TrueCrypt: WTF?

Description: After covering the week's most interesting security news, Steve and Leo look back upon and analyze the past seven days of insanity which followed the startling surprise "self-takedown" of the longstanding TrueCrypt.org website, and of TrueCrypt itself.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-458.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-458-lq.mp3

SHOW TEASE: It's time for Security Now!, and today's episode is truly a mystery story. Last week the developers of TrueCrypt put up a message on their site saying, "Go away. No TrueCrypt here. No TrueCrypt for you." What happened? Steve Gibson doesn't know for sure, but he's got a very plausible explanation. And we'll let you know what to do going forward. TrueCrypt: What the Heck? Next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 458, recorded June 3rd, 2014: TrueCrypt: What the Heck?

It's time for Security Now!, the show that protects you, your loved ones, and everybody you know against the bad guys out there on the Internet, so the guys who would steal your passwords, steal your privacy. Mr. Steve Gibson is the guy in charge here. He is the editor in chief at the GRC website, and the creator of SpinRite, the world's best hard drive maintenance and recovery utility, the inventor of the term "spyware," the writer of the first antispyware tool. He's been doing security for a long time and is a trusted source for all of us. Thank you for being here, Steve.

Steve Gibson: Eh, and it shows, yes.

Leo: Well, and actually...

Steve: I lost a week.

Leo: Yeah. I bet you did.

Steve: This TrueCrypt thing took a week out. So I'm so excited. I posted back in the SQRL group this morning that, after the podcast today, I would be back in. We left things a little up in the air. What we're deciding on is where we want SQRL to look for its users' identities. So, like, go look in the My Documents folder, then we'll look in the program's own execution directory, and then look in the current working directory. That way it'll support portable, if you, like, bring it on a USB stick, and you've got your identity there on the stick with you, SQRL will be able to find it there. If you put it in the My Documents folder, that's sort of like the universal per user folder, so that way it'll handle roaming profiles and multiple user systems. So we're at that level of detail now where the rubber is in contact with the road.

But then this happened on Wednesday. And I was watching you, in fact you know I was watching you because I was in the chatroom noting that the signatures did match between TrueCrypt versions. This was immediately after the world learned that something had happened to TrueCrypt. And we were all scrambling around trying to figure out what. And basically...

Leo: This is right in the middle of This Week in Google. And, you know...

Steve: Right.

Leo: ...of course everybody going to the TrueCrypt.org page looked at it. And I think a lot of people said, oh, it's TrueCrypt's been abandoned. And my first reaction is, well, I would like to see more. This sure looks like a bad hack, just the way it's written. The idea that the folks who wrote TrueCrypt would propose Windows BitLocker as the alternative, I mean, essentially TrueCrypt was written because they didn't trust BitLocker or any closed-source solution.

Steve: Actually, it was written well before BitLocker.

Leo: All right. But there were encryption solutions from Microsoft before BitLocker that nobody really wanted to use. And so it seemed odd to me. And the way it was written, also, the reference to the fact that Microsoft had terminated support of XP puzzled me. So my initial reaction was, well, we don't know. But, boy, I'd like to see some explanation of this before I would jump to the conclusion that it's real.

Steve: Right. So for our listeners who don't know, this episode of Security Now! is titled "TrueCrypt: WTH," or WTF, depending upon how...

Leo: Choose the letter of your choice.

Steve: ...family-friendly you wish to be. I went for "F" because, boy, it's been quite a week.

Leo: Well, and that's the reaction we all had, like, what happened?

Steve: Yeah. And, well, so that's - you and I are going to talk about this ad infinitum, at length, in the second half of the podcast, after we talk about a little bit of news that managed to squeak its way through the TrueCrypt hubbub during the course of the last week. I want to talk briefly about news of troubling, sort of this general troubling misuse of the U.S. Computer Fraud and Abuse Act; the fact that Chrome has tightened up its security in an interesting way. We'll talk about the Brian Williams-Edward Snowden interview which occurred the day after last week's podcast, last Wednesday. Some strange news about the Zeus Botnet and CryptoLocker which is, like, more not news than news, but everyone's covering it because CryptoLocker is now a buzz term. I want to talk a little bit about the WWDC that happened yesterday, and so my favorite new planned features in iOS 8. A quick note about "Halt and Catch Fire," which aired for the first time two nights ago, on Sunday night. My absolute final comments ever about shaving. And then we'll talk about TrueCrypt.

Leo: No, no, you don't have to paint yourself into a corner there. You may speak about it again.

Steve: Well, it's actually - I've done my job. And it's been enough time now that people who heard me raving placed their orders, received their product, shaved their face - I presume it was their face - and tweeted me their reactions. So I have four that I had this morning. And since I produced the PDF, two more have come in. And so we will - my point being that, if people don't get it by now, they're not going to, so I'm not going to belabor the point any. And then we'll talk about TrueCrypt. So, I think, a great podcast.

One of the things that is really troubling is when legislation is written to pass, which often means it's blunted. So you can have someone who wants to put together a law. And I guess this is often referred to as sausage-making because it's a process you really don't want to witness because you don't end up wanting to eat the product after you've seen where it came from. So somebody has a great idea about a new piece of legislation. And they write a nicely assembled piece, and then there are objections to it. Always it's going to be the case that there will be other people who are sort of mildly against it, or maybe they're against the person who wrote the legislation, so they just don't want to, you know, they just want to mess with them. So what ends up happening is you get legislation which oftentimes is too open to interpretation. And "open to interpretation" is an attorney's favorite phrase because it means everybody gets to argue while the meters are running, and that's what keeps them in business.

So we have a situation like that with the so-called U.S. Computer Fraud and Abuse Act. And I picked up on an interesting article in The Guardian in this last week which had the - it talked about this Act, and it said that "Security researchers say they've been threatened with indictment for their work investigating Internet vulnerabilities." And it cited a number of people, H. D. Moore, for example, who essentially left the sphere of work for a while. I mean, he was a little bit on the darker gray side of neutral - I think he was darker than 14% gray - because he did the Metasploit Framework, which joyously adds exploit demos the day that they're released and making them very easy for less technically inclined hackers or capable hackers to leverage.

But there are other instances where this legislation is really damaging security. For example, it's often the case now that some researchers will find a flaw in some

company's product. They will do the responsible thing, which is notify the company of the flaw. And we're always on this podcast covering the fact that companies often do nothing. I mean, they're busy; they're working on their next version; you know, it's like that's not good news. So just sort of there's some tension that is created - typically because of the company, their receiving company's inaction over some number of months - created between the people who find a problem which they believe is important, and I know all of our listeners would think it was important, you know, things like backdoors in router firmware, for example, or webcams that you can log into without a password anywhere on the Internet and look in people's bedrooms and nurseries and things. I mean, really, you know, stuff that ought to be fixed.

So what happens, then, is that after some number of length of time of inaction, the people who are in the know say, look, this is important. You seem unwilling to fix it proactively. We've got to go public with this. At which time the company's attorneys immediately turn around and threaten to turn these people, the researchers, into criminals under the Computer Fraud and Abuse Act. And this is the problem is that, unfortunately, the way this is written, it is absolutely feasible to bring a lawsuit. They may not ultimately win. But as we also know, just the threat of suit is often enough to cause people to not take any further action.

And what's now beginning to happen is that researchers in academia are looking at the choices they have ahead of them. It's like, what should we research? And there are areas where they know they're going to be stepping into this problem, very much like the DMCA, which is again a huge problem for people who want to explore the cryptography of DRM content. Here we're talking about fundamental security architecture, the implementation of security on the Internet.

