SECURITY NOW!

Transcript of Episode #501

Listener Feedback #209

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-501.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-501-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. I'm here. We're going to talk about the latest security news. Yes, another survey of bad passwords coming up. And then 10 great questions from you, our audience members. Stay tuned. Security Now! is next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 501, recorded Tuesday, March 31st, 2015: Your questions, Steve's answers, #209.

It's time for Security Now!, the show where we protect you and your loved ones online, your privacy, your security, with this guy right here, the Explainer in Chief, Steven "Tiberius" Gibson. And he is here once again to both put us all on edge, and then make to us feel better about...

Steve Gibson: Starting into our second set of 500 podcasts.

Leo: 501.

Steve: The first set of 500 is done. I should mention to people who are wondering where 500 is, I just realized, when someone sent me a tweet saying, "Hey, Steve, when are you going to put up 500," that I never got the transcripts from Elaine. In 500 episodes, this has never happened. She is sort of in a perilous strange world where there's like snow on cactuses. I've seen photos from her porch. And I don't know where, but, I mean, there's, like, huge windstorms. There's electricity going out. I make the small bandwidth audio because she had been bandwidth constrained, but I think her satellite provider fixed that - I'm looking up at the sky because that's where the satellites are - fixed that a while back. Yeah.

Leo: They're up there somewhere.

Steve: Anyway, I mean, but there's a lot going on in her world. She's also got all kinds of animals all over. There's a menagerie, and they're getting sick and having to go to the vet, and things are happening. So I sort of, I get a little sort of side channel of this through the last 10 years that we've worked together. But never has she gone radio silent.

So what I'll do is, after the podcast - and so the reason I didn't post 500 is I'm sort of a - I have a system. And when the transcripts arrive from Elaine, that's my trigger to assemble everything, convert to PDF, convert to HTML, get everything up and posted. Anyway, I will put everything up minus the transcript for 500, and we'll hope that she's okay and that I hear from her. We will be transcriptless for a while one way or the other, but I will alter my routine so that I can operate without the transcript trigger to cause me to post this.

Leo: How do you do that? Do you do a little cron job or a watch folder or...

Steve: You know, I often say that investments in infrastructure come back to reward you over and over and over. And I have something - if I can do it here. I'm going to type - I'm going to move the microphone down so you can hear it, and I'm going to type "get 500."

Leo: Oh, you mean you can actually publish it in real time. He's going to - okay, there's Steve is now typing "get 500."

Steve: I guess you couldn't hear it. It said "Auto-initiating media workstation monitor."

Leo: Auto-initiating media workstation monitor.

Steve: Anyway, so that's a system that I wrote that periodically checks for the appearance of the audio from you guys. And so that alerts me to when the audio appears. But here's what is just so dumb. And this is a little bit like the cobbler's kids going without shoes. Everybody else in town has shoes, but not the guy that makes the shoes. I have manually done this 500 times.

Leo: Wow.

Steve: Manually ran a - I have a Perl script that converts the text file from Elaine to a CSS-laden HTML document. Then I bring that up in IE because its PDF printing is better than the other browsers, print that to a PDF, then collect all of those and blah blah blah blah blah. Anyway, all manual, and so stupid. I mean, again, if I had invested in a little infrastructure - I just didn't think this was going to go for 10 years. Like I said, oh, you know, maybe, you know...

Leo: A few, five, whatever.

Steve: Maybe Leo will get tired of The Cottage and, you know. Or it'll be a heat wave, and we'll just shut down and never start up again. I just didn't know. Had I known, I would have done things differently. And here I am, manually doing this. And at this point I'm so busy I don't want to take out any time to do it, so I'll just keep posting it manually so that I can get SQRL done and then SpinRite 6.1. I have a whole big growing list of things I promised myself I would do once SpinRite 6.1 is published.

Leo: Wow. Good for you.

Steve: We'll hope that happens.

Leo: That's awesome.

Steve: So this is a Q&A. Last day of the month. Tomorrow is April Fools. And it's a good thing the podcast did not fall on April Fools.

Leo: Thank god. I hate April Fools. I hate it.

Steve: I do, too. I do, too, because you have to, like, look askance at everything that you see.

Leo: Yeah, can't trust anything.

Steve: Wonder, like, okay, wait a minute, you know. And invariably every major site will do some April 1st goofy thing.

Leo: Not me. Not me.

Steve: And we have eschewed that in the past, eschewed or something like that, and we will in the future. So anyway, Q&A #209 for the beginning of our second set of 500 episodes. Now it's clear to me this is never-ending, so I...

Leo: Do a little infrastructure, will you?

Steve: Do a little infrastructure building.

Leo: When you say infrastructure, it's really writing software or writing a script;

right? It's not - you're not laying concrete.

Steve: It's like, I can look up right now and see total SpinRite sales for the day so far, total from yesterday, total week to this time, total for last week, month to this time, total for last month and so on. I've got bar graphs. I've got bandwidth stuff. I've got all kinds of metrics going on so I can see exactly what's happening. And so these things I built. And oh, well, you know, now "Yabba dabba do" comes out of my phone when I'm like in a restaurant. And only my brother-in-law thinks it's annoying. Like when I was up for Mom's birthday, he picked Jenny and me up at the airport. And a "Yabba dabba do" came out of my pocket, and he says, "That's annoying." It's like, okay. So I silenced the phone for my visit. But anyway, so, yeah, that kind of infrastructure stuff is fun.

Leo: Cool.

Steve: That kind of stuff, yeah. It's very much like your timer. That's something you invested in, and now it's just going to give back, you know, as you said, wasn't that big a deal. But what a convenience to have that. So what I've seen is those kinds of investments just pay off so well. Or like the little buzzer I have that lets me know when UPS is walking to the front door. Just, you know, so that the package doesn't get stale.

Anyway, not much happened, although some interesting things. We've got to talk about the GitHub and GreatFire.org DDoS attack because the mechanism of this is really interesting. There was a very disturbing vulnerability discovered in routers that are often used in hotels, convention centers, visitor centers and so forth. A really fun analysis of 10 million passwords. You're going to want to find the link low down in the show notes, Leo, and get that PDF, or get that page ready because...

Leo: All right.

Steve: Just tons of interesting stuff there. And this being a Q&A, we've got 10 great questions, comments, and thoughts. And I was a little bit light on the front end because there's some deep stuff we're going to be doing in the Q&A, some questions that deserve some good attention. So I think we have, for 501, a great podcast.

So the picture of, well, we'll skip the picture of the day. The first page of the show notes I always put something up if I can.

Leo: I love this one.

Steve: It's sort of a fun - yes.

Leo: You sure you want to skip it?

Steve: Well, we'll be coming back to it.

Leo: Oh, okay.

Steve: And we'll just note that "monkey" is now number 15.

Leo: Moving down.

Steve: Yeah, it is, you're right. I think it was 10 for a while; wasn't it?

Leo: Yeah, "monkey" should be higher.

Steve: Yeah, "dragon" has now taken monkey's place at number 10.

Leo: Oh, okay. We're not talking the Chinese zodiac here, folks.

Steve: Yeah. And there are some that we cannot pronounce on a family-oriented show. But anyway. Okay. So there's been something going on for about a week. Apparently the first site to come under problems was GreatFire.org. And we've not talked about them before. But as I understand it, their mission is to make available Internet content which is otherwise blocked by China's censoring Great Wall Firewall.

Leo: I think we have talked about them.

Steve: Well, but maybe not in any more depth than just that.

Leo: No, because they were the ones who - what did they do? They identified something - oh, I'll have to go back in time. But, no, I'm pretty sure we have talked about them.

Steve: Okay.

Leo: Anyway, they're not merely a place to echo this stuff. They are monitoring what China is doing with this Great Firewall.

Steve: Ah, okay. So they're using Amazon's cloud content delivery network in order to host the content that they are mirroring and making available. And since many of China's own businesses also use the same CDN, it turns out that it's extremely difficult, and so far has actually been impossible, for the China censors to block these guys, the GreatFire.org people, because they're using a common CDN to many huge Chinese properties.

So then on March 26th - and so this GreatFire.org had been having problems with attacks like for a week or two before this hit GitHub. And when it hit GitHub, it sort of came of course to the tech community's awareness because GitHub is a resource that is beloved by many of us. There's, for example, I don't know how many SQRL projects are currently hosted on GitHub. But, like, 40 or something, I mean, it's amazing. You go to GitHub and put "SQRL" in, and although some were quickly created and have been abandoned, many are underway because I'm talking with their authors on a daily basis at this point.

Anyway, so the following message was posted on the 27th, which was last Friday. GitHub posted: "We are currently experiencing the largest DDoS (distributed denial of service) attack in GitHub.com's history. The attack began around 2:00 a.m. UTC on Thursday, March 26, and involves a wide combination of attack vectors. These include every vector we've seen in previous attacks, as well as some sophisticated new techniques that use the web browsers of unsuspecting, uninvolved people to flood GitHub.com with high levels of traffic. Based on reports we've received, we believe the intent of this attack is to convince us to remove a specific class of content."

And as we'll see in a minute, GitHub hosts some tools which are used for circumventing Chinese censorship. And so they are, based on the attacks they were seeing, it looked like they were under attack because they were hosting those tools. So how this attack works: Millions of global Internet users, so this is not just Chinese users, although China is where the problem is injected. Anybody anywhere in the world visits thousands of different websites, hosted also anywhere in the world, whether inside or outside of China, will randomly receive malicious JavaScript which is used to launch an attack against GreatFire.org's websites and, more recently, GitHub, which is hosted - and this is content hosted by Amazon cloud services.

It turns out that, much in the same way that people - like many, many, many websites use Google Analytics. When you use Google Analytics, you embed an invocation of some of Google's JavaScript into your website so that the pages you display invoke that script. That creates a fetch from your browser to Google Analytics and allows them to see who you are, thanks to cookies, and where you are, and basically supply analytical information back to the webmaster, the owner of that website, using very nice Google-based UI.

