Listener Feedback #186

Description: Steve and Leo discuss the week's major security events and discuss questions and comments from listeners of previous episodes. They tie up loose ends, explore a wide range of topics that are too small to fill their own episode, clarify any confusion from previous installments, and present real world 'application notes' for any of the security technologies and issues we have previously discussed.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-452.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-452-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We do have a couple of updates for iOS and for Macintosh. He'll be talking about those. They just came in over the wire. But we'll also give you a chance to ask your questions. Eight questions, eight answers. We haven't done that in a while. Steve Gibson and Security Now! are up next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 452, recorded April 22, 2014: Your questions, Steve's answers, #186.

It's time for Security Now!, the show that covers you and your security online, protects you and your loved ones. Here he is, the security Explainer in Chief himself, Mr. Steven Gibson. Hi, Steve.

Steve Gibson: Hey, Leo. Great to be with you again. So we've done, what, two podcasts largely, nominally about Heartbleed. We've got some interesting questions about that and some other stuff. And but we're going to do a Q&A. We're going to get to some questions because there have been a lot of them, and a lot of sort of similar topics. So as I always do, I chose some that were representative. And we're also going to talk about, briefly, Ladar Levison's appeal ruling, unfortunately; the fact that the backdoor, the blue box router, consumer router backdoor that we talked about around Christmas came back to bite us for Easter, that the backdoor was not as tightly closed as we had hoped and assumed.

There was an interesting piece about Google working to bring PGP-style end-to-end encryption to the masses, and I wanted to take a moment to explain why that's entirely possible, for the same reason that it's possible for LastPass to be secure, if Google chooses to do it.

Leo: Oh, I hope they do. But they won't, of course, and we know why.

Steve: Exactly, because they want to look at your email.

Leo: If they encrypt it, they can't read it.

Steve: That's a problem. And then also I want to talk about how the BSD project just forked the OpenSSL project and some of the early results from that. And also I just wanted to mention, in the Q&A, we'll finally follow up on the issue of jailbreaking iOS because that was the last little thing that even with three podcasts devoted to iOS security, I kept promising we were going to get to, and then events overcame us. So many people have continued to ask, hey, Steve, what was that about? Tell us about that. So, yes, we do that this week. So we've got a great podcast. And, oh, baby, I can't even tell you about next week's podcast. It is going to be oh, so good.

Leo: Wait a minute. Now, don't just tease us like that.

Steve: It's the revocation podcast.

Leo: Oh.

Steve: And there's a page which I tweeted which basically it is, in a single page - and it's not small because it's thorough. But if anyone wants to understand it, it's now online. And next week's podcast will be about that and additional things like Adam Langley's unfortunate Imperial Violet blog post this Saturday, where he said, no, don't enable revocation.

Leo: Oh. I really wanted to ask you about that. But that's for next week.

Steve: Yes. So that's next week. I've got a dialogue open...

Leo: You're going to keep us hanging for a week?

Steve: ...with him and another great security guy on the Chromium Project, Ryan, and a ton of interesting news. And one thing that would be interesting would be for our listeners who were Firefox users to enable revocation checking and the must check, the second option. I did talk about it briefly last week. That essentially puts Firefox into what's known as a hard fail mode where, if it cannot affirmatively verify that the certificate is still good, it won't display the page. Well, and the argument is, oh, you know, it won't work. It'll break you. Nothing happening, I mean, it's a disaster and a catastrophe. Well, I've always had it on, and I've never had a problem. So it'd be interesting to get a larger experience base from that.

I mean, it is true there are absolute known problems with that. If you're behind a so-called "captive portal," like you're in a hotel, and they make you agree to their terms of service, and if that connection is secure, and their own portal is blocking your browser from verifying the security of the certificate they're giving you, then you're in a Catch-22. And so it's certainly the case that this can fail. But so you turn that checkmark off, agree to the terms of service, and turn it back on if you want the best security available.

So anyway, prepare for next week because it's going to be a doozy and, I think, really, really good. Basically I've, in the last week since we talked about this, I've spent, unfortunately, not working on SQRL because I believe this is important. And my goal is to raise awareness because I've read Internet engineers saying, well, nobody really cares about it. But the reaction to my revoked.grc.com page was turmoil because people assumed it was working. And so it's not that people don't care, it's that they don't know. And so after next week everyone's going to know.

Leo: I'll give you another data point, and then we'll move on. Remember we were talking about the idea that you can force certificate checking on OS X, but you have to do it in the Keychain access app. And I noted that they really try to prevent you because they gray it out. This is OCSP and CRL, the two different databases of revoked certificates. And the default is "best attempt." But if you hold the option key down, you can say "require for all certificates" on both of these. This fundamentally breaks OS X. The reason it's grayed out is basically you can't use anything. The app updates stop working. The store stops working. So I've put it back to "best attempts" because, as much as I'd like to do this, it doesn't work.

Steve: Yeah. And the Chromium guys understand that. One of the reasons, well, anyway, so I don't want to preempt next week. But those terms, basically what we're going to do is we are going to walk through the two decades of history from what is a CRL, what happened with it, what were the problems; then the development of OCSP, the Online Certificate Status Protocol, what has happened with it; what's been going on since; and what is the solution? Because unfortunately Google is wrong about their position. I understand their stance, but I'll substantiate this.

Leo: Next week.

Steve: So it's going to be a good one.

Leo: Next week, not now.

Steve: Well, because we just have...

Leo: We've got so much other stuff.

Steve: ...only so many hours we can spend.

Leo: No, don't say that, because then I'm going to get all the emails from people saying, "Can you give Steve 18 hours a week because I think he really...." And, you know what, if you want to do a second show, you know who to call.