And so what's beginning to happen is that attorneys are waking up to the fact that there's this really broad - and again, that's the problem, really broad law which they can pretty much at will just wield against anyone who upsets them. And the truth is it dramatically weakens the effective security on the 'Net. And I don't know how this gets fixed in the long term. I mean, maybe, maybe something has to happen, probably, that's just generally the way these things work, something has to happen which is dramatic where we then, after the fact, forensically look at, okay, wait a minute, you're saying that people knew about this, and they didn't talk about it? Well, why didn't they talk about it? Oh, it's because they were under threat of lawsuit from the people who didn't want them to talk about it over this law. Well, then we have to change the law. And so we're probably stuck with this.

And what's interesting is that - I did a little more research into the background of this. Congress wants to make this stronger. I mean, again, our congressional lawmakers barely understand how any of this works. And so they're trying to take this in the wrong direction, rather than the right direction, that is, of making it more onerous and more of a broad brush. And the problem is that there are, you know, researchers do have to act in some ways like bad guys. I mean, not with evil intent. But, you know, we're all generating packets.

And so the question is, are these packets being used as probes? And with no intent to actually conduct criminal activity? Or are they being used for the purpose of crime? I mean, that seems cut and dry. The problem is, companies that don't want their own mistakes to come to light can use this in order to keep those mistakes from coming to life. And obviously that does not improve the Internet security. So anyway, I thought that was - it was interesting. And as I looked into it more, I've just found myself sort of shaking my head, thinking, well, okay.

Leo: And it's also been used kind of capriciously and with heavy-handed, sometimes overzealousness by prosecutors. Aaron Swartz, of course, was prosecuted under this Act.

Steve: Yes, yes, exactly.

Leo: And that was an example of somebody being really over-prosecuted. But nevertheless - and it cost him his life. He killed himself as a result. So, yeah, I think that there's no question that this Act is wielded inappropriately often by prosecutors. Some of that, I think, comes from a fear of computing, just from a general fear of technology and a real ardor in kind of getting the bad guys, the hackers. You know, our friend Randal Schwartz was put in jail...

Steve: Yes.

Leo: ...for something very similar. He, you know, the thing is, it's always a little bit gray. It was gray with him.

Steve: Exactly.

Leo: Where, you know, he probably was going where he shouldn't, but he found a vulnerability and reported it immediately. Nevertheless got arrested for it. And that happens, you know, another case is a friend of ours, Adrian Lamo, who similarly, you know, he was pen testing. But, you know, he broke into The New York Times. And, you know, technically it is a crime. And so - and it's often a gray area. But I think probably the worst - I agree with you, it's bad for security, and it is often overzealously prosecuted. And Aaron Swartz is the perfect example, where he really didn't do anything wrong.

Steve: Yeah, and, I mean, we've sort of touched on this before, but the only solution I can imagine is to use anonymity. And it's unfortunate that you have to do that because then you don't get to use whatever credibility you have in the industry to say, look, I'm an honest-to-goodness researcher, and I found this, and you need to pay attention to this. And so you have to use anonymity and try to get their attention. But it is the case that you probably need to protect yourself from - and it's interesting because you said they're afraid of technology. And you're right. But almost certainly these are also companies massively profiting from the technology. And you only want to do...

Leo: Well, I'm thinking of the states attorneys general and the law enforcement people. The people who actually prosecute the crimes I think often over-prosecute because they just are a little terrified by the whole - we know how the FBI treated Ladar Levison.

Steve: Yes.

Leo: They're overzealous, I think.

Steve: Just, yeah, just came stomping in and demanding way more than they actually need.

Leo: And this gives them a tool that they can use inappropriately.

Steve: Yeah. So what did you think of Brian Williams interviewing Ed Snowden?

Leo: I haven't watched it yet. I'm waiting for your opinion.

Steve: Okay.

Leo: I've read, of course, all the reviews and the synopses. But I haven't actually seen it.

Steve: Yeah. And it didn't really - I think the one piece of news, and I guess I call it news because I kept hearing it repeated over and over and over for days afterwards, was Edward's assertion which The New York Times, or, I'm sorry, NBC was at least in part able to verify was that he had brought his concerns to the attention of upper management.

Leo: Right, right.

Steve: And so he'd, like, he's worked within the system, within channels, and he asserted numerous times, ongoing dialogue, you know, he was explaining to them why he felt this was completely extra-constitutional behavior that was really beyond the pale. And aside from that, I just got another sense of what an articulate, thoughtful person this guy is. He did assert, because Brian asked him, would you like to come home to the U.S.? And Edward said yes.

Leo: Of course.

Steve: I can't, but I would like to. Of course I would like to. So...

Leo: And I think at some point it might be appropriate for him to come home and face the music. I mean, it's a risk, as we know from Chelsea Manning. But...

Steve: Yeah, I mean, I guess the concern is whether - what kind of justice he would face.

Leo: I understand. But this...

Steve: Everyone says, "Come back and face justice."

Leo: At this point he ain't gonna be spirited away to Guantanamo Bay and have a secret trial. He's too public. So if I were him, I'd make a very public show - I bet he does this - a very public show of saying, all right, I'm going to come home and defend myself. And be very public about it. And I think the press would make sure that he got as fair a trial as he could.

Steve: We do have his temporary asylum expiring shortly. So it's coming up on a year now when Russia said yes, you can stay for a year. And it's not clear what's going to happen afterwards.

Leo: Yeah. And he's got lovely skin, I'm told. He's very, very...

Steve: Well, I guess he's having a good time.

Leo: He's got a lot of time for spa treatment.

Steve: Does caviar do that for you?

Leo: Oh, very good for your skin, yeah.

Steve: Okay.

Leo: All that fish oil.

Steve: So there's just a strange story. And I don't - this is annoying because, if we knew more, maybe it would be interesting. It's getting headlines. But nobody seems to be saying anything. So nationalcrimeagency.gov.uk. The Guardian picked it up. I've seen it in several places, you know, picking it up. And that is that, from what I've been able to see, a federal court in Pittsburgh, of all places, decided to allow the FBI to redirect the automated requests by victim computers for additional instructions - these are botnet computers, so we're talking about the well-known, it's called the GameOver Zeus Botnet.

And of course we've talked about Zeus a lot, and also CryptoLocker. And what's significant about this is that the good guys cannot interfere with computer communications without themselves breaking the law. I remember being in a discussion with the attorney general years ago during something, I don't know if it was Nimda or Code Red or what it was. But it was one of the worms. And we were inquiring whether it would be legal to disinfect machines that had been infected by this. The deal was...

Leo: Without their owners' permission.

Steve: Right. Right. The traffic being generated by whatever this was, I don't remember the incident, but it was not spoofed. So anybody sniffing and collecting packets on the Internet was getting the actual IP addresses of the infected machines. And so there was a huge temptation to build a disinfector, which would sit there, and when your IP or your 'Net was probed by an infected machine, it would send something back to disinfect it, which we had the capability to do. Except it was illegal. Even though these machines were already infected and were spreading an infection, it was against the law for white hats to disinfect the machines.

So apparently what happened is - and this is where this is all fuzzy because there's no details in these stories. But there's like this weird - people in the U.K. are being told they have two weeks to protect themselves. It's like, what? Whaaa? So the only thing I can figure, from reading everything I have, is that what the court did was provide a two-week interruption, that is, permission for the FBI, through the DoJ, the U.S. Department of Justice, to interdict the communications of the botnet for a fixed period. Can't do it forever. We're going to give you two weeks. I mean, I don't know what good that's going to do. Maybe we'll get more news or information a week or two from now. So we'll keep our eyes open. But a lot of people were tweeting, saying, hey, what is this about? And so I did all the research I could find and plowed in, and this is my best guess. So we'll see if that turns out to be the case.

Now, we've often talked about Zeus. And I was talking about it in relation to banking and building firewalls between accounts. And I did pick up essentially an echo of what we've been talking about. In one article it mentioned that security researchers estimated that between half a million and a million computers worldwide are infected with this GameOver Zeus botnet.