Well, it turns out that the Chinese site Baidu, B-A-I-D-U, has the same thing. There is Baidu Analytics code which is just h.js. They want to keep it small. And so what happens is, when sites anywhere in the world are hosting Baidu's Analytics code, much for the same sort of purpose that other sites might use Google, that means that the people visiting those sites, who are also located anywhere in the world, will cause their browser to fetch Baidu's h.js JavaScript. Somewhere inside China, before the traffic gets to the Baidu servers, some entity, and it's now believed to be Chinese authorities, are intercepting the incoming query for the h.js file and replacing it with attack JavaScript.

So the users, users all over the world, their browsers retrieve this maliciously intercepted and replaced Baidu Analytics JavaScript with code which has been analyzed, I've got links in the show notes to the deminifying and analysis of this script, which uses the web browsers to then pull web content from two different URLs, GitHub.com/greatfire and GitHub.com/cn-nytimes, which is a Chinese language version of The New York Times, which the Chinese government objects to people having access to. They block it and don't allow people to have access to it.

So what has been seen was a phenomenal amount of web traffic. And the problem with this sort of attack is that it is not a protocol, like a lower-level protocol-style attack like a TCP flood, where you can say, oh, okay, like a TCP SYN flood. This is a higher level HTTP protocol attack, meaning that valid queries, there's nothing invalid about them except the number of them, valid queries are being made over HTTP to those pages at a level that essentially creates a denial of service, completely floods the servers. They have been, while this attack has been going on, they've been seeing 2.6 billion requests per hour, which is an increase of 2,500 times over the normal level of traffic that they see.

Leo: It has to be an amplification attack; right?

Steve: Well, yeah. I mean, the problem is, once this script starts running, it itself, the one instance of this malicious script starts making queries over and over and over and over.

Leo: Of course. It's amplifying itself. Right.

Steve: Right. So people all over the world...

Leo: Oh, gosh.

Steve: ...are using, are going to sites all over the world that are getting this malicious script into their browsers all over the world that are then just firing off queries, bang bang bang bang bang bang bang bang bang, using JavaScript to pull the content of these pages, thus creating a devastating and very hard to block, I mean, basically you have to do upstream protocol-level blocking. It requires some sophisticated filtering. And the problem is this is expensive. That is, even though these requests are coming in very fast, because the content is being served by Amazon, Amazon's CDN is designed to handle serious traffic. Consequently, this is costing GreatFire.org $30,000 per day in Amazon billing because Amazon is a pay-go, you know, you pay as you use it. And suddenly their apparent usage has spiked.

Now, one question is whether Amazon will get onboard and not actually charge for the bandwidth which is being consumed. GreatFire doesn't know yet whether they're going to do that or not. But apparently they have asked Amazon if they would consider forgoing this because this is not legitimate bandwidth, this is clearly an attack against these guys.

Leo: Wow.

Steve: But anyway, I just thought this was interesting. It's an interesting hack which is difficult to fix because Baidu is saying we're not doing anything wrong. We're not seeing anybody asking for our analytics services any longer. No, because that's being intercepted, and those queries are returning malicious code from someone, some intermediary. And it has to be Chinese authorities because you've got to be able to intercept major pipeline traffic inside of China, and at the protocol level. I mean, again, because this is not down at packets, this is protocol. That's a much more difficult hack to pull off.

So anyway, I thought the mechanism was interesting because basically it turns innocent - thousands of innocent users browsing the web are now having their web browsers running malicious JavaScript, that they innocently got, which is pounding on two sites that those browser users would probably wish not to be doing, if they had a choice.

Leo: Yeah, amazing.

Steve: Yeah. Now, one thing could happen. This is the sort of thing where, for example, Google could respond by pushing out an update to Chrome that is aware of this behavior and won't execute that JavaScript or, for example, will detect that redundant queries are being made in fast order to GitHub properties and do a rate-limiting on something. I mean, it's the kind of thing that the browser could do. Otherwise it's going to require some sort of filter to be put in somewhere. Wow. Really interesting. And it's another consequence of the fact that our browsers are running executable code that they receive from any website they visit, which is a mixed blessing. We love the convenience of that. But as all of these conveniences that we enjoy, they can be turned to a dark purpose, as well.

A security site, Cylance, discovered something very disturbing, which is that a very popular router, manufactured by a company called ANTlabs in Singapore. The router family is called InnGate, I-N-N Gate, as in an inn, a hotel. And for whatever reason, these are extremely popular routers used by hotels, convention centers, visitor bureaus and such, to manage traffic and allow people to get online. I mean, there are thousands of them in the world. What the Cylance folks discovered is one of the most basic sorts of flaws you can have, which is an open rsync daemon running on TCP port 873 on all of those routers. Excuse me.

Leo: Rsync is used by UNIX heads to do backups.

Steve: Yes.

Leo: I use it. It's great.

Steve: Yes. And in this case it's a little too great because it turns out that the instance of rsync included with the InnGate firmware is incorrectly configured to allow the entire file system to be read and written without authentication.

Leo: Oh, that's not so good.

Steve: So a remote unauthenticated attacker can read or modify any file on the router's file system.

Leo: Wow.

Steve: So this is TCP port 873 running rsync. You can, if you were to scan the 'Net and find listening ports 873, you then go to your Linux box, and you type rsync, space, and the IP address, you get a listing of the root of that router's file system.

Leo: Wow.

Steve: I mean, it's just like that. So there have been some interesting reports of attacks on hotels where - I remember there was - some Asian high-end hotels were having their visitors compromised after visiting. And it turns out that - so all kinds of things can be done with this. In a quick scan of the Internet, the Cylance folks found 277 immediately accessible devices in 29 countries. On their page a ways down, on their blog page talking about this, there's a map of the world, and it looks like most of them are in the U.S. A hundred of them were quickly found throughout the United States.

So one of the problems - yup, there's the map. One of the problems is that the severity of the issue is escalated by how little sophistication is required for an attacker to exploit it. I just explained in a sentence how anybody can do this. I mean, that's how bad it is. And, I mean, I'm not giving anything away because this is all over the security community, and people are scrambling now to deal with this. But an attacker exploiting a vulnerability would have access to launch attacks against guests on the affected hotels' WiFi. Targets could be infected with malware using any method, from modifying files being downloaded by the victim or by directly launching attacks against their now-accessible systems. Given the level of access this vulnerability offers to attackers, there's no limit, apparently, to what they can do.

And, for example, if an attacker had compromised an InnGate device at a hotel with the vulnerability, they could obtain shell access via SSH, which is running, by the way, in those routers, in order to get root access, and then run tcpdump, which is also present on these devices, to dump all the network traffic going through the router. Any unencrypted traffic, anyone who is, for example, seeing session cookies that are not being maintained over a secure connection, but as used to be the case, remember like with Firesheep, where you negotiate your logon credentials securely, but then the site drops you back. High-profile sites are no longer doing that. You know, Facebook and Google and most others are now HTTPS all the time.

But any other site that is not keeping you secured throughout your entire session means that your session cookies, your logon authentication, is freely sniffable by this kind of an attacker. So, again, we're well past the time where any site that uses persistent logon can ever allow nonsecure, nonprotected session cookies to happen. And unfortunately, hundreds of thousands of sites still do that. Those that haven't deliberately switched over are still vulnerable to that kind of attack. Or, for example, a tool like SSLstrip, which removes the S's from the HTTPSes in order to prevent HTTPS from ever having a chance to get started, could strip out even a site's best effort in order to protect users.

So anyway, this is a bit of a sobering vulnerability, and it's one more reason for our listeners to remember that any places like open WiFi or hotel WiFi, or even hotel wired networking, if anyone still uses wires in a hotel, these are just really unsafe places not to really be secure. The solution is to always use a VPN in those circumstances. Use a VPN. Even though we know that your traffic will be coming out of the VPN server at some single location, you just want to wrap a secure tunnel around your local traffic until it gets out, like out into the wider Internet, and away from this last-mile region where there are just so many potential vulnerabilities like this.

There is a site, WP Engine, which is a major WordPress hosting site that did a beautiful, I mean, just a really fun analysis of two different password databases. They pulled them together and did a very nice analysis, one of the nicest that I've seen in a while, that I thought it would just be fun to talk about for a second. For example, because of the depth of information they had about the account holders, they've been able to give us some demographics that we haven't seen before. For example, they did a chart of the birth decades from, okay, of the 10 million total accountholders whose passwords they have, 220 of those accounts included date of births. So they were able to show a chart. And the highest incidence of date of births of people in this 220,000 subset nearly looks like maybe 95% - wait, this doesn't sum to a hundred.

Leo: That's not percent. It's 95,000.

Steve: Oh, it's in thousands, it's in thousands, okay, right, 95,000. So but it is...

Leo: Percent would have been more useful because this is just raw numbers. I mean, there are probably more users from that cohort.

Steve: We do know that it's of 220,000, so 95,000 of 220,000. So what's that? That's less than 50%, but not way less. Anyway, for example, the born in the 1980s is more than twice as many as the next greatest, which is born in the 1990s, which is about half that amount, and a little bit less than that was 1970s. And then it rapidly falls off. So sort of interesting demographic. Also, where they had gender information, it was almost exactly two-thirds male, one-third female. That is, 485,000 credentials that had gender information, two-thirds male, one-third female.

And then we have, of course, the ever-popular list of the 50 most-used passwords. Still solidly in first place is our all-time favorite, "123456." And the runner-up for first place, of course, is just "password." And it's amazing that sites still allow you to enter that as a password, or even 123456. It just, you know, these must be old sites because I don't think you could do that today.

Leo: No.

