SECURITY NOW!

Transcript of Episode #496

Listener Feedback #207

Description: Leo and I discuss the week's major security events and discuss questions and comments from listeners of previous episodes. We 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-496.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-496-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Oh, yes, of course we're going to talk about Superfish and Komodia and what it really means. And of course nobody better to do that than Steve Gibson. Then 10 of your questions; 10 of his answers. A great Security Now! is coming up next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 496, recorded Tuesday, February 24th, 2015: Your questions, Steve's answers, #207.

It's time for Security Now!, the show where we cover your security and privacy online with the guy who knows more than anybody else out there and is a great Explainer in Chief, Mr. Steven Gibson of the Gibson Research Corporation. Hi, Steve.

Steve Gibson: Yo, Leo. Great to be with you again, as always. And we have such a great show that I considered dropping the Q&A to next week, except that next week we may even top this one. As I mentioned a couple weeks ago, "60 Minutes" did a segment which was really unnerving, where they had a group on, talking about in this instance car hacking, who disabled the brakes on an unlabeled vehicle. It had a manufacturing label on it, but they, like, blacked it all out with tape because they didn't want to embarrass the maker of this car.

Leo: Although, as you pointed out, if you knew anything about cars, you probably could figure it out; right? Yeah.

Steve: Yeah, I'm sure it's, like, everybody except me knew what that was. All of the hybrids sort of look the same. They've got this weird sort of hybrid look to them. And so this was, I think, one of those. And they told Lesley Stahl, who was doing a segment on

"60 Minutes," to stop in front of the orange cones. And so she puts her foot on the brake and runs right through the cones.

Leo: Yeah.

Steve: Anyway, it turns out that this group that were, I guess they were a subcontractor or hired by DARPA to do the research, are fans of the podcast. We're going to have them on next week to talk about...

Leo: What?

Steve: Yeah, to talk about...

Leo: Cool.

Steve: ...carjacking. And actually they wanted to broaden it a little bit. They're also big fans of and [clearing throat] I guess have done some work on drone hijacking.

Leo: Oh.

Steve: So we're going to talk about vehicle hijacking, vehicle hacking. And because they're - it's funny because when we were talking, going back and forth in email, the guy said, well, one of the main techies managed to get one line into the "60 Minutes" program. You know, it's like, yeah, that's a problem. And I said, oh, yeah, I said, you know, we've all done interviews where they interview us for half an hour and then almost all of it winds up, as they say, on the cutting room floor.

So but for this podcast, because we're techies, we're going to get the whole scoop on, like, what they actually had to do to take over the cars, and what the state of that is. One of these guys is ex-NASA and knows all about what it takes to do formal proving of security because they had to do formal proofs of, like, space shuttle software correctness in order to get, you know, not to have, like, oh, wait, we need a patch for this? No, we can't have any patches on this space shuttle software. So a great podcast next week, and everyone's going to want to catch it.

But this week, of course, we had not only our regularly scheduled Q&A #207, but major, like, everybody was buzzing. In fact, Twitter was really unusable for me for a day while everyone who follows me wanted to make sure that I knew about what I'm calling Lenovo's big mistake because I think that's probably the best way to characterize it. And then we have another stunner from Edward Snowden. We've got TrueCrypt back in the news in a good way; the fact that the HSTS support is gaining its final major adherent; some tidbits and follow-ups; and then, of course, 10 questions from our listeners. So not a dull second, I think, this week.

Leo: Yeah. You've got people glued to their sets. Do you use a set?

Steve: It's funny because I asked my sister years ago, when my niece and nephew were in high school, I said, "What channels do they watch?" And she says, "Oh, they don't watch TV."

Leo: No.

Steve: "They watch their laptops."

Leo: Yeah.

Steve: They cut the cord, like, long ago. So you need to turn the volume up on your laptop.

Leo: Okay.

Steve: And play this YouTube link which is the first thing on the show notes here under "Komodia."

Leo: Oh, lord, lord, lord.

Steve: It's only about a minute. It's about a minute and a half.

Leo: Okay.

Steve: But it is just a hoot.

Leo: All right. YouTube.com, let's look.

[Clip]

MALE VOICE: So you want to develop a network interception application like parental controls or anonymizers. Maybe you want to do it yourself, or you've already got a working proof-of-concept on a virtual machine supporting one or two browsers. Now the fun begins. You've got to ensure you're supporting all the current OSes and the 64-bit flavors. What about the five leading browsers? And you'll want minimal conflict with the top 40 antivirus products. Could be you want to support HTTP decoding and SSL decrypting. And that's going to get really complex. You could skip doing all this QA, but do you really want your clients doing the QA for you? Twelve to 18 months go by, and finally you can get to work on your core application.

You know, there is an easier way. Introducing Komodia's Redirector [Leo yells], the network interception SDK that allows you to develop your solution instantly. It's used by more than 100 clients, including some Fortune 500 companies, to develop parental control software, anonymizers, game acceleration, and other custom solutions. By using Redirector, you can focus on your core application without getting into technologies like LSP or WFP. With a simple-to-control interface, you can intercept website traffic and network applications [Leo: Oh, my god] from any programming language.

So where do you want to be in the next year and a half? Slaving away with the QA, or launching your product? Make the right choice. [Leo: Oh, my god] Komodia's Redirector. [Leo: Oh, this is...] Get your free 14-day trial now.

[END CLIP]

Leo: Oh, this is not a joke. That's an ad for Komodia.

Steve: It is a professionally produced, high-quality ad, basically saying we're producing an SDK which will keep you from having to roll your own. And what happened was that, among many other companies, if we believe them a hundred others, a company called Superfish said, well, we can barely get out of bed in the morning, so we're going to use the 14-day free trial that Komodia is making available, and we're going to wrap our product around that because, boy, that really sounds like it will do the trick and save us all that time independently developing that ourselves.

Leo: Can you explain what it did, too? I mean, I, you know...

Steve: Oh, yeah. We're going to get there.

Leo: Good. All right, all right.

Steve: Oh, yeah, yeah. And then along comes Lenovo, who, like so many companies today, is adding crap to their product. I mean, I'm having to - every time I update Flash, I've got to prevent Adobe from installing a trial version of Norton Antivirus on my computer. It's like, turn that off. I don't want that. But unless I'm careful, I get it. And we've talked about all of the crapware, which is probably the best term for it, which is being installed on stuff, retail things that we purchase.

Leo: Did you hear...

Steve: I've heard everything you've said about it since you heard about this.

Leo: ...what Rene Ritchie pointed out from the How-To Geek, remember, How-To Geek did a great piece where they used CNET's Download.com to download a file, and the top 10...

Steve: Top 10 Downloads.

Leo: ...loaded with stuff. But they just updated that piece to say, and by the way, two of the adware programs that you get by using Download.com have Komodia in them.

Steve: Yeah, yeah. Okay. So...

Leo: So this is everywhere.

Steve: So my take is that, first of all, it was going to come to light sooner or later, and Lenovo happens to be, as we know, they're now the number one PC producer; right? Or they were until last week. And they unfortunately chose to preload Superfish onto people's systems, and Superfish uses the Komodia SDK, the ad for which we just heard or saw, in order to pull this off.

Now - so, okay. First of all, the greater concern, the sort of overriding concern is that, to some degree, this is a generic response to the same problem the NSA has been complaining about, that is, the NSA has been complaining about - or in general law enforcement, the three-letter initial organizations. Law enforcement generically has been increasingly upset that, in their term, the problem of the Internet going dark because we are increasingly bringing up security. We've got efforts that'll be going online a few months from now from the EFF, the whole adding the technology to essentially make encryption free. It's traditionally not been free because you have had to purchase certificates of varying grades and quality and repute from certificate authorities. And in order to drop the friction of going to TLS to zero, the EFF is going to be doing this "we all encrypt" effort to essentially automate with your server the process of getting and maintaining an SSL certificate.

So even before that, you know, there's been major efforts to move us to security. Google, to their credit, has been pushing this, and maybe overly pushing it, but still pushing it. And the whole HSTS, the HTTPS Everywhere effort, and all of that.

Leo: As somebody said, five years of progress in securing your transactions out the window in one fell swoop. I mean...

Steve: Okay. So the point is that the Internet is going dark, and law enforcement has been affected by this. But so, too, have other services which we have traditionally relied upon. For example, antiviral software is also doing this. Antiviral software is installing a certificate in our browsers in order to crack open our secure connections in order to do AV scanning inside of SSL tunnels.

So I sort of want to put this in context. We're going to talk about what an extra unbelievably awful job Komodia has done. But the overall view here is that things that we say we want, no one is saying they want visual discovery, which is the Superfish product, which was wrapped around or layered on top of Komodia's odious HTTPS proxy. But I'm seeing HTTPS proxies now being installed by AV software because that's the way they're choosing to solve this problem in order to have visibility into the increasingly SSL/TLS connections that browsers are making.

Leo: Yeah, but I did not ask my antivirus to watch my SSL streams. Who, I mean, do you really want that?

Steve: That's the default. When you install...

Leo: I don't - but I didn't ask it to do that. Well, I don't use antivirus. But if I did, that's not what - I want to look at why do they want it to do that?

Steve: Many, well, AV tools, as you know, for years have been filtering our Internet connections, trying to catch this stuff before it gets into our computer. And so if we're saying yes, we want you to monitor our use of the Internet; we want you to block, you know, downloading things. We want you to see that something coming in is bad, really on the fly, before it has a chance of landing and being executed. That's what we're asking for. And so that's what these things are doing now.

Leo: So we are asking for that.

Steve: We really are, yes. And I found myself - I'm thinking, what was it I just - it might have been Malwarebytes. I turned that off because I knew to turn it off, but it was on by default. And so this is the way this problem is being solved.

Leo: Is it possible to do this safely?

Steve: Okay. So let's talk about that. The jargon that we saw explode across the Internet was "man-in-the-middle attack." And while it's technically true, the man that's in the middle is installed on your computer. So this really wouldn't, if done correctly - if done correctly. And that's the key. It's unnerving that all of the certificates that you see when you look at websites are actually signed by your AV. And there is a tremendous responsibility on the AV product to do it correctly because it is so much easier to do it incorrectly. And that's the path that Komodia took. And that's what's actually mostly upset security researchers because, for example...