Leo: Half a million, wow.

Steve: Between half a million and one million. So as many as a million. And that approximately 25% of the infected computers are here in the U.S. They said, "The total losses worldwide are unknown, but we believe that the losses exceed $100 million to U.S. victims." And finally it finished by saying, "Because many of the victims are small- and mid-sized businesses, their accounts typically do not have the same legal protections afforded to consumer accounts, so such losses can be devastating." This was exactly where we got a couple weeks ago when I realized that I was talking about my perspective as a small business, and you knew that that seemed wrong from your own experience, Leo. And indeed, consumers have protections which small- and mid-sized businesses don't. And so..

Leo: And others outside of the U.S. often don't, as well. We should - many of our listeners are outside the U.S. These are relatively recent U.S. banking laws.

Steve: And unfortunately this Zeus botnet is around and thriving because there's such tremendous incentive for the bad guys to get this onto people's machines. It is very banking website aware, so that it watches what you do, and it knows which sites you're going to and how to steal your credentials on the fly. And so this thing gets into your

computer. You do your online banking, and this thing says, thank you very much, and drains your account.

Leo: Holy-moly.

Steve: And if you're a small business, your money's gone. There isn't government protection, and there's no recourse. The bank - and from the bank's standpoint, they saw you withdraw the money. They saw you transfer the funds. So, and there are typically no records left. So how do you prove to the bank that this wasn't you? And in any event, the bank doesn't care. You transferred your money away, or someone did, but as far as the bank's concerned, they obeyed a lawful request for a money transfer. And it's just gone. And, I mean, that I got directly from agents of the FBI who I was talking to in the early days when I was setting up my eCommerce site. I had a chance to say, okay, from your experience, guys, what are the gotchas?

And the first thing out of one of my local contact's mouth was absolutely be careful about wire transfers because that's what we see over and over and over is - and so that's where I set up firewalls and just turn transfers off on any accounts where we park cash, that are not, you know, checking accounts, where we're dynamically moving cash in and out. So again, to our listeners, please be careful. And that is, you just don't want to get one of these things in your machine, especially if you're a small business. And so we'll find out what this two weeks is, maybe, at some point.

Okay. So I watched your coverage yesterday of the Worldwide Developers Conference...

Leo: Thank you.

Steve: ...that Apple did, and iOS 8. And I was - I'm very excited about 8. They're giving us features which I think they can probably do securely. I like the idea of starting from something that, if anything, is too closed, and then carefully opening it once you've built a security infrastructure that allows you to open it securely. And we had our famous three-episode podcast on iOS security, where we looked at this incredible infrastructure which has been built and is present in iOS 7 now.

So what they're beginning to do now is Apple hears what people are asking for, and they're looking over at Android and seeing what people are able to do on that platform and recognizing, okay, we need to be a little looser here, but not sacrificing security. So with the base of security they have, they are now going to be allowing a sort of a controlled interapplication communications. So applications will be able, under user control, to surface some of their user interface elements in other apps.

So, for example, you could bring up the little Send To panel in Safari. And in addition to sending it to email or iMessage, you could send it to other apps. And, for example, it'll be really interesting to see what LastPass may be able to do. We may finally be able to get, and in fact I've already got a dialogue open with Joe, and he hasn't seen this enough in depth yet. But it looks like Apple will be making the so-called DOM, the Document Object Module - wait, the Document...

Leo: Yes, that's correct.

Steve: Object Module? Sounds wrong.

Leo: Model, model.

Steve: Model, that's it, model - available. And essentially it's this beautifully standardized hierarchical description of a web page. And you can traverse it. You can explore it. And, for example, in there will be the form fields which are, you know, to be filled out by LastPass. And so it may very well be that we get, for example, in Safari on iOS, the same level of LastPass integration that we are enjoying on traditional browsers on our desktop operating systems. So that would be really nice. So they're essentially curated, sandboxed, interactions is the way Apple is describing them. So, again, not a free-for-all, but more interapp communication, which I think is going to be a great move forward.

Leo: I think it's well done because the receiving app and the sending app have to cooperate. And in effect the sending app is a standalone app, it's called an "extension," that operates as a standalone app and is - I think they've done it exactly right, and I think you make a really interesting point, that if you start from security and work that way, it's much easier and better than to start from insecurity and work towards security. So I don't think - I think what they're doing, you know, judging from what I've read, is the exact way to do it.

Steve: Yeah.

Leo: Yeah, it's exciting.

Steve: And, Leo, custom keyboards. Oh, be still my heart.

Leo: Finally. Well, frankly, that was why I left iOS, so...

Steve: Oh, god, I know. And it's why I've got my little Typo keyboard, my BlackBerry clone. I mean, I tried the Fleksy app. And it's like, okay, so I could kind of type onto a Notepad. But then I've got to copy and paste everything over into the other app where I want it to go. So now we're talking about user-installable third-party keyboards. So, yay. The idea of being able to experiment with something better than this ridiculous keyboard where, if your fingers, after you press the button and it goes click, then your finger slides off it, and it doesn't register? It's like, what? You know, that's the definition of the iOS default keyboard now.

And I guess they've added something called QuickType where it knows who you're talking to and learns the kind of communication, like how formal you are with that contact, and then does word prediction and posts the words above where you're typing. And so when you see the word you've given it the first few characters of, you say, oh, yeah, that's what I meant, and you just tap that in order to type the whole thing in. So anyway, so that's being moved forward. But, boy, I love the idea of an ecosystem of attempts to improve the keyboard. Yay. So maybe that would be great. I mean, the problem is I'm going to want the big phone, and of course no more Typo keyboard on the

big phone. So I'm going to go back to an onscreen keyboard. But it looks like I may have a choice of which one. So I'm jazzed about that.

And they're opening Touch ID, as we hoped they would, to third-party apps. So that's just great. What that essentially means is that an app will, I mean, like a perfect example is SQRL. The thing I need for SQRL on the iPhone is for you to continually reauthenticate so that a sibling or a friend hasn't picked up your phone and is using it. Thanks to the fact that SQRL is able to authenticate for you, the thing we need is you to authenticate to SQRL.

So the idea that SQRL would be able to say - pop up the little fingerprint image, and you go, oh, yeah, and put your fingerprint on Touch ID, and then there's a secure communication. And I'm sure Apple will have done it right. We have no documentation yet that I've seen, but that'll be coming shortly, where there's a secure communication to authenticate that you are who you are. And then that gives SQRL permission to authenticate on your behalf to the website you're visiting. So, and, I mean, all the other apps that want to have on-the-fly reauthentication will be able to leverage this. So that's just wonderful news.

Leo: Yeah.

Steve: And I also saw something that didn't make the headlines. But per-app battery usage. Isn't that cool.

Leo: Yeah, well, it's something everybody else has had. But it's time. A lot of these things, I mean, to be fair, are things that Apple's playing catch-up on, like keyboards and so forth. But it's good.

Steve: Yeah. And I'm happy that Android is there. I'm happy that Android is there to apply pressure to Apple because I want...

Leo: Well, then Apple's going to integrate it in a more elegant way, et cetera, et cetera. I mean, the per-app battery info is what it is. There's no elegant way to do that. It's something they ought to have put in there ages ago because otherwise you can't figure out what is draining my battery?

Steve: Right. And I am also glad that the whole HomeKit initiative, I would, again, because Apple is Apple, um, because they're not advertising based, and they're not Belkin or Linksys, that are just using other people's firmware with backdoors that even they don't know about, I'm really comfortable with the idea of Apple getting into the home automation market and bringing their security model and their approach into our home. I would prefer them, frankly, than anybody else I can think of. So I just think that's good news.

Leo: Yeah. They, of all the people out there, they're the ones who could actually make this finally work.