Steve: In most sites. They just won't let you get away with that. Third most popular, add a "78" to the end of "123456." Then of course "qwerty" is there. Anyway, the link is in the show notes for anyone who's interested, or it's just WPEngine.com/unmasked. So if our real-time listeners are curious, or somebody listening to the podcast near a web browser, WPEngine.com/unmasked - U-N-M-A-S-K-E-D. As we were noting earlier, "monkey" has fallen down to the 15th place. Actually just looks like it's been pushed down by a lot of simple numeric passcodes, or passwords, which is disturbing. "Letmein" is still there in number 16. For some bizarre reason, "michael" is number 20. "Mustang" is 21. And it sort of goes on there. So the top 50 are those we've seen before.

I got a kick out of the next thing, which was they looked at when a site tells you that you must have a number, you know, a digit in your password, a password cannot be all alphabetic, you must have a digit. Well, what do people do? There's one of these heat maps showing where the area is the number of times it occurred relative to the others. And almost a quarter, just about a quarter of them is just the digit "1." So if a site says, oh, no, you can't just use that, you've got to have a digit, so someone says, okay, fine. Oh, actually there is a percentage. It's 23.84% simply put, literally, the digit "1" after it. Now, there's a slightly more creative group, 6.72%, instead use the digit "2." We drop to about half that, to 3.86%, who just jump right to the digit "3." They don't mess around with "1" and "2" first. And so on.

But again, it's some interesting demographic breakdown. And it actually sort of goes almost numerically, "123," then "12," then "7," then "5," then "4," then "6," then "9," then "8." But basically, if people are told they have to have a digit, they just tack one on the end, and almost a quarter of the time it's just the digit number "1" to satisfy the requirements.

And then the last thing, scrolling way down, Leo, the keyboard patterns. This is the other analysis that I thought was really interesting, that I hadn't seen before, which is they looked at passwords and mapped them to where the keys were on the keyboard. And so this page shows the 20 most common keyboard patterns occurring within those 10 million passwords. And as we already said, of course, there's "qwerty," which is just the first six keys in lowercase across the top row from left to right of the keyboard. And then the second most common is you keep going a little bit further. Maybe you think, okay, well, eight letters. Or six is not enough; I'll go for 10. So then they keep going to "uiop," which actually is all of the top alphabetic characters on the keyboard.

Then some people get a little more clever. The third most common is "1qaz," so that's coming down the leftmost diagonal from "1" down to "z," so the upper left to the lower left. Then, since they haven't really gotten critical mass yet, they go back up and do "2" and come down the same, you know, the adjacent diagonal to "x." So that's the third most common. And then the fourth is they don't go up into the numerics at all. They probably will after they get spanked for not doing that. But those guys just go diagonal staying in the alphabetic. Anyway, it turns out that there are many of the passwords that have been found that look a little bit gibberish-y, also look sort of suspiciously familiar. And it's because in fact they are just derived from some sort of a linear sequence of key tops on the keyboard.

Leo: I'm trying to figure out this number 20, which, by the way, he actually poses as a puzzle.

Steve: That's interesting.

Leo: I haven't read the answer.

Steve: You're right, that's completely...

Leo: "Adjmptw." It is alpha, I know it's alphabetical order. And it is from left to right, "adgjmptw."

Steve: I don't see a pattern.

Leo: Yes.

Steve: A-D-G skips every other. But then J doesn't, otherwise it would be K. You're right, Leo, that's - okay. And apparently that's happened many times. That's the 20th most frequent occurrence. Oh, there's an unimaginative one, "mnbvcx." So that's the lower row of alphabetic right to left. Boy, we really need to get away from passwords.

Leo: We need to get away from keyboards.

Steve: We really need. And then we've also got - they analyzed on iPhones with the keypad the 20 most used, what they called "key walk." Or am I confusing that with what's above? No, because that's keyboard, and they are definitely showing it on the iPhone, although they're not showing, they're not breaking it down in the same fashion. But so the idea is that people are doing a simple series of 10 keypad entries and creating passwords that way also.

And then what is this, the 10 most common word selections in 10 million passwords. So again - oh, they broke it down by category: fruits, animals, anything begins with...

Leo: Colors, noun, verb. Ilove..., my..., days of the week.

Steve: Yeah.

Leo: Number one name in passwords, John. Number two, David.

Steve: Number one superhero, of course, Batman. Number two, Superman. Then Ironman. Who's Hawkeye? Hawkeye Pierce?

Leo: No, come on, don't you know your superheroes?

Steve: I don't know my superheroes.

Leo: Yeah, yeah. Well, Hawkeye's a superhero.

Steve: I would have thought Spiderman would be higher up there. Who's Gambit? Is Gambit a superhero?

Leo: You're asking the wrong guy. I'm not...

Steve: Thor I know. Wolverine I know. Punisher and Cyclops.

Leo: Yeah, yeah.

Steve: Okay. Anyway, I just thought that was fun. And, you know, nothing very surprising. As I said, we've got to get away from those passwords.

Okay, now, Leo, this is completely random, but it's time for Miscellany. I picked up this on the news. There was some conversation about it this morning when I was making coffee. And this is the Active SETI project. Up to now SETI, the Search for Extraterrestrial Intelligence, has been passive, meaning we've been using big ears, looking, scanning the skies, using supercomputers, famously using distributed, like there's a SETI screensaver where your system's unused cycles can go to perform signal analysis, trying, basically filtering the random noise, looking for any kind of coherent information. Now they're starting to talk about transmitting. And I'm not sure how I feel about that. What do you think about that?

Leo: Why wouldn't - what, are you afraid that they're going to attract attention?

Steve: Yes. I don't think...

Leo: Oh, come on, Steve. What? Why would you let anybody - no, you've read too much science fiction.

Steve: I guess maybe. But Gibraltar Earth was a scary series.

Leo: I know, I know. Okay. Let me explain this whole thing to you. It's very simple. We're too far away. We're all too far away from one another.

Steve: Yeah, we really are way on the wrong side of the tracks.

Leo: Everybody is.

Steve: We're out on an unsociable galactic arm, yeah.

Leo: Until some race discovers faster-than-light travel, we're stuck here. Wormholes, schmermholes. We're stuck here. They're stuck there. Let's hope it stays that way. But they could still radio us.

Steve: Yeah. So if you're saying that, then I don't think I want to wave a red flag, you know, they're...

Leo: They can't get here. And by the way, the big issue is the speed of light. Those signals we're sending out?

Steve: Oh, no, no, no.

Leo: They may respond thousands of years from now.

Steve: Oh, no, this lightspeed limit, that's not a problem.

Leo: We're only going to reach nearby stars with the signals.

Steve: Well, we, but not they.

Leo: Oh, you think they've come up with FTL or wormholes or something.

Steve: Well, everybody else, everybody else has it except us, Leo.

Leo: No one has it.

Steve: We still have fur on our bodies.

Leo: It turns out you can't do it.

Steve: Yeah.

Leo: Maybe Einstein was right.

Steve: If it can't be done, then I consider that a blessing because I think we need some separation here.

Leo: Well.

Steve: That just, you know, look at this dumb little planet of our own and what a problem we're having dealing with our own race among ourselves. I just, you know, I don't think I want to light off a big flare and say, "Hey, we're over here." I think we'd just better...

Leo: You're familiar with Fermi's Paradox; right?

Steve: Yes.

Leo: Well, the paradox is, if you make completely normal kind of assumptions, that we are on a typical star, there are billions of stars in this galaxy that are billions of years older, with a very high probability, in fact we know this to be fact now, many of those stars have Earth-like planets.

Steve: Yup.

Leo: The issue is interstellar travel because he asks, why is it, given all of that, we haven't been contacted? Well, I'll tell you why.

Steve: Right.

Leo: We're too far away from each other.

Steve: Right.

Leo: That's my opinion.

Steve: Which I think is a blessing.

Leo: It's not a bad thing.

Steve: I don't think it's a bad thing.

Leo: No, no. In fact, that's a science fiction meme, practically, of the alien race come to chew us up and spit us out.

Steve: Yeah, yeah.

Leo: It's a cookbook. Are you ready for questions? Is that it on that list? You've got one more, one more.

Steve: I have one question in the spirit of the Q&A from a Filipe - you think it's Filipe or Filipe? He is listening, so...

Leo: I'd say Filipe, but maybe Filipe.

Steve: So Felipe, he sent a tweet saying, hey - because he saw the show notes and saw that I had picked up on his question that he had sent in the mailbag. He had a question about SpinRite, running SpinRite on brand new drives. And we've talked about it a few times before. But he said, "Hello, Steve. Thanks for Security Now!. Here's a quick question. Do you recommend running SpinRite on brand new HDD or SSD drives? What level would be adequate to find a possible problem with new drives? Cheers."

And I'll reiterate that I, well, first of all, I and Greg and everybody who knows runs SpinRite on brand new drives. It's just part of what you do. Back in the early days of SpinRite, Compaq Computer Corporation over-purchased drives and ran SpinRite on them all and returned those that appeared to be the weakest. The drive manufacturers didn't like that behavior, but they weren't going to argue with Compaq because Compaq was such a huge OEM purchaser. So Compaq was using SpinRite to prequalify the drives on the dock before they even accepted them. And of course we all remember those days where drives had a list of defects marked on the label, but the OEMs that were using those drives never bothered logging those defects into the drive.

My point is that today the world is very different, but no drive manufacturer does anything like running SpinRite on their drives because they can't afford the time. It takes, I mean, they're mass-producing drives. What they rely on is the technology that they have built in, the huge amount of error correction, and the ability to spare out sectors. Now, they do sort of a pre-assembly analysis and look for defects on the surface. So there are already sectors that have been taken out of service. But they don't have time to run the entire drive again once it's been assembled.

So the experience of myself and Greg, who of course is a SpinRite user himself, and hundreds, thousands of SpinRite users, is have a new drive, just give it to a machine and let SpinRite loose on it. Just Level 2 is fine. Just do a read pass over the drive with SpinRite looking very carefully so that the drive is able to, once it's been assembled, this will be the first time the drive has actually read its own sectors because that doesn't happen in the factory. And SpinRite will have an opportunity to work with the drive and take any sectors out of service before you start putting data on it.

