v1.2 support because it was available well before OpenSSL was. So many Apache servers may be using this: GNOME, CenterIM, Exim, WeeChat, Mutt, Wireshark, slrn, Lynx, CUPS and gnoMint, among others.

Leo: Wow. It's everybody. That's everything.

Steve: Yeah. I mean, it is an alternative SSL package that has huge usage. And now we know that all, I mean, everybody using these needs to look for updates. This just happened. It's going to take a while to integrate the updated TLS into the specific packages. But it has to be the fact that it's coming because it needs to get done. Basically we have a certificate authentication bypass in another very heavily used open source library. Whoops.

Leo: Don't use GnuTLS.

Steve: And it was found by an audit.

Leo: Don't use GnuTLS.

Steve: Well, and the problem is it's not something that end-users use. It is a library built into all these other applications. And, I mean, otherwise it's robust and solid and great. And it's feature-packed. It supports all the state-of-the-art protocols. It's a great library. But a mistake got made and, happily, got fixed.

Leo: Actually, according to Howard Chu, who discovered it, it seems like a pretty messy piece of code. He says: "I see the code makes liberal use of strlen and strcat when it needs to be using counted-length data blobs everywhere. In short, the code is fundamentally broken." So it's not going to be a goto fail fix.

Steve: No.

Leo: Use a different library, I think, is the key.

Steve: The problem is OpenSSL looks very much the same. If you look at some of this code, which has evolved over a long period of time, it has, I mean, it's really sad-looking code. It's just like, okay, well, let's just hope it works.

Edward Snowden did another document dump. The story got picked up, of course, by TheGuardian.com that has been one of the major carriers of his document releases over time. And this one discloses - this is news from last week that I saw, and it's like, oh, goodness. It's a little problematic. The UK's spy agency, GCHQ, it turns out, was deliberately collecting webcam images from many millions...

Leo: Oh, no.

Steve: Oh, yes, of Yahoo! users with the help of the NSA. The program, unfortunately, was called Optic Nerve. And the biggest problem, Leo, was that there was so much nudity in the images that were being captured that it created a problem for their surveillance program because all of the people in GCHQ wanted to be looking at these pictures. So it started, it was in prototype form back in '08, after which it went live. And we know at least as of 2012 it was still active. TheGuardian.com in their story reports - the URL says gchq-nsa-webcam-images-internet-yahoo. So I imagine if you google that phrase, gchq-nsa-webcam-images-internet-yahoo, it'll come right up, TheGuardian.com.

They said: "Documents dating between 2008 and 2010 explicitly state that a surveillance program codenamed Optic Nerve collected still images" - what it did is it took, to manage the bandwidth, I mean, you can't just be recording the video streams of millions of Yahoo! users because that would strain even the NSA's data storage capability. So they took a frame every five minutes from all these people as a sort of a - and they also said it was a tradeoff for, like, paying some heed to privacy concerns.

So "...collected still images of Yahoo! webcam chats in bulk" - this is bulk collection. This is not targeted. This is all of the Yahoo! webcam chats they could collect. They were snapping a still frame every five minutes, "...and saved them to agency databases, regardless of whether individual users were an intelligence target or not."

Leo: Awful.

Steve: I know.

Leo: Awful.

Steve: So much for metadata, bulk metadata collection. This is photo collection.

Leo: And of course David Cameron in the U.K. has introduced Internet filters for all ISPs, just so you can't see this kind of stuff, but they're collecting it in their spy bureau.

Steve: And it's highly attractive to be, like, to be browsed through.

Leo: Well, I'm glad - yeah. Web1013 in our chatroom says it shouldn't be called "Optic Nerve," it should be called "Optic Perv."

Steve: Yeah. "In one six-month period in 2008 alone, the agency collected webcam imagery - including substantial quantities of sexually explicit communications - from more than 1.8 million Yahoo! user accounts globally." Yahoo!, of course, their response you can imagine. I mean, they were livid, saying that this was "a whole new level of violation of our users' privacy."

"GCHQ," the story goes on to say, "does not have the technical means to make sure no images of U.K. or U.S. citizens are collected and stored by the system." I mean, it was just sweeping up everything, completely independent of its point of origin. And they're disclaiming responsibility, saying, well, we can't filter. Sorry, we don't have the means. And the story says: "...and there are no restrictions under U.K. law to prevent Americans' images being accessed by British analysts without an individual warrant. The documents describe GCHQ's struggle to keep the large store of sexually explicit imagery collected by Optic Nerve away from the eyes of its staff, though there is little discussion about the privacy implications of storing this material in the first place."

So they're not saying, oh, you know, we're not sure we should be collecting it. They're saying, yeah, we want it, but the problem is how do we get the nudity out because we wish that our staff wasn't looking at all of that.

The story says: "Optic Nerve was based on collecting information from GCHQ's huge network of Internet cable taps, which was then processed and fed into systems provided by the NSA. Webcam information was fed into NSA's XKeyscore search tool" - that we've talked about before - "and NSA research was used to build the tool which identified Yahoo's webcam traffic." So that says NSA was doing the tap filtering technology, which of course we know from our coverage here that they certainly have.

And finally: "Sexually explicit webcam material proved to be a particular problem for GCHQ. As one document delicately put it: 'Unfortunately, it would appear that a surprising number of people use webcam conversations to show intimate parts of their body to the other person. Also, the fact that the Yahoo! software allows more than one person to view a webcam stream without necessarily sending a reciprocal stream means that it appears sometimes to be used for broadcasting pornography." And that's what GCHQ was collecting. Basically, the number I saw was 11% of the Yahoo! webcam stream was sexually explicit. And so...

Leo: 11%. Well, that's not too bad.

Steve: 11%. So that's what people are - 11% of people are using it for sending that sort of content to each other. And unfortunately it's being snapped every - a frame of that's being snapped every five minutes and stored and viewed.

Leo: That's the interesting thing.

Steve: Wow. In other sort of interesting news, two German freemail sites, web.de and gmx.net, have both begun tricking Firefox and Chrome users into removing AdBlock. So they're free email sites. They don't want their users blocking ads.

Leo: Sure. That's how they pay for the site.

Steve: Yeah. But rather than detecting it and telling them to disable AdBlock, what they're doing is they're faking a Mozilla security alert.

Leo: Oh, that's terrible.

Steve: I know, at the top of the screen. It's designed to look exactly like the little bar that we see sometimes when Firefox wants you to verify something.

Leo: Little yellow bar, yeah, yeah.

Steve: Yup, yup. And in fact, if you click the link and scroll down, you can see a sample of it. And then if you click on their smaller picture, you get a larger one. I meant to embed one here in the show notes, but I just forgot when I was putting it together. So it presents a false browser alert. If you click on it, it takes you to a page explaining about the "dangers" of using adblocking software, saying that it filters the content of pages, which of course it does by design, I mean, adblockers do by design, and induces false security alerts, which, okay, is what's...

Leo: This was a false security - induces our false security alerts. Holy cow.