Steve: Yeah. When I heard that Google had toyed with the idea of putting ads on the NEST thermostat, I thought...

Leo: That was never - that was a false rumor. But...

Steve: Good, good.

Leo: That's absurd.

Steve: The last thing I want to have to do is put adblocking software on my thermostat.

Leo: Google denied it. That was a false rumor. But I think that the larger point is that this will be an alternative to people who don't trust Google. Many of the things that you do with Google now, like Google Drive, will have its equivalence on the ecosystem, on the Mac ecosystem. And for people who prefer privacy, this will be a good choice.

Steve: So did you see "Halt and Catch Fire"?

Leo: I haven't. And, you know, I've seen mixed reviews. And I know you were less impressed than you thought you'd be.

Steve: Yeah. I saw the first five minutes. And the good news is it got better. So I did watch it, and I think it's promising, and I am hopeful. And it was fun. There is, you know, nonsense. I mean, obviously I'm watching this and could do what these clowns are pretending to be doing on the show. And he's got some huge board of the white plugin jumper boards covered with LEDs, and he's reading hex off of it. And the first example he gives is of reading a "B," and unfortunately it's showing the hex for "D." It's like, okay, well...

Leo: That's pretty bad. But are there any really bad, like, oh, this is so...

Steve: No. Overall, overall they're carrying the theme. I mean, this is Compaq.

Leo: Is it, do you think? I mean, is it...

Steve: Oh, yeah, yeah. No, this is definitely...

Leo: Do they say Compaq, or is it just...

Steve: No, but they call themselves...

Leo: It's a roman clef.

Steve: They call themselves Cardiff something. And I don't know if I remember that, or if that's just...

Leo: Cardiff is a familiar name, yeah.

Steve: Yeah, I'm kind of thinking maybe that their Cardiff predated Compaq. But it is absolutely the BIOS. The engineer says, he's, like, taking this IBM PC, slides the familiar gray steel cover off and pulls the boards out. And so he's the engineer to sort of the wheeler and dealer sales guy. And he's showing him how it's all made from standard, off-the-shelf components except this one chip. Now, it was a little bogus because they had showed four of them, and he said, and we don't know which one it is. You think, well, yes, you do, it's the one with the copyright notice on it there.

And so they, like, remove this chip. And they show, like, using some solder sucker in order to pull the chip off, even though they later removed it from its socket. So it's like, okay, well, that didn't quite make sense, either. But so there are little things like that. But, so, fine. And but then they manually read out the 64K BIOS, word by word, and wrote it all down. And I don't know why they wrote it down because then the next thing we saw was that it was scrolling on the screen of a TRS-80, and then it was being printed out of a printer, and then they were holding and kissing the disassembly of it.

Leo: Well, that's not cleanroom reverse-engineering.

Steve: No, no, no. And the other weird thing is that, I mean, I have the source from IBM. They published it in the IBM PC Technical Reference. It was my bible while I was writing, first FlickerFree, and then SpinRite. So IBM was completely open about this. So what happens is IBM gets wind of this, and the attorneys between the two firms talk, and now they go out and find this hacker chick who was the one that was having sex in the first five minutes of the first, well, actually, of this episode. And she's going to be the cleanroom engineer who has never had any contact with a PC at all. And so she's going to write from scratch an API-compatible interface BIOS.

And that's the story of Compaq. I mean, that's what Compaq did. They did have the non-cleanroom version. And actually, as a backup, as I remember it was a backup, they had the cleanroom version in case they needed it. So anyway, it's a - I would recommend people watch it. I'm sure AMC will be airing it this week. And then plug it into your DVR and see where it goes. I'm definitely going to be watching. I thought it was fun.

Leo: Good.

Steve: I got an interesting note from Jason in Portland, who clearly knows his way around machines, saying that SpinRite isn't obsolete yet. He said: "Steve, lately I've been wondering if SpinRite hasn't been rendered obsolete by SMART and sophisticated drive technologies. Could it be that SpinRite's supposed miracles are simply because it forces the drive to take a hard look at itself, and all the repairs are being handled by the

drive itself? I wondered if booting from another drive and running" - and then he gives a, he says, a `cat /dev/sda, which of course is a Linux term for the hard drive, and he's then piping that to /dev/null, so basically he's just sending the entire drive to nowhere, which basically causes the machine to read the entire drive. He's wondering if that "would accomplish the same thing as a Level 2 scan by virtue of reading every sector."

Well, I already know that it doesn't because what SpinRite does that that won't is it shuts down all of the fancy stuff that ignores errors in the drive specifically to find them. So that's one of the big reasons that this is not the same as copying the drive to another drive. SpinRite gets in and basically makes the drive as stupid as possible in order to make it as sensitive as possible to problems, which would otherwise continue to be missed.

Anyway, he goes on, saying: "I found the answer last night. I updated GRUB, my bootloader, some time ago and just noticed that my seldom-used Windows drive was no longer on the list. I tried a few tricks to get it to boot, but Windows would always partially load, and then the system would reboot. So I reached for SpinRite and ran Level 2 on the drive. When it was complete, I tried rebooting again, but the symptoms were unchanged. Next, I got a little more aggressive. I started Level 5 and went to bed. When I awoke, the scan was about one-third complete, but I tried booting the drive anyway. It was a success. Windows loaded right up!" He has an exclamation point.

"I finished the scan for good measure, ran `update-grub2' from my main OS, and I'm back in business. This experience shows that drives aren't yet smart enough to take care of themselves. Thanks for the great product and your wonderful shows with Leo. Jason, Portland, Oregon." And, Jason, thank you for sharing that.

Leo: Thank you.

Steve: And indeed, there is still a lot of magic, as I mentioned midway through that, that SpinRite is doing, way more than just letting the drive read itself. It gets, you know, it's much more proactive.

Leo: Somebody in the chatroom insists I ask you about the new programming language that Apple debuted to replace Objective-C, Swift. I said you will have no opinion because you're an assembly language programmer, and anything higher than assembly language is dreck.

Steve: Well, so, yes, I'm not going to use it. But I certainly salute them for, as an aid for all the people who, I mean, I think this is a very interesting move forward for the C-like language. For example, I saw one of the security guys at Google picked up on and really liked the fact that data types were no longer allowed to overflow silently.

One of the things that happens is, for example, you define an unsigned integer, you know, a uint in C, and you can add two uints whose sum will not fit in a uint, and it wraps. And it does so silently because C, despite its high-levelness, is still a relatively low-level language, which is really the way it was designed, as sort of a high-level assembly language to write UNIX in, in the old days, back at AT&T and Bell Labs.

So Swift doesn't let you do that. Swift notices that this addition overflowed, which otherwise produces a bizarre value. But in Swift it says, whoa, you know, raises an

exception, and so the programmer is able to catch that. And other things like, I mean, I listened to you...

Leo: There's lots of stuff like that. No pointers. Yeah. I think they've attempted, as the Department of Defense did with Ada, to create a language that encourages good programming practice, obviously. It's always possible to write an exploit.

Steve: Now, it is, well, and I guess one of the things that's a little troubling is we don't know what the adoption is going to be. The good news is you can smoothly transition. You can write new things in Swift and add it to your existing Objective-C blob. Somebody starting from scratch might just want to start with Swift.

Leo: Oh, yeah.

Steve: But so there's no obligation to program in Swift. And I do love that playground, too, the fact that they have an...

Leo: It's cool, yeah.

Steve: Yeah, an interactive mode where you could just, like, build code and watch it go without having to do a whole project recompile.

Leo: And you saw us, as we were talking about Swift, too, one of the features, and I'm not surprised Apple put this in, is that switch structures, which are case statements, don't fall through at the end. You can't have, you know, you have to have a result.

Steve: Have a default.