So, yeah, I do think it makes sense. And the same thing is true for SSDs. As we have seen, they're a little more like hard drives than we expected because they're cramming so much density in. Oh, and I just saw an announcement, too, it was a little scary. Some manufacturer has decided to increase the number of levels stored on an MLC drive, a multi-level cell drive. Normally, on what used to be called an SLC, a single-level cell, you would either store a high charge or a low charge, that is, one bit of data, only two voltages. But in order to double the density, in order to get twice as much data in the same space, the somewhat lesser reliable drives, our so-called MLC, they store four voltage levels - zero and a quarter and a half and, well, I'm sorry, between zero and a quarter, between a quarter and a half, between a half and three quarters, and between three quarters and full. So four different ranges. And that gives them two bits of data stored in a single cell.

So it's a clever hack that doubles the density. But as you can see, you have to be able then to discriminate the voltage in a cell much more exactly than if you only have to tell between empty and full. Now you've got half, you know, partially empty and partially full that you need to be able to determine. Well, some manufacturers decided they're going to go to three bits, which is way aggressive because now we're talking eight different voltages in a cell. So we'll see how that turns out. It looks like SpinRite's going to be busy with SSDs for a long time to come.

Leo: And thanks to the Urban Dictionary we have solved the riddle, "adgjmptw."

Steve: Uh-oh.

Leo: There was a little bit of a hint in that it's in alphabetical order. It's the letters you get when you press "1" through "9" on your keypad or cell phone.

Steve: Oh, that's what that was. And that's why they showed the cell phone down below.

Leo: Cell phone, yeah.

Steve: Yes.

Leo: Clever, hey?

Steve: Interesting. But don't anybody use it because it's now in all of the cracking dictionaries. Everything...

Leo: Apparently...

Steve: Go ahead.

Leo: The reason it's in the Urban Dictionary, it is also used as a term of frustration when your texting conversation has come to a bitter end.

Steve: Who does that happen to?

Leo: You mash, you go [frustrated noises].

Steve: Wow. Okay.

Leo: Let us give you some questions, my friend.

Steve: Great.

Leo: I think you feel brilliant, smart, ready to answer these.

Steve: Energized.

Leo: Energized. And, after all, since you picked them, I presume you know the answers to them. Starting with Question #1 from Joe Pracht, and that's how he says you pronounce it, in North and South Carolina. That's a little bit of a mystery, but we'll just leave that to you, Joe. He writes, and this is a long one, he's recovered CryptoWall files without paying any ransom: Steve, I am a network and systems administrator for a large nonprofit covering North and South Carolina. Ah, you have given us the answer. We have had two XP computers infected by CryptoWall. We have a Group Policy block in place for CryptoLocker and are working to remove all XP machines from the network. However, in both cases we had the users disconnect the computers and ship them over to us. During Episode 496, Listener Feedback #207, Joe Meedy wrote you with a question about CryptoWall and made the statement, "I've read that CryptoWall makes a copy of your data file. It encrypts it, then deletes the original file." So, smart man, Joe Pracht.

Steve: Uh-huh.

Leo: He thought about this. He said: I created a full image of the infected drive - that's always the first thing to do, just image that sucker off - and then used R-Studio to attempt the recovery of deleted files. I'm not promoting R-Studio over other products, he writes. This just happened to be one our department had a license to. The recovery brought back deleted files of all types. I contacted the user of the initially infected laptop to discuss some of the files we found. I mentioned a picture of kids at a Japanese steak house, and the user was ecstatic. Not all files were recovered, but we recovered enough to make the user very happy. Thank you and Leo for the last 10 years. I'm a longtime listener and can't wait for the new show every week. Joe Pracht. Wow, that's a great story.

Steve: Well, yeah. I thought this was important to share because this demonstrates that, clever as these CryptoWall/CryptoLocker crypto bad guys are, they're making a fundamental mistake, and that is they're not overwriting the unencrypted files.

Leo: Shhh. Ixnay on the overwrite-ay. You're worried about connecting with aliens. You're giving away the store here.

Steve: Well, unfortunately it's out of the barn anyway. And the bad news is, or the good news is, until they fix this, this is a hot tip for recovering some of your files if you don't want to pay the ransom. The problem is it won't recover them all because new files are being created, which is the bane of deleted file recovery. We all know from the old days of the DOS E5 flag in the directory that, when you delete a file, all that really happens is that it's removed from the directory, and its clusters are returned to the file system.

But DOS has always allocated in a sequential manner. That is, it sort of, in order to reduce fragmentation, it allocates in a forward moving through the drive cluster sequence so that it tends not to immediately overwrite recently deleted content, or actually recently created content. It's creating it in an ongoing fashion. So you have vulnerable unused space which anything could say, oh, look, this is free. Let's take it and write data in it.

But still, what this says is that these guys have missed a trick, the bad guys have. If they were doing it right - and, well, for example, throughout my SQRL code I never release a buffer that contained anything sensitive without wiping it first. So I wipe the buffer, then I release it so that, if it goes anywhere - so the memory I'm no longer possessing that anybody else could get has nothing sensitive in it. And so the mistake these guys have made is in not scrubbing the contents of the file prior to deleting it.

Now, I don't think it probably really matters because most people are going to see the CryptoWall/CryptoLocker ransom note, and it's kind of all or nothing. They're going to see that the files they can see that they care about are all scrambled. They're going to probably pay ransom. However, for our listeners, it's worth noting, and this is why I really thank Joe for posting this, that for now, at least, the files that are unencrypted are just deleted. And if you use some good undelete software, there's a good chance you can get a chunk of them back, at least.

Leo: Hmm. But that's only CryptoWall, not CryptoLocker.

Steve: Well, we don't know about CryptoLocker. He did use both terms in his note. He said "recovered CryptoWall files." And then he does say "Group Policy blocks in place for CryptoLocker." And he also talks about XP machines. So there's a little bit of confusion here. But so I guess my point is, if you are ever faced with a, for example, a family member who says, "Oh, my god, I desperately need this one file, this is the only thing I care about," that kind of scenario, you might take a look to see if it's just sitting there undeleted after crypto whatever has done its deed.

Leo: Question 2 here from Daniel Aleksandersen in Oslo, Norway. He says new Windows versions are auto-sharing WiFi passwords: Windows Phone 8 have shared network passwords by default with Xbox, Skype, and Outlook Friends. This innovative feature is now enabled by default in the latest Windows 10 preview builds, as well. Is this at all secure? Granting random people access to an internal network sounds not ideal. What is going on? I didn't know about this. What is this?

Steve: Yeah. And this is really interesting. Actually, either last week or the week before I shared a SpinRite story where the user who sent this actually had a two-part, and I didn't share the second part, but it lodged in my brain. And he was clearly, he was obviously a listener. And his experience, which I'll now share, was a little more worrisome. He had a friend who was using Windows 8 at home. And one of the guy's machines crashed, his Windows 8 machine. Oh, yeah, I think it was last week because remember I talked about a Windows 8 data recovery. Well, that was the email that also went on to say that that machine that crashed was fine, and he used SpinRite to recover it. And he then allowed that user onto his home WiFi network to do whatever. Then, separately, later, that person brought a different machine over that was having some weird problems, maybe infected with malware? It was on his network.

And so, first of all, this is behavior that Apple has been using among their iOS-enabled devices. I've seen this myself. There are a couple restaurants I frequent where the management has given me their password. In fact, there's one where they wouldn't give it to me; but they said, well, "Let me have your iPad. We'll enter it. But we need to keep it secret because we don't want everyone to be using our WiFi." But, you know, they like me. So they entered the password secretly into my iPad, and my phone was then on their network. So Apple was bridging and is bridging WiFi passwords through iCloud or iGlue...

Leo: It's probably continuity. It's probably handoff. I would guess.

Steve: Well, yeah. And so what's happening here, Windows 8 is apparently doing the same thing. And that is that - and it's now enabled by default, says Daniel, in the Windows 10 preview builds. I don't know for sure whether it was something you had to turn on in Windows 8 or not. But the idea is, if you have a household with Windows 8 or 10, and you give one machine your WiFi credentials, they're glued together, and the other machines are able to get on the network. And so the issue Daniel is raising, and actually it was last week's SpinRite testimonial guy who said, wait a minute. The person whose one system I permitted onto my network could have brought back a different infected machine, and it's already on my network.

Leo: Steve, this is what LastPass does. Apple does this through the Keychain, the chatroom is saying. Of course they do. You can share your Keychain. But that's what LastPass does. Right? If you memorize a password on one machine, it's on your LastPass, and you can use any other machine with that password. That's what every machine does, everything does.

Steve: Except that - no, no, that's completely different because LastPass bridges browsers together so that you're able to log into another site on a different machine with LastPass. But to me that's different than any machine in a cluster just, like, knowing the private credentials of someone's WiFi network.

Leo: Well, it's done securely. It's through Apple's Keychain mechanism. There's a Keychain...

Steve: Right. Well, okay. So, see, but that's not the problem. It's not the security of the password sharing. It's the side effect that an unauthorized machine...

Leo: Right. Well, you should consider that, when you're giving somebody a password, you're giving them the password, not the machine. The user.

Steve: And now we know that you're giving their network of, yeah, their network - yeah, okay, I see what you mean. Good point. So, for example, sort of the same thing, the user may have entered the password privately into the guy's Windows machine, although there are ways to pull those out in hex, so it's not like...

Leo: Yeah. That doesn't happen that often.

Steve: ...it makes it secret, yeah.

Leo: Does it really? The guy says, "Give me your machine, I'll log you in." I wouldn't give them my machine.

Steve: Well, and of course we already know that the right solution is to have a second guest network which is not on your internal network and is protected because that's...

Leo: But just consider, if you give enter a password on a machine, you're giving, not the machine the password, the user the password.

Steve: Right.

Leo: And most systems - Android, iOS and Macintosh, now Windows 10 - have the capability of automatically sharing because we all use multiple devices.

Steve: Right.