Steve: I appreciate that, but I've got to - and see, at the same time, everyone's saying, well, where's SQRL? Where's SpinRite 6.1?

Leo: I know, you've got other things to do.

Steve: What about that longest repeated string thing? It's like, oh, my god, there's just - it's just me. So...

Leo: One thing, just one more thing on Heartbleed, from the Fixer. Would SQRL have been susceptible to Heartbleed?

Steve: That's Question No. 2.

Leo: We'll get to it. Coming up.

Steve: In today's Q&A.

Leo: All right. All right, Steve. Let's get to these...

Steve: If he has your medical records, then you do hope he knows a few...

Leo: He ought to, yeah. Let's get to the news.

Steve: So okay. So the bad news is my reading of this, sort of reading between the lines, is that Ladar could have used a better attorney because the...

Leo: He saw this coming, though, by the way, when we interviewed him. He was very afraid of this.

Steve: Okay, yeah. So he did lose on appeal the contempt of court charge, which was filed against him or brought against him, after - which essentially was the FBI's reaction to his printing out his SSL key on paper in, what was it, 5-point type or something, like 11 pages of gibberish. So he technically complied with their order to turn over the keys, but didn't make it easy for them.

And so Seth Rosenblatt, who reported for CNET, sort of summed it up nicely. He said: "The appellate court didn't comment on the substantive issue in the case, whether the

government had the right to demand the encryption keys that would allow them to observe all traffic of a targeted email account. Instead, the appeals court ruled that the Internet privacy issues raised in Levison's appeal were not clearly articulated while he was defending himself in district court. The appeals court said Levison should have brought forward his claim that the government was exceeding its authority under U.S. 'pen register' and 'trap and trace' statutes before being charged with contempt of court by the district judge last summer."

And you know pen register is like one direction. That's phone numbers that call in to you. And tap and trace is the reverse. It's records of phone numbers that are called by you. And anyway, so in the judgment, Judge Steven Agee wrote - there was a three-judge appellate panel. And he said: "Levison's statement to the district court simply reflected his personal angst over complying with the pen and trap order, not his present appellate argument that questions whether the district court possessed the authority to act at all."

So, I mean, this is complex stuff, and you really need to have an on-the-ball attorney. And to me this sort of feels like this wasn't argued correctly. But I did really like - Ladar was quoted saying: "Freedom is the ability to do something that somebody else disagrees with, to make a choice that somebody else wouldn't make. The problem with disrupting our right to privacy is that, at the same time we do that, we disrupt our right to free speech. And without the ability to speak freely, a democracy is no longer a democracy."

Leo: Love it.

Steve: So, yeah.

Leo: Love it. And by the way, Ladar did a great interview with us on Triangulation, our kind of big thinkers interview show which Steve has appeared on several - many, many times. But Triangulation 125 if you want to hear his conversation. And he did talk a little bit about his concerns about how this would go in court. Now, he does have a lot of legal help, I believe, but maybe not the right. I don't know. TWiT.tv/tri125, if you want to see it.

Steve: Good. Yeah, an ACLU attorney, Brian Hauss, the ACLU filed an amicus brief on behalf of Ladar, like putting their two cents' worth in is the way that works. And Brian said: "The court focused its decision on procedural aspects of the case unrelated to the merits of Lavabit's claims. On the merits, we believe it's clear that there are limits on the government's power to coerce innocent service providers into its [the government's] surveillance activities. The government exceeded those limits when it asked Lavabit to blow up its business - and undermine the encryption technology that ensures our collective cybersecurity - [for the purpose of getting] information that Lavabit itself offered to provide." And remember that Ladar had complied...

Leo: He complied, yeah.

Steve: ...with several more limited, reasonable requests for specific individual access. What they were saying was, no, we want your master site key. And it's like, oh, my goodness. I mean, he reacted the way any responsible person would. It's like, no, that's

- you can't have that. You don't need that.

Leo: Because that would allow the government to spy on every bit of mail coming and going into his servers.

Steve: Well, I mean, it's - I mean, I would say no, too. And I would do what I had to do. You simply - that's just unreasonable. And unnecessary. I mean, it really is. You're making an assertion, an implicit assertion which you can no longer honor if that's the case. And then what are they going to say? Oh, thanks for the keys. Now you can't change yours. It's like...

Leo: Right, right. Basically they're making him spy for them. They're forcing him. It's appalling.

Steve: It's worse than that.

Leo: It's appalling.

Steve: Yeah, because if they were...

Leo: Making him lie for them.

Steve: If they were making him spy, he would at least know what he was doing. They have taken, they have subverted the privacy of everyone who uses his server.

Leo: Horrible. Horrible.

Steve: Yeah.

Leo: I can only hope that this will be further appealed. He has a legal defense fund at Lavabit.com. I encourage everybody to donate.

Steve: Is it still there? Or was it Dark Mail?

Leo: Last time I went, yeah, maybe it's Dark Mail now because last time I went to Lavabit something else happened.

Steve: Yeah, and I think his security certificate has been revoked.

Leo: Yeah, no. But he did that, no, he revoked it.

Steve: Right, because it had been - he had lost control of it because the FBI had it.

Leo: No, this links to his Lavabit Legal Defense Fund right there. And he even has a bitcoin link. Maybe I'll just give him some bitcoin. But that's at Lavabit.com.

Steve: Okay. So it must be that you can't go secure? Can you go...

Leo: No, no. And he did that on purpose, remember, so that you'd get this warning.

Steve: Yes.

Leo: And I think he even has - let's see. And presumably, if you were using Lavabit as I was, yeah, cannot connect to the real Lavabit.

Steve: And, boy, am I familiar with what that looks like now.

Leo: We've seen those a lot, haven't we lately? Yeah. At least that works.