Steve: Yeah. So the good news is Mozilla's security team are looking into it. And you can tell it's a fake notice because it scrolls up with the page. A real alert is being presented by the browser, so it doesn't scroll away when you scroll the page under the real alert. But this one scrolls off. So it's like, okay, guys, you know? And we, you and I have talked about this. No one has a problem, if a site detects that you're blocking their ads, and they're ad supported. I completely endorse the idea that they have the right to say, hey, you need to turn off, make an exception, whitelist our site for ads because that's how we make our money. That's how we provide you this free webmail service. No one's going to complain about that. Or, if you do, go change providers. But don't be slimy like this and convince people that, I mean, because what you're then doing is you're removing that, of course, from their browser, if they are confused by this and do so, which is, you know, it's wrong.

Leo: That's a good point. It's not just for their site. It's for everybody.

Steve: Right. And the fact is, as we've covered, adblocking actually increases your security to the degree that ads are sneaking malware into your system, which we've been talking about as a problem.

Leo: Apparently the - this is a German-language article from Heise Online. But apparently they have stopped after protests, both companies.

Steve: Oh, good. Oh, good to know. Well, I hope we gave them a little more - a little heat to that decision.

Leo: We have stopped. Steve Gibson told us not to.

Steve: So after we talked about it Tuesday, but in no way related to us talking about it last Tuesday, Mt. Gox did formally declare bankruptcy. So Mt. Gox is gone. Tech Crunch carried an interesting piece written and, full disclosure, written by Brian Armstrong, who's the CEO of Coinbase, which is an alternative, an alternative exchange for Bitcoin of good repute. Coinbase, you know, my favorite ZeroBlock app for iOS lists Coinbase among - there's like a set of four now that you can rotate among, and Coinbase is one of them. And it looks like, I mean, these guys are doing everything they can to be a 100% standup exchange for bitcoin.

So his piece was - he brought up a couple points I just wanted to share that I thought were worth mentioning. His piece was titled "What's NOT being said about Bitcoin?" This was covered by Tech Crunch on Friday, February 28th. His point that he makes I think is really good. And Brian focuses on the notion of Bitcoin not being as much a store of value as an enabler for an open payment system, which I think is really a great point. He says: "An open payment network is a game changer," and that what was needed was a system to manage payments in an open fashion, that is, some means of preventing duplicate spending so that, when you spent money, you couldn't respend the same money and thus create fraudulent transactions. Until Bitcoin came along, we didn't have a means to do that. There was no open network solution that didn't involve a centralized clearinghouse. Bitcoin provides that.

Leo: Boy, that's a really interesting contextualization of it that I hadn't really thought about.

Steve: Yes, yes.

Leo: And I think that's a fair way, if you characterize it that way, that really has value.

Steve: Well, and for example, Overstock just noted that in the last two years they've done a million dollars of transactions in bitcoin, and that the average Overstock.com cart, the bitcoin average cart size is $216, which is 30% higher than for people using dollars, U.S. dollars.

Leo: People spend more bitcoin.

Steve: Yes. They spend more equivalent U.S. dollars if it's in bitcoinage than in actual dollars. So, I mean, as that spreads to retailers, as retailers begin to understand that people spend more money if they're shopping in bitcoin, I mean, a third more, that's dramatic. I mean, that's world-shaking. That changes things.

So Brian, quoting from the article, Brian said: "Around San Francisco, New York City, and other major cities across the globe, bitcoin acceptance is rapidly moving into brick-and-mortar shops, restaurants, and even professional businesses like dentists and law firms." He said: "Consumers are paying with a quick scan of a QR code or using technologies like NFC and Bluetooth Low Energy. Merchants are enjoying instant transactions at lower fees, and this momentum will only accelerate in 2014 with thousands more companies beginning to accept Bitcoin."

So anyway, I just - I liked that notion, reframing it as a transaction. Remember last week when I was looking at Blockchain.info, and I encouraged our users to go look, click those links down in the lower right of the home page of Blockchain.info, where they're, like, enumerating the transactions and the dollar flows. And it's like, oh, my god. I mean, this is not just strange people in dark bedrooms with mining machines glowing and wondering whether they're minting more bitcoinage than the power they're consuming. I mean, there is an active transactional market that exists now, that's actually happening as people experiment with this Bitcoin network.

My concern, of course, I mean, I'm wondering, does this thing scale? Because when I, like, tried to catch my wallet up to date, and you need to download gigabytes of past Blockchain in order to get yourself current. And it's like, okay, what's going to happen in 10 years? How does that happen? So I have seen that there are solutions to that in the works and ways of not having to drag the entire history of the Blockchain with you. But at the moment that's what you do if you just use sort of the standard brain-dead Bitcoin Wallet.

But anyway, I think Brian's point is Mt. Gox hurt the community somewhat, but didn't kill it. He ends up saying: "Mt. Gox is in no way the end of Bitcoin quite the opposite, in fact. Just as the closing of Silk Road in 2013 led to the biggest boost in value of the bitcoin to date, weeding out immature companies and bad actors is paving the way for a legitimate bitcoin marketplace. While it may be coincidence that during the Mt. Gox debacle, Coinbase hit one million consumer wallets, it is also representative of what legitimate Bitcoin companies have known through the big ups and the low lows," which is that "Bitcoin is fundamentally the best payment system for the Internet era."

And on the heels of that, just now, if you click that link for Flexcoin.com, Leo...

Leo: Whoopsies.

Steve: Yeah. F-l-e-x-c-o-i-n dotcom, wiped out by theft. Their web page now says: "Flexcoin is shutting down. On March 2, 2014, Flexcoin was attacked and robbed of all coins in the hot wallet. The attacker made off with 896 bitcoins" - that's something like $650,000 - "dividing them into these two addresses." And they list the two addresses where the bitcoins were transferred to. "As Flexcoin," continuing to read their web page, "As Flexcoin does not have the resources, assets, or otherwise to come back from this loss, we are closing our doors immediately."

Leo: Oy.

Steve: Yeah. They said: "Users who put their coins into cold storage will be contacted by Flexcoin and asked to verify their identity. Once identified, cold storage coins will be transferred out free of charge. Cold storage coins were held offline and not within reach of the attacker. All other users," meaning those who were storing their aggregate $655,000 worth of value online, "will be directed to Flexcoin's 'Terms of Service' located at Flexcoin.com/118.html, a document which was agreed on upon signing up with Flexcoin." And we already know what the Terms of Service says. It says, "We're not responsible for any loss of your coins, but we are sorry."

And then it ends, saying: "Flexcoin will attempt to work with law enforcement to trace the source of the attack." Uh-huh. "Updates will be posted on Twitter as soon as they become available." So the lesson here is, once again, do not entrust your large store of bitcoins to an online service. You just can't at this day and age. You have to store your wallet, I mean, don't even trust it on your computer. You store your wallet offline. Right now this is immature. The market is immature. And, I mean, this was the advice we ended up with, talking about this last week, is transfer coins in when you need to do exchanges, but otherwise...