Leo: Well, there's also the larger issue of, yeah, Komodia is a man in the middle on your system. But it then passes it on to Lenovo or someone, a third party. So it's acting as a man in the middle for somebody else. Presumably your Norton is not doing that.

Steve: Okay. So the reason everyone's sort of trying to gloss over this is it is complicated. So what any of these things do is they put a certificate in your OS. Now, we should note, this is also what corporations are all doing now because again, they have no choice. Their network's users' traffic is probably mostly today, but probably all in some not too distant future, going to be over secure connections. The corporate IT guys were becoming increasingly blind to what their own corporate IT traffic was. They want border AV. In order to do border AV today, in 2015, you need to crack open SSL connections. And sadly, there's no next level of encryption unless you do something special. That is to say, unfortunately, credit card information and usernames and passwords are being protected by SSL.

I don't want to put a plug in here for SQRL, but of course SQRL doesn't even need SSL to be secure. SSL is optional in SQRL because its own security is so strong, it doesn't rely on that. So using a different authentication system than just usernames and passwords and trusting the SSL encryption does protect you from this. But within a corporate environment, you probably have a certificate.

And in fact I've had the SSL, can't remember what I called it, page on GRC now for quite some time, for a year or so. When I realized this was going on - oh, SSL Fingerprinting. When I realized this was going on, I created a page to allow people to see whether anything was intercepting their certificates. Yup, there it is, Fingerprints. Because GRC has a view of unfiltered certificates, the actual certificates from the sites. And if yours doesn't look the same because a fingerprint cannot be forged, then something is interfering with yours. So nobody behind an AV system which is changing their certs will see the same fingerprint. Nobody using one of these Lenovo laptops or, unfortunately, any of these other hundred products, whatever they are, that is using Komodia, is going to see the same certificate.

Okay. So we want to believe that corporate AV proxies are doing a good job. We want to believe that AV, that is, the AV products we purchase and now probably pay an annual license fee, are doing a good job. Microsoft has a privileged position in Windows in that they don't have to do this in order to provide real-time Internet connectivity filtering, and their stuff does this. The problem is they're always a little bit behind the curve. It takes them a few months to, like, add awareness to this. And I did hear just today that they've added awareness for Superfish to whatever brand of AV they're now offering. So that was a pretty quick response. And fast response is what they're trying to do with their solution.

Okay. So what did Komodia do wrong? There it's sort of hard to know where to start. The first thing they did wrong - okay. So to understand the architecture, in order for these to function, your browser must have the public key of the certificate authority. That's the way the CA system works. So when we talk about the "root CA store" in any of our operating systems - Windows, Android, Mac, doesn't matter, iOS - the root CA store, those are all of the public keys belonging to the certificate authorities who have signed the certificates of remote servers. So when the remote server sets up a connection with us, they have - they've signed the certificate with their private key that they protect, I mean, that's the crown jewels of any of our certificate authorities. They absolutely protect it to their death.

And in fact they protect it so much now that the certificate isn't signed with their root, it's signed with a sub-CA because they don't even want to expose their actual root certificate to even their own signing process, they're so obsessed with security. Nothing gets their private key. So all we have is the public keys. But as we know, that allows us to verify the signature, that is, allows the web browser to verify the signature on the certificate for that site that we have received.

So if you're going to do an HTTPS proxy, if you're going to crack open SSL/TLS connections in order to see in them for whatever reason - in the case of corporate IT to filter the Intranet's traffic before it gets to you in order to look for malware; in the case of AV which you have installed on your machine, that AV tool installed a certificate in your root CA store - because they are going to - the only way this works to intercept, they're going to spoof the certificate from the remote website. When they see you wanting to create a connection, an SSL connection, they intercept that attempt, manufacture a certificate on the fly which they sign with their private key.

And this is the big weakness of all of these systems. That private key, which is never supposed to leave the depths of a real certificate authority, it must exist in that proxy in order to create spoofed remote server certificates, SSL/TLS certificates on the fly. Now, a good implementation of a proxy will create a unique public key to put in your browser, that is, to put in your root CA store. It'll create a unique key pair, a public key and a private key, every single time. The worst thing that Komodia has done is to reuse the same private key throughout their entire product suite.

Leo: So not only is it visible on your machine, but everybody's is the same.

Steve: Yes. And the password that protects their private key that's in the Superfish software installed on Lenovo laptops and a hundred other software products is "komodia." It took Robert Graham three hours of poking at this thing...

Leo: It's a great story, by the way. Read his blog post about how he found the password. It's great.

Steve: Yup, yup.

Leo: He says, "I used ghetto tools."

Steve: Exactly. So basically he just said, okay, maybe the simplest dumb thing will work.

Leo: And it did.

Steve: And it did. Yeah. And he basically...

Leo: So it's the same - now, what is the purpose of the password as opposed to the key?

Steve: The purpose of the password is supposed to be to - you're supposed to have the password in order to decrypt the certificate's private key on the fly.

Leo: Ah. Oh, okay. Well, that's why you'd want to use the same one for every machine. Simplifies the code.

Steve: Well, not only the same password. That would be bad enough.

Leo: The same key?

Steve: But the same key.

Leo: The private key.

Steve: Now, what does this mean? This means - this is like the Death Star scenario. This means everybody who has any version of the Komodia-based software, a hundred companies including Fortune 100 companies, we don't know, you know, doing parental control software - again, these companies do not roll their own. They say, ah, you're right. We saw the ad on YouTube. We want to save ourselves 18 months of painful cross-platform, cross-browser, cross-everything development. We'll take the 15-day free trial, get the SDK, fire it up. Oh, look how easy it is. Drop it in, off we go. We don't have to do all that. And every one of these products, based on this, has installed the same public key in the root certificate store on all these platforms.

So that it's very much like the Hong Kong Post Office problem except this is worse because at least the Hong Kong Post Office's private key is hopefully unknown. In this case, the private key, it took Rob Graham three hours, 180 minutes of just sort of trying stuff, and he now has it. In fact, it's on Pastebin. Everyone now has it. There's a link to it in the show notes, and all of our listeners...

Leo: In case you want it.

Steve: ...now have it, in case you'd like it. And this allows you to do anything you want on any of those machines. You can now create your own certificates for websites which all of those machines will trust.

Okay. Now, Part 2 of how bad this is, is that during this connection setup, it's now created a fraudulent certificate to make your browser happy. Now it turns around, and it connects to the remote server in order to make the connection to that remote server's HTTPS. Unfortunately, it's got the worst set of security ciphers ever seen. It still has 40-bit, four zero, 40-bit RC4 and MD5 hash as the cipher. Which, you know, which everybody can crack. It does 40-bit - four zero bit - DES, not even 3DES, just once. One DES.

Leo: What? No.

Steve: I mean, these are ciphers from...

Leo: Is this a high school kid that wrote this? What is...

Steve: ...that everyone stopped using. Even I stopped using them 10 years ago, they're so broken. And this thing supports them all. So all that anyone would have to do is be connecting - now, okay. You would have to still have a server on the other end that agreed to this craziness. But this demonstrates how bad this software is, that it's willing basically to drop all the way down almost to no encryption in order to connect to the remote server and complete your connection.

So there is a site that I link to in the show notes here, Filippo.io/Badfish, which anyone can use. It takes a few seconds, and it will check your system for whether or not you've got Superfish, Komodia, there's another one called PrivDog which has come to light through all of this. That's another - it's a piece of software people install on their computers, not knowing what they're doing and how bad this actually is. And Leo, you just did it, and...

Leo: Now, I'm on a Mac. I'm not vulnerable. Or am I?

Steve: Well, Komodia says you are. Mac was one of the platforms.

Leo: So they make a tool for Mac developers, as well.

Steve: Correct.

Leo: Now, okay. So you raise a very important point, which I'm glad you did, which is that these so-called man-in-the-middle attacks, these self-signed certificates that companies put on there, are often used for good purposes. But it does point out that you have to trust, if it's your antivirus, that you have to trust that, not only are they trustworthy, but that they've implemented it in a trustworthy way, or didn't borrow Komodia code or something.

Steve: Or they may even have the best of intentions.

Leo: With the best of intentions, right.

Steve: They could also have bugs. They could have, like, for example, SQL Server, who thought that a database would cause such a problem?

Leo: Right, right.

Steve: You know, because...

Leo: So should we just eliminate self-signed certificates? Is that a bad idea?

Steve: I think it's really a bad idea. I think that, now, Windows apparently has some hooks in it. And I've not taken the time to dig in. But I remember when Microsoft was going to be doing this. They talked about making hooks available for traffic filtering specifically for AV. And I don't know why people are not taking advantage of it. But people aren't. They're just doing this.

Leo: So they don't have to be doing - you don't have to do this.

Steve: No. I don't think you do. Because Microsoft still is smarting from those antitrust days, and they didn't want to have features in Windows that were exclusive to them. As Microsoft has crept into the AV filtering business in Windows, which they're now solidly in, they've had to make those same hooks available to other vendors.

Leo: Is there a way to go through your certificates on a system and see what certificates...

Steve: Yes.

Leo: And delete ones you don't want?

Steve: Yes. You can look at your root store. And I'm trying to think, what was it that I - oh, I know. It was on Jenny's laptop. Jenny's laptop got a bunch of crapware installed on it, both hers and her mom's. And I went through and deleted the - and it was doing this. In fact, it may have been, I'm afraid to say this, I think it was Malwarebytes. After I removed Malwarebytes, it left its public key certs behind. And so I went into the Windows - it's called personal - I think it's the personal certificates. Windows divides them up into different places. So you can see the ones that have been added. And they're pretty obvious that, I mean, they're not like DigiCert and Global Trust and Symantec. Or actually Symantec may have installed some, too. I think I have seen - although Symantec, it's VeriSign now, so that may be why those are there.

Leo: So would it say "Komodia" if you have a Komodia cert on there? Would it say it's from...

Steve: It probably does. It probably just says hi, you know, we're Komodia, trust us. It's like, oh, no.

Leo: On the Mac, you know, you just go to Keychain Access, and you can see both your personal certs easily and other certificates that are installed, and you can go through those and remove those. So on a PC it's a file that you look at?

Steve: No, you've got to go into...

Leo: Is there a tool?

Steve: ...admin tools. You can go in, it's the Certificate Manager. And I think...

Leo: Ah, okay. So you go to the administrator - you can right-click on your computer, select Properties, and then bring up the Certificate Manager there.