Leo: And that would prevent the GoTo: Fail bug that bit Apple so hard. And I think there's no accident that they did that. And you, by the way, and there are no gotos, but you could still probably do something really stupid that's...

Steve: Well, and I'm not a big fan of an autotyping language, or a soft typing language, because I think that's one of the things a programmer should really do and understand is deliberately establish the type of their variables because then the compiler can help you so much in making sure that you abide by the type conventions that you set. Now, there's a tradeoff. I mean, for example, JavaScript is just - it just uses up - it just, you know, type abandonment, essentially. And so, and maybe there are, like, pragmas that you can use to turn off those things in Swift, to say I don't want autotyping, I want to, you know, be forced...

Leo: Oh, no, it's a typed language. It's much more a typed language than something

like Python, which really is truly autotyped. I think that Swift solves that problem very nicely and elegantly.

Steve: Well, it does say that it autotypes. So it sees, from the context of your usage, it decides what type is.

Leo: Right. But that's the - the usage is the declaration. And it cannot change. So casting and the kinds of things you do in C routinely can't be done. And that's always a problem is casting a string into something that it can't be cast into or vice versa can cause crashes and can cause insecurity. And this, I went through the - I've been going through the whole, as you saw, all 850 pages of the language definition.

Steve: Yes, I did, I watched you. You refused to end.

Leo: I wouldn't stop.

Steve: All of your MacBreak guys were there.

Leo: I wouldn't stop.

Steve: We've got 867 pages.

Leo: We're almost there.

Steve: We're going to finish.

Leo: And then last night I spent a considerably longer time with it and the language itself. And I'm pretty impressed. But, you know, we'll see. There's always a chance for...

Steve: And these days we have a language every week. I mean, we've got...

Leo: Well, that's true. This will get adoption, and I'll tell you why. Objective-C, which was really from NeXT and is only used on Apple, got adopted because it's the native language.

Steve: Right.

Leo: This will get adoption. It's close enough to C and to any language. If you've

ever written software in a high-level language, it's very, very understandable and straightforward. So absolutely this will get adoption. And Playground alone, it's got an excellent debugger. The IDE Xcode has always been very nice. You assembly language guys, you can keep doing your own thing, but...

Steve: And we will, proudly.

Leo: I'm sure you will.

Steve: Okay. So as we mentioned at the beginning of the show, if there's any listener here who doesn't already know, we were all - we, the industry, the Internet community, were shocked when on Wednesday of last week the longstanding TrueCrypt.org website disappeared. TrueCrypt.org started redirecting to a TrueCrypt subdomain. I think it was a subdomain. Maybe it was down in the directory of SourceForge. And there was a really screwy-looking page, I mean, no nice formatting, no stylesheet, just sort of scroll down.

And first it starts off by warning us. There's a first line in red: "WARNING: Using TrueCrypt is not secure as it may contain unfixed security issues." And then, Leo, as you mentioned, it made sort of a strange tangential reference to how Windows support for XP had stopped in May of 2014.

Leo: That's like saying somebody put mayonnaise in my sandwich. It's a non sequitur. It doesn't...

Steve: Exactly.

Leo: Has nothing to do with anything.

Steve: It has no bearing. Yeah. And then saying, oh, and besides - I'm blanking on Microsoft's...

Leo: BitLocker.

Steve: BitLocker. BitLocker is available for Windows. Well, it's like, wait a minute. Not all Windows. I mean, only...

Leo: Right, Windows Professional and...

Steve: ...the Pro and the Enterprise.

Leo: Yeah, yeah.

Steve: Yeah, so not...

Leo: And really [TrueCrypt] was created because they didn't trust proprietary encryption, whole-disk encryption. So it's just very odd.

Steve: Yes.

Leo: Why would they send you to the thing, the very thing they were created to eliminate?

Steve: Right. So since then I have done three blog postings, one each day - one Wednesday, one Thursday, and one Friday. The Thursday one was a little controversial. I made up a letter as if it was written by the TrueCrypt guys. And I said, here is an imaginary letter from the TrueCrypt authors or developers. And so that generated some dialogue. But I spent some more time looking at the code and looking at the license which they took from version 3.0 to 3.1. And what was very clear to me was that, while this may have come as a surprise to us, it was not a surprise to them.

This was - it was a huge effort to essentially neuter the fully functional TrueCrypt v7.1a into what they called 7.2. And what they essentially did was they put throughout the code a warning notice saying that it might be insecure, and they removed the ability to create new TrueCrypt volumes. Essentially, all 7.2 can be used for is mounting existing TrueCrypt volumes, presumably for the purpose of removing TrueCrypt.

And in fact it looks to me, reading through the code, that the process of encrypting an existing drive was interruptible and restartable. In fact, I know that was the case. But it doesn't look like decrypting it used to be. They added interruptibility of decryption based on looking at the differential between the old files and the new files. So, I mean, they added, they did some serious work in order to pull this off.

So I did three blog postings. Then the next day, on Thursday, I popped onto Mike Elgan's Tech News Today show, and also Tom Merritt's podcast, to talk to both of them about this and what I was thinking. And the big problem was that, as listeners to this podcast know, I'm sort of anti-conspiracy theory. I mean, it took me a long time to acknowledge that - now I'm blanking again on the government-sponsored worm that went over and got the [Iranian] nuclear centrifuges. Stuxnet. Everyone was, like, oh, you know, Stuxnet. We were all guessing in the beginning about what Stuxnet actually was. And I was reluctant to acknowledge that this thing was nation-state born, as it does now seem that it was, we and the Israelis working together to pull that off.

So, similarly, I mean, the first thought was, was this a site defacement? Did someone hack the site in order to put up this wacky page? I mean, it seemed that hard to understand initially. And we now know that's not the case. The source, the people who manage and run SourceForge have received no indication from the developers that their site was hacked, that they don't want this to be there, that this was a mistake. And we're now seven days into this. So we know that this was deliberate. We also know that it's deliberate from, as I said, the extensive reworking of the 7.1a code to create 7.2.

My final piece of work was to create a TrueCrypt archive on GRC, which now exists and will forever exist. So no one need worry that TrueCrypt is going to go away. I have it. And under the Other item on the main menu it says TrueCrypt Archive, and there's a

page with all of the 7.1a materials. All the builds...

Leo: But if you believe this page, why would you want to use it?

Steve: Well, because I'll explain the page. I do believe the page, and I believe I understand the page.

Leo: Part of the - we should explain that a lot of this is complicated by the fact that the developers are, and still are, anonymous. And so no one, I mean, there's no guy you can call up and say, hey, guy, what the hell's going on here? Because no one knows who they are; right?

Steve: Okay. I don't think that's germane. But okay.

Leo: Well, it's germane in the sense that, because they're anonymous, there's no way to say, "Are you still alive? What is going on?"

Steve: Well, you don't know that we've heard from them, and we have. So I'll get to that in a second. So...

Leo: Well, again, if you're anonymous, I don't know how you prove that they're them.

Steve: And, Leo, anyone can have any conspiracy theories they like.

Leo: Not a conspiracy theory, though. It's just that, well, I'd like to hear what you have to say about it. I just - and I know you considered having the key, the hash, the MD5 hash to prove this is from them and all of that. What we just...

Steve: Well, Leo, just look at the code. There's no other explanation for what was done between 7.1a and 7.2. I mean, I'm looking at evidence.

Leo: Okay. Then why would we want to use TrueCrypt if that's true?

Steve: Because it's been just fine for the last two years. And so, okay.

Leo: Keep going. Because I'm interrupting you. So go ahead. But I did want that one fact in there, that the developers of TrueCrypt are not known. We don't know who they are.

Steve: Nor have they ever been.

Leo: Nor have they ever been, right.