Steve: Good for him. Well, really, I tip my hat. He's just done...

Leo: He's a freedom fighter. He is a freedom fighter.

Steve: He has been and continues to be a very important part of this conversation, which I argue we have to be having. This is a good conversation. I mean, just for the sake of asking these questions.

Leo: And you may excoriate Edward Snowden as a traitor, and there are those who do that. But there is no one who can say anything negative about Levison. He acted with absolute integrity. And he is fighting for freedom. I mean, there's just no question in my mind.

Steve: Well, and it's not even clear that there was anything there. It's not like there was a huge pot of gold that he was hiding.

Leo: No, he's just doing the right thing.

Steve: All that happened was that Snowden referred to it once as, like, a secure email

provider, which I argue is an oxymoron at this point. And maybe he had an account. But it's like they could have just said, "Give us that account," and he would have said okay. Anyway, yeah. And we've seen instances where the FBI seems - I think it's just these are very technical issues, and someone probably said to someone else, go get his keys, and that's what someone did. We've seen instances, remember, where, like, whole servers were ripped out of racks and removed, even though they hosted hundreds of other people's domains. And it's just like it's a bit of a blunt instrument in some cases.

Leo: I do refer you to the interview because he talks about that, as well. And some of the people, the agents that he spoke to were clearly clueless. Some were not. Some knew exactly what they were doing. And he's of the opinion they knew exactly what they were asking for.

Steve: So around Christmas time we covered the story that was broken by Eloi Vanderbeken of Synacktiv. That's his company. He was the guy who discovered that port 32764 was open in at least 24, we know of 24 models of routers by Netgear, Cisco, Linksys, like Cisco brand and then the Linksys Cisco, or, yeah, the Linksys Cisco and also Diamond routers. And remember that 32764 is an interesting number because exactly half of 64K is, which is really 65536, half of that is 32768. So this is four ports down from the middle of the theoretical port range.

And what we discovered, and we covered it at length around Christmas, was that there was a server open, listening for incoming TCP connections, such that anyone who found a router with this open could give it this amazing collection of commands, like dump your firmware, dump all of your settings, dump your admin password, your admin username and password. It's like the keys to the kingdom just sitting there. And so of course that caused a big brouhaha. And in January patches were available to close that.

Well, Eloi - I don't remember now the story exactly. But he was - I don't remember now, I'm confusing whether it was at Christmas or just at Easter. I think it was at Christmas, where he was at his relatives' house, and he discovered this in their router. Anyway, for whatever reason, he recently took a look at the firmware. And one of the, I mean, the arguably best thing about open source software, and I got a bunch of flak for my recent position on open source. People who are in the religion believe that I wasn't sufficiently devout. And anyway, that's another topic.

But one of the best things about it is, when you suspect a problem, it is available to go check. And which is - so there's a benefit I would never argue with. Like, for example, that's exactly what Heartbleed gave us was the guys discovered the problem, then they looked in the source code, which was available, and said [gasp], and completely understood the nature and the reach and the extent. No reverse engineering was necessary. So that's very powerful. But we also know that the fact that it's open doesn't automatically make it secure. This same event teaches us that.

But in this case what's cool about the routers, and I originally said this four months ago, is that there's now mature tools for reverse engineering the firmware. The typical manufacturer sourced firmware is not open source, it's magic. It's not DDWRT, which is a beautiful open source alternative, which at this point, especially after you hear this story, everyone should be installing because, even if it isn't perfect, it was written by people who had pure intent, which is what I believe is behind TrueCrypt and OpenSSL. I mean, these are well-meaning people. The problem is what it appeared to be four months ago is that the manufacturer deliberately put this backdoor in routers. Okay. A month later, in January, that's closed. That no longer works.

Eloi sucks out the firmware of an updated router, uses these very nice mature reverse-engineering tools, and takes a close look at it. And what he discovers is the backdoor is still there. Taking a close look at it, he finds that there's a server running, not for TCP, but for Ethernet. So what happens is, when the router is booted, a raw socket is opened on the same port, on 32764. It's not, however, bound to TCP. It's just raw, meaning that it's just going to give you - it's going to give the listening service whatever comes in. And, okay. And this is kind of confusing because I just said it wasn't bound to TCP, but I referred to a TCP port. So what it's doing is it's looking for raw Ethernet packets with an ether type of hex 8888.

Now, the ether type is a 2-byte, 16-bit piece of information in every Ethernet packet header. And remember that, like in our local area networks, we've covered all this in years past, all of this fundamental technology. So anybody who's interested who hasn't been listening from day one, this may spur your curiosity. We've got podcasts completely explaining all this. So the idea is that a TCP packet is - TCP cannot move over Ethernet without sort of an envelope, without a carrier. So Ethernet is the carrier. And so inside the Ethernet packet will be like an IP packet. And then inside the IP packet will be which type of IP packet, and it's TCP. So it's sort of this, you know, the Russian dolls, sort of stacked wrappers or enclosed wrappers.

So 8888 is unused. And, for example, 0800, that ether type, 0800, says it's an IPv4 packet like we have out on the Internet. And, for example, 0806 is ARP, the Address Resolution Protocol. So that gives you some sense, so that the ether type determines what's inside. So what we have now in these routers, and maybe even more of them, we really don't know, is an Ethernet backdoor that listens for a unique and not otherwise in use ether type - not IP, not ARP. It's its own thing.

And if the payload, if such a packet arrives at the router, and the payload contains the MD5, which is a hash, Message Digest 5, it's pretty much retired because it's had security problems, but it's useful for this. It's sort of a key. The MD5 hash of the router's commercial model number, that's the magic cookie that the packet contains. And if it's a packet type of 201, which is just another header, then the system launches the original TCP listening backdoor that has even been enhanced with some additional commands since then. There's some stuff to flash the router lights.