Leo: And you're right, because LastPass doesn't monitor WiFi logins, so it doesn't happen there. But you could, I mean, it could if you wanted it to, I guess. But that's only because LastPass doesn't monitor that. Keychain does. All passwords are stored on the Mac in your Keychain. Question 3. Shall I move on?

Steve: Yeah.

Leo: Okay.

Steve: Yeah, no, I completely - I'm agreeing with you. These systems are built into the OS; whereas, for example, Last Pass is running in the browser. And so it has limited visibility. It can't see WiFi passwords because it's browser script.

Leo: Somebody says 1Password will.

Steve: Oh, interesting.

Leo: Yeah. And Apple will because Apple stores passwords in the Keychain, which is actually a better place than storing them in the clear in the browser or somewhere like that.

Joe Kouba, Tucson, Arizona wonders how, when brute-forcing, do we know when we've got successful decryption? Steve and Leo, I've been listening since Episode 0. Thanks for all the work you put into this. The passion alone is what makes listening an enjoyment. I'm technical in the area of computers, but I've never gone deep with encryption beyond what I learned in computer science classes in college, so my question might sound silly, and I'm hoping there's a simple answer I'm just not understanding. If I have a file that's encrypted, and I'm attempting to crack it by throwing computer power against it, how does the computer know when the file is cracked? How do I know I have plaintext?

For example, say I have a text file that contains "Hello, World," but when it's encrypted it's just garbled text. It'll be obvious when it's properly decrypted since the file will become readable. But how do you know a file is cracked if you don't know what's inside? The data may not be obvious as some English words. It might contain data that by itself also looks like garbled text. So actually this would be the perfect example, if I'm encrypting a binary file, how do I know when I've decrypted it?

Steve: Right, right.

Leo: This has long puzzled me. How you know when it's cracked? Thanks for your time.

Steve: So this is a perfect follow-on to the discussion we've been having in the last couple podcasts about encryption and authentication because we were using the example of that web-based service, which was initially performing no authentication, only encryption. And the problem was it had no way of detecting the wrong password. And remember, Leo, we showed that little Windows Notepad of crazy unicode gibberish when I generated that by giving it the wrong decryption password. It could not detect it, so it just gave me gibberish rather than the proper ASCII back out. And so that allowed us to talk about the need for authentication.

It is absolutely crucial, and all proper encryption systems also authenticate because, as a friend of the show, FireXware is his handle, explained also, there are well-understood means for replacing chunks of encrypted files with other chunks and altering the file that's encrypted. So encrypting a file is never enough. You absolutely also must authenticate it. That is, what that does is, it verifies that it is the same file that was encrypted; nothing about it has changed. And, for example, SSL and TLS do this. And all good encryption does. SQRL encrypts the SQRL IDs, and I use a hybrid, an authenticated encryption technology which provides both. And so when you enter your password, it decrypts your identity. And the first thing it does, well, actually it decrypts the password and then checks the authentication of the encrypted contents to verify the password.

So, Joe, the answer is, encryption that is correct, that is done right, will always also be authenticated. And as I've also said, the authentication you do after you encrypt - which means that, when you're decrypting, you first authenticate. So any brute force decryption would first be a brute force authentication. You would take a trial password and see whether the file authenticates for decryption with that password. If not, you know you've got the wrong one. The second it does, then you've independently verified that the password is correct, and then you can proceed to decrypt with absolute confidence that you've got the right password, and the decrypted result will be correct.

Leo: Very clever.

Steve: Really neat.

Leo: Murray, Murray Wood in Vancouver, wonders about UEFI secure boot and a crashed hard drive: Steve, thanks for the UEFI explanation last episode. I'm wondering how, if Secure Boot can't be turned off - which it usually can, but I guess maybe the new machines you can't, I guess it's what we were talking about, huh - would one go about replacing a hard drive? I assume that the UEFI databases are stored somewhere on the drive. If you have a hard drive crash, the new drive would need to be initialized somehow. It seems an initialization utility would be needed, and wouldn't that defeat the purpose of UEFI? I look forward to your answer and enjoy the show.

Steve: So a lot of interesting questions about UEFI, of course, from last week, because the discussion stimulated, as we would expect, questions for the following week's Q&A. Okay, first of all, I wasn't clear that the databases are stored on the motherboard. They're in nonvolatile memory in the UEFI firmware.

Leo: You need a secure store for them; right?

Steve: Correct.

Leo: Yeah.

Steve: Exactly. So they are not on any hard drive. So because the UEFI secure boot exists separately from whatever it might be booting. The whole initial warming up phase is done just on the motherboard with its firmware, with its expansion ROMs, verifying signatures and hashes and everything, before it even worries about like what to do next. So it's all going and up and warmed up before that happens.

Now, if it turns out that the drive has crashed, then it's like, okay, how do we recover this? Well, in the same way that you do now because, for example, the Windows 8 or Windows 10 boot DVD is also signed. So you're able to boot a recovery DVD, a recovery disk through UEFI from the system's CD drive or a thumb drive or whatever because it, too, is signed. And so UEFI will check the signature on the boot media whether, I mean, regardless of what it is. And if it works, it's just as golden as booting the original crashed drive. You're up and going.

Leo: Very sweet.

Steve: Yup.

Leo: Martino Jones in Ypsilanti, Michigan dual-boots Windows and Linux under UEFI: Hi, guys. I'm sure others have also sent in emails about this already. However, I thought I'd share just in case. I didn't know you could do this. I currently have Linux installed on my machines. My laptop is running Ubuntu with UEFI turned on, and Fedora is dual-booting with my Windows 8.1 gaming desktop. This also has UEFI turned on. On my desktop I have Windows installed on one HDD and Fedora on the other. The Grub2 boot allows me to boot with UEFI into Linux, so I never have to turn it off. Love the podcast, watching and listening for a couple years now. Happy 500th!

Steve: Yeah. So I wanted - I wasn't clear enough. Actually we sort of ran out of time talking about the non-Windows side of UEFI Secure Boot. This has been around long enough that there are, like, the major installations of the various flavors of Linux all have their security certificates worked out and available and installable in order to certify them for UEFI Secure Boot. And the reason I chose Martino's note is I wanted to make sure everyone understood that this in no way even - it's not clear, if you can't turn it off, how you'll be able to add Linux to a system with Secure Boot enabled. But I still imagine you probably can, as long as all the tools you use to do that are signed. And signed tools are now available across the board for many of these major flavors of Linux.

This might be a bit of a problem if you were, like, hooked on some really minor sort of backwater unknown brand of Linux, rather than the main, major builds. But all of them have got UEFI Secure Boot available. There's a little bit more you have to go through. But as Martino shows, once you do that, it's really not in your way at all. So I didn't want anybody to feel put off that UEFI was going to be locking them out in the future. It really won't.

Leo: Good. That's good news.

Steve: It just gives us protection.

Leo: Bob in Florida wonders whether the Chrome miniLock file encryption extension is any good: Would Steve be able to review miniLock file encryption in Chrome? I'd love to hear his thoughts on an app that seems very good to me. I haven't tried this. I'll have to. I was looking for a file encryption app to be able to use with a Chromebook. It will actually work with any system, and I found the website, miniLock. io. It's got to be JavaScript; right?

Steve: So it is 100% wonderful.

Leo: Oh, neat.

Steve: It is, first of all, it's by our friend Nadim Kobeissi. He's the guy who is famously behind Cryptocat. And we remember Cryptocat. He's a Ph.D. candidate, getting his Ph.D. in applied crypto. MiniLock.io is his recent project, and it is as beautifully done as anything could be. He has had it audited. Matthew Green and a number of other cryptographers have been involved in the project, have looked at his spec. There has been a full code audit. The architecture is really well thought out.

So here's what it is. First of all, Chrome browser and Chromebook, so the Chrome OS, compatible. It uses my favorite elliptic curve crypto. It's the same crypto that I chose and am using for SQRL, that is, the Daniel Bernstein Curve25519 crypto. The beauty of that is that, and this is why I chose it for SQRL, actually it's the reason SQRL exists, is that that crypto allows you to have a deterministic private key. And what that means is that - whereas, for example, with RSA, you have to randomly choose primes and then use those as your private key. The point is, randomly choosing isn't deterministic. That is, you can't immediately go to one.

With the crypto that I use and that Nadim is using, this Curve25519, you're able to say, "This is my private key." That's the key to SQRL is that, when you take your identity, your SQRL identity, and mix it with the web domain, the result is your private key, every single - and every time you go back, it's the same private key. That's the secret. And Nadim has leveraged this same characteristic in a very clever way.

Here's how it works. You use your email address and a passphrase. He checks the entropy of the passphrase to make sure it has at least a hundred bits of entropy, using an entropy-measuring formula that somebody else worked out. So you use your email address, which is easy for you to remember, and a passphrase, and it uses them both just to get sufficient entropy. That directly creates your private key under the Bernstein elliptic curve crypto. And the private key creates the public key. The beauty of this elliptic curve crypto is that, for example, unlike PGP or RSA keys, which are massive, these keys are 48 characters. They are tweetable, for example.

Leo: That's nice.

Steve: Yes. So with miniLock.io, so here's how it works. You load it in your browser. You fire it up. You put in an email address, your email address, and a passphrase. It gives you your public key, which you can then give to anybody who you want to enable to send files to you. Because it's 48 characters, you can tweet it. You can email it. You can even read it to them over the phone. I mean, these keys are so small that they're way more convenient. So, and the point is, that is a public key generated from the pairing of your email address and your secret passphrase. That enables anybody...

Leo: It doesn't like mine. It says it's too weak.

Steve: Ah, well, yeah, got to...

Leo: Yeah, 123456789.

Steve: Oh, darn. Keep going, 1011121314.

Leo: It's suggesting popexaggerationlagunedetectionsmustilymotivationscairo. So there you go. I'll have to get another phrase.