Leo: And do it fast. Move quickly.

Steve: Yes, don't leave them sitting around. It's just, I mean, the podcast covers the huge problems websites have keeping hackers out. I mean, it's a weekly topic. And there's nowhere hackers want to be more than where the money is. I mean, who was it that famously said, "That's why we rob banks"?

Leo: Willie Sutton.

Steve: Because, yeah, that's where the money is.

Leo: The bank robber.

Steve: So it's why they're robbing websites, Bitcoin currency exchanges, is that's where the money is. And the problem is we also know how difficult it is to create really secure websites. If you do anything that is convenient, like, oh, using jQuery or SQL Server on the back end, I mean, you immediately open yourself up to attacks on those things that you're using to make your job easy. And it means you are responsible for filtering to a level that it's like, oh, well, maybe we're going to get around to that soon.

I mean, the news that we have heard from Mt. Gox is not encouraging. The nice-sounding guy who founded it and put it together was aware that there were problems for some period of time and talked about getting to them when they had a chance. And as we understand it now, those problems are what sunk them, that there were people using that double withdrawal scheme that we talked about a couple weeks ago to essentially drain them over time. It was happening. So, ouch. At this point it really looks like there's a future. But it is, as I have said, it's still early days.

And a little bit of a problem with Threema. This is, of course, the alternative, very secure protocol, instant messaging app. Someone posted a workaround for their four-digit PIN used optionally to protect sensitive data. And it's an interesting hack. So in Threema you can say I want to protect sensitive data with a four-digit PIN. And Threema says that we will delete the sensitive data after 10 failed attempts. So 10 attempts to guess a four-digit PIN which you've randomly chosen is one of 10,000 numbers. So, okay, that's pretty secure, given that 10 strikes, and you're really out.

It turns out they made a mistake in programming that. They've been notified. I'm sure we'll see a fix shortly. The mistake is that they don't delete the data after your 10th failed try until after you respond to the dialogue which informs you that you've had your 10 tries; your sensitive data is being deleted. They should delete it first, then bring up the dialogue where they say we deleted it already, sorry. I don't know exactly what the dialogue says. But you have to click Okay.

Well, it turns out that the way iOS operates, and this is on the iOS platform, while the dialogue is present, it's what's termed, at least in the Windows world, as a modal dialogue. A modal dialogue is one of those where the rest of the UI behind the dialogue is locked. It's frozen. It's modal, meaning that all user interface, all user input is directed to that dialogue. So you're unable to communicate, to talk to, to do any UI events to the app that's floating behind the dialogue that is in front and on top. But if you close the app and rerun it, the app comes up and then begins the process of building, rebuilding the modal dialogue. Until it does, the underlying app can receive user input.

And so what this guy who discovered this problem found out is that he could try 10 times, and then he gets the notice that the data is going to be deleted. Unfortunately, it isn't a notice that we have already securely deleted the data. And so what he did is he shuts down Threema, starts it up again, and he can squeeze another guess in before the Okay comes up. And then he does it again and again and again and again. So it does allow you to bypass the 10 strikes in order to guess the PIN. And if you were sufficiently patient, you could eventually find it. And I wonder what happens? I didn't think it through. If you found it, but it's still displaying the Okay dialogue, I wonder how you actually use Threema?

Anyway, I don't know. But interesting little glitch with iOS and the way they handle their Okay. And it's a reason why you have to be very careful about this kind of stuff. I've encountered this, for example, with SQRL, where you want to - one of the options is to have SQRL delete its knowledge of your password when you screen save or you suspend. Well, you don't want to do that upon resumption from suspending or screen saving. You want to do that when the app receives notification that's in the process of happening. So these do have security implications, and you need to really think this through and be careful about it. So we know that the Threema guys, their hearts are in the right place. And I'm sure they will fix it, if they haven't already.

Speaking of SQRL, I posted at the beginning of the weekend my intention in the newsgroup, the grc.sqrl newsgroup, to think about internationalizing and making SQRL multilingual. The response was swift and very positive. A number of people who read the newsgroup and have been following along said, oh, yes, yes, yes, yes, yes, I hoped you were considering doing that. I've just a few panels left of UI to design, and then the UI as far as I know is completely laid out. And it's turned out very nicely. I'm really happy with the way it looks.

And so as I then immediately start writing code, of course the question is what do I do about other languages? So I'm going to explicitly put all of the text strings of the product in a separate file and allow that file to be edited. Originally, I was thinking I would just create a simple text file that anybody who wanted to create a different language version of SQRL, and who spoke English and another tongue, could open the file and just replace the strings with their localized language version, provide it back, and we'd be able to use it.

Well, it turns out that there's a better solution. There's a group called Crowdin.net. Actually, crowd-sourcing translation is all over the Internet. I mean, it's just something that makes sense. I found an online service that I really like. I'm very impressed with it. It's called Crowdin, C-r-o-w-d, as in crowd, i-n dot net. And I wrote to them yesterday, told them what I was doing, and asked if we would qualify for their free open source and academic license, and they said, oh, absolutely. We're delighted that you have found us. I don't know, couldn't really tell if they knew me and SQRL, but they sounded like they did from their reply. So what this will allow is essentially it'll create a platform for sort of managing the internationalization of SQRL's user interface. Many people have since said, hey, I've got three languages I want to do and so forth. So it looks like, with relatively little effort and very little slowdown at all in the development, we'll be able to get SQRL internationalized.

So what I will do is, as soon as I get the UI finished, I'm going to build the UI of SQRL first so that we can essentially overlap that. While I'm working on gluing all of the inner technology to the UI surface, that will allow me to publish the user interface through Crowdin.net and then invite all of our listeners who speak other languages to go over there and, spending whatever time they're able to, to work with everybody else who is doing it and convert the user interface to whatever languages they know. That will allow me then to pull all that back and to create individual language versions of SQRL. So I'm really excited about that. I think that'll help it a lot.

Leo: That's exciting, yeah.

Steve: Yeah. And I did get a nice note from a Caleb Marble in Rockford, Illinois, very short. And he said: "Rockford, Illinois (Save me from this place)." So I don't know why. Has the weather been bad in Illinois? I guess maybe - this was dated February 1st, so certainly this time of the year maybe he was buried under snow. Anyway, he just said: "Thank you for a fantastic product."

Leo: By the way, yes, the weather's been bad in Illinois, really bad.

Steve: Ah.

Leo: Yes.

Steve: "Thank you for a fantastic product," speaking of SpinRite, of course. He said: "In the few months I've used SpinRite it has recovered four drives from failure, including one hard drive for a local nonprofit whose" - and he made up an acronym, VID, meaning Very Important Documents - "VID dating back to 2002 were stored on a single shared NAS (Network Attached Storage) from the early 2000 era with no backups." He says, in parens: "(I later introduced them to Carbonite. Thank Leo for me.)" Then he said: "Thanks again. Another satisfied customer." And Caleb, thank you for the report and for letting me share it.