Steve: So they didn't disa- yeah. Correct. So I put up a page on GRC where people could go to get TrueCrypt. My belief is that these guys, what became clear to me in looking at this, reading the comments in the code, looking at their website, and then I posited my own theory that made the most sense. And then that was confirmed by their communication with some people they had spoken with before, who were then tweeting back and forth with Matt Green at Johns Hopkins.

And essentially what happened is that they just got tired. They got done. They'd been running this thing for 10 years. We know that they weren't making much money. They were asking for donations and weren't receiving them. It's a thankless job. Imagine putting together a Trust No One encryption system where, if you lose, if you forget your password, there's no one you can turn to. And so, in response to that, they develop a system where they make you burn an ISO and boot the disk in order to prove that you've got the recovery key and so forth, which is I think reasonable precaution for something that is going to be a Trust No One solution.

So what seemed to me entirely reasonable was - and seems still - was that these guys who, and I did read that they started off in their mid-20s, 10 years ago, so now they're in their mid-30s, were just sort of winding down. 7.1a, the most recent release from February of 2012, so 27 months ago, hasn't changed because there's no bugs. I mean, it's finished software. Millions of people use it. It's robust. And the one thing that it has that was brought to my attention is cross-platform compatibility, that is, it runs, the same volume will mount on Windows or Mac or Linux because it carries that across all three platforms. So it is a useful, I believe, robust piece of software.

And so my theory is, imagine that these guys decide they're done. I mean, they created what they wanted. They have a piece of code. And now out comes Windows 8. Their code doesn't support the GPT. It only supports the master boot record. And there's the whole Windows 8 safe boot complication and all of that. So they're looking, if they want to continue to move this forward, at a substantial effort in order to give this thing Windows 8 compatibility. And they look around and think, you know, why? We're done.

Now, the argument has been made, completely reasonably, that they could have changed the license and turned this over, sort of formally, in an announcement that they're retiring. I can understand them choosing not to do that. It has always been the case that the TrueCrypt license was more source-available than open source. It wasn't GPL, so it is not in a lot of the standard GPLed OSes. Many of the OSes have their own proprietary full-drive encryption solutions, and they use those. But they're not cross-platform, and that's one of the nice things about TrueCrypt.

So looking at this code, I mean, I was amazed at the quality of the coding. It is lovely work. It is beautiful. And I could easily see them saying, you know, we've just been listening to what happened after the same decade of span with OpenSSL and what a catastrophe the OpenSSL source code became. We don't want that happening to our TrueCrypt code. So we want to just say it's over. Don't trust the...

Leo: Well, why not say all of that? Why be so cryptic, if you will? Why not just write that?

Steve: They're not me. Now, what they did say - so Steven Barnhart is somebody who has interacted with these guys before. Early in the morning on Thursday he wrote two letters to "David," with whom he'd communicated. And I've seen David's last name, but I can't pronounce it. It's got a bunch of extra unicode stuff over some of the letters, umlauts and so forth. And he received two responses later in the day, which he then shared with Matthew Green. And so in a series of Twitter exchanges which I picked up on, he quoted TrueCrypt developer David saying, "We were happy with the audit."

So one of the questions was, you know, were you, well, I should mention that one of my own pie-in-the-sky theories, this wasn't actually one that I was promoting. I believe these guys just decided to be done with it. And this whole go over to BitLocker or other native platforms is them actually not wanting responsibility for the future of TrueCrypt. They want people off of TrueCrypt. They never want to think about it or hear about it or be asked about it or bothered by it or expected to support it again. And in practice, the only way to make that happen is to have people stop using it.

So David wrote, as reported by Steven tweeting to Matthew, and Matthew subsequently asked for email headers in order to do some of his own detective work, "We are happy with the audit. It didn't spark anything. We worked hard on this for 10 years. Nothing lasts forever," is a quote from the email that one of the developers is believed to have said. Then Steven, paraphrasing, said the developer personally feels that forking is harmful. Quote, "The source is still available as a reference, though."

So they're really not wanting people to take this and despoil it. And frankly, from what I've seen, that is what would happen. I mean, what they have is immaculately - I'm stunned by the quality of this, and the idea that they pulled, they held this thing together over a span of 10 years, and this is what it looks like today. Steven then said, "I asked, and it was clear from the reply that he believes forking's harmful because only they are really familiar with the code."

Leo: I understand with a security product like this you've got to be - I can understand why they'd want to be careful with it. But this is not how open source projects work. But okay.

Steve: Well, this is not how - I don't know what that means.

Leo: This is not typical...

Steve: This is not an open source project. And this is, I mean, this has been uncharacteristic from the beginning.

Leo: I'm confused. Is it not - is the source code not open?

Steve: No. It's available, but not open.

Leo: Because of the license.

Steve: Right. And the anonymity of the developers. I mean, that's another thing that's atypical of open source. So this is not how...

Leo: Yeah, for a security program I can understand that, yeah.

Steve: Yeah. So for you to say this is not how an open source project works, I don't know what that means. I mean...

Leo: Well, if it's not open source, then it doesn't mean anything.

Steve: Oh, okay.

Leo: It's on SourceForge. The source code is open. The license is a little odd. I'm not clear that it's not not an open source, either.

Steve: And the license is a little encumbered, too, because remember - do you remember ScramDisk? Because I remember using ScramDisk. That was the progenitor of this. There was something called ScramDisk, and it later became TrueCrypt. And if anyone's curious, Wikipedia has a good historical look at this because there was a security-related company; someone left and apparently lifted the source code and then repackaged it; and then the TrueCrypt guys assembled themselves. And then attorneys were talking, and then so they backed away, and for a while they were using other pieces of sort of quasi-available licensing. So it's not nearly as clean as a written-from-scratch creation. Anyway, so Steven said, he said, "I asked and it was clear from the reply that he, David..."

Leo: It was funded by GPL for a while.

Steve: Yes, true, true.

Leo: Yeah. It's such a strange story all along.

Steve: It is. And he said, in effect, so they believe forking's harmful because only they are really familiar with the code. Then Steven tweeted, "Also said no government contact" - because, again, they were asked explicitly, did someone approach you? Because one of the wild theories was the so-called "warrant canary." People were conjecturing that this weird behavior was a consequence of their receiving some sort of a National Security Letter or something, and so this was their way of doing a Ladar Levison, essentially, la Lavabit, and just killing it in order to send the message that it's insecure.

As I was going to say, talking to Mike and Tom on Thursday, when they were, like, soliciting alternative theories from me, I said, well, okay. I mean, you know, imagine that this is - we know that Phase 1 of the audit is through, and nothing nefarious was found, although I was careful to make sure that our listeners understood that this also wasn't an

audit of TrueCrypt, this was just the startup boot phase, get the system going code, and it all looked fine. Nothing bad was found. And second phase, which, by the way, is still going to happen, and so that's one of the other cool bits of news here is that the IsTrueCryptAuditedYet effort continues. So we're going to get an audit of the 7.1a code, which everyone is using currently and that is still available. And I fully expect it to turn up no problems. But we'll see. And we would have gotten an audit one way or the other.

But one of the theories was, why would this have happened if you didn't just believe that they're done, as I do? It would have been, well, that there was going to be something found, or they worried there was going to be something found, or they would have known there was going to be something found; and they wanted to, A, maintain their anonymity, and also get the hell out of Dodge and not still be around when it hits the fan. So anyway, I don't think that's the case. I think it's going to audit cleanly. But Steven asked, and David the developer responded, no government contact except one time inquiring about a support contract for TrueCrypt, which does sound like some random branch.

Leo: Sounds exactly right, yeah.