So what does this mean? First of all, if you had such a router, Leo, I could not send that packet to you because Ethernet doesn't cross the Internet.

Leo: It's not routable, they say.

Steve: Correct, it's not routable. The idea is Ethernet is used within a LAN, and it encapsulates the IP data. But out on the Internet it's IP traffic which is going from routers. Although the routers are typically linked with some encapsulation protocol, like it might be Ethernet, and it often is. But the point is the Ethernet gets stripped off, and then the IP packet is carried into the router. It routes around. And then a new Ethernet packet, a new Ethernet wrapper is put on if it's going to then transit across an Ethernet link.

So we call the WAN side of our routers the Ethernet, the Wide Area Network. But actually - and it is that from our perspective. But from the perspective on the other side of the router, it's the ISP's LAN. So what this does is it gives - it nicely constrains the attack surface or the attack neighborhood from being anyone on the Internet to being anyone

on the ISP's LAN, which is still not no one, and you still don't want this. But it does mean that it's not possible to scan the Internet. That's how we know, for example, that there were 6,000 routers that had this problem, is someone scanned for port 32764, and 6,000 routers said, yeah, I like connections on that port. Well, after...

Leo: Yeah, come on in.

Steve: Yeah, I'm listening.

Leo: What do you want?

Steve: Yeah. I got a whole list of commands I'm happy to respond to.

Leo: Oh, lord.

Steve: So for those routers that were updated, that got closed. But what we now know, there's no question now that this wasn't left over. I mean, even though someone deliberately designed this, it could have been argued that they forgot about it and left it in.

Leo: It was a test mode or developer mode or something.

Steve: Very much like the famous Windows metafile mess that I got myself involved in because, looking at it, it was clear to me somebody put this in there on purpose. That's all I ever said, not that it was nefarious.

Leo: Malicious, yeah.

Steve: It's just that it was there. And even Mark Russinovich agreed with me. He looked at the code, and he said, yes, this looks purposeful.

Leo: Some things are mistakes. Some things are actually purposeful.

Steve: Well, and also, Leo, the point I made was back in the beginning, I mean, Windows metafile, that's an ancient format. And that was before anyone even thought about security. I mean, it wasn't even an itch in Microsoft. So they thought, hey, wouldn't it be cool if we could put code in a metafile. Now the idea would just, I mean, it just makes you shudder to think of putting code, deliberately put code in an image format. But once upon a time that seemed like a cool hack. Anyway, so the point is that we didn't know for sure that this just didn't happen to get left in by mistake. Now there's no question.

Leo: Really.

Steve: This is an undocumented deliberate administrative backdoor into routers.

Leo: That's terrible.

Steve: Such that - I know. Such that the ISP, with no knowledge of their users, can send any router, any customer router on their network, that Ethernet packet. And that causes the system to start up the sys config manager, it's SCMGR. It starts with the -f flag, which causes it to then bind TCP protocol to that port. Then the ISP - essentially that - so this is like a knock. We've talked about port knocking. This is a type of a knock packet where the door is closed until you send a special packet, like knocking on the door, and this opens the door. So...

Leo: So this is a deliberate backdoor.

Steve: Yes. We now know without question this is a deliberate backdoor that has been - because people found the previous backdoor. And arguably that was really bad. That was an Internet-wide backdoor. This is not good because it's not clear to me that other customers on the same LAN could not arrange to attack the other customers on that network. That is, I mean, routing is complicated, and there could be VLANs and other sorts of barriers to divide an ISP up. But what we know is that this thing was discovered. It wasn't removed. They increased the security so that you have to knock on the router first with a packet that cannot come from anywhere on the Internet. It has to come from somewhere on the ISP's LAN, which is what we see as the WAN. But if it does...

Leo: It has to come from the gateway router?

Steve: No, because that would be - a gateway router is still an IP router. It's got to come from an Ethernet switch or an Ethernet injection.

Leo: So would this work with DSL? Or would you have to be on a cable, I mean, this seems like an odd setting.

Steve: No, I can see that somebody must want this. Somebody said, like, this was...

Leo: You're thinking ISP wanted this.

Steve: Yes, because it's only useful for an ISP. So this allows an ISP to, I mean, benignly to, like, help you to do tech support.

Leo: Yeah, right.

Steve: Like the new things, the new commands, flash the lights. And so the ISP can't see if the lights are flashing. So the idea is you get on tech support with your ISP, and they're trying to diagnose why you don't have connectivity.

Leo: Do you see the lights flashing?

Steve: Yes.

Leo: Sorry. I can't resist. But, no, I actually had that happen, exact thing happen.

Steve: Yes. We're going to make the lights flash.

Leo: But why is this a vulnerability, then, if it's not over TCP/IP? It's really only a vulnerability to your ISP; right?

Steve: Well, it's not...

Leo: It's a limited vulnerability.

Steve: Yes. And I wanted to make that clear. I wanted to explain the nature of this is it is only somebody on their network. But it's very likely, Leo, that my neighbor could do the same thing, and that's a problem.

Leo: Ah, because he is on your network, yeah.

Steve: Exactly. He's on the same WAN segment as I am.

Leo: See, and I think that's not a DSL thing. I think that has - sounds to me it's a cable modem. But I don't know.

Steve: It depends upon what happens at the DSL head end and the way the networking is. And you're right, I couldn't answer that definitely, either. I'm sure there are people who do know.

Leo: It strikes me that this is a cable thing. No?

Steve: Well, cable is definitely Ethernet.

Leo: Yeah. I see on my cable, I see on my segment, other people on my segment. I know that.