Steve: So the idea is that, when you use your email address and your passphrase to create a public key, that allows you to receive encrypted files from anyone. So you send this to somebody, you tweet it, for example, out into the universe, or tweet it to a friend, or send it through a text message, whatever you want to do. They then use miniLock in a similar fashion. They drop a file that they want to securely encrypt and send you.

And we're talking world-class security. I mean, this is absolute unbreakable bulletproof security. They drop the file on the browser, having given it your public key. It encrypts that file so that only you are able to decrypt it. And it can do multi-way, it can do one-to-many encryptions. So you could drop many different people's public keys in to encrypt that one file so that that group of people, the individual members are able to decrypt it. The way he designed the file system, the header, and nothing about the encrypted file gives away the identities of either party at either end. So it's also anonymous and proof against tracking. You then take the resulting file and send it to these people. And only the people who have the matching private key, which is always regeneratable from their email address and their passphrase, are able to decrypt the contents of the file.

So it is really nice. I spent some time, as you can tell, digging into it and understanding what's been done. And so for Chrome and Chrome OS, for Chromebooks, it's a terrific little, absolutely bulletproof, properly designed, state-of-the-art crypto file exchange tool.

Leo: This is so cool. That is really great. So now if I want to send this image that I just miniLocked, I'd need somebody's public ID, and I could just paste it in. It automatically puts mine in. And it's okay for me to show this; right? Because I could tweet it and everything. That's a public key.

Steve: Yup. Yes. That is the full public key. And what that does is that allows anybody who has it to encrypt a file that only you can decrypt. So that allows them to encrypt something, knowing that you're the only one who can decrypt it. And so unless you give your credentials away, it's a secure one-to-one channel between two people. And also you're able to drop multiple identities in and multiply encrypt the file.

Leo: That is so great.

Steve: Yeah, it's just - it's beautifully done. So, Bob...

Leo: You're right, this is neat.

Steve: Thanks for asking, Bob. I'm glad you - it had crossed my radar a few months ago. I remember looking at it thinking, on, this is correct. But I just, you know, I got distracted by something shiny and never got back to it.

Leo: Well, now I'm going to tweet out my miniLock key; right? And this way people can just search through my Twitter stream, and they'd have it.

Steve: Yup.

Leo: Gosh, I love that. And that's all they need to send me private stuff.

Steve: Correct. Now somebody can encrypt a file to you, and only you are able to decrypt it.

Leo: Love it. Put this on your blog, too, I guess.

Steve: Yeah, I mean, exactly, you can post it publicly. You might put it on your private web page. It's like, hey, anybody who wants to send me something, use miniLock, and here's my public key.

Leo: It's probably not as easy as doing, like, PGP if it's built into your mailer. But you could use this for email, too. Just create a text file, miniLock it, and then attach it, and then send it.

Steve: Sure, sure. I guess what I like about it is that PGP is sort of like, it's there whether you need it or not. My use case is more like, hey, I want to scan some financial documents and send it to my accountant, but I don't do that every day. So this allows you to very handily have absolute state-of-the-art public key encryption at your disposal. And use it only when you need to.

Leo: Nice. Very, very cool. It doesn't, you know, PGP is useful for signing and other things. But this is great. Andrew Stevenson, Dorset, United Kingdom. He found tracking protection in Firefox. He's looking at an article on Ghacks. The full link is in the show notes. Firefox users right now can enable tracking protection by going to about:config, searching for privacy.trackingprotection, and double-clicking on the setting. That'll set it to TRUE, and it's enabled. Mozilla is going to be adding this feature in private browsing mode from Firefox 39 and up. But if you're still using an older version of Firefox, again, about:config, double-click privacy.trackingprotection. That'll set it to TRUE.

[ghacks.net/2015/03/27/firefox-39-tracking-protection-for- private-browsing-mode]

Steve: So this is very cool. There's been a project underway for some time now, like more than a year. A group of ex-Google engineers got together and created something called Disconnect because they were upset about tracking. It's not so much - and this is sort of where we have to be careful with our terms because it's not about seeing ads. It's about having large databases of profile assembled by little tracking bits scattered all over the place. And there are companies, as we know, whose entire purpose is building portfolios of information about individuals.

So at the moment we're at Firefox 36.0.4. And as you said, if you go to about:config, as we know, there's like a bazillion little twitchy configuration things. So in the search box, put in "privacy.trackingprotection." That'll show you two items that come up and match that string. The first one is the privacy.trackingprotection which you can just double-click in order to toggle it, and it'll flip to TRUE, and the little TRUE goes bold. What this does is something that will be surfaced in the UI with Firefox 39, so a few major versions from now. But it's available to all Firefox users today.

What this does - and so right now under Privacy in the Options on Firefox, the only option you see is that sort of unfortunately failed Do Not Track. I had high hopes for it. I liked the idea of browsers saying, yeah, please don't track. We always knew that it would be a choice for trackers to decide whether they wanted to honor it or not. So what you see right now under Privacy, under the Options dialogue, is an option for that checkbox, "Ask trackers not to track me." Coming in 39, that gets pushed down, and there's one above it that says "Turn tracking protection on."

What that does is the Disconnect folks have a very comprehensive block list of tracking companies. This doesn't block advertising. This blocks just the aggregate trackers. And Mozilla has built that into Firefox, and we can enable it today. So basically we're getting a significant feature in Firefox. It will be in the UI with Version 39, where you just go Options and Privacy, and you'll see it. But we can turn it on now. When you do, the little icon in front of the URL will show you, it turns into a shield if you've gone to a site where any of the block list domains were attempted to be queried by your browser.

So I of course turned it on, and I went to The New York Times. Bang. Up came the shield, indicating that at least one domain was blocked from tracking me at this site. And if you're curious, you can use the Web Tools Console under the Security Tab to see a list of all the blocked sites which were prevented from tracking you from the page you were on.

So Andrew, thanks for the tip. I had not seen this yet. I'm delighted to know that it can be turned on now, and everyone gets the advantage of it. And then eventually, when we get to the major Version 39, everyone will be able to see it right there in the privacy tab of the options dialogue. Very cool. Oh, and I forgot to mention, in tests it speeds up the median page load time by 20%. Just blocking all the other queries, the tracking queries that your browser would otherwise be making gives you a 20% speed boost across the board.

Leo: I almost said "Chapter 8." Sometimes it feels like that. Question 8. Styles Bytchley - can that be his name? - in Toronto wonders - maybe it's pronounced...

Steve: Well, it's spelled B-Y-T, Bytchley, B-Y-T, yeah.

Leo: Bytchley, Styles Bytchley in Toronto wonders why can't UEFI protect against Equation Group's HDD BIOS rewrite? Been with you since day one, podcast definitely something I look forward to every week. I'm a proud SpinRite licensee, and it has saved me. You guys are great, Happy Birthday, blah blah blah. By the way, Happy Birthday. I forgot.

Steve: Thank you.

Leo: Was it a good one?

Steve: And Leo, I just wanted to make sure you knew how much I appreciated this 10 years of podcast. Some...

Leo: I thought so.

Steve: Of course, I mean, I tell you guys.

Leo: I guessed that.

Steve: Okay. I wasn't explicit. And I was thanking our listeners, and someone tweeted me saying, hey, you didn't thank Leo.

Leo: Don't listen to the tweeter.

Steve: I thought, well, I thank Leo all the time. But, you know...

Leo: You don't know.

Steve: Just for the record, this is great.

Leo: Some people use - I know that, of course. Some people use Twitter as a way to stir up trouble. Just ignore that. Styles Bytchley writes: I just finished listening to Episode 500 on Secure Boot and UEFI, and it seems like there ought to be a method for this to prevent the hard drive firmware attack you were describing in 495. Wouldn't a hard drive that had been infected by that Equation Group hard drive firmware rewrite have something different about its driver/hash/fingerprint that could be detected by UEFI as different than the original whitelisted item? Cheers, Styles.

Steve: Alas, no.

Leo: No.

Steve: I loved the idea because it was clever. And it sort of - it was a nice intersection with this notion that UEFI sort of, like, absolutely protects us. Now, having said that, it would definitely complicate things because the idea is that, well, I mean, it'd be interesting to see if the hard drive firmware rewrite could get around Safe Boot. That is, the Safe Boot cannot prevent the hard drive firmware from being rewritten because that happens after the system's booted, when some malware somehow arranges to run and get in and issue standard ATAPI, which is the AT attachment API, the ATAPI commands that allow drive firmware to be updated. Drive firmware can typically be updated by commands to the drive.

So that's what the whole Equation Group's project is about is, remember, subverting our hard drive's firmware to perform on-the-fly sector replacement. The problem is that, if it then tried to do that, that is, the next time the system boots, it could run afoul of UEFI if its replacement sectors weren't also signed. So then we've got to wonder, okay, well, if someone has the capability, like a state-level actor, one of the three-letter agencies, to get a replacement firmware into, like, across-the-board families of drives, as we saw when we were talking about this a few weeks ago, couldn't they also arrange to get that replacement sector signed so that UEFI Secure Boot would function, especially in the world we're moving to where Secure Boot is just going to be the way things are in the future. And I have to sort of imagine that that probably wouldn't actually stop them.

So unfortunately, Secure Boot doesn't protect the drive's firmware from being altered. And I wouldn't bet that it actually prevents the altered drive contents from being able to run, either. I think they probably can sidestep that, especially when we're trusting so many CAs. I mean, we're just - that's the problem with the trust being as broad and deep as it has become.

Leo: You're getting the word out. I got a call on the radio show on Sunday, somebody asking me about CAs. They didn't know it was called "CA," but they kind of understood what was going on. And they said, why are there so many of them? And there's a bunch in foreign countries. I said, "Like the Hong Kong Post Office?" He said yeah. And but I showed him and the audience how to check the certificate of a website. So he understood what the CA was up to and why it's important, not only that the certificate match, but that the CA actually be trustworthy. I wish I could have given him a tool to remove CAs.

Steve: Yeah, we're going to get to that in Question #10, as a matter of fact.