Steve: Yeah, but it's not surfaced on all systems. Sometimes you have to go into the Run dialogue and go certmgr.msc or something like that, in order to get to the - but certainly you just Google how do I get to Certificate Manager in Windows, and there's lots of stuff there on the 'Net.

Leo: Sounds like anybody who listens to this show should be doing that. You're all sophisticated enough to do that.

Steve: I really - yes. I absolutely agree.

Leo: Wow.

Steve: And for what it's worth, the show notes have a ton of more links to all of these things we've been talking about. Get your own copy of the private key if you want and so forth.

Leo: Where do you - yeah. I searched for Certificate Manager on this $59 tablet, actually, and it popped right up. So I can just...

Steve: Oh, good.

Leo: I can just run that with a $59 tablet. Okay. Good. People should probably do that.

Steve: Yeah. So...

Leo: It may break some software, though. It might break your antivirus.

Steve: Yeah, I would say see whether what you see looks like something you want. For example, in corporate IT you don't want to be deleting the certificate that your gateway AV has installed, or you won't be able to get on the 'Net at all. I mean, you'll quickly know that was a mistake. So don't discard these with abandon. I know that our listeners have been having fun with this ever since we've been talking about how - I want to say "rich" and "deep" the certificate store, the root store has grown. There are people who are seeing, who are experimenting, our listeners experimenting with how few they can survive with. And the fact is it is a very steep exponential decay, where you go from 450 down to 10, and pretty much all of the Internet that you care about is being served by the 10 largest certificate authorities on the 'Net. And then it just, you know, nobody is, I mean, how often are you actually encountering a certificate signed by the Hong Kong Post Office?

Leo: Right.

Steve: Maybe never. But it's there. So you could - that's the kind of thing you could safely delete.

Leo: Yeah, and I see, for some reason, some weird certificates in my Apple, as well. I might want to just take those...

Steve: Eh, I know.

Leo: Do not confuse Komodia, which is K-O-M-O-D-I-A, which Comodo, with a "C."

Steve: Actually, Comodo is in the doghouse, too.

Leo: What?

Steve: They're the people - yes, Comodo, the CEO of Comodo is involved with this PrivDog tool which some people are feeling is even worse than Komodia. And I thought it was interesting that Comodo, who is unfortunately a certificate authority, they have another branch, or I guess it's Comodo themselves, who are selling software which is doing some of this same stuff.

Leo: [Expelling breath]

Steve: Yeah.

Leo: I should have asked. We had Gregor Freund, as you know, on yesterday.

Steve: Yeah.

Leo: He was the guy who created the first popular firewall product, which we recommended...

Steve: ZoneAlarm.

Leo: ...heartily for many years, ZoneAlarm.

Steve: Yeah. We didn't recommend the color. The color scheme was really annoying. But...

Leo: The bright red.

Steve: Oh, my god. Orange, orange and red.

Leo: Did it make a noise? I feel like it might have made a noise, too, like [harsh buzzer sound]. Anyway...

Steve: Well, I think it was just when your eyes saw it and there was, like, some neural feedback of some sort because it was, like, ugh.

Leo: But would even - that's more than 10 years old. Would that have used, would all firewall software do this kind of stuff?

Steve: No.

Leo: No.

Steve: Because that was just looking at IP addresses and packets and blocking where they were going to. And that's on the outside of the...

Leo: It didn't have to inspect the stream and the contents of the stream to do this.

Steve: Correct. It's the so-called "deep inspection." If something's doing deep inspection...

Leo: That's what gets you in trouble, yeah.

Steve: ...the only way to do it is to crack these things open. In fact, one of the other consequences of this we've talked about is that this also breaks all the caching that ISPs are doing. And there has been some rumblings that it may before long become a requirement for customers of ISPs to install the ISP's public key in their root store to allow the ISP to crack open your secure connections for the sake of caching in order to decrease their bandwidth. Because the problem is this absolutely, you know, SSL creates a one-to-one association between your browser, hopefully, and the remote server. And that's having a real impact on ISP bandwidth. Let's hope that never happens.

Leo: Oy, oy, oy.

Steve: I mean, that's - oh.

Leo: And where would we get your show notes? People are listening, going, okay, I want your show notes. Is that at GRC.com/securitynow?

Steve: I always tweet, I tweet the link just before the show. So it's in the Twitter stream, and it's always the same format. And they're always linked. So, okay, so you go to GRC.com/sn and then look at the show notes for last week, which are the third - it's the third icon. And then just add one to the number. That is, the URL is just, you know, it just increments.

Leo: Okay.

Steve: It's something like - I don't have it right here in front of me. It's, you know...

Leo: @SGgrc on Twitter. Just go to Twitter.com/SGgrc, and you'll see the link there.

Steve: And you'll see it right there, yes.

Leo: Yeah. I'm so scared now. I don't want...

Steve: Yeah, I mean, it is...

Leo: I'm willing not to have anything do deep packet inspection on my system.

Steve: I agree. I think...

Leo: I can live without that, thank you.

Steve: The problem is with doing that comes great responsibility.

Leo: Right. And I don't trust anybody.

Steve: And Komodia demonstrates how irresponsible it's possible to be. And the problem is, even well-intending AV tools, we're requiring, we're hoping that they're not going to be making any big mistakes.

Leo: Right.

Steve: And unfortunately this is a lot to verify. And I agree with you, Leo. I think it's better just to say, no, I don't want anything cracking my traffic open.

Leo: I want my traffic to Amazon and my bank to remain encrypted. Of course, if you're using PGP, if you're using your own personal encryption layer...

Steve: Another layer.

Leo: ...you're safe.

Steve: Exactly.

Leo: But you're not with Amazon and your bank because they don't support that.

Steve: No. Right. In fact, that's a nice segue into our next story because The Intercept dropped the news from another tidbit from Edward Snowden, that GCHQ, the U.K.'s equivalent of our NSA, in cooperation with the U.S. NSA, infiltrated the network of Gemalto. Gemalto is not quite the sole source, but for all intents and purposes the sole source, of the world's SIM cards. They produce two billion SIM cards per year. They're a multinational firm incorporated in The Netherlands. Their clients are AT&T, T-Mobile, Verizon, Sprint - the big four in the U.S. - and 450 other wireless network providers around the world. Gemalto operates in 85 countries, has more than 40 manufacturing facilities, and they've got a major one in Austin, Texas and a large factory in Pennsylvania.

So essentially what we learned - and this is the slide, this is the picture of the week on the front page of show notes is the slide that Snowden captured and has revealed. Essentially, their network was infiltrated some number of years ago, and GCHQ was bragging that they now had all of the private keys in all of the SIM cards that Gemalto has been producing.

Leo: And how many is that?

Steve: That's all of them, essentially.

Leo: They sell two billion a year.

Steve: Yeah.

Leo: So it sounds like it's all of them.

Steve: It's pretty much all of them.

Leo: All the U.S. carriers use Gemalto. Everybody does.

Steve: Yup. Everybody. When I saw the name, it's like, okay, I know the name because that's where the SIM cards come from.

Leo: There's no other company? It's all Gemalto?

Steve: It's basically Gemalto. There are some others like, you know, because there's always room for one more.

Leo: You've got to admire the NSA. I mean, they've obviously hired the best hackers they could afford.

Steve: You know, Leo, I've been approached at earlier phases in my life, and I remember thinking, eh, working for the NSA would be boring. I was wrong about that.

Leo: No.

Steve: I don't think...

Leo: Only the smartest people work there.

Steve: Yeah. I mean, they've got mathematicians, but it's the hackers at the NSA who are busy.

Leo: Well, you remember that scene in "Good Will Hunting" where they try to hire the mathematic genius of - and then he says, why should - they said the question you should ask yourself is why shouldn't I work for the NSA? And he has actually a long - it's on YouTube - but very good answer.

[Clip from "Good Will Hunting"]

WILL HUNTING: So why do you think I should work for the National Security Agency?

NSA AGENT: Well, you'd be working on the cutting edge. [Leo: Yes, you would.] Be exposed to the kind of technology that you wouldn't see anywhere else because we classified it.

Leo: What year was this? This is like 1998? '97? We did not know how right they were.

Steve: Oh, boy.

Leo: And how right Will was, if you listen to his answer.

Steve: So with the stolen encryption keys, "intelligence agencies can monitor mobile communications without seeking or receiving approval from telecom companies or foreign governments."

Leo: Basically, that's the keys. That's the keys.

Steve: Yes.

Leo: They can get anything they want.

Steve: "Possessing the keys sidesteps the need to get a warrant or a wiretap and leaves no trace on the provider's network that the communications were intercepted. Bulk key theft such as this enables intelligence agencies to unlock any previously encrypted communications they had already intercepted, but did not yet have the ability to decrypt. As part of the covert operations against Gemalto, spies from GCHQ, with support from the NSA, mined the private communications of engineers and other company employees throughout multiple countries. Gemalto was totally oblivious to the penetration of its systems and the spying on its employees. Gemalto has refused to comment, other than to say that they had no prior knowledge that the agencies were conducting the operation" against their network.

Matt Green, our cryptographer friend at Johns Hopkins, explained to The Intercept, that broke this story, "Gaining access to a database of keys is pretty much game over for cellular encryption."

Leo: What you want to know is what SIM cards does the NSA use? Because I bet it's not Gemalto.

Steve: Yeah, see, the reason this was a perfect segue that you brought us into was you mentioned that the problem with the lack of trustworthiness of TLS, that is, of HTTPS now, is that we're relying on it for protection of in-the-clear data - credit card numbers, usernames, passwords. When I'm looking at my credit report, it was delivered over SSL, and I'm looking at my Social Security number, and I'm answering questions and confirming things to the website. So that is our sole wrapper of encryption. And so it is not TNO, obviously. We never claimed that it was.

But the good news is that we do have TNO solutions that are essentially encrypting within our cell phone communications. So when you use the encrypted verbal and text communications, them having the decryption key for the wrapper of that, that is, the external tunnel no longer helps them. So this is why having iMessage encryption, even though it's of dubious value with Apple maintaining the keys, I would argue that for most communications iMessage is safe. You just can't absolutely depend upon it. You need to use something like Threema or one of the other tools where you're carrying the obligation of managing the keys, but the flipside is nobody else has them.