Steve: Yes, yes, yes, yes. And so the takeaway here is there's never been a better time to switch to Tomato or DDWRT. Switch to one of the firmware packages that your router supports that enthusiasts have developed for themselves and to share with everyone else. Perfect example of valuable open source, where even if we're not sure exactly what it is, it sure beats this thing.

Leo: But I guess we should also say that most routers are not compatible, unfortunately, with these third-party ROMs. So you're going to check and make sure yours is. And DDWRT I found out is on ASUS routers. They actually not only are compatible, but some of them actually run DDWRT. And I wish more companies would do this. Why are you [indiscernible]? They run it out of the box.

Steve: Ah, nice.

Leo: Let other folks, let other people do this because - why are you developing software? Stop it.

Steve: One of the reasons is they're trying to have extra checkboxes on their feature list, and fancy features, and oh, we're going to check for firmware in the background and update and so forth. And so they're trying to differentiate themselves. And we just want it to be a reliable piece of blue plastic.

Leo: Right. In this case blue.

Steve: So, the VentureBeat covered - they broke the story, as far as I could see, that Google is researching ways to make encryption easier to use in email. And what they're specifically - Google is apparently exploring bringing PGP to Gmail. And I'm excited because within the Gmail ecosystem this is entirely possible because now we have clients that are able to run crypto code. That's what LastPass does in order to turn all of our passwords into a pseudorandom noise blob that we're able to send to LastPass, and they keep for us, and they give us the cloud cross-device linking through that and the ability - they're also cloud backup in case our machine craps out on us, and we set up a new machine, and we're able to get all of our passwords back.

Nothing at all from a technology standpoint is preventing Google from adding true end-to-end encryption to Gmail. A user could have their client, their browser, use the platform's random number generator, which is hopefully good. Also maybe get some from Google and other places, put a whole pile of randomness together, harvest entropy to create a key which Google never knows about in the same way we do with LastPass. And that is their PGP key. And it is encrypted and stored with Google, but Google never has the decryption key because it's based on things local to the server.

And then all users of Gmail could just sort of have this happen. I mean, it could be completely transparent, in fact. It's probably, you know, they'll roll it out slowly. It'll be a

checkbox you turn on and so forth. And it's of course a problem when your email leaves the confines of Gmail because, if it continues to be PGP-encrypted, then we're back to the same problem. But for Gmail users, first of all, they could receive PGP email. I mean, for example, the classic problem that Snowden had with - I'm blanking on the reporter's name - Glenn, Glenn Greenwald.

Leo: Glenn Greenwald, yeah.

Steve: Where Glenn was like, kept resisting installing PGP, which is the only way that Snowden was willing to communicate. That could all go away. So to me this is exciting. I mean, this offers us, within the Gmail ecosystem, transparent, user-to-user, true end-to-end encryption, where Google is not holding the keys. So it doesn't have the vulnerability, for example, that iMessage does, where Apple is our key holder. And it would allow users who wanted to generate PGP security to do so painlessly in Gmail without learning or knowing anything, and then emitting their email to somebody who's a security guy outside of Gmail, who could then receive that standardized envelope, PGP envelope, and access it. Or, similarly, in the Glenn Greenwald model, to receive a PGP-encrypted email from someone who had encrypted it for them.

So this is neat, now. Of course, VentureBeat ended their story saying, unfortunately: "Don't expect Google to set up site-wide end-to-end encryption, however. For Google to monetize Gmail, it must be able to scan messages in order to serve targeting ads to users. It's an advertising business," writes VentureBeat, "after all." So that's a good point.

Leo: It's really an interesting gut check for Google. What's more important to you, ad revenue or supporting PGP? We had Vint Cerf on again yesterday, and he reiterated that the only change he'd make maybe in his invention of TCP/IP is, at the time that they did this, in the early '70s, public key crypto was known among the spy industry and was known privately. It was not publicly known. And had they known about public-key crypto and had they had access to things like RSA, they would have used it. They were using DES with symmetric encryption, which as we now know is not enough. And he said, "We would have put in strong encryption in TCP. We would have made that possible, a possibility." And he works at Google now, and I'm sure he's lobbying to do this. What about Hushmail? That is a PGP service. It's webmail using PGP. You know about Hushmail?

Steve: I haven't looked at it. I know of it.

Leo: Yeah. And the way Hushmail works is you pay for it. It's 50 bucks a year.

Steve: Right.

Leo: And it's strong encryption. It's PGP.

Steve: Do you ever see Google going for any sort of a pay-for model?

Leo: They'd have to because obviously they're not going to monetize - I think this would be a way for Google, which makes plenty of money, to win some goodwill, to say you could check a box here that turns on PGP. We encourage you to do so, even though we won't monetize. Maybe what you say is, and we'll charge you five bucks a year for that.

Steve: And it's not like it can be transparent because, if you turn that on, and then you send that email to your mom, she's Glenn Greenwald all over again. It's like...

Leo: Well, that's the problem.

Steve: Honey, I just got a large blob of noise. What is this? It's like, oh, sorry, Mom, I forgot to turn that off for you.

Leo: Phil Zimmermann started Hushmail, the creator of PGP.

Steve: Oh, yeah. In fact, I've got a great quote from him later on about the OpenSSL stuff.

Leo: And I think that's what Ladar and Phil are trying to do with Dark Mail.

Steve: Yes. And it is a hard problem because we're trying to add this afterwards, and you've got the problem of people who aren't up to crypto.

Leo: I've used Hushmail, and I have a Hushmail account. I like Hushmail. But it's not easy because, if you want end-to-end encryption, you've got to get the other person to use Hushmail.

Steve: It's got to be both ends, by definition, yeah. And about Gmail, it's because it is Gmail. I mean, it's not like it's some small also-ran email provider. It's Google. It's like, oh, my goodness. If they were to do end-to-end encryption, it would change the world, yeah.