Steve: Exactly. And then David said BitLocker is, quote, "good enough," unquote, and Windows was original "goal of the project." So again, this sounds a little defensive, which I believe. I mean, that makes sense to me that they're - I just think they're tired of this. I mean, I just think - and notice we haven't had - there's been no motion, no activity for 27 months, and they've probably been under pressure brought by Windows 8 and the safe boot and the need to support the GPT partition type, in addition to the MBR, the traditional MBR that only goes up to 2.2TB because it only supports 32-bit partition extents. So they probably faced the need to do something. And then they just sort of said, you know, why? We're done. And then finally, quoting the TrueCrypt developer David, David said, "There is no longer interest." And...

Leo: I can see how, you know, this happens a lot in other open source projects where people just get burned out, and they just go, you know, this is no fun.

Steve: Yes, yes.

Leo: I don't want to do this anymore. Now, if it were really open source, they could hand it off. In fact, they could have gone into the - they did modify the license in 7.2. They could have modified the license to make it open-source compliant.

Steve: Yeah.

Leo: I guess what they were afraid of is insecure forks.

Steve: I think that's exactly right.

Leo: You know what I'm saying?

Steve: Well, and Leo, also, OpenSSL, as we know from people who've looked at it, I mean, objective people have looked at it and just retched at the quality of the code. And I can't even - I can't express to you how gorgeous this thing is. It is a big project, and it is immaculately written. I mean, I don't know who they are. But wherever they are, someone is probably paying them really well to write beautiful code. Because they are fabulous developers. I mean, I would trust it just because of the way it looks. I mean, it just - it looks like my source. I mean, it's just - it's beautiful code. So anyway, I think that's the story. There is a new site, TrueCrypt.ch, which is over...

Leo: Incidentally, that's not exactly what the audit said. I understand that that's your opinion, and I certainly trust your opinion. But the audit was not completely praiseworthy on the code. Right?

Steve: Okay.

Leo: Well, I mean, I remember them saying that - okay. "Overall the source code for both the bootloader and the Windows kernel driver" - which is not all of it, obviously, this was just the first stage of the audit - "did not meet expected standards for secure code. This includes issues such as lack of comments, use of insecure and deprecated functions, inconsistent variable types and so forth."

Steve: Hmm.

Leo: That's what the audit said.

Steve: Right. Well, I'm using it. And I know a bunch of other security-savvy people are using it. And I feel it is no less safe today than it was seven days ago, before any of this happened. I think it's always been safe. And the audit will proceed. We'll know probably in late summer...

Leo: I now understand the urgency of the audit. I'm really appreciative of the need for an audit.

Steve: Yeah.

Leo: So I'm glad they're going to continue that.

Steve: Yeah. And, okay, so there were three tweets that came from @OpenCryptoAudit. The first one was: "We are considering" - and this was Friday. "We're considering several scenarios including potentially supporting a fork under appropriate free license with a fully reproducible build." That's the first tweet.

Second one was, like, nine hours later: "More details on our work with the critical infrastructure initiative." And then they said we are continuing forward with formal cryptanalysis of TrueCrypt 7.1 as committed and hope to deliver a final audit report in a few months. So we'll get an audit. TrueCrypt will continue to exist. My guess is that it will continue into the future. It's probably going to have to be renamed because the developers have not turned the name loose. So we'll have to come up with some other name for it.

And the license, I did not - I spent more time in the code than the license. But there was, as you mentioned, Leo, change to the license in going to 3.1. And the license did always allow people to use portions of TrueCrypt in their own code as long as they made it clear that that's what they had done. And I was trying to cram so much of this into my head that I don't remember now which direction that went in, whether some of that language was removed.

Leo: They stripped out attribution language. They stripped out a requirement that you link to their website. I think that isn't indicative of anything. It's kind of them saying we don't want anybody to mention us again.

Steve: Yeah, exactly. I think...

Leo: In fact, it would support your argument, the modifications. The problem is we - I want to say "they made," but that implies that it's the same people who wrote TrueCrypt, and I don't know that that's conclusively proven because they are anonymous. That's my point.

Steve: True. And to support the contention that they are different people, we would have to know, we would have to understand why the true developers have not spoken.

Leo: Right. Maybe they're dead. I don't know. I mean...

Steve: They probably got killed.

Leo: We just don't know. That's the problem.

Steve: Okay, yeah. But, I mean, Occam's Razor is where we are. And that's always where I go.

Leo: Yeah, well, I think the bottom line is nobody should use TrueCrypt anymore because obviously, for whatever reason - you think you could continue to use 7.1.

Steve: Absolutely. I think everyone should continue to use it.

Leo: Are there alternatives? Are there credible alternatives? That's what I'm most curious, I mean, if I've been using TrueCrypt, I'm going to decrypt the drives and go somewhere else. Remember, Steve, they said, it's the first line, that it's not secure.

Steve: That's what Microsoft said of Windows XP, and you know how I feel about that.

Leo: No, but I'm saying the guys who wrote TrueCrypt said, "Warning: TrueCrypt is not secure." So you believe them, but you don't believe that line?

Steve: No. I believe that what they're saying is that they are no longer going to fix security issues.

Leo: I wish they'd said that. I wish they'd said, hey, we're tired of this, we're not going to do it anymore, use something else, go away, don't bug me, kid. That would have been clearer.

Steve: But they actually want people off of it. That's my point.

Leo: Well, they could have said that.

Steve: They did. Look at their page. That's what they're doing, they're scaring everyone away from TrueCrypt for no reason except they want to wash their hands of it. They don't want support calls. They don't want responsibility for its future. So my point is that, eight days ago, no one had any problem with it. I don't think this changes anything.

Leo: Well, that's true. And I would - yeah, we'll have an audit; right? That'll tell you.

Steve: Yeah, exactly.

Leo: It is still, if it's not open source, at least it's open code. You can read the code and...

Steve: Yes. And when we were talking about it before, we were talking about how, for example, how difficult it is to build the thing. You need, what was it, v1.52 or something of Visual C?

Leo: Right, it's very weird that way, yeah.

Steve: You know, you need some archaic version of this, and you need - and there's other weird things, too. Like they went through and changed U.S. to United States. And I thought, well, why expand the abbreviation? I don't know. But, you know, so I did look. I

looked at really what they changed. And somebody spent a lot of time doing this, which again is the other thing that makes me believe this is the developers because you have to understand it in order even to neuter it the way they did. And that's not as simple as stealing someone's security certificate.

Leo: No, that's a good point, that's a good point.

Steve: I mean, this was a huge, huge amount of work.

Leo: I think it's fairly safe to say it must be the original developers who created 7.2.

Steve: Right.

Leo: I mean, I think that's fairly reasonable.

Steve: Right.

Leo: It's still a puzzlement. So are there alternatives out there? I mean, I know that there's a fork of it that is open source, it's BSD, FreeBSD licensed. But it's Linux only.

Steve: Yeah. And that's the problem is I don't think there's anything else which is cross-platform. There are various single-platform alternatives. And I guess they're worth looking at. My feel- see, I think this is going to survive. I think that the financing now exists, thanks to the Linux Foundation and the crypto auditing project. They're going to audit TrueCrypt, and we'll have an answer within a few months, later in the summer. They're probably going to find exactly the same level of quality, which I think is sufficient. It works. There are no known bugs. They're probably not going to find any backdoors. I don't think that's why these guys did this. They did it as a labor of love. And they finished. Then I think it'll get renamed, and it will continue to live as a cross-platform, open source drive encryption solution.

So I'm not going anywhere. I'm not looking for anything else. I'm completely happy with it through the summer until we get the result of the audit. And, you know, we would have had that either way. And essentially, this sort of creates a firewall between the original developers and a new batch. And I just hope that the new batch does as good a job as the existing guys have done. I think there will be an 8, probably. And the first thing on their list was going to be to make it support Windows 8 and the GPT and super-large hard drives.

Leo: Wow. It's a really strange story.

Steve: I just think it's...