Leo: Questions for you, Mr. G, if you are ready.

Steve: Our Listener-Driven Potpourri No. 184.

Leo: Oh, I love it. Question 1 from Ernie Moreau in Kelowna, British Columbia, Canada. He says: "I'm part of the problem." Oh, no. Ernie, not you too. Steve and Leo, I had five of the 45,029 servers that were used in the NTP monlist attack. Whoa. Which really frosts my apples. When I first came across the attack, it presented itself as our servers maxing out our bandwidth, which could happen legitimately if we're really busy, although it did seem suspicious. So I allocated more bandwidth, an extra 200 bucks, or at least until I could investigate the issue.

I used to use the NTP (Network Time Protocol) daemon running on our servers to update my servers' clocks. That's how people use it, of course. Something ran in the background. I didn't worry about it. Never gave it a second thought. Well, I'm not running it as a daemon anymore. I've set it up in a cron job to run every so often to update the clock. There has to be a better way, but I don't want to be part of the problem. I've attached the notice I received from my Internet service provider. Thanks for all you do. You make a big difference in bringing secure practices to the masses. Or masses of geeks, then to the masses of the masses. Ernie Moreau, proud SpinRite owner. I don't see it attached, but I'll take his word for it.

Steve: No, I have it here. It was long, and I won't read through the whole thing. But I was very impressed. First of all, one thing I didn't mention when we were talking about any kind of network, any kind of Internet reflection attack, is that the attacker, we understand that the attacker is spoofing the IP of the intended target when they send packets to, for example, an NTP server, giving it the monlist command. That's a UDP packet which is carrying as its return address the target's IP. So they send the packet to the NTP server, which responds to the victim. But what that means is all of the incoming traffic to the victim is carrying the legitimate IP address of the NTP server. That is not spoofed, and in fact is not spoofable. It's because the server is behaving itself as it expects it's supposed to. It believes it's responding to a legitimate request. So that means that anybody at the receiving end of the attack knows all the IPs of the servers that essentially were DDoSing it.

So Ernie's letter says: "Hello. We've received a complaint against your server at IP address...." And he has that blanked out in what he attached, but the IP is given, which is how the note was sent. "Please ensure that this is dealt with. If you fail to fix the issue and respond to this ticket within 72 hours, your server may be shut down. A public NTP server on your network, running on IP address" - and there it is again given - "participated in a very large-scale attack against a customer of ours today..."

Leo: Interesting.

Steve: "...generating UDP responses to spoof monlist requests that claim to be from the attack target. Please consider reconfiguring this NTP server in one or more ways." And this thing goes on. It was very comprehensive. And I was very impressed that he received this kind of notice. And presumably this is the response which, at least to this attack, we don't really know what's being done in all the various cases, but it is the case that you could record all the IPs flooding you and, if you're able to reach the person in charge of the IP, send them a letter like this and see if you can get them to fix the problem. Certainly, I mean, if Ernie's reaction to being the host of a small fraction of this attack is any example, everyone involved in inadvertently generating bandwidth is aware of it because their bandwidth just goes crazy.

Leo: Yeah, it all makes sense, yeah.

Steve: Yeah, their own local connection is being saturated as their servers are trying to respond at this 400x magnification of the attack. So, very neat. Thanks for sharing with us.

Leo: Yeah, yeah, and admitting it [laughing].

Steve: Yeah.

Leo: Aldo in Chicago has a question about port-forwarding: Thanks for all the hard work on the show, Steve. Here's my problem: I have three Xboxes in my house. Each has a static IP. My router has turned - I have turned off UPNP in my router. So I manually configured the router to forward the various Xbox live ports to an Xbox. That Xbox passes its network self-test just fine. However, the other two Xboxes report NAT being Strict because I'm not port-forwarding. Actually, it's Strict because he doesn't have UPNP turned on. Is there a way to forward the same ports to multiple IPs? Oh, that's an interesting question.

Steve: Isn't that?

Leo: Little background, that UPNP is a Microsoft technology, Universal Plug and Play, specifically for Xbox. And Xboxes on networks where you've turned off UPNP will complain, and you won't be able to, I think, join games others have started or, no, you can't start your own game, something like that.

Steve: Well, and we know that for people who are security aware, security conscious, they can disable Universal Plug and Play and then map a collection of ports - statically mapping, it's called - through their router.

Leo: There's quite a few, too, as I remember for the Xbox.

Steve: Yeah. So what they do is they add mappings where they say any incoming traffic to my public IP on port whatever, 4362...

Leo: Do you want to know? It's 88 UDP, 3074 UDP and TCP, 53 UDP and TCP, and of course port 80, the HTTP port on TCP.

Steve: Yes. You have...

Leo: So you have to have those ports open and forwarded.

Steve: Now, so the idea is that Universal Plug and Play would allow the Xbox to do that for you. It would allow the Xbox to say to your router, hey, I need those ports on the outside sent to my IP. So here's the problem. The problem with Universal Plug and Play we've talked about a lot, is that unfortunately it allows malware in your network, and we know there now is malware that is Universal Plug and Play aware, to do this. So Aldo's got a problem in that he's got three Xboxes, and he wants Universal Plug and Play turned off for security. He's figured out how to map things. But obviously - so the problem is in your router you have to have it go. You say any incoming traffic on this port goes to this internal IP. It'll be 192.168.0.12 or something, whatever he's got his Xbox set for. The problem is it can only go to one IP. So here's the solution: You put your three Xboxes behind their own NAT router.

Leo: And route to the router.

Steve: With Universal Plug and Play enabled. And then you give that internal router a fixed IP on your external router, and you statically map the required ports through to the internal router.

Leo: Clever.

Steve: And that allows you to have multiple Xboxes, essentially all doing Universal Plug and Play mappings internally, which is safe because it's just going to be them on that little internal sub-network. They're all sharing the single IP to which the external traffic is aimed. And that ought to solve the problem. And, you know, routers are cheap these days.

Leo: Routers are cheap, and you don't need to use a particularly good router for this.

Steve: Yeah, use one that you don't want to put on the outside, that you don't trust out on the real world.

Leo: Right, right. Lou Rubinfield, wandering around Pennsylvania, explains why he wants to obfuscate passwords, the dot dot dots we've been talking about: I listened to your recent podcast about passwords and the ridicule you've made about ongoing obfuscation of passwords when they're being entered. I have one very good use case: I'm often presenting via projector in meetings and need to log onto a website or application. When doing so, I frankly don't want to broadcast my password to all those viewing. Just one good reason to keep obfuscating. Love the podcast. It keeps me thinking on my commutes, and of course I use SpinRite regularly and therefore have no horror stories to share.

Steve: So I wanted - this is on behalf of Lou and also the other thousand of you...

Leo: Wow.