Leo: Yeah. Everybody would use it. So maybe this is a way Google could kind of get some goodwill and give up a little revenue.

Steve: And also it could be way from deployment. It could be that VentureBeat picked up on an internal, I mean, Google has running many internal projects. Some, and I know you and Gina have talked about it, they kill after a while because they never get off the ground. They go, oh, well, you know, we were just checking that out.

Leo: Tons of that.

Steve: Yeah. So also in the news is the fact, or the news, that the OpenBSD project has forked the OpenSSL project to - and be careful how you pronounce that word "forked" - OpenSSL to create LibreSSL.

Leo: Oh, how funny.

Steve: Uh-huh.

Leo: Oh, how funny. Because, you know, the same thing happened with OpenOffice and LibreOffice. "Libre" is the word for open source in most countries, not the U.S.

Steve: So if you click that link, that LibreSSL.org link, Leo, because their site is a little bit - is a kick. Oh, and here we have the trash being taken out. You want to take a break?

Leo: I guess so. It isn't much of a website. In fact, it looks like it's a heavy reliance on Comic Sans. Doesn't really give you some confidence here. Is this a joke? This must be...

Steve: No, no, that's - no. It says: "At the moment we're too busy deleting and rewriting code to make a decent web page. No, we don't want help making web pages, thank you." Okay. So also the other one is the OpenSSLRampage.org, that second link there. And this is the "OpenSSL Valhalla Rampage." And they're calling it "The Purge."

Leo: And they're using the Heartbleed logo, I note, on the page. It's a [indiscernible].

Steve: Oh, yeah. So, okay. So what is this about? This is, in all seriousness, this is interesting; but it does terrify me because, despite all of its warts, OpenSSL is amazing for all that it contains - and those little posts there, Leo, are fabulous - for all that it contains. And it's time tested. Yes, we just all went through a huge upheaval because one line of code was left out to check the bounds on a buffer. But that's a mistake. And that can happen. So here's what the OpenSSL guys are doing. I mean, I'm sorry, the OpenBSD guys. Having split off, forked the OpenSSL project, they are now going through and literally, as I said, they call it "The Purge," hacking and hewing. As of, I mean, like almost immediately, 90,000 lines of C code have been removed, reducing the overall line count by 150,000 lines because they've also done some reformatting of where the curly braces are, as we were talking about C formatting.

Their point is that there is so much stuff that is just not needed in OpenSSL. So, for example, OpenSSL supports, of all things, VMS, which is the old DEC operating system, which is now owned by HP. And I thought, it does? And I went over and looked at HP, and there they're talking about, oh, yeah, how VMS has OpenSSL, and it's got 0.9.8

something or other, one that was, like, current. And so that's something that apparently VMS is still in use, and OpenSSL is the way they get their security.

So these guys were saying 99.99% of the community does not care about OpenSSL's support for VMS. And it's hard to argue with that. 98%, they say, do not care about Windows. What they care about is POSIX support so that it can be used with UNIX and UNIX derivatives, all of the *.NIX machines. They don't care about FIPS compliance because of course that's a mixed blessing, as we know. And so these guys were saying, even with all of that, the code base is, I mean, this has been done already in LibreSSL, and the code base is still API compatible.

Leo: Unless you're VMS or, you know.

Steve: Oh, no, no. Absolutely. So Windows will no longer...

Leo: Or Windows, yeah.

Steve: LibreSSL will not offer Windows compatibility, nor VMS compatibility. But the point is it will be, apparently, vastly smaller. And their goal is to go through it and root out all of the stuff, the cruft which it has sort of accumulated as features were added that, well, yeah, okay, I could see that, on a full moon, when that is also leap year, that that might come in handy. But, gee, is it worth having that in everybody's copy of OpenSSL? Eh, probably not. So, I mean, OpenSSL even has its own "printf" implementation.

Leo: Really.

Steve: And one of their - yes. One of their comments is "Pretty much no one needs to write their own printf implementation." Because they can just obviously bind to the C runtime that's got printf. But OpenSSL has one, in case you didn't have one around or available. So that gives you some sense for it. So the reason I consider it a mixed blessing, as I said before, is that it's like, oh, boy, hacking and hewing in there, it's very easy to make a mistake, very easy to introduce some subtle incompatibility. So it's like, yikes. I guess my recommendation is, great that this has happened. Now give it 10 years. Because, boy.

And I thought I had a quote, I don't see it here now, I ran across - it might be somewhere else, a really clever - I think it was in the Q&A, a Phil Zimmermann - yes, it is in the Q&A, a Phil Zimmermann quote about this notion. So we'll get that when we get to it.

There is a very important piece of security news relative to updates. Everyone listening to this, if you haven't already, who is running a Mac with OS X needs to update now. Apple released this morning, and such a short time before the podcast I couldn't get them into the show notes, so I quickly generated, after the show notes were already formatted and PDFed and emailed to everyone and posted on the site and everything, I created another file that goes through what happened.

So we normally don't do this with Apple because we normally don't have access to this.

This was actually posted on GitHub by a guy, Frederick - I hope I got his name right, I don't have it here - who is at Whisper Systems, who posted this, who found this and posted it to detail what's going on. It affects iOS to a much lesser degree. So iOS doesn't, you know, there's an iOS update and a Mac OS X update. And I'll note, I think we were talking about this before we hit record, Leo, on the podcast, that it does restart your iOS device; and you'll find Bluetooth reenabled, as always after one of these updates. So turn Bluetooth off if you don't need it. That's standard advice because having Bluetooth on opens an RF attack surface for your device you'd just rather not have. We don't know of any problems with Bluetooth's stack until we know about them. So...