So just to finish on the topic of the GCHQ and the NSA and cell phones, 2G was the original platform. Remember, and I've talked about this through the years, I remember, like, telling my attorney, when I was using first-generation analog cell phones, like we'd stop the conversation at some point, and I would say, "Wayne, I'll call you back on a landline once I get to the office." Because I knew from my own experience you could just get a cheap police scanner, and it would scan the frequencies that cell phones at the time used, and you could hear people's conversations - and some of them were, I mean, it was entertaining - because there was no encryption.

Then we went to 2G, which is the current, still the dominant platform globally. And that encryption is trivially cracked. So you don't really even need the keys for 2G. The NSA can cut through that like butter. And it is still the dominant platform. But 3G, 4G, and LTE, that's not crackable. You need the keys for those. And now we know GCHQ, with the help of the NSA, basically attacked the Gemalto network, got it infiltrated, did what they call "implants" on a number of their servers, and have exfiltrated the database that relates the SIM card to its private key, which you otherwise would not have. They make that data available to the carriers, that is, the carriers have to have that in order to decrypt what this SIM card is doing, that is, they have to know the private key of the SIM card. Nobody else is supposed to know. Well, now we know that, as you said, Leo, got to give them credit. They have all of those.

Leo: They're good.

Steve: And understand, too, that the other thing, the other factor here is that SIM cards were never introduced originally for privacy. They were introduced to control billing fraud.

Leo: Oh, that's interesting. Ah.

Steve: Because billing fraud - yes. And so the whole SIM card supply chain never really had security as its focus. It was to bring fraud down, which was rampant in the early days of analog cell phones. So when they went to SIM cards, it was a billing hook. But as a consequence, not that much security surrounded the whole supply chain from one end to the other. And as we know, the weakest link in security is what will get attacked and cracked.

Leo: Yeah.

Steve: TrueCrypt, an update. This was a tweet actually by Matt Green, who has been overseeing the audit of TrueCrypt. The good news is the audit of TrueCrypt, that final version 7.1a, is underway.

Leo: Oh, good.

Steve: So we'll remember that late in 2013 they brought out the very first crowdfunded audit of TrueCrypt and raised $70,000. Part 1 was finished, and it examined only the boot and the startup, the initialization process, and came out with a clean slate. Then

Part 2, which is the much more challenging part, was going to be the detailed look at the cryptography of TrueCrypt from the symmetric encryption through the random number generator and basically everything else.

But then we all got blindsided when late in the spring of 2014 the TrueCrypt authors decided to throw in the towel and pull the plug on the TrueCrypt project. After recovering from the shock of that, taking a look at where things stood, talking to attorneys and so forth, they've decided they're going to move forward. A group called the NCC Group's Cryptography Services has the contract to perform the Phase 2 audit. And Matt wrote that they will be evaluating the original TrueCrypt 7.1a, and are to begin shortly. However, to minimize price and make the donations stretch farther, they've allowed the start date to be flexible, he said, which is why we don't have results yet. But it is underway. And that's the one they're going to be doing.

A lot of people have asked me, what about this or that spinoff? And for what it's worth, the attorneys have examined the license agreement, and all the spinoffs are illegal. It is not legal to do anything with the source code. All we can do is look at what we've got and continue using that. So people may or may not care about the legality of that; but it's a little dicey, then, taking cryptographic software from somebody who you know is breaking the law and who knows they're breaking the law.

And finally, IE is the last browser to adopt HSTS, the strict transport security for HTTP in Windows 10. It's in the technical preview, either now or coming. But IE was the last browser in heavy use that did not support HSTS. And there are a couple interesting things. There is a site, HSTSpreload.appspot.com. That's a site that allows anyone who has a server that wants HSTSpreload to add it to the Chromium list which all the browsers are now using. Firefox and Opera and even IE will be using the official Chromium preload list. GRC.com has been on it now for quite some time.

And in fact, if you put GRC, or, yeah, GRC.com into HSTSpreload.appspot.com, it'll verify. It'll say, oh, GRC.com is already in the list. And there's another cool thing you can do with Chrome, a fancy URL. If you put in chrome://net-internals/#hsts, that takes you basically to a browser of the Chromium preload list. And you can then put things like GRC in, GRC.com. And, oop, it'll pop up and show you, yep, you're in the browser. And it allows you to easily see who's in and who's not. But you can also use HSTSpreload.appspot.com, not only to query the official global list but to submit your own server for inclusion.

And the reason that's important, just to finish this topic, is that the one weakness of HSTS is that the very first time a browser goes to a server which is issuing HSTS headers, if that first contact were intercepted, and the "S" was removed from the HTTPS, then they could downgrade your security, and the user would never know. Once that contact has been made, though, over an SSL/TLS connection, HTTPS, then the server will give the browser permission that lasts for, like, a year. It's three million something minutes, I can't remember the number now, but it's a long time. And GRC has been issuing that header for quite a while, ever since I went to HSTS, that is, HTTPS Everywhere. And of course an increasing number of sites are doing that now. Once the browser has that, then it knows to silently upgrade any HTTP connection to HTTPS. And then the man in the middle never has a chance to actually intercept and interfere.

A couple of little bits of miscellany and errata. In talking about HTTP/2 last week, I had a lot of fun talking about the Lempel-Ziv compression and how that worked and how, by using a single compression context across all of the streams, you automatically got header compression. That's true. And HTTP/2 does that. But the surfacing of CRIME, which was an attack on compression, people will remember that there was a very clever way of probing the data being compressed by changing it. By using essentially compression artifact, using the change in size that zlib, the GZIP library, used, it was possible to reverse-engineer the data that was being compressed.

So after zlib was already in use for SPDY, CRIME occurred, and they backed out of using standard Lempel-Ziv-style compression and went to essentially the same thing, a shared context, but where specific headers are pretokenized, essentially, because the headers are just not - they're so well-known, things like user agent and cookie and URL and so forth that are so well known, those are pre-assigned to tokens. And then there is a context maintained which is such that subsequent streams only talk about the deltas from the stream before. So I just wanted, for the sake of completeness, to correct that.

Also I wanted to mention that "Citizenfour," which is the movie I talked about after staggering out of the theater and being so impressed with it, it is now airing on HBO. It's just shy of two hours long. It won an Oscar on Sunday for the Best Documentary. It scored an 8.3 out of 10 on IMDB, and something I've never seen before on Rotten Tomatoes, 98%. So, and I think the movie is absolutely worth seeing it. People who listen to the podcast will remember that the way I described what I learned from it was I came way with a much better appreciation of the notion that even though I may not be super concerned about privacy for myself, I came away with a much better respect for the fact that privacy is a right; than I should respect other people's concern for to a greater degree than I think I had before I saw the movie.

So again, I commend everyone to check it out. It's the story of Edward Snowden, but really well done, I mean, basically they had the foresight to always have a camera running, from their very first meeting, when Laura and Glenn walk into the hotel room and have no idea who this guy is. I mean, absolutely none. We get to see all that. There's a lot of stuff that has not been seen before. So I thought it was really good.

Leo: It's on iTunes and Google Play, as well, if you don't have an HBO subscription.

Steve: Ah, good. Good, good, good. And have you seen it yet, Leo?

Leo: I have not. I will watch it tonight.

Steve: I really recommend it. It really is, I mean, it's, I mean, we sort of think, my god, more Snowden? How could there be anything we don't yet know? And it turns out I know as much as we all do, and I was really impressed. It's really worth - I think it's a very worthwhile movie.

Leo: Speaking of privacy, you know, and you to some degree, I to a much greater degree, I'm not worried about my privacy because I'm kind of living in public anyway. But I agree with you, I think we certainly should consider that. Somebody in the chatroom, and I just wanted to circle back a little bit to the Komodia thing, said, "Well, Leo, you're not concerned about privacy. Why would the Komodia thing bother you?" It's not merely privacy. It's a security issue, as well.

Steve: Yeah.

Leo: We should emphasize that. It's not merely that somebody could see your stuff. It's that it's so poorly implemented...

Steve: Yes.

Leo: ...a bad guy could take advantage of it.

Steve: Yes. And that's exactly it, is that we don't even know, no one's even looked yet at what bugs this thing might have. But imagine - I'd be surprised if it didn't have buffer overruns in it. How could it not, with it being such a piece of crap? So that anyone using it, surfing the 'Net - and by the way, it's going to be obvious to a server that you're probably using Komodia because who else would be advertising this cipher suite? I mean, this ridiculous...

Leo: No one else would do this.

Steve: Oh, my lord. I mean, it's either a browser from prehistory, you know, IE2, or it's Komodia in 2015. So you could have probably no problem at all sending malware back up that connection, taking over the user's computer. And, boy, you sure do not want their root certificate, for which everyone on the planet now has the private key, you don't want that matching public key in your root store of your computer. Wow.

Leo: Okay. I'm sorry, I didn't mean to distract you from...

Steve: No, I'm glad.

Leo: "Citizenfour," everybody should watch. I'll be watching it tonight. And I was so pleased to see that they won the Oscar. That was great. That was amazing.

Steve: Yeah. Yeah. So I just - I am now through with number three book in "The Expanse" series. I finally know why it's called "The Expanse." That's all I'll say about that. Apparently some people felt that I did a bit of a...

Leo: Spoiler? Bit of a spoiler?

Steve: Yes, yes, yes. Yes, thank you, a bit of - no wonder I couldn't remember the word. I hate the idea that I would have done that. But when I went to search for a word that I used, I was shocked that it was so far back in the book. I thought the word, the term I used was something we encountered much sooner. So I really apologize. If anyone felt - actually it turns out that what I said was more of a tease that isn't exactly correct. Or, I mean, even though it seemed like a spoiler, it actually isn't. So I'll let you figure out what that means. I did also learn that the 10-episode Syfy series coming out sometime later in 2015 is just book one.

Leo: Wow.

Steve: I was wondering, like, how much they had stretched it, how much they were going to do. So that's cool. Now, I'll say of book - because I said that book two was better even than book one. Book three, I think, it felt like it had been stretched out a little bit. I mean, there was a lot that happened. It was fun. I like the guy's writing style. And actually it's a pseudonym for two people. Leo, why do people do pseudonyms or, no, what do they call it?

Leo: Well, you know, Stephen King famously did because he wanted to write books in the genre Westerns. So he wrote them as...

Steve: Pen names is what I was trying to think of.