Steve: Oh, yeah. I mean, it was half of the email that I received was people telling me their own reasons why it's important for them to have passwords obfuscated. And one I read was interesting. He said, you know, it helps people understand that this needs to be a secret. And I thought, well, that's kind of an interesting take on it.

Leo: Yeah, that's fair.

Steve: I hadn't thought of that one before. But mostly it's all the instances which people were explaining, of which Lou's is perhaps the most obvious. It's like, yeah, clearly, if you bring up a web, you want to log onto a website during a presentation in an auditorium, and of course famously during the Sochi Olympics there were all these people who had their - the camera crews were coming into control rooms and various places where their WiFi passwords were written on the whiteboard. It was like, uh, whoops. Or up on the screen.

So, and that's why the tradeoff I'm using - oh, come to think of it, though, iOS still has a problem, then. If you were logging into a password under iOS in plain sight, people would not see the password after you had entered it. But if they were quick, they'd see the characters as you were entering them, so the way iOS shows you the one you've just typed. And of course they did that as a compromise for the keyboard, which is such a problem.

Leo: So that does no good at all. I mean, you might as well just unobfuscate.

Steve: Right. Anyway, and what I'm doing in SQRL, of course, and you can see this on the operation page that's up right now, is I make it very simple to show the password if you want to see it. But by default, it will always be blanked. So you'll get big dots as you're typing it in. But underneath it there's a link for clearing it, if you want to start over, and there's a link that says Show. And after you click it, of course, it flips over to Hide. So you're able to toggle...

Leo: That's the way to do it.

Steve: That is. That, to me, that ought to be there in every case is the option to show it if you know it's safe. If there's no one looking over your shoulder, if there's not an auditorium looking over your shoulder, and you've just - and especially a complex password on a touchscreen. Good luck entering that. I mean, I can't enter my own WiFi password. It's impossible. I use cut-and-paste in order to enter it on my various devices.

Leo: Patrick Warn in Georgia, Vermont - that's confusing - wonders about SQRL and FIDO U2F: I was hoping you'd take a few minutes on a Q&A episode to compare and contrast SQRL - your solution - with solutions like a YubiKey with FIDO U2F support. Do they work together? Compete? Answer different problems?

Steve: So I'm not an expert yet on FIDO. I'm going to have to be because obviously I'm sure I'll have to put up a page on SQRL, like how does SQRL compare to FIDO, since FIDO is the acronym, F-I-D-O. It's kind of funny that we've got a name for a dog and a squirrel as the two acronyms.

Leo: Which one chases the other?

Steve: Here, FIDO. Here, FIDO. Stands for Fast Identity Online, is what the FIDO acronym stands for. FIDO comes in two completely different flavors, which is the first thing that begins to make it confusing. There's U2F, which is Universal Two Factor, and UAF, which is, I don't know what it is, Universal Authentication Framework? I don't know. I just made that up. UAF, whatever it is, that's what the other one is. And they are not the same. Essentially, U2F is sort of what Yubico and Google have been doing. And of course they famously joined FIDO recently, which is sort of where FIDO adopted the U2F alternative to the UAF, which was what was always in the FIDO project and what they were doing.

I have been a little curious, but I cannot say I've done a deep dive. I know a few things. For example, the U2F work with YubiKey and Google, from a technology standpoint, it has to be a second factor. That is, it cannot be a single factor. And so that's one huge difference right off the bat with SQRL. Remember, SQRL is designed to completely eliminate usernames and passwords. SQRL assigns you a unique, really long token. It's 256 bits, which is the equivalent of a 77-digit number. It just makes one up at random for you, for every site you visit. It's a different 77-digit number right out of the crypto. So that both identifies you and authenticates you.

And so SQRL just - the login just disappears if you're using SQRL. The problem is the Universal Two Factor, U2F in FIDO, and I verified this in the specs, they don't use my clever approach for synthesizing public key pairs based on the domain that you're visiting. That was really the invention, I mean, the light bulb that went off in my head was, wait a minute, if I use Dan Bernstein's elliptic curve crypto, it allows me to use anything as a private key. Well, that means I could use a hash of the web domain name as the private key. Which means that I can create private keys from web domains in a deterministic fashion so that every time I go back to the same domain, I get the same private key. No one knows what that private key is except inside SQRL, and that's derived from the user's master key.

Unfortunately, U2F and FIDO don't have any of that technology. They don't do that. They, when you go to a website, they create a standard key pair using random numbers. The problem, though, then is that you need to hold onto all of these private keys. And you end up with this keychain problem. So the way that they solve that is they encrypt the private key and give it to the website to hold, which is kind of bizarre. But that's what they do. So that when they go to a website, they say give me my encrypted private key, which they then decrypt. So now they have the private key associated with that website. And then they go about proving that they're the owner of the private key in the same way that SQRL does.

So what they've done is they've offloaded the storage problem by having the website hold all of the - having each website hold the private key for them. But what that means is they have to identify the user first in order to get the website to give them the proper private key for the proper user, which is kind of a kludge. But it also means it can't be a single factor.

Now, the completely different technology is the UAF side. And it's so complicated, I can't figure it out. I've looked at it. I'll spend more time on it because obviously I do need to figure it out. Someone knows how it works somewhere. Although apparently no one has it working yet, it's so complicated. And I shudder at the idea of needing to write the server side of this. But one of the beauties of SQRL is that the server side support is almost, I mean, like almost no crypto at all. It is trivial to implement, which I'm glad for because I would like SQRL to win. The UAF side is, like, unbelievably complex. And unfortunately it is based on the P-256 elliptic curve which came straight to us from the NSA.

Leo: Oh, no.

Steve: Yes.

Leo: That's a shame.

Steve: It's based on the curve where magic numbers were provided. We even know the guy's name at the NSA who said, here, use this number. And there's no explanation for why.

Leo: Well, because we know it, that's why.

Steve: Yeah. Both Bruce Schneier has said no, you cannot trust this curve, and Bernstein has a page where he calls the curve "malleable" because, I mean, it's just like, no. And unfortunately that's the crypto in FIDO.

Leo: Of course it is.

Steve: And so you've got to, again, you've got to wonder. There's influence in these critical protocols, and it looks like FIDO is under the influence of the NSA. So that sort of maybe puts a nail in its coffin.

Leo: Yeah, huh?

Steve: We'll see. But SQRL isn't and explicitly doesn't. And I'm working as hard as I can to make it work.

Leo: We'll use the charitable interpretation that the authors of FIDO didn't know it was compromised and trusted NIST.

Steve: Right. And we'll know that when they change it, which they haven't yet.

Leo: Oh, okay. Oh, there you go. John McDonald in Monterey, California. He says: Save Windows XP. I have three computers on a local area network. One's Windows 8, one's Linux, and one's XP. They're all connected by RealVNC. After April, of course, the XP machine will be vulnerable, I know. If I keep it on the LA - LA? - but never use - oh, maybe he means LAN, L-A-N - but never use its browsers to go online - oh, this is a question I get a lot. Will it be safe from hackers? It's on a LAN, but it's not going online itself. Can a hacker get to the XP machine when the Windows 8 machine or the Linux machine goes online? I want to keep the XP machine on the LAN so I can use its RealVNC connection to see the Windows 8 laptop's small screen on the XP desktop's larger screen, and control the Windows 8 laptop from the XP desktop. So - oh, no, come on. Did he just say this? Will the XP machine be able to use Google Drive securely? Did he just say that really?