Leo: Apple points out they use it for location granularity.

Steve: Right, right. You get better location. And I, you know, it's like, uh, okay.

Leo: They did the same thing on the WiFi. They don't want you to turn off WiFi.

Steve: See, but WiFi I can understand. The only way I can see Bluetooth granularity would be if you're near beacons at an Apple store. Or, I mean, the point is you need to be with Bluetooth range of something with a known Bluetooth location, like other Bluetooth people, maybe. Anyway, I just - it gives me the creeps to have it on as a security guy, except - because I have to have it on for my beloved Typo keyboard on my phone. But it's off on all my pads because I just don't need it.

So, okay. So what happened? Both for Mac OS X and iOS, for Lion, Lion Server, Mountain Lion, and Mavericks, so this one goes back a ways, there is an attacker in a privileged network position can obtain website credentials. Now, this is really interesting. I mean, I don't know that this doesn't affect everyone everywhere on everything. I mean everything. I mean Windows and Android and BlackBerry and everything. Because get a load of this one.

The description is: "The set cookie HTTP header could be processed, even if the connection is closed, before the header line was complete. An attacker could strip security settings from the cookie by forcing the connection to close before the security settings were sent and then obtain the value of the unprotected cookie. This issue was addressed by ignoring incomplete HTTP header lines." Now, okay. So I read this in the late morning, and I thought, what? And, okay, I've written a lot of set cookie headers in my time. And reverse engineering this, it's like, okay.

So a set cookie header, the normal format is the phrase "set hyphen cookie colon space." Then you have a "name equals value pair," that is, the name of the cookie equals the contents of the cookie. And you can do that a few times to set whatever cookies you want. Then you have an optional term which specifies that this can only be delivered over HTTPS. And I don't know if it has to be last, but by convention it is last. And so what these guys realized was - and I regard this as theoretical because I don't even know if you could actually do this because it would have to be that the first part of the header line ended the packet it was in, and the security phrase which is normally tacked on the end was in the next packet, because that's the only way you could sever the connection in between packets. You can't sever the connection in a packet because then the packet's not going to validate.

So it's like, okay. That's why I earlier tweeted, it's like, hey, folks, update iOS when you

get around to it, but don't be in a panic about this one. So interesting theoretical problem, but seems to me really obscure. But at the same time I don't - so in the case of Apple's systems, both iOS and Mac OS X, they would allow this incomplete header to be processed, even though it didn't have its normal carriage return line feed ending at the end, which is what they're now doing. This is what the change is, is they would have to - they wait till they get that formal end of line carriage return linefeed to say that we received the whole header.

They were processing a partial header, and someone figured out that, if you could snip it right at the exact right character, which has to be on a packet boundary, I mean, which also has to be on a packet boundary, then you could get the cookie which may well contain, for example, your session, it might be your session cookie, which you definitely want to keep secret. It could make it nonprotected, not protected by SSL. Then you would connect again. Then you would spoof the server, and the browser that had stored the cookie unprotected would give it to you over HTTP. Again, it's a convoluted exploit.

Leo: It's kind of like Firesheep, though; right? Once you've got the session cookie?

Steve: Yeah. But again, Leo, you've got to, I mean, talk about standing on one foot, touching your nose, jumping up and trying to click your heels together three times.

Leo: This is the kind of security flaw we like. If there are any good security flaws, this is that.

Steve: Yeah. Like I said, yes, update iOS. But don't, like, rush home to do it. You'll be okay. Okay. That was one of a number. There is a problem, not in iOS, only in OS X, in the core services UI agent. And this sounds potentially more problematical. Which is why I said absolutely do update OS X. I've updated all of my Macs immediately. I mean, they were turned off, but I turned them on in order to update them. "Visiting a maliciously crafted website or URL may result in an unexpected application termination or arbitrary code execution." That's bad.

So they describe it as: "A format string issue existed in the handling of URLs." And this is, like, bad. "This issue was addressed through additional validation of URLs. This issue does not affect systems prior to OS X Mavericks." So they did something that broke URL validation in a way that allows a malformed URL to potentially perform an arbitrary code execution. That's bad. That means links on web pages, links in email, the link itself could be malicious. So, ouch. Patch. Patch that, everybody.

In the font parser, again, another potential problem. "Opening a maliciously crafted PDF file may result in an unexpected application termination or arbitrary code execution." And then Apple says that "A buffer overflow existed in the handling of fonts in PDF files." And we already know from all of the coverage of this in Windows, those are not good. "This issue was addressed through additional bounds checking. This issue does not affect OS X Mavericks systems," so previous. So this was something they fixed at OS X which does affect earlier ones. And then one that is new for Mavericks, and this is bad also.

This is in the Image I/O library: "Viewing a maliciously crafted JPEG image may lead to an unexpected application termination or arbitrary code execution." They said: "A buffer overflow issue existed in Image I/O's handling of JPEG images. This issue was addressed in this patch through improved bounds checking. This issue does not affect systems prior

to OS X Mavericks." So again, a new problem introduced in Mavericks. And there's a problem with the Intel - anyway, I think everybody's got the idea. Update. This is important for OS X.

Intel graphics driver problem. There are a number of these that are local, that are for that reason much less concerned than, like, clicking on a link on a web page, and you're pwned. So in the graphics driver a malicious application can take control of the system. So that's not good, but not a remote cross Internet problem. I/O Kit kernel. Also in iOS there was a problem where a local user could read kernel pointers which allowed a defeat for Address Space Layout Randomization. And this actually is a perfect instance of the kind of bug that could be leveraged into a jailbreak. And we'll be talking about jailbreaking in today's Q&A. So again, that's something you want to fix. And so on. So anyway, definitely worth doing.