Leo: Yeah, so he wrote as Richard Bachman because he didn't want people to say, oh, it's a Stephen King novel, and then say, oh, it's a Western. So that would be one reason. Why O. Henry used O. Henry instead of his real name is beyond me. I don't know.

Steve: So people just choose to for some reason.

Leo: Privacy? Security? I don't know. That's...

Steve: Anyway, so this is a...

Leo: I don't know.

Steve: The book says it's by one guy, but Wikipedia knows better, it's a couple guys. Anyway, so three is absolutely worth reading. I've not started into book four yet. Mark Thompson said that I would be a little less impressed with four. I don't know what that means yet. But so far I really, I mean, I like the characters. I really, you know, it's great. It's not at the very, very pinnacle of the books we've recommended, the Hamilton stuff, the Michael McCollum novels, the Pournelle-Niven stuff. I mean, that's world-class. This is maybe just a notch below. But when you're looking for something - and they're on Audible, too. So...

Leo: I'm putting them on my list.

Steve: Consider that, yeah.

Leo: I have two credits waiting for me.

Steve: Actually, you've already got the first one.

Leo: Oh, that's right, that's right. That's the one where he's drifting through space debris, looking for stuff to scavenge; right?

Steve: There's definitely good stuff. So, and I got a fun note from someone who apparently isn't too familiar with me because he referred to me as Dave.

Leo: Dave? Hey, Dave.

Steve: So his name is Fred Elbel. And the subject was "Testimonial for SpinRite." And this came through Sue. So he says "Dave: This is a testimonial for SpinRite. You may place the following text on your website, but you may not publish my email address." And nor will I publish your use of my name, Fred. I guess his name is Fred. My name, of course, is Steve. So he's not a podcast listener. And so in fact this explains it. He says: "After doing a backup of an old XP system, it failed and would not boot because of a boot drive error. I wasn't sure that the backup had even completed okay. Then I remembered reading about SpinRite and wondering what the heck it actually did. After reviewing the material, watching the videos, and reading testimonials, I decided to give it a shot. I created a SpinRite CD image on another system and booted it up on the bad system. I then ran it in Level 2 against the bad drive. The next morning the computer had powered off, so I ran SpinRite again. It completed in an hour or so. At that point the computer booted and ran flawlessly!"

Leo: Wow.

Steve: "SpinRite is a fantastic tool," he says.

Leo: We agree.

Steve: "And is certainly worth the purchase price. Fred, Denver, Colorado, USA." And for the record, my name is Steve.

Leo: Thank you, Fred. Thank you, Dave. Steve Gibson, he's here to the rescue.

Steve: You know, an interesting attack, I've sort of been thinking while you were talking, would be anybody in an open WiFi environment - hotel, caf, whatever - could intercept someone's HTTPS connections and just send them back a Komodia-signed certificate.

Leo: Oh. And [crosstalk] on there.

Steve: And if their, well, if their browser accepts it, then that just says, okay, good, we get to filter their entire conversation. And it's just that easy. I mean, so the man-in-the-middle attack is, to the degree that there was something going on with a third-party server - and you're right, there was - it was never exactly clear what, although in all of the futzing around that Lenovo and Superfish did, they talked about, oh, shutting down the third-party data polling or whatever the hell it was doing. So there did seem to be some skullduggery about doing something with the data that they were intercepting. But even absent that, having that Komodia cert on your system really does open you for an attack, just that simple. Anybody in the middle can just grab a connection, see if your browser accepts it. If not, oh, nothing gained, nothing lost. But if it does, bang. They completely own all of your secure traffic.

Leo: Terrible.

Steve: Yeah.

Leo: And a lot of people just say, yeah, sure, whatever, okay. Must be Microsoft. Question No. 1 in our listener-driven potpourri, from Kai Harder, Frankfurt, Germany. He's noticed a new phishing trick: Steve, I recently received a clickbait spam email. "You can see my private photos here," said the link. Examining the email source, as you recently advised, told me the link led to a shortened URL. I was about to delete the message when I saw an unsubscribe link near the bottom. Sure enough, same URL. Maybe this is old - of course. Maybe this is old news to you, but I wasn't familiar with this scheme before. You regularly warn people about not clicking links in email, but you've got to include "unsubscribe" links in here because this is a really nasty one that people might not think of. I am always clicking the unsubscribe link.

Steve: Actually, I am guilty of unsubscribing to email that is unsolicited from people I know. So, I mean, it is dangerous. But you get on mailing lists, and you're hoping that they're going to honor your unsubscribe. So, I mean, it's things like Twitter. Twitter just - I keep, you know, they're trying to commercialize and incentivize and amortize and monetize Twitter. And so I'm getting crap from them that I don't want.

Leo: All the time.

Steve: And so, yeah, so I trust them to honor an unsubscribe. But the reason I grabbed this was it is - this is the way people are getting CryptoWall on their machine. This is the way that GCHQ is penetrating the employees of Gemalto. So ostensibly good people are using it to get in and get the information they feel they need. And we know that bad guys, I mean, this is the vector. It is social engineering email. And I just liked this because you get email you don't want. And, so, boy, even if you're smart enough not to click on the nasty private photos link, you see, oh, look, but I can stop getting more of these. Uh, er, you know, and before you know it, your drive's encrypted.

Leo: Yeah. Well, I do it on a Mac, so that's probably all right. Or a Chromebook. That would even be better; right? Do it on a Chromebook. And then, even if that unsubscribe link leads to something nasty...

Steve: Yeah, well, and I, of course...

Leo: And don't give any credentials, either.

Steve: Right. I do it with Flash, Java, and JavaScript all disabled, so there's that, as well.

Leo: But if you don't, I mean, if you click "unsubscribe" and it says, oh, and by the way, what's your credit card number, I would not give it to them.

Steve: Yeah.

Leo: That'd be another thing.

Steve: That would be a bad sign.

Leo: Bad idea. Juun Pei in Mountain View, California, had a question about embedded spy firmware in hard drives. We found out about that this week. Did we talk about that last week?

Steve: Oh, yeah. Actually...

Leo: We did.

Steve: ...we did cover it in the podcast, yup.

Leo: Because it was the Kaspersky security conference.

Steve: Exactly.

Leo: Yeah, yeah.

Steve: That was underway just as we - it was, like, on day two, and they had announced it on day one.

Leo: So with Kaspersky finding that spook firmware in Western Digital and Seagate drives, among others, do whole-drive encryption tools such as the trusty TrueCrypt defeat the ability of the spooks to see what's on a compromised machine? I've been reading what others are saying on some 'Net forums. The idea is firmware like this would use any OS it was aware of to do its bidding. From what I know about computers, as soon as firmware like BIOS loads the OS, the OS takes over. So how can firmware even listen for incoming connections to the machine or exfiltrate data? Also, if that's possible, wouldn't simple packet captures reveal that's what's going on?

Steve: Okay. So this is an instance of us not having enough information.

Leo: Right.

Steve: One of the things that is important for people to understand, because I've been flooded with people asking if SpinRite could, like, fix this, it's that Kaspersky did not find the spoof firmware on any drives.

Leo: Oh.

Steve: Because you can't.

Leo: Oh.

Steve: What they found was the evidence of it in the dropper, which is where all of the awareness of those different drives' makes and model numbers was.

Leo: Yes, yes.

Steve: So because the firmware, you can't see it. It's write-only. The API, the firmware API doesn't let it be read back. And once it's written, it protects itself from being overwritten. So you can't even refix it by updating from the manufacturer's good firmware. It gets in there, and it protects itself because it's got all the keys. It's the firmware that decides what the firmware should do.

Leo: So it's like a rootkit in that regard.

Steve: It is, exactly. Oh, and by the way, I forgot to mention that Komodia makes rootkits also.

Leo: Yeah, that's handy. The full-service hacker, yeah.

Steve: In fact, a link that we didn't go to in the show notes that I meant to, the Komodia site immediately went offline. And I don't know why. They immediately declared they were DDoSed, except that DDoSed page came right up. And then, but I had already seen the site because I, you know, thanks to my Twitter followers, I was clued into this thing immediately. I went to Komodia because I drilled down, figured out where the real problem was - it was with them - browsed around their site, and then a while later when I went back it was DDoSed. Ah, but I remembered. The old Internet archive. And so I went back, and I have archived links in the show notes. Their product list is a hoot and a half. Because they're just, oh, yeah, SSL interception, we've got that. Rootkit, you'd like a rootkit? We've done all the hard lifting and the rootkit. Just click here, and you get your rootkit for two weeks' free trial. So, yeah.

Leo: Do you think it's just some guy, like a 19 year old who says, ah, this would be a good business?

Steve: Boy, I bet he's been making some money. But I think that game's up, yeah. I mean, you saw the ad. The ad was super...

Leo: That's well done.

Steve: Oh, my god. I'm sure that even if - say that the Superfish people said, hey, Lenovo, we'd be happy to pay you to install this crap on your laptops. And the Lenovo engineer says, uh, we're really not sure. Who is this Komodia? Oh, go look, check out the YouTube ad. These guys are on the up and up. Oh, okay, fine. Yeah. So anyway...

Leo: What a world.

Steve: What I wanted to say was that we cannot see the firmware on the hard drives. No one, not even Kaspersky, has been able to see the firmware on the hard drives. All they ever found was the installer. So we don't actually know much about it. We know what firmware could do, and we believe that it substitutes the boot sector for its own. So when the boot sector which the BIOS jumps to in order to start things going, when that happens, the firmware could supply a different one.

Now, the question is, would the drive being encrypted render this ineffective? And I'm strongly inclined to say yes because there was this - there was also this sense of which file systems that firmware was aware of. It knew about FAT and NTFS and EXT2 and a couple others. But the fact that it was file system-specific makes me think, well, then there were a lot that, you know, it might have a problem if what's physically written on the drive is encrypted. It's not a file system it knows how to interpret.

On the other hand, if it was smart enough to hook the OS after the TrueCrypt bootloader had loaded, maybe it's smart enough. Or maybe, I mean, maybe they have this as a TrueCrypt-defeating mechanism, which could be possible. We just don't have enough information because unfortunately no one has ever been able to see into the firmware. All we saw was the thing that was able to install it into the drive.

Leo: Yeah. Wow.

Steve: Yeah.