And then Evelien Snel in Eindhoven, Netherlands says: We all know what is going to happen. Support for XP will end April 8th, and it worries me a lot. Windows XP is a central element in my everyday business workflow. Not all the software I use will run without problems on Windows 7. Been there, tried that. I guess I'm not the only one with this problem. Can you think of ways to keep XP and mitigate the risks of not getting updates? Maybe I can isolate a PC from the big bad Internet and have it access my internal LAN, again, only. Thank you for all your great podcasts. Prom proud SpinRite owner and listener since No. 1, Evelien Snel.

Steve: So, okay. The honest truth is I think concern over lack of security patches for XP is overheated and overblown.

Leo: Oh, good. Oh, that's good news.

Steve: I do.

Leo: Why do you say that, Steve?

Steve: Well, I haven't used any since SP2.

Leo: Patches?

Steve: Yeah.

Leo: Why not?

Steve: My machine never liked SP3. It broke it in some way. And I removed SP3. And what was that, many years ago?

Leo: Oh, yeah.

Steve: And I haven't...

Leo: It was notorious, by the way. You should try it again, because I think they fixed it.

Steve: Well, I guess my point is I'm just fine without patches for XP because I do all of the other good things. I get no spam. I don't click on links in spam. I am very careful with what I do. I use Firefox, famously, with NoScript turned on. And one of my laptops stopped being able to update. There were some - and many people, this happens to many people where some update gets stuck, and it keeps saying that it's going to reinstall this update. I've spent countless hours trying to unstick this laptop, like looked everywhere. I can't do it. And it's like, okay, well, seems to be fine. I use it. I'm careful. So, I mean, so I really do believe that people should not be freaking out over the idea that they're not going to get their monthly feed of patches from Microsoft.

Now, we just saw last week in the statistics that we shared about the virtue of not running as an administrator, 100% of 2013's problems that involved Internet Explorer were blocked if you were not running as an admin, 100% of them. So, and we don't know for a fact that it's going to block all future ones. But it blocked most of the problems, just not being an admin privileged user.

Leo: But only 92% of the problems in general. Right?

Steve: Right. So I just wouldn't hyperventilate, everybody. You and I famously, Leo, don't run third-party AV tools on our machines. I'm just careful with what I do. And I don't - this is not the normal advice I give people. I tell everybody run antivirus because I think it's generally a good thing to do. But if you behave yourself, I mean, it just isn't like your machine will immediately become encrusted with malware the moment Microsoft stops feeding your machines its monthly update.

Leo: I'm not sure I agree with you on this one, Steve.

Steve: Oh, okay.

Leo: I shouldn't disagree with the famous Steve Gibson.

Steve: No, I recognize it's a...

Leo: It's contrarian. You understand that.

Steve: Yup. And maybe it only applies to somebody who really understands the dangers. But I've never...

Leo: But the real issue of these exploits is that they, I mean, the reason exploits are an issue is because they don't require user cooperation, that they take advantages of flaws in the operating system.

Steve: Well, typically, the major vector, we've seen Flash exploits, we've seen PDF exploits, and we've seen browser exploits. So those are...

Leo: Those you're not - I'm not worried about you with those, obviously.

Steve: Right, well, I mean, that's really it. That's where all the problems are coming from.

Leo: Yeah. You wouldn't get CryptoLocker if you didn't get fooled by the phony PDF and so forth.

Steve: Right. Or lick the link in email that said, oh, look, we have a payroll update for you that you weren't expecting. It's like, what? Wait a minute? I'm not expecting that. And so you would click on the link, and it runs the malware.

Leo: So, okay. But to answer these guys' questions, if a machine's on the LAN but not actively going on the Internet, are they vulnerable?

Steve: No. You're not vulnerable. I would say increase your security, switch over to - many people ask, hey, how do I change my existing account to a non-admin because I'm all set up right now, all of - my username and all that. I can't create a new account and reinstall everything. And you don't have to. You create another account, give it admin privileges, and then change your main account to standard user. So you just demote it to lower privileges. So I would say do that. I would say, if you are an AV user, certainly third-party antivirus isn't going to stop functioning. And we did hear that Microsoft is going to continue supporting the whatever it is, the little green house that we've got.

Leo: Security Essentials, or Defender.

Steve: Yeah, Security Essentials. That's going to continue for some time, too. Yeah, so I just - I don't see it as the end of the world. It's 34 days, by the way. I've got my little down counter here.

Leo: Do you not think it's the case, though, that the bad guys have got exploits in their pocket that they're not going to release till after April 8th because they don't want Microsoft to fix them, and after April 8th a vast trove, I would imagine, of effective exploits will be released.

Steve: We'll be covering it here. And I don't - we'll have to see, either way, if that's the case. I mean, I would never suggest that somebody who isn't security aware, I would never suggest some random user using XP and Internet Explorer, who's using a laptop and clicking on every link and every email that they encounter, do this. But, I mean, for example, both John and Evelien are clearly security conscious.

Leo: Yeah. They're asking the right questions, yeah.

Steve: They have a huge investment in the configuration of XP. And it's not out. They're not walking around in open coffee shops and exposing it. They have situations where they just sort of want to know will it still be safe. And my point is it won't start to crumble the moment Windows stops sending it its monthly updates. It is still there. It's still a robust, very mature operating system. And while it's true that we see that mistakes Microsoft is making generally reflect all the OSes all the way back, these things generally are things you have to go and get. They're problems you have to seek out in one way or the other, clicking on links, going to malicious sites, getting Flash or old versions of Java, running old versions of Java. So my feeling is, if you remove Flash from your browser, you don't have Java running, you use Firefox with NoScript so you're not running scripts - especially, I mean, some of these machines they're not even doing web surfing from. They just want to be able to use the machine. I don't see any reason not to.

Leo: I'm trying to think of a counter example, of something that you might get just by - remember the old days of - was it Melissa or Stuxnet and others which - they were network worms; right?

Steve: Right.

Leo: You're not concerned about a network worm?

Steve: No, because now we've got, since SP2, we've got the firewall turned on. Everybody's behind NAT routers. NAT routers protected you. The only way you could get a worm is if you had a machine directly on the Internet with - and this is pre-firewall, either third-party add-on or finally when Microsoft turned it on in XP, or you turned it on before SP2 in XP. There you really had exposed ports, which of course is why I did Shields UP! was to let people know, like, oh, my god, these ports are actually open and exposed to the Internet. But those days are really behind us.

Leo: Wow. I'm going to have to readjust my thinking because I've been one of the people loudly banging the gong to get off XP.