Leo: Well, there you go. As if we planned it. Grayson Palmer in North Carolina wants to send links in email.

Steve: I love this email. This is just so wonderful. Okay.

Leo: I'm responsible for security at a regional retail company. I have educated our employees to be very careful about clicking links in emails - good man, Grayson - especially when the email is unexpected, or if the hover-over help shows the link is really going to take you to StealMyCredentials.com. Now, when I want to send them emails with hyperlinks, I get back a bunch of, "Hey, you told me not to click this, Grayson. Is this safe?" That's a good sign they're aware of the dangers. I like that part. I also get some, "You know, should you be sending out hyperlinks when you've been telling us not to click on them all this time, you hypocrite?"

Of course what we've really been telling them is not to click on hyperlinks in emails from unknown senders, or ones that have other telltale signs. Is there a simple "best practice" for this? I would love to be able to send out hyperlinks in emails, and yet somehow be able to show non-technical users it's okay for them to click. You know, maybe a little padlock symbol like you see in browsers. Or am I just relegated to saying, hey, you've got to trust me, it's from the internal security team email address? An example is a recent effort my team went through to point out to users where the Help & Resources page is of the IRS, where they can learn a lot about avoiding tax scams. And he puts the link in here, but I'm not going to click it. What do you think?

Steve: And so I love this because, okay, several things. The short answer is, no, there's no solution to this. And in fact he mentions, unfortunately, telling them not to click on links from people they don't know or trust.

Leo: That isn't even it, is it.

Steve: Correct, because many times, you know, the way email spread was that a user's email account either, like at Yahoo! or in their computer, would get compromised, and the malware would send out email. And, you know, so like you get email from your mom saying, "Oh, honey, this is the funniest video I've seen in years, check it out." And then Mom provides you the link, except this never came from Mom.

Leo: Right.

Steve: So you can't - so the idea of trusting anything doesn't work. The notion of some sort of a symbol or something doesn't work because anything can be spoofed. That's the problem. So I sort of loved this Catch-22 that Grayson got himself in because he's trained his people properly, so much so that they're, like, refusing to click on the links that he sends them. The problem is there is absolutely no way in that channel, that is, in the email channel to tell people to trust it because it's the channel itself that is not trustworthy. And the way I'm phrasing this is the key. That is, you have to use some other channel.

Specifically, I imagine that he has control of a web page. And so what you could do is tell people in the email, manually go to this web page. Don't click on a link to it. I'm not giving you a link to it. He could, like, give them a non-link and say type this into your web browser. And there you will find the IRS help in filing your taxes link. So give them a page within the company where they can go to get things. Maybe make it a fixed page, so it's always the same page. And they could, like, create a shortcut in their browser. And so then Grayson could update that page with the things that he's, like, easy access links for things. And that way it stays around. It's always there. They're able to tell their friends about it if they want to share it.

And essentially this is another way of looking at that really wonderfully pithy bit of wisdom, which is never do anything you didn't go looking for. Remember that? Brian Krebs was the first person who suggested that. And that is, when a website says, oh, you need to update your Flash, wait a minute. You didn't go looking for a Flash update. It was offered to you. Never accept anything which is offered to you. Only do things that you initiate, that you go looking for. And so in this case the readers of the email would themselves go looking for the page that Grayson has put up, saying go check out this page because there you're going to have links you can trust, not in the email channel, but in using a different channel because you just can't trust.

Leo: Or say "google this phrase."

Steve: Yeah. That would be good, too, yes. And we do that often on the show. Google this, and you'll go to the right place.

Leo: Right. Or you could use "Let me Google that for you," the website. You've seen that; right?

Steve: No.

Leo: It's a snarky way to respond to an obvious question. Chatroom knows it.

Steve: Oh, yeah.

Leo: It's lmgt...

Steve: Is salt bad for you? Let me google this for you.

Leo: So what you do is you go to "let me google this" - I have to google that.

Steve: Oh, it's the abbreviation?

Leo: Yeah, it's the abbreviation. It's lmgtfy. So then you enter in the phrase you want. Is salt bad for you? Then click the button. You could either do "I'm feeling lucky," by the way, or I would do a Google search. And now it creates a link, which I'll copy. And you could put that link in the email and train them that, only if it says lmgtfy.com. And then, by the way, they can kind of verify because, when you hit it, it goes to Google and types it in for you and then presses Search. So I think that works.

Steve: Wow.

Leo: Lmgtfy.com. Or just type the prose words. Go to Google and search for quote something.

Steve: Yeah.

Leo: Yeah. Because Google, if you give it a specific search, will give you the right page.

Steve: Yes. For example, if you searched "IRS help and resources"...

Leo: Perfect. You'd get that page.

Steve: ...which is the body of that URL, it's going to be the first thing that comes up.

Leo: It's why we don't even really have to give out URLs anymore. You do this now in the show. Sometimes you say, oh, well, just google, and you'll find it.

Steve: Yup.

Leo: It's amazing.

Steve: It really is. And I heard you on one of the other podcasts talking about how...

Leo: They're very powerful.

Steve: ...amazing it has become. I mean, it is the front door. Google is the front door to the Internet.

Leo: It's the Internet for people now.

Steve: Yes, it is. In fact, I've had very, you know, my less techie friends, who don't understand what a browser is, it's like they don't know that Google is not the Internet.

Leo: No.

Steve: They just - that's how they find everything.

Leo: If they want to go to Yahoo!, they don't go to the browser bar and type Yahoo.com. They type "yahoo," hit return, and then click the first link. And in fact that's why one of the top search terms, the most searched-for terms on Google, is "Yahoo." Or at least it used to be. I went to Google, they have a great - one of the buildings at Google, this was years ago, it had a great big screen that would show you the top searches in real time. And Yahoo! was like, boom, boom, boom, boom, boom, boom.

Steve: Oh, neat.

Leo: That's probably changed since then.

Steve: Yeah.

Leo: Yeah, nobody goes to Yahoo! anymore.

Steve: I hope so.

Leo: Last question, and this is the one I actually would love to know the answer to, it comes from Wayned in the middle of Southern Maryland. He wonder why users can't allow or disallow CAs, Certificate Authorities: I'm listening to SN-500 return to a question I've had in the past: Why cannot we select which Certificate Authorities we want to allow? We could go through and edit the list to disallow CAs, but it takes a bunch to eliminate all of the ones that make no sense. Since I live in the U.S., I speak English and a little French, I don't know that I will ever, ever, ever need to use a certificate issued by Hong Kong, Serbia, Japan, et cetera. I may feel the need to allow the France CAs; but then again, maybe not. Can you explain what's going on? Maybe this is the guy who called the radio show because that's basically the question in a more technical form.

Steve: So, yeah. I guess the best way to phrase it is that we were never supposed to care about this. The designers designed it to be hidden and perfect. And unfortunately it's, we know now, it's not perfect. And so its imperfection is causing people who wish it were less hidden to want to have some access to it. They, for example, Wayned understands that it's the ridiculous four or 500 CAs that we trust which is causing some weakness. But at the same time, the sense I get is that the danger hasn't yet risen to the level where it makes sense to go through the trouble. There are definitely listeners of this podcast who have decided they're going to run the experiment. I get email from them. I get tweets from them. They turn off all their CAs. They disable them. And you can do that, down in the plumbing of your operating system. You're able to move them to...

Leo: Can you pick and choose? Or...

Steve: Yeah.

Leo: You can. Oh, I didn't know that.

Steve: There's a whole Certificate Manager, for example, in Windows, where it looks a little bit like your registry. It's tree structured, and there's trusted providers and untrusted providers. And you can drag them all down, and then your browser starts having a fit because you can't do anything. But then one by one you move the certificates that are causing the fits back into the trusted category, and those will be DigiCert and GlobalSign and VeriSign and GoDaddy and Comodo.

And what you'll find is not surprising. Very few CAs cover the bulk of the sites you visit. And so very quickly the alarm bells start tapering off, to the point where after having maybe 20 or 30 CAs, you're not seeing alarm bells anymore, yet you no longer have the other 370 certificate authorities being trusted. So it's absolutely possible to do it. It isn't made easy because you can get in trouble. You could delete the certificate by mistake, for example, I mean, Microsoft doesn't want people digging around in there. It's just supposed to all just sort of be automatic and magic.

So there isn't a super friendly user interface to it. They haven't made it simple. And I guess my point is that it's not anything that I have bothered to do or that I would even recommend. Enthusiasts who want to experiment with it have done this. And their results are pretty positive generally. As I said, you add about 30 to 40 CAs, and everything seems to work just fine, even though you're no longer trusting all the other ones in the planet. Which might be a benefit.

But I guess my point is that the actual danger, the true danger that we appear to be in from trusting all the others just doesn't seem to be that significant. With Chrome out there, Google instantly spots fraudulent certificates. And the CRLSet gets pushed. Everybody scrambles around for a day, and we get a news topic for the podcast. But nothing happens. I mean, it's not the end of the world.

Now, the only place where it might be interesting is if law enforcement were using some of those off-brand CAs to sign their fraudulent certs when they want to intercept our traffic. That is, if we winnow the CAs down to only those that we know and look legitimate and actually seem to be signing certificates that we encounter on the web, then it very might well be that we're no longer trusting the CAs that other - and I don't just mean U.S. agencies because, remember, no one will be surprised if Chinese intelligence is able to get any kind of cert it wants from China. And as we know, the CNNIC or, yeah, CNNIC, the China NIC, they're providing certs. I'm sure Chinese intelligence is able to get a cert for any domain that it wants.

Leo: Wait, wait, I can't delete the China cert. I can delete a lot of certs, but I can't delete that one.

Steve: Ah, interesting. Sticky.

Leo: That's an Apple or Chrome thing. This is the Chrome cert store. Maybe I did something wrong. Oh, these are - maybe these are system certs. I need to go to...

Steve: Yeah, again, it's not mean to be simple.

Leo: Yeah, no.