Leo: At this point I would trust BitLocker or FileVault, the Apple solution, over TrueCrypt. And it just seems like it's risky, given that it's the first line on the page, that you shouldn't use it. It's risky to continue using it. After the audit, maybe then everybody will feel fine doing it. How hard is it to write this kind of software?

Steve: It's hard not to make a mistake. I mean, I'm learning, as I'm writing SQRL, because I am absolutely determined not to make any mistakes, you have to think about security constantly. And it is so difficult to take an attack posture with your own code. It is so much easier for somebody else to take an adversarial position and say, okay, how can I get around that? And that's what we've been doing in the GRC newsgroup, in the SQRL newsgroup, for the last six months, is looking at this. And other people have been looking at my proposed protocols and solutions and trying to come up with any way of attacking it, which has been invaluable for me.

Leo: Your skill is to do this, to write the code. So I agree with you. You need a review board of crypto experts to do this right. And to write the code you'd need to be a skilled programmer and a crypto expert, I presume. Or is the crypto pretty straightforward?

Steve: The crypto is now very straightforward. They've got all kinds of wacky stuff that's mostly legacy. For example, it used to use cipher block chaining, CBC, as its mode of encryption. Then it switched to LRW, which I think Steve Adler or somebody Adler, maybe it was Mike Adler, had rights to. So it wasn't completely free. But now they switched to XTS, which is the state-of-the-art block-style encryption for hard drives. Yet they're still supporting CBC and LRW for backwards compatibility. So that could finally all go away. And you'd also need to have kernel-level programming because you've got to have a kernel side. And so that's 32-bit and 64-bit kernel for the various platforms that support both bit sizes. And then userland side for creating drives and creating containers and decrypting folders and all that. I mean, it's a big, complex project. But certainly not insurmountable.

Leo: I think it would be a really great crowd-sourced project to, now knowing what we know - in fact, if I had known that TrueCrypt was written by anonymous developers and had never been audited, I might have been a little less likely to recommend it. Maybe do this right with a crowd-sourced project, get some very accomplished programmers, make sure they have enough money to feed and pay the rent, and then nominate a board of experts, of crypto experts to review it.

Steve: So here's what we know. We know it is solid and stable and bulletproof and bug-free. We know it doesn't crash. It doesn't trash your data. It has none of that misbehavior. We also know that law enforcement hates it because they are time and time and time and time again unable to crack TrueCrypted drives. We've covered many of those stories on this podcast. If you use a good password, nobody can get into it. What more do you want?

Leo: Well, again, it's hard for me to ignore the statement at the top of the page.

Steve: There's the evidence. That's the evidence. Who cares? This is the evidence, Leo. It is bulletproof. It doesn't crash. No one can get in. That's the definition of a perfect hard drive encryption tool. And this page didn't say that eight days ago, and the program hasn't changed.

Leo: Yeah. Okay. It's your show. You recommend it. That's fine. I just - I think when the devs whom you seem to believe say using TrueCrypt is not secure, I take that as written. I agree with you. It's robust. It's time-tested. But we don't know, we still don't really understand what happened.

Steve: You want me to read what they said again?

Leo: Yeah, I'm reading it right now. "WARNING: Using TrueCrypt is not secure as it may contain unfixed security issues." That's in red at the top of the page. What additional statement could they make that would negate that?

Steve: How about, "We worked hard on this for 10 years. Nothing lasts forever," and "There is no longer interest on our part," which they're reputed to have said. Anyway...

Leo: I know. I understand what you're saying. I do.

Steve: I have no problem with us disagreeing.

Leo: And I wish there were an alternative that we could say, with clarity, "but fortunately you don't have to keep using it." But apparently there isn't.

Steve: Well, I'm not using BitLocker. I'm not using BitLocker. My god. You don't think that Microsoft has a way of responding to a request from the government? I know that TrueCrypt was written specifically to thwart that, and we have example after example of it doing so successfully. That's all I want from a whole-drive encryption tool. And again, we'll have it audited in a few months.

Leo: Right. Okay. So I look forward to the audit. I think the audit - I suspect you're right, that the audit will come back and say, no, it's clean. But, I mean, I think you're probably right on that case. You know what, I don't use TrueCrypt, so it's not up to me. If you used TrueCrypt, I think you did it because you were of the paranoid variety, and you really don't want to take a chance.

Steve: Well, yeah. If you have a laptop that's traveling around, and you actually really have the crown jewels on the laptop, then you cannot trust the encryption built into the IDE drives themselves because we know that that can be bypassed...

Leo: We know that's not secure, yeah.

Steve: ...by the manufacturer. So your only choice is to use a third-party tool, one whose reason for being created makes sense to you. And the reason for TrueCrypt makes sense to me. So, yes, my laptops all have TrueCrypt, and I know that they're secure.

Leo: The best way to comment on this is to go to GRC.com slash what?

Steve: Feedback.

Leo: Feedback, I knew it was an F, and ask your questions there. We with any luck will be able to do a Q&A episode next week; right? Is that the plan?

Steve: Oh, yeah, I think we should, absolutely.

Leo: That's the plan, anyway.

Steve: I think so.

Leo: You can also go there and find SpinRite, the world's finest hard drive maintenance and recovery utility, and all of Steve's freebies he gives very generously on his pages of all sorts of stuff. GRC.com. 16Kb audio is there; transcriptions of each and every episode are there. The show notes now are also there. GRC.com. We have full 64Kb MP3 and even HD and SD video of the show at our website, TWiT.tv/sn. And you can also subscribe in iTunes and all the other podcast clients. Or just get the app, and that way you'll get it each and every week automatically.

Steve: And we should mention that we decided not to pursue the BT Sync or BitTorrent or whatever because it's important for the downloads to get counted because that's what Podtrac tracks, and that allows advertisers to know how many listening ears and viewers we have. And we would lose that if it was all turned into a homogenized BitTorrent Sync network. So it's really better for us if our listeners will download the individual episodes.

Leo: If you wanted to do a BT Sync of the show notes or of Elaine's transcriptions, something like that, no problem at all. We would ask that, if you download a show, it's best if you download it from the TWiT.tv website. All the episodes, every one of the 458 episodes is there. And that way they get counted, and that way we get...

Steve: Yeah, and I even have my own links at GRC are redirecting through Podtrac also. So...

Leo: Yeah. And that's, by the way, an innocent redirect. What happens there, just so you know, is it goes through the - it does go through the Podtrac servers, where they count individual IP addresses, once and only once. In fact, more than that, they

compare it against a master list of IP addresses to eliminate spoofed IP addresses, to make sure each address is unique and real. And then they use that as the count. And that's the count the advertisers use. That's what we bill advertisers with, and it's the count that we supply to advertisers and to hosts like Steve. The show is doing pretty well.

Steve: And the show generates a really good number.

Leo: Yeah, it's a conservative number compared to other podcasts who use basically download numbers. This is not that. This is a very accurate count. And you have cracked 100,000 listeners, Steve, which puts you in a very...

Steve: No kidding.

Leo: ...very rarefied space.

Steve: Yay.

Leo: And I would hate to see that number any lower. I want it to go up.

Steve: Very nice. I'm glad to know that. That's neat.

Leo: Yeah, and, I mean, I don't know how you could - I don't know what we could do to vet that redirect and so forth. Podtrac is not collecting those IP addresses. As I said, what they do is they count them.

Steve: Right. It's only for the sake of not double-counting downloads.

Leo: Right, right. No salesman will call, I promise. Hey, thanks, Steve. We'll see you. We do this show every Tuesday, 1:00 p.m. Pacific, 4:00 p.m. Eastern time, 20:00 UTC at TWiT.tv. You can watch live. If you can't get here, of course you can download it. But either way we hope you'll be back next week. Thanks, Steve.

Steve: Thanks, Leo.