Leo: Steve in Utah is looking for a low-level formatting tool: Steve, with all the wonderful free stuff you have, I'm wondering if you've ever written a low-level formatting program that can be used on my hard drive. If so, please tell me what it is called. And if you haven't written one, well, where could I get something like that?

Steve: Well, let me think. Have I ever written...

Leo: You have to think, I'm sure.

Steve: ...a low-level formatting tool?

Leo: What could that be? What could that be? Is it called Spin, Spin [crosstalk]...

Steve: Yeah, maybe just the preeminent one of the entire history of human mankind. Of course, famously, that's what SpinRite 1 was. It was the low-level formatting tool to end all low-level formatting tools.

Leo: Not famously enough, apparently.

Steve: Yeah. So, yes. I saw this, I said, oh, this is too much fun.

Leo: It's great.

Steve: Yes. Now, what Steve of course wants is - I'm not sure because what happened is that SpinRite's low-level formatting was so good that, in order to be that good, it had to do a whole lot of other things that weren't about low-level formatting. That is, it had to have fabulous data recovery because once I low-level formatted the drive, I was never going to be able to recover data again. So I had to recover it first.

Leo: Right.

Steve: And then drives stopped being low-level formattable, yet SpinRite continued, not being a low-level formatter anymore because it couldn't be, no one could, but being a data recovery tool. It just sort of changed because it was always a fabulous data recovery tool, and that part lived on. So Steve, I'm not sure why you want low-level formatting. It sounds like you want to zero the drive. And the low-level format command does still exist in drives. Anything, any low-level formatting tool - just google "low-level drive formatting tool," you can find one. When viruses are particularly upset with you, they will trigger the low-level format command, and your drive will start going tick, tick, tick, tick, tick, tick, tick, tick, tick. Now, it isn't low-level formatting. It is zeroing. So if that's what you want, they are all created equal. There's no way for one low-level formatter to be better than another because all any of them can do is issue one command to the drive, which is "format," and the drive will go off and do that.

Leo: So it's in the drive firmware.

Steve: Yeah. It's supported by the firmware. And no drives low-level format themselves because they all have all of this high-level intelligence on them. They've got sector pools, spares pools. They've go servoing information. They've got all this management architecture actually living there on the surface of the drive. So no drives today actually perform a low-level format. They have converted it into a "write zeroes to the drive." And that's what they do.

Leo: Yeah. Very good. So, Steve, you've got one, you just didn't know it.

Steve: Yeah.

Leo: Tom in Indiana, he's a little worried about HTTP/2, which we talked about last week. Will there be a way for browsers to still block ads, scripts, unwanted audio and video completely? That is, will browsers be able to block the remote web server's speculative push? If not, those of us with bandwidth limits, for instance on wireless, you know, 4G networks, and expensive service are going to get hit hard. Blocking BS is easy now. A program just modifies the page. But it seems now that a user will have to eat and pay for that BS, even if it is not displayed as output, just it's cached.

Steve: Great question. And I didn't get into the absolute weeds in our discussion of HTTP/2 last week. So a couple quick notes. Each endpoint, the browser and the server, are able to, well, by spec, they exchange a settings frame which allows them to set their connection policies. So the browser could constrain what the server is able to do, that is, not give it so many streams that it's just got free streams available to arbitrarily send things down to the browser.

The other important thing to note, which I didn't mention, is that the same origin policy still applies to server push. That is, remember that we're making one connection between the web browser and the server at one domain. So the same origin policy is what we've talked about often in the way browser pages are restricted, like script from a given origin cannot just arbitrarily go and get resources from some other origin, from some other domain name. It's only able to make queries from the same origin.

Well, same origin applies to server push, meaning that there's no way for other domains' content to get mixed in or pushed to the browser. But finally, and most importantly, the mechanism that implements server push is actually called a "push promise frame." The push promise frame is sent by the server to the browser, saying I recommend that you allow me to push the following content. And that push promise identifies the resource. So the browser has to not decline the server's suggestion. And it can. For example, it might respond, "No need, I've got it in my cache."

So that's the way we solve the problem of an aggressively pushy server, saying, hey, let me send you everything else on this page. Instead, what it sends is it sends intent to push, essentially, and the browser can very quickly decline those that it doesn't wish to have. Those could be ads that it doesn't want to take the time. But remember, ads are typically going to be coming from a different origin anyway. So there wouldn't be ads coming up that same stream unless the same server was wanting to serve ad content, for example, to solve the problem of cookies not being tracked across domains and so forth, which it's conceivable some servers would. But more importantly, the browser has to say, yeah, I don't know about that. Go ahead. Send it when you have a chance.

Leo: Yeah. I'd like to see that. Send it along.

Steve: Yeah. Yeah, when you have, yeah, when you've got some spare time.

Leo: Joe Meady, I'm sorry to say, in Stratford, Connecticut, had a CryptoWall encounter. I'm sorry to hear that, Joe. Steve and Leo, love Security Now! and the other shows on Tech TV. I like it that he called it "Tech TV." That's exactly what it is. I watch them on my Roku. It's tech for your TV. While I was watching The Tech Guy show live on Sunday, my wife calls down to me from the home office and says, "Oh, hey, Joe..."

Steve: Joe.

Leo: "Joe, I can't open the QuickBooks file. I get an error that says I may have the CryptoWall virus?" So I ran up into the office and, sure enough, she's got the virus. I shut down the computer immediately, while it was in the process of encrypting her files, and I managed to stop it from going any further. What a terrible way to interrupt a live TWiT show, he says. I didn't add that, he added that. I booted up Ubuntu -oh, Joe, thank goodness you're there - so I could copy off any data that was not compromised. Fortunately - oh, Joe, I'm loving you even more - my wife was not an administrator. So I'm wondering now how she got it. I've read that CryptoWall makes a copy of your data file. It encrypts it, then deletes the original file. I was wondering if I could recover some of the files by undeleting them. Is that possible? Probably he gave his wife the administrator password. And so when...

Steve: Actually, what we know is that CryptoWall cannot delete the, what are they, the system restore copies. So you can still get infected with CryptoWall locally, but it cannot make the deep deletes that your system really needs.

Leo: If you're not an administrator.

Steve: Right.

Leo: He said, I'm wondering if I could recover some of the files by undeleting them. Is that possible? Can you recommend a good tool to use to undelete files? I think she got the virus from an email from one of her clients, possibly a family member.

Steve: Yup.

Leo: I'd like to track down the source of the infection. Can you give me any guidance on that matter? I was thinking about using a virtual machine to track down the email or website. Unfortunately, not all her pictures were backed up. Oh, sigh. She's pretty upset and even considered paying the random. I told her not to. Her machine was an XP machine. This happened while I was in the process of getting her new hard drive up and running with Windows 7. One more day, and I would have been done. Oh. Any help is appreciated.

Steve: Okay. So the best thing to do for Joe and anybody else is BleepingComputer.com.

Leo: Great site.

Steve: Yes. That's the site. Rather than trying to go through all the possible variations, this does put you in self-help mode, but clearly Joe is up to the task. I mean, look at everything else that he did, firing up Ubuntu and getting all the not-yet-fully-encrypted files off. There you will find forums, and they've got one devoted to CryptoWall where you'll find a bunch of, unfortunately, people who are in similar straits, and pointers to all the best tools and solutions and commentary, write-ups on the state of the art in how to deal with it. CryptoWall just came out with version 3.0, which uses Tor more deeply. It gives people more time. I think 2.0 only gave you, like, five days. This one gives you a longer grace period. So it's not gone. It's still bad. But BleepingComputer.com, that's where you want to go find the right forum to tackle CryptoWall, and we'll just turn you loose with that, Joe.

Leo: Do you have an undeleter you recommend?

Steve: I don't have any favorite undeleter. Yeah, I don't know.

Leo: It all started with Norton; right? That was the first one I used. I think that's one of his first...

Steve: Yup, that was...

Leo: That was his first program.

Steve: That's what put him on the map, Norton Undelete.

Leo: Remember that well.

Steve: Yup, Peter Norton.

Leo: On the original PC, I don't know if it's still true in Windows, it was an upside-down "E." They replaced the first letter of the file.

Steve: E5 was the character.

Leo: It appeared on the screen as an upside-down "E."

Steve: Yup.

Leo: And that's how you knew it was deleted. That's all it did.

Steve: Yup. Well, actually it...

Leo: I guess it would remove it from the catalog or something; right?

Steve: Yeah. Well, no, it overwrote the first character in the directory.

Leo: Right.

Steve: And then it released all the chain of allocation.

Leo: Right.

Steve: The allocation chain in the FAT. But it didn't overwrite the first cluster, which was part of the directory. So when you wanted to undelete, you'd go through and look for the E5s. Oh, and there was a way to tell what that was. Or no, no. I think it would prompt you, what do you think the first character of this file was? And you'd go, oh. It would say, you know, blerch urity, and you'd go, oh, security. And so that would give it the name. And then what it would do is it would have in that directory entry the first cluster of the file. So you could always get that back if you hadn't done much writing since then.

But the other thing the FAT tried to do was it wrote in a go through the drive forward and then loop back to the beginning and overwrite. So even so, it would generally not be writing over what you had done. And so then it would sort of heuristically look at the file allocation table and check to see whether that cluster was free, and was the one next to it free, and what about the one after that, and after that, and after that? And so it would try to rebuild the original contiguous block of clusters to end up meeting the length of the file, which was also stored in the directory. So there was a lot of information there. Peter did the work and created the world's first undelete utility and made a lot of money.

Leo: Mark Goldstein, Northern Virginia, notes the world is changing. Mark, wow, really? Oh, I shouldn't be sarcastic. For the past 15 years I've been in cybersecurity and privacy. Whenever I would meet someone at a cocktail party - whatever happened to cocktail parties? That's another thing gone; right, Mark? They would ask what I did for a living, and almost immediately after hearing they would suddenly realize, I think I need to freshen my drink. But lately, with cyber-something on the front page news, now everybody wants to ask questions about security and encryption and so on. Even the U.S. President is talking about encryption and the need for government monitoring, metadata and such. We cyber geeks are becoming much more relevant now. Have you noticed that, Steve? You get access to cocktail parties?

Steve: Ah, yep. When I'm involved in computers and security, people are like, oh, Internet security, oh, yeah.

Leo: Ooh, very nice.

Steve: Yeah. That's a big thing.

Leo: Very nice.

Steve: Then they go freshen their drink.