Steve: It's not for your common user.

Leo: You should know what you're doing, yeah.

Steve: Yeah, you really do need to know. Otherwise you break your computer.

Leo: Well, you just - you'll get a lot of - you can go to sites where the cert won't be there, and you'll get - if it'll let you on the site, it'll just warn you; right? Or...

Steve: Well, it depends. For example, if the site is an HSTS site, you cannot bypass a certificate warning there. It will not let you. But otherwise you're often able to say, oh, yeah, fine, you know, I trust you anyway. Go ahead.

Leo: Google actually, and this is what I told people, as you said, on the show, has such a good system for certificates, that if you just click the icon, the padlock icon, it'll say this is good, this is safe, you're all right. I mean, I think that's very - I think they do a good job, in Chrome, anyway.

Steve: Yeah. So I guess in conclusion I would just say it is absolutely true that a user could curate the CAs they choose to trust; and that, as you'd expect, very few CAs cover most of the market, especially in any region of the world. You just don't need to trust that many. And in fact, it's like a very sharp falloff, where you end up with trusting 40, and now that seems to be all you ever need except maybe the oddball case here or there, which might actually be good not to trust one of those. So, but the issue is then it becomes a management headache, something you have to kind of keep an eye on. You're going to get errors from time to time. And I have to wonder if it's really just worth the effort, if the problem is bad enough to worry about it. And we are in the process of fixing this problem with other certificate technologies that are in the process of coming online. So it's certainly a question that is interesting.

Leo: Speaking of interesting, I was just clicking on the cert at Bank of America in Chrome, and it says, "Your connection to BankofAmerica.com is encrypted with obsolete cryptography."

Steve: Ooh hoo hoo...

Leo: The connection uses TLS 1.2. So that's Google doing a little bit...

Steve: Yeah, it's using RC4, and that's the problem is Google does not like RC4, 128-bit RC4, and SHA-1. So not only are they using a bad cipher, but they're also using an SHA-1 cert. Now, I am, too, because as we know there's nothing wrong with SHA-1. In fact, just yesterday I had my favorite CA group, DigiCert - a domain that I use sort of for background plumbing stuff is GRCtech.com. And it was expiring next month. So I asked DigiCert to make me two certs, an SHA-1 that's good through the end of the year, and then an SHA-2, that is, SHA-256, that I'll be using after that, just because I want to continue using SHA-1 certs all the way through 2015 so that I'm not setting off alarms in Chrome, yet everyone is still able to get to GRC, because people using older OSes cannot get to newer sites that have non-obsolete crypto. Although I do think it's time to turn off RC4. One wonders, Leo, why...

Leo: A bank.

Steve: Well, not only the bank, not only why they're offering it, but why your browser wouldn't...

Leo: Accepted it.

Steve: Well, why your browser wouldn't have chosen, why your browser wasn't offering something that they would have chosen first. You might try going to SSL Labs and put that Bank of America into the SSL Labs site and see what the order, the cipher suite order is because that's weird. You would think that your - certainly your browser is offering much more modern ciphers. So it must be that the cipher ordering at Bank of America is really bad, like either they don't have any strong ones, or the strong ones are at the bottom of the list, and they should be at the top of the list.

Leo: Yeah.

Steve: Because there's no good reason you should be getting that kind of...

Leo: Where does it show the list? Is that at the bottom? Oh, here it is, cipher suites, yeah, number one. There it is. TLS RSA with RC4 128 SHA.

Steve: Oh, my lord.

Leo: That's number one. That's the preferred list, the preferred...

Steve: Oh, my lord.

Leo: Huh. Huh.

Steve: And the good ones are below it.

Leo: Below it.

Steve: Oh, goodness.

Leo: That's really surprising, frankly.

Steve: Wow. Wow. In fact, that's almost a little spooky, Leo, like, okay, that's really very difficult to understand.

Leo: This is BankofAmerica.com. I'm not going to some weird site.

Steve: Wow.

Leo: Yeah. That's not good.

Steve: That's sort of hard to - that's hard to understand. Yeah, and look at all the better ones that are below.

Leo: Yeah.

Steve: And so your browser certainly offered a mature list. And from that, the BofA said, eh, we'll go with RC4, it's fast.

Leo: Let's go with the crappy one.

Steve: Yeah.

Leo: I got this. I got it handled.

Steve: Wow. Wow.

Leo: That's, boy, you know, you have to - I'm kind of in some ways not thrilled about SSL Labs, especially giving poor grades to my friend Steve Gibson and so forth.

Steve: I'm back at an A+.

Leo: You're back to an A? All right?

Steve: Yeah.

Leo: But it's really, I mean, maybe the grades, they could downplay the grade thing because that's kind of like it's too broad a brush.

Steve: And I understand, you know, Ivan Ristic is behind SSL Labs. He's a great guy, got a neat book out about securing SSL and TLS. But his thing is ranking security, and I think he's doing a good job. I'll take some dings right now while I stay with an SHA-1 cert. I've already got SHA-2 certs waiting in the wings. I'll deploy them on New Year's Eve of 2015, and then I'll go from an A to an A+. Right now he's saying, eh, Gibson's using a weak signature. It's like, well, okay. I mean, weaker, but plenty strong. So but Ivan's service, I love SSL Labs.

Leo: Yeah, I mean, my point is grades don't tell the whole story.

Steve: No.

Leo: But it's really great to be able to look at things like encryption suite order.

Steve: Oh, my, yes.

Leo: I didn't know I could do that.

Steve: Yes.

Leo: That's fascinating.

Steve: Yes. And that's...

Leo: And a little disturbing.

Steve: That's, I mean, that's actually - that's concerning. I mean, you kind of wonder how that could happen by mistake is what I'm saying. It's like, whoo, wait a minute. Although who really cares about decrypting BofA traffic, I don't know.

Leo: There's nothing important going on there.

Steve: Really, no.

Leo: If I notice any big wire transfers in the next few days, I'm going to blame you guys at home; all right? Hey, thank you, Steve Gibson. GRC.com. That's the place to go for your 16Kb versions, transcriptions as soon as Elaine digs herself out from the cactus snow.

Steve: Yes. I haven't heard from Elaine. So here's the deal. I will be posting the audio of the podcast shortly after I get it. So even before the transcripts come, normally I wait for the transcripts, and I do it all at once. But I'm not sure where Elaine is. So I'll get 500 posted as soon as we disconnect now, and 501 a few hours from now when you guys get the audio edited.

Leo: Yay. And by the way, my main banking is done at this place, USAA.com.

Steve: Yay, A+.

Leo: A+. And if you go down and look at the cipher suites, they don't even support...

Steve: Nope.

Leo: ...the weak cipher suite at all.

Steve: RC4 should not even be in the list anymore.

Leo: On the list.

Steve: No.

Leo: What is Bank of America up to?

Steve: That's why I'm saying, Leo, that's sort of beyond bizarre. That's really - that's difficult to excuse.

Leo: Hmm. What was I saying? Oh, GRC.com. Now, if you have questions for future GRC.com/feedback, if you want to get SpinRite, that's the best place to do that, the world's finest hard drive maintenance and recovery utility.

Steve: Only place to get SpinRite.

Leo: You can make Steve's pocket go "Yabba dabba do."

Steve: A late birthday present. And by the way, some people did buy copies of SpinRite for my birthday. Which, you know...

Leo: Oh, that's a great idea. Everybody wins.

Steve: Wow, I mean, I really appreciated it. So I couldn't say thank you to everybody, so thank you so much. My Twitter stream went crazy. After I thanked people for thanking me, then it even went more crazy. So I thought, okay, I'm just going to shut up now because it's self-fulfilling. So thank you, everyone, for the birthday wishes last week. I really appreciated it. And especially for the presents, wow.

Leo: Yeah. You got presents? Well, SpinRites.

Steve: SpinRite, that's what I mean.

Leo: But, see, that's a present that everybody gets a present.

Steve: And one guy I did reply to. He said, you know, "I bought a copy of SpinRite to thank you for the 10 years of the podcast and for everything you do." And I said wow, I said, you know, I don't...

Leo: That's nice.

Steve: I'm a little uncomfortable with charitable contributions, but someday SpinRite is going to save him, too.

Leo: Think of it as a guy's taking you out for ice cream. You get ice cream; he gets ice cream. Everybody gets ice cream.

Steve: And I'm going to keep SpinRite going for, you know, it's already been going for 30-plus years. It's going to keep going. So it'll keep paying people back.

Leo: Yay. And don't forget SQRL is there. If you want to get rid of passwords, help do it. Tell your favorite website, "Use SQRL, use SQRL, use SQRL."

Steve: We're on our way. We're on our way.

Leo: On our way. I have other versions of this show, high-quality audio, we even have video if you want to see how gorgeous Steve and I are. You can get that at TWiT.tv/sn or wherever podcasts are stored and forwarded, lots of places to get that, including our own apps. We don't do them, but the TWiT apps on all the platforms. But whatever you do, make sure you subscribe. That way you'll get every episode. No more voting on the Podcast Awards.

Steve: Closed down last week.

Leo: Alea iacta est, the die is cast. We'll find out soon whether you won. Of course you did. What else? Oh, if you want to watch the show live, you can do that, too. It's Tuesdays, 1:00 p.m. actually 1:30 p.m. Pacific, 4:30 p.m. Eastern time, that's 2030 UTC, live.twit.tv.

Steve: Ah. I just checked to see whether I had heard from Elaine because I sent her a note. She says, "I'm fine, and I did send it, and I've just forwarded that email. Perhaps your mailbox was just swamped with birthday messages," says Elaine. So we're all good to go. I will get her - and it didn't come through again. Interesting. Something seems to be catching it. So I'll let her know I didn't get it. But anyway, so it looks like Elaine's fine, transcripts are happening, and we'll be up to speed again.

Leo: Good. Very nice. Thanks, Steve.

Steve: Thanks, Leo.

Leo: See you next time.

Steve: Thanks, buddy.