Steve: Well, I mean, there isn't a reason not to. It's time to move to Windows 7. I will migrate myself. I mean, I'm hearing other people say I tried to use 7, but things I use gave me a problem. So I've got - I have a machine ready to configure. I will start moving my stuff over and just sort of take it easy and see how it goes.

Leo: See what happens, yeah, yeah.

Steve: Yeah.

Leo: Yoram Snir is our next correspondent, from in Potomac, Maryland. He wonders about a SQRL selfie. That's, well, that's a good question.

Steve: Yeah.

Leo: Yeah, sure. You betcha. Hi, Leo. Hi, Steve. I've been listening for five years. No need to say more. Yeah, that's true. I'm developing an iOS application which accesses a centralized server and requires authentication. What happened? It went dark. I'm reading it, and the machine went to sleep. Oy, oy, oy. Now, don't read my password. Oh, good, it's dots - which accesses an authorized server and requires authentication. I would like to allow the user to, A, start using iOS without any sign-in, the app without any sign-in. Oh, sign onto the centralized server, I guess. And B, start the App on another owned iOS device, then use the first device to authenticate the new device. I hope you're following all this.

Do you think SQRL can be utilized for such a solution? I would guess that server-side implementation, plus some iOS framework, can create a very strong solution for me and many others without the need for SQRL password management application. In other words, can my app - I don't - I think there's something, there's a typo. I don't understand what he's saying. In other words, can my app be an SQRL management app for itself only?

Steve: Yeah. Now, I hate the visual that's associated with this term.

Leo: The selfie, the SQRL selfie.

Steve: No, no, not that one. The notion of a headless SQRL.

Leo: That's what you're going to do right now.

Steve: Because, yeah, that's sort of what you want. And the answer is absolutely. No one has ever proposed that before, and I hadn't thought of it before. And in fact I wouldn't recommend that you use anything to do with SQRL, although you might use some of the open source that'll be developed. But essentially you could just use that core crypto that I documented in the very first days of SQRL, that is, use Dan Bernstein's elliptic curve, the Ed25519. That's the signing part.

Essentially, the idea would be that you have your app generate a good random number, and it uses that with the server to generate a unique identity. And then the server challenges the app by sending it something unique, a nonce, which the app signs using its private key, and the server verifies it with the public key. Basically it's the underlying SQRL technology, but you really don't need all of the other paraphernalia that SQRL uses. You just use that elliptic curve crypto core, which is all open source and well documented. And it's a terrific way of doing in-app authentication without any sign-in. So you can absolutely re-use that sort of the spirit of SQRL in that fashion.

Leo: The spirit of SQRL.

Steve: And you would call it a headless SQRL.

Leo: A headless...

[Talking simultaneously]

Leo: ...the spirit of it.

Steve: Yeah. Because the SQRL dies.

Leo: He's dead.

Steve: All that's left is the spirit.

Leo: It's the spirited SQRL.

Steve: Okay.

Leo: Andrew McGlashan in Melbourne, Australia has a correction about fresh JAVA installs.

Steve: Yup.

Leo: Steve, I tested your theory of a fresh install of Java and found there's a problem. I deliberately changed my Java security to allow it to be used in a browser. In other words - by the way, in most browsers now it's off. By default, Java has to get approved before it can be used in a browser. That's a good thing. Then he uninstalled Java. So he disabled the security and uninstalled Java, downloaded a fresh install, and reinstalled it. It did not disable Java's use in browsers and was fully enabled until I went back to the Java control panel applet to fix it. Please let everyone know. Cheers, Andrew. That makes sense because it's not Java that's flipping that bit, it's the browser that's flipping that bit.

Steve: Well, actually it's both of them.

Leo: Oh, okay.

Steve: What he's referring to is that I reported my experience a few weeks ago where there was something I needed Java for, and I'd completely removed it from my machine. And so I downloaded the latest Java 7, lord knows what version it was, directly from Oracle and installed it on my system in order to have Java available here. And what I reported on the podcast was I got this very comforting dialogue that warned me that Java was not enabled in my browsers. And so I incorrectly assumed that that meant Oracle was doing the right thing, that they were now, on fresh Java installs, not by default enabling it in browsers. Now, we still don't know, if a system has never encountered Java before, which way that goes. But what Andrew wanted to point out was that when he had previously had Java enabled for use in browsers, and that is a Java setting now in the Java Control Panel, which you get under the Windows Control Panel...

Leo: Oh, okay. That's a different thing, then. I see.

Steve: Yes. So you've got two sides. You can sort of think of it as like the Java is the server and the browser is the client. So the browser can disable its use of Java, or Java can disable any use by browsers. So you can sort of do it at either end. So I erroneously assumed that a fresh install was saying we're not going to be in your browser unless you explicitly tell us to. And so now we don't know for sure one way or the other. But it is definitely not what I thought was...

Leo: That makes sense now. There's two mechanisms going on, and that's - I get it now. All right.

Steve: Yes. So thank you, Andrew, for the update.

Leo: Yeah. So, yeah, just go back when you install this stuff and verify that it's turned off in the browser, which is the only safe way to have Java.

Steve: Well, actually turned off globally in your system. I mean, remember it was a few versions ago, many versions ago actually, where they gave us a switch in the Java Control Panel where we could just turn it off so that browser plugins are all disabled.

Leo: Ah.

Steve: And so but do it both ways. Turn it off in your browser, and turn it off in your system. So you can use Java applets, Java applications, but you cannot bring up Java in your browser because, whoa, that's just not safe.

Leo: A lot of people use Java applications because they play Minecraft, which is probably the single most common use.

Steve: And speaking of which, Leo, my god, is that popular.

Leo: Yes. You noticed.

Steve: Holy - I was amazed. 400,000 downloads in one day? And then it's got 100 million users.

Leo: Yeah.

Steve: Wow.

Leo: It is. And funny, because it's just this eight-bit game, just a silly eight-bit game.

Steve: Incredible.

Leo: Yeah. But people like it. What's nice about it is it's generative. You're usually building stuff, which I think is really cool.

Steve: Yes.

Leo: John-Charles, that's his first name, in Chicago, Illinois recommends a free Trust No One alternative to Hamachi: Tinc. Long-time listener, 2006. Show among the many things that changed my life and steered me into the software industry, where I am now gainfully employed. I'm in the process of catching up on episodes, and I heard in a recent Q&A a gentleman bemoaning the loss of the free version of Hamachi.

I, too, faced a similar challenge when Hamachi dropped Linux support, or at least made it very frail some years ago. At the time I found Tinc, a no-configuration VPN, T-i-n-c. It's free and does require at least one node to have a known IP. But all other nodes are auto-configured. They find each other through the one common node. It's not like OpenVPN in that it is fully decentralized. Once a node has connected to the network, it sets up direct connections on all other edge nodes. But it is very fast and reliable for me to keep a connection to my father's computer in a different state. Not the computer. The computer's in a different state of the union. Not a different state of physical...