Leo: Yeah. It's a short conversation, I've noticed.

Steve: You finish a sentence, yeah. They don't run.

Leo: Mike Woolard in Ohio wants to use SQRL in a multiuser environment: My company is looking into implementing SQRL with our web applications. More than 30 percent of our customer service calls are password related. That's probably low for some companies. And SQRL should zero that. We offer a SaaS - Software as a Service - B2B solution, where the users of the application are generally not tech savvy, and in an environment where multiple users tend to share workstations to use our application. That would make sense. They'd have to have separate passwords, each of them. Because of the password issue - oh, they also tend to share accounts. That's not what we want. Well, that's one way to get around it. Why have extra passwords?

We want to avoid having the user logon and off Windows to use SQRL for authenticating onto our sites. I see from SQRL's documentation there's a flag that can be set to always ask the user to select an identity. And he's reading up on SQRL at GRC.com/SQRL. Just wonder how the details - how this works, if you've had any other suggestions for implementing it in an environment like this. Thanks, and I am a SpinRite user, he says, for five-plus years. That's an interesting use.

Steve: Yeah. Actually, I anticipated this because of the, I would say, the multi-SQRL household, but actually the multi-user household. You know, many places have a shared machine. Sometimes the computer is yours, and there's no one logging into it. That's like the case for me, my main workstation and so forth. But many families have accounts for all of the different people in the household. With SQRL, you can, first of all, have multiple identities. You don't need one for yourself. That's the whole point of SQRL. And in fact I go through some lengths to make sure, if they're creating another identity, that it's actually for another person, that is, I don't want them to think, oh, I need to create an identity for every website. So no, no, no, that's the whole point, one ID, maybe for the rest of your life.

But you wouldn't want your son to use your SQRL identity. Now, unless you gave him your SQRL password, he couldn't because you need - you are prompted for that in order to make sure that you didn't walk away and leave the computer on and with SQRL able to stand in for you to authenticate you all over the Internet. So there's two choices. You can change the identity that SQRL will use. So you can create multiple identities for multiple people, and name them, you know, Johnny, Dad, Mom, Susie. And if you realize that, when you sit down, oh, SQRL is set to Dad, and you see that very promptly in the first dialogue, and if you don't change it, your password, your password won't work for Dad's SQRL identity. So you can change it.

But it occurred to me, in an environment where people might be frequently changing, setting an option in SQRL so that it always, it proactively asks you to select from a dropdown list box which user do you want to authenticate as, that's how we handle that. So it's one additional step. But it's very quick, and that just allows you to say, yeah, I'm Joey. And then Joey types in his password, and off he goes with SQRL, with Joey's SQRL identity being the one that will authenticate him on the Internet. So, Mike, it's entirely operable in your situation. It'll work.

Leo: Excellent. A good use for it, in fact.

Steve: Yeah.

Leo: In a previous episode, Steve - this comes to us from Steve in Alton, Illinois. In a previous episode you mentioned that DDoS attacks could mostly be prevented if all ISPs checked for traffic having origination IP addresses that are outside their network - spoofed traffic - attempting to exit from the network. I'll rephrase this. If ISPs simply said no packets can come from our network without having an origination address within our network.

Steve: Perfect.

Leo: So I'm wondering why this seemingly simple precaution isn't being done. Would this stop all of these attacks or just certain kinds? For instance, if the server being attacked was within the same ISP as the attacker - well, I'll let you explain it - or they used UDP or SMTP instead of TCP/IP, would doing this cause any other networking-type problems with, for instance, VPNs or other types of forwarding? Why aren't ISPs doing this? Or is it just because it would be expensive? Why, Steve, why? I want to know the same thing. Why not?

Steve: Yeah. I mean, many in the industry are wondering. And this is the tendency that Google is being so proactive about over on the security side. It'd be wonderful if they had some way of enforcing this with ISPs. It is just laziness. It doesn't seem to be a big problem. It's called egress filtering. You filter the traffic egressing from your network. Ingress is coming to you. Egress is leaving. And so all you have to do is drop packets that do not carry a source IP originating from that network.

Leo: And I can't think of a single reason why anybody would legitimately be doing that.

Steve: There is none.

Leo: There's no reason why something coming from your network shouldn't have an IP address within your network.

Steve: Correct. Now, it does not...

Leo: It doesn't because they're spoofing IP addresses.

Steve: Exactly. That's the only reason. It is a breakage of the fundamental IP layer for anyone to put an address other than their own in the outgoing packet. And in fact you can't do it. OSes will not let you do it. At the software level, you have no control. And...

Leo: This was why you were upset about raw rockets in Windows XP.

Steve: Exactly. You should not have that control.

Leo: But if you're a Linux user, you can do it, obviously. I mean, their software, the operating system will let you do it. Windows Server will let you do it.

Steve: UNIX has always allowed it, if you had root privileges. Even there, UNIX original guys said nobody should log on as root. Nobody should be able to overwrite the source IP in outbound packets. And so if you're not logged on as root, and you can't overwrite the source IP, you can't do it even on UNIX.

Leo: Yeah.

Steve: But the problem was raw socket technology was there, and bad guys get root privileges. Then they can do it. So the final piece of Steve's question is does this completely solve the problem? No. But, for example, the traffic, recently we were talking about DNS reflection attacks where you send a little query to a DNS server, and it sends a big response back. You want to spoof your source IP when you send a DNS query out to Google's DNS servers so that it sends it back - it thinks it's sending it back to you. It sends it back to your victim, instead.

Where this won't help is TCP. So, for example, if HTTP queries are being used to flood a server, an HTTP server, with valid requests for pages, those cannot have a spoof. So TCP flooding would be blocked by egress filtering. The fact is all ISPs should be doing egress filtering because it blocks all the SYN floods and the DNS spoofing attacks. It blocks a huge class of them. Not all, but enough that it's worthwhile, and it's so simple to do. They just don't because they didn't do it last week or the week before, and everything seems fine.

Leo: You know, the same, there's kind of an analogous situation with port 25 and spam. For a long time, ISPs would allow you to basically serve, have an email-sending device on your local PC and spam people.

Steve: Right.

Leo: And eventually - and one of the reasons they didn't disable port 25 is because it would cost them a lot of money, they thought, for all the tech support calls they'd be getting. Why can't I use port 25? I don't think they got any, but let's say that's why they didn't. Eventually, just public opinion forced them. Most ISPs now block port 25; right? They don't...

Steve: Right.

Leo: They don't allow you to have your own mail server.

Steve: Right, 25, and the Windows filesharing ports, those are all blocked, 135...

Leo: 139, 135, yeah.

Steve: Yup, 135, 139, 443. You just can't use those because they've been so troublesome in the past.

Leo: Right. And it is probably the case when you're a Comcast, and you have literally tens of millions of customers, that even if one tenth of 1% call your service, call your help line because something's not working - but I can't think of anything that would be broken by preventing that.

Steve: And we can see the quality of the service that Comcast provides.

Leo: Oh, yeah, all right. Well, that's one of the reasons; right? It's expensive to give you good service.

Steve: Exactly. They can't afford to put UNIX gurus on the phone.

Leo: Right. Also very important mitigation would be to get rid of these amplification attacks. NNTP servers and so forth need to be patched.

Steve: Yes, yes.

Leo: Because you still probably don't want to use your own IP address when you're launching an amplification attack. But...

Steve: Oh, no, it'll just come right back at you.

Leo: Oh, that's right. You need to spoof that there, yeah. You're going to have to spoof that.

Steve: I just flooded myself, oh, my god.

Leo: That's what I would do. Let me just see, I'd like to try an amplification attack. Wait a minute, where did my Internet go? Good point. You have to spoof the IP address. Peter Sysak, S-Y-S-A-K, in Ontario, Ottawa, Canada, wonders about SpinRite parallel operation: Just bought a copy of SpinRite - smiley face - to use for various purposes, one of which being to maintain my five 2TB NAS drives. Five drives, each 2TB. I'd like to know if SpinRite will process the drives in parallel. In other words, can I plug in all five, or two, and have SpinRite work its magic in parallel? I'm currently running drives one at a time, so doubling up or more would be terrific. Because, as we know, SpinRite could take a while. Thanks, Steve. Keep up the great work. I emailed you guys a while ago and mentioned I had started listening to Security Now! from No. 1. I am catching up. Let's see. We're at Episode 496, soon to hit our 500th episode. Where is he? Episode 157. You've got a way to go, dude. Can't wait to get caught up.

Steve: Hey, but he's going to be fully tuned up by the time he gets there.

Leo: Yeah, good news.

Steve: Okay. So as soon as SQRL is wrapped, and I should say, I haven't been talking about it much, but we're making great progress. There's now a full login demo account system is online. A SQRL protocol diagnostics is online. There's a Mac OS X client is up and running and in beta. iOS client is up and running. Drupal, and there's some X. Is it Drupal 7 or, no, Drupal 7, is that something?

Leo: Drupal 7 is the latest version, yeah.

Steve: Okay. Then we have that is up line. A test server, a command line client is in the works. I mean, it's all happening.

Leo: Wow, that's great.

Steve: So once that's behind me, my logic is to do something to hold everyone over while I completely rewrite SpinRite. So 6.1 is sort of the catch-up. That's why I'm making it free. It will reduce the time to run SpinRite on one drive to about 1TB per hour. So that means a 2TB drive will take two hours. So that's way faster than we've had. But it will not do multiple drives at once because none of the architecture in none of the existing infrastructure in SpinRite supports that. That is the absolute wish list, one of the main features of 7 will be that I will spin up and simultaneously run SpinRite on every drive you have, and they will all be running at 1TB per hour. I can do that. I'm writing it in assembly code. The hardware can do that.

So nothing is going to stop me. The reason I'm not doing it first is I need to bring SpinRite up to where it should be so that people are okay while I work on 7. And 7 will have a GUI and be file-system aware and all kinds of stuff. But that's a big project. So 6.1 will make it way faster and practical to run on today's multi-terabyte drives, whereas it's still painful. But everybody gets 6.1 for free.

Leo: Nice. Finally, Question 10, Andy in Alabama. He wonders, what are you doing with Java on your site? What? Steve, love the podcast, and I've heard you talk often about how unsafe Java in the browser is. I just noticed the Big Number Calculator on GRC requires the user to have Java installed. Hey, I understand it's served over TLS from your site, TNO aside. But wouldn't having Java installed open users up to malicious applets on other sites? With the recent JavaScript optimizations, is there some reason you don't use asm.js or something like that?