Steve: Being.

Leo: Being, yes. If you have not researched Tinc, I would strongly urge you to do so as it solves many problems seemingly faced by many in the SN community, and it is completely Trust Nobody. Thanks for the wonderful security advice and hours of entertainment. Proud owner of SpinRite, long-time listener John-Charles.

Steve: So it's T-i-n-c hyphen V-P-N dot org [Tinc-VPN.org]. And I was not aware of it before John-Charles mentioned it. And it looks very nice. I have not messed with it or played with it, but I wanted to bring it to our listeners' attention. It is 10 years old, so it's been maturing quietly on a back burner somewhere. And it's got broad cross-platform support. It looks like it's got very good RSA certificate-style security. And it uses what they call a "mesh network," meaning that, as he said, you don't have one central server, but you do need one of the nodes to have a known IP. All the other ones connect to it. And they find out about each other from the one shared node, and then they establish direct connections to each other.

So these guys refer to it as a mesh network, where your traffic ends up going directly point to point. It's described also as a "zero configuration," or no configuration VPN. I would not describe it that way. But I'm impressed. They have walkthroughs for configuring under different platforms. Anyway, I just wanted to raise a flag for listeners of ours who are technically savvy, and I know that we have a huge base of technically savvy users. This may be a solution worth taking a look at. It looks really nice. They've got installation packages. It's, again, cross-platform, and all the bells and whistles. So thanks, John-Charles, for making us aware of it. It looks like a great alternative.

Leo: Tinc. Glenn - I guess this is it. This is our last one.

Steve: Mm-hmm, yeah.

Leo: Glenn in Maryland. He says don't trust Foxit: I've been using it as my PDF reader since it was recommended here. Yeah, we've recommended it for years. I was just notified of an update, after installing it found my homepage and default search had also been changed, and two additional programs had been installed without warning. I bet not without warning, but okay. I'll try it. During installs I watch for extra junk. Well, he says that it's since become so common. And perhaps worst of all, in researching what happened, Foxit says the extras will protect you from download software changing your homepage without permission. This is very common, by the way, that the bad stuff says I'm going to protect you. Foxit and all its friends - reason is, we took over the home page so no one else can.

Steve: Right.

Leo: Foxit and all its friends have been removed from my system. I guess I'll give Firefox PDF viewer a try. That's too bad.

Steve: It is. Now, the Firefox PDF viewer is not great. I mean, it's there, and it kind of works. Really has font rendering problems and worse printing problems. So, and it's a memory hog also. When you bring up a PDF in it, it's just like, wow. You see it burn up memory. So, I mean, I consider it still in its early stages. There are other plugins. So maybe something other than Foxit. But anyway, I did want to give everybody a heads-up about Foxit because it has been our goto suggested PDF plugin for Firefox for so long. And it's unfortunate that, I mean, given - again, you haven't verified this. I've not verified it independently. But I wanted to...

Leo: It's frankly not surprising. We've talked about Download.com and how the hitherto useful downloader...

Steve: Oh, god, it's awful.

Leo: Yeah, Michael downloaded - Lisa's son downloaded something from Download.com, and I had to remove, I think it must have been 10 pieces of spy, you know, not malware per se...

Steve: Junk.

Leo: Junk from the system. And by the way, I'm mad at you. This program, Rails, is stealing my life.

Steve: Ooh, you got a circle. So you're able to create other - you can create Y connections to send the trains around in both directions.

Leo: Oh, yeah. What I've learned, the last thing you want to do with this game is make a right angle. Everything should be - everything's got to be loop-de-loops; right?

Steve: Isn't it wonderful, Leo?

Leo: Yeah, well, yeah. "Wonderful" is one way to describe it, if by "wonderful" you mean taking over your life so that you can't live and do anything anymore, yeah, wonderful.

Steve: Oh, I know. I know. It is so wonderful.

Leo: It's really a fun game. It's, what was it, a couple of bucks, Rails. There's Android...

Steve: $2.99?

Leo: $2.99. Android, as well as iOS. And the Android version is exactly the iOS version. There are similar games to this, if you look for railroad traffic control games, even Flash games on the desktop. And I know you don't do Flash. But this is really good. I've been running it an awful lot lately.

Steve: Oh, I'm glad, so glad.

Leo: How far have you gotten? It's hard.

Steve: I've stayed down, wanting to see how high I can get my score.

Leo: Yeah. You're trying to get all the stars; right? Yeah.

Steve: 435, I think, is the most I got on Level 1.

Leo: Yeah. Well, and then you get these stupid little slow guys. And then there's these gray trains that can't be routed at all. You have to get them directly to a station. And, oh, man, I hate this game. I blame you, Steve. See, I'm trying to find a good extension. You always want an escape hatch for any...

Steve: Yes, exactly, where if you need to route a train down a spur.

Leo: Yeah, there you go, there you go. See, because I got another one coming this way. So quick, quick, get that yellow little push cart out of the way. Aagh. I hate this game. I hate you. No, I don't. I love you dearly, and I thank you for introducing me to yet another addiction. There are so many that I owe Steve Gibson for, including cabernets, coffee, and Rails.

Steve joins us every Tuesday, 11:00 - I'm sorry, 1:00 p.m. Pacific. Let me pause that game. I was doing pretty well there. 1:00 p.m. Pacific, 4:00 p.m....

Steve: Don't want to lose it.

Leo: Don't want to lose that one. Ugh. 1:00 p.m. Pacific, 4:00 Eastern time, 21:00 UTC on TWiT.tv. Next week we're going to talk about the Apple security document.

Steve: Full deep analysis of this fabulous document that Apple finally deigned to publish and disclose. I'm really pleased about it. And we'll have a great podcast.

Leo: Good. Good, good, good. So do watch here live. If you can't, though, on-demand versions made available. He has some on his website, GRC.com, 16Kb audio for bandwidth-impaired, but also transcriptions, which are really great, thanks to Elaine Farris, does a good job on those. He also has, of course, SpinRite there, GRC.com. SpinRite's the world's great hard drive maintenance and recovery utility. ShieldsUP!, to test your router. UnPlug 'n Pray, all sorts of stuff.

Steve: DCOMbobulator.

Leo: DCOMbobulator. Does anybody still use that? I don't know. It's there, though, in case you need it.

Steve: It gets downloaded.

Leo: Does it really?

Steve: Yeah.

Leo: Wow. Somebody's still running Windows 95, I guess. Lots of stuff, up-to-date stuff as well as the good old days. And passwords and all sorts of good things. And diet information. If you want full quality audio and video, we have it here, TWiT.tv/sn for Security Now!. And of course Security Now! is carried on every podcatcher and Stitcher and iTunes and the Xbox Music Store and all those places, so you could subscribe there. Just get every episode when it comes out. It's nice to get all the episodes. Get the full collection. And that's at our website, TWiT.tv/sn. Steve, thanks so much. We'll see you next time.

Steve: Thanks, Leo.