Steve: So it's a really good question. I have really no defense. I was using that Big Number Calculator, which I did not write, but, like, all the time when I was doing things like Password Haystacks and Perfect Paper Passwords, all those times where I wanted to know what is 2^326. And I don't want, like, oh, well, it's about 10^77 or something. I want digits. And this thing gives them to you. Anyway, it's just so cool that I decided to grab the Java and put it on GRC.com where I had easy access to it. And I thought, well, since I want it, maybe it would be handy for other people.

So, yeah, I guess to the degree that it requires Java - and, I mean, I have Java on my system. I need it for all kinds of things. Eclipse is Java-based. Other things that I use are Java-based. I mean, you know, real Java. But I'm also using Firefox with NoScript, and Java in the browser is completely disabled. In fact, there's no plugin for Java in Firefox. And I've got Java security turned up from, you know, Sun Java's control has it disabled in the browser. So it's possible to turn Java off for the browser now, in latest versions of Java, so that it's not a problem for the browser. You would need to turn it on for GRC if you wanted to use the Big Number Calculator that I've got there. But it's possible to have it installed and keep it away from your browser.

So, oh, and I also saw this because I saw the note about asm.js, and I got a big kick out of Microsoft poking Google in the eye because asm.js is the Mozilla solution for superfast scripting, superfast JavaScript. Google does not offer it. Microsoft is supporting it in Windows 10 and in the next version of Internet Explorer. So I thought that was - I saw that and thought, oh, okay, yeah.

Leo: That's interesting.

Steve: Maybe Chromium will consider it.

Leo: Yeah.

Steve: Yeah, it's neat. The idea is that it is a subset of Java which can be compiled much more efficiently. Java is a so-called "automatic language," where things like first use of a variable defines its type, and it determines a lot from context. And it automatically allocates storage and then has a garbage collector to determine as best it can when that's no longer necessary. All of those things impose a huge burden on the interpreter and end up slowing your code down. But you can use a proper subset of JavaScript that deliberately eschews any use of those things. And, boy, you can then compile that down to something that screams, which the beauty of that is you're still writing in JavaScript, hundred percent legal JavaScript. Or I should say the compiler is because you'd rather take something like maybe a C# compiler and compile that to asm.js. And then you've got something written in a high-level language which also runs on all the browsers that support asm.js.

Leo: I'm a little confused by the interchangeability of Java and JavaScript. You're writing Java in JavaScript? Is that what you're saying?

Steve: No, I'm sorry, you're right. His note brings them...

Leo: You've confused the hell out of me here.

Steve: Yeah. His note says...

Leo: Why don't you use asm.js, which is a hyper fast JavaScript.

Steve: Which is also...

Leo: And you could have recoded Big Num Generator. But you'd have to rewrite it. You didn't write the original.

Steve: Correct. I think he assumed I wrote it because it was on GRC, even though on the site I do have a link to where I got it from and gave credit to the original author because the guy did a great job.

Leo: We should also point out that it's fine to put Java on your system. Disable the browser extension.

Steve: Correct.

Leo: Disable browser support. Use Java locally only. And then Java is not any more dangerous than anything else.

Steve: No. Java, in fact, is a very mature nice language. The problem, the mistake Sun ever made was in saying, oh, think of how amazing it'll be if people can just download Java applets and run them in their browser. Yeah, somebody called Adobe thought that would be really good, too.

Leo: Well, you know, I have to say they did a lot about sandboxing. I mean, they knew that there was a risk to that, and they thought they were writing something safe.

Steve: After a huge amount of pain.

Leo: Yeah.

Steve: Remember how much pain we went through, I mean...

Leo: Right. Well, because of ActiveX we knew that the idea of running code in a browser is a bad idea.

Steve: Bad, bad.

Leo: Local code downloaded to your browser is a bad idea.

Steve: Right.

Leo: Anyway, it doesn't matter because that's ancient history. Nobody uses Java anymore.

Steve: No.

Leo: Anyway. And can I ask you one question, Question 11 from Leo in Petaluma, California?

Steve: Yeah.

Leo: We were talking about this on MacBreak Weekly, and I was curious about FaceTime versus Skype and whether one is more trustworthy than the other. Both claim to be encrypted, do they not?

Steve: Oh, I would - FaceTime, hands down.

Leo: But that's because you trust Apple over Microsoft.

Steve: Well, it's because I trust at least Apple is making all the right noises. We know that Skype has been compromised. We know for a fact that Skype security can be monitored under request from the NSA. I mean, if they shut down the relay servers so that now that no longer exists, Microsoft did a number of things specifically to allow eavesdropping on Skype conversations.

Leo: Ah, okay, okay. So, yeah. Because one of the things I remember is Gary Kasparov, who was a world champion chess player.

Steve: Yeah, chess.

Leo: He ran for President of Russia, and he said I will only use Skype because otherwise the Russian police are spying on me. He trusted it.

Steve: That's the old Skype.

Leo: Yeah, it was old Skype. It was pre-Microsoft's acquisition. So you're saying Microsoft has compromised Skype.

Steve: Proactively compromised Skype in order to make it a, I mean, and there was some logic to it. I mean, they explicitly said we need to be able to comply with telecommunications warrants if we're served with them. So under those circumstances we will. Whereas Apple is at least still saying we will refuse those.

Leo: Okay.

Steve: I mean, again...

Leo: It doesn't strike me as exactly a technical difference. It's more of a belief that one company is going to protect you more than...

Steve: If I had to, yeah, if I were asked which seems more trustworthy, I would say FaceTime is currently. But if you really needed it, you'd want to layer a third-party TNO application on top.

Leo: Right. I mean, use Secure Voice or Silent...

Steve: RedPhone or Silent Circle, yeah.

Leo: Yeah, because those are open source encryption technologies, and you know those are secure.

Steve: Yup.

Leo: Whereas here I just - it strikes me you're saying - not you particularly, but the people in general are saying, well, I trust Apple. And I think that that's a - I love Apple. I'm not saying don't trust them.

Steve: Oh, and I've been screaming about the fact that I...

Leo: And Trust No One is trust no one, not trust no one except Apple.

Steve: And the problem with iMessage, for example, is key management. Apple does your key management. And if you're outsourcing your key management, you no longer have TNO.

Leo: Right.

Steve: Because they could easily, I mean, when I'm sending an iMessage out to multiple people, Apple has sent me all of their public keys, which I use to multiply encrypt the message out to each of them.

Leo: Right.

Steve: Nothing prevents Apple from tossing the NSA in there. There's no visibility into key management. That's the problem.

Leo: Yeah, and even the EFF in its scorecard which we've mentioned before...

Steve: Great, great site, yup.

Leo: ...talks - and Cryptocat is all green checks.

Steve: Yup.

Leo: FaceTime, I have to point out, is not all green checks.

Steve: Yup.

Leo: But neither is iMessage, Kick Message, a lot of solutions. There are really only a few that are, you know, OTR encryption is the way to go, and only a few - Signal, RedPhone we mentioned, Silent Text we mentioned. Telegram, TextSecure, there are only a handful. And FaceTime's not one of them.

Steve: No. I mean, it's going to be the tools that are explicitly privacy enforcing.

Leo: Right. Okay. I just wanted to clarify that. Mostly I wanted to understand, is it a technical thing, or is it kind of a - it's almost political. It's who you trust, who you believe.

Steve: The difference between them, yes.

Leo: And we don't know of a technical - we suspect that Microsoft is, well, they even said they are, so I guess...

Steve: Yeah, they have said they are.

Leo: The problem is secret courts. That's the real problem...

[Crosstalk]

Steve: Yeah, yeah, exactly.

Leo: That's the real problem.

Steve: Yeah. Well, and arguably an outlaw law enforcement. If you're hacking Gemalto's network in order to steal the private keys of the SIM cards, that's outlaw law enforcement.

Leo: Right.

Steve: I'm sorry.

Leo: It is.

Steve: You know? Yeah. And you're doing it to circumvent, oh, the court system that was put in place for allowing you to obtain those keys legally. You're saying, oh, it's too much trouble. We'd just rather have them all. Trust us.

Leo: Yeah, well, the other thing they say is, it's okay, they're not U.S. citizens, so who cares?

Steve: Yeah.

Leo: Who cares? They don't count. Ah, what a world. Steve Gibson...

Steve: Indeed.

Leo: ...thank you for making this a much, much safer world for all of us. If you go to GRC.com you can find this show, 16Kb audio of it, full text transcripts written by Elaine that make it very easy to follow along. He also has SpinRite there. That's his bread and butter, the world's finest hard drive maintenance and recovery utility.

Steve: Used to be the world's finest low-level formatting utility, but began to do...

Leo: Can you do low-level formatting through SpinRite?

Steve: I took it out.

Leo: Turned out the [crosstalk] stuff was better.

Steve: Yeah. I mean, in fact, somebody had an old, old machine, and neither 5 nor 6 would run on it. So I sent him a copy, actually I opened, I have a copy of 3.1 on the shelf up there. I opened the box and took out the diskettes. And then I think I mentioned how hard it was to get the copy of 3.1 to him because email is so tightened down these days. So finally I put a link on the server and said, here, click the link, then tell me when you have, and I'll take it off.

Leo: That's right.

Steve: So but that one will still - that will do low-level formatting on a drive from the '80s.

Leo: I love it. If you want audio and video of this show, you can get it from our site, as well, TWiT.tv/sn, YouTube.com/securitynow, iTunes, wherever you get your podcasts. Or get the app. There are podcast apps, yes, but also there's TWiT apps on every platform. We don't make them. We thank our third-party developers who do that. You'll find them there. Steve is @SGgrc on Twitter. Follow him there for updates all week long. It's a good place to ask questions, too, or GRC.com/feedback. In two weeks we'll do a feedback episode. What are we going to do next week?

Steve: Car hacking.

Leo: Oh, that's right, we've got the car hackers themselves.

Steve: Yes. We've got the guys who did the actual hack and can tell us how they pulled it off with all the kind of detail that our audience wants.

Leo: Nice. Can't wait.

Steve: Yes, yes, yes.

Leo: Thanks, Steve. We'll see you next time.

Steve: Okay, my friend.