WebKit had 16 things found and fixed for iOS. So, and they're potentially exploitable. So, again, update your Apple devices. And I thank Frederick for this great information. We normally don't get this from Apple. So it's interesting to have that for a change.

Oh, and our friend Simon Zerafa tweeted something that just caught me at the right moment, and so I put it under "Security Funnies." He tweeted: "Due to a security leak, your biometric data may have been compromised. We recommend you change your fingerprints as soon as possible." So, yeah.

And I did have just a short tweet from one of our listeners, and a follower, who actually sent it to @GibsonResearch. It said, and this was at 5:31 in the morning, so early, I don't know if that's my time or his time, on the 19th. He said: "Just recovered a 500GB drive. After 28 hours working that out, SpinRite brought it back to life. Another happy customer." And his name is "mar," and @mtropyancheno is his tweet handle. So thank you for tweeting that. I appreciate that.

Leo: I like his handle.

Steve: And sharing it with our - yeah, it's neat - with our listeners.

Leo: Leo Laporte, Steve Gibson. Eight questions which, you know what, now in hindsight that seems like a right number.

Steve: I had a feeling.

Leo: Question No. 1 from Casey Elisha...

Steve: And you'll love them, Leo. A lot of them are short.

Leo: Short's good. They can be tweets, that kind of thing. Casey Elisha Bailey in Bellevue, now, this says Bellevue, Nebraska, via Twitter: Steve, wasn't there going to be some quick talk over jailbreaking at the end of the iOS Security episodes that time ended up running out upon?

Steve: Yes, indeed. So, okay. In fact, as I was listening to your commercial with one ear, I was thinking, you know, my comment about the patch that just was issued where a local application was able to obtain kernel pointers, and even then I said, well, that's kind of the way a jailbreak would work. I wouldn't be the least bit...

Leo: This is to block jailbreaks, yeah.

Steve: Yes. I wouldn't be surprised if, no, I mean, if that update blocked an effective jailbreak.

Leo: They do that all the time.

Steve: Yes. And so that's a perfect example. So in the three weeks that we covered iOS security, we basically looked at this soup-to-nuts, amazingly aggressive security architecture which Apple has put in place where, from the moment the power turns on and the boot kernel checks the signature of the code in memory before it loads it, and then it loads it and runs it, and it checks the code of the next thing, so you get this chain of verification, replete with all the other checks that we talked about. This is specifically to prevent things like a jailbreak, to prevent applications from doing anything wrong.

So the question is, how then can they? And it was - and I'd never really focused on jailbreaking. It just wasn't something I had looked at before, and we hadn't had any - I had had no need to go dig into it. But reading these documents, I found myself thinking, well, how do you jailbreak in that case? And the answer is exactly what we would expect. It's one thing to have a bulletproof, perfectly worked-out design. But you also have to have a bulletproof, perfectly worked-out implementation.

And we know how, I mean, OpenSSL is an example. And in TLS we're still finding little edge cases where TLS, it's like something we didn't think about in the protocol. So it's possible that the design can be wrong independent of the implementation being right; or the design can be solid, and the implementation can be wrong. So jailbreaks leverage mistakes, even tiny mistakes, infinitesimal mistakes, little chinks in the implementation of the architecture that we covered which looks absolutely bulletproof.

And a perfect example is the one we read where a local application manages through some oversight, either in design, which is still possible, or in implementation, to get some information that allows it to bypass the mitigations that Apple has put in place. And thus a jailbreak. It gets some foothold and is then able to, like, for example, if there was any way that an application could get enough privilege to change one byte, like one byte in storage, for example, the byte that verifies the signature. Somewhere there's a byte that is doing ultimately, after all this fancy amazing crypto, it comes down to a conditional branch, equal or not equal. And all you have to do is change a bit in order to change that from an equal to a non-equal. And in doing that, in changing that bit, you have broken the signature. But now you've also changed the test to make sure it's broken, rather than to make sure it's not broken.

So that gives you a sense. If there's any way to change what Apple intends, that's all it takes. And so it's been a cat-and-mouse game. The major work that Apple did when they released v6 was primarily focused on security improvements to prevent jailbreaking. And the people in the community who were either watching the jailbreakers or enjoyed the challenge of trying to jailbreak iOS and iPhones, ended up concluding that iOS 6, the

mitigations in place significantly raised the bar on what it took to jailbreak, and that many of the old tricks didn't work. And many of the bugs which were once reliably exploitable just no longer are.

So basically that finishes our coverage of iOS and jailbreaking. And I like my little example of just one bit flips the sense of the jump so that it verifies the signature. Now it verifies the non-signature, which you also get because of the change of the bit. And then, I should say, then you go make other changes that you want to in the code to give you the access that you want. That's why you then have to use or reboot the phone; and, oh, look, it comes up broken.

Leo: Somebody's saying that's what you do to break, or used to do to break copy protection on disks. You'd use a hex editor. You'd look for the copy protection code, and you'd just jump.

Steve: Yeah, exactly.

Leo: You just say "jump around that."

Steve: There was a really, really good friend of mine who had a dongle-protected vertical application program. It was for, like, embroidery graphics or something, I mean, really obscure. The dongle literally got fried. I saw it. It was like charcoal. I don't know what happened, but something catastrophic. And the company was gone. They were out of business. And he was actually one of my very best friends. And he said, "Steve, we bought this. Here's what's left of the dongle. But we can't get it replaced. The company is gone."

And so I rolled up my sleeves - those were back in the DOS days with SoftICE - and stepped through and found the test and just removed the branch, and then it worked. I mean, that's the problem is that all software has an Achilles heel like that. And this is why Apple's challenge is so great is they have to be perfect. And they're getting much closer to it.

Leo: Somebody's saying also in the chatroom that no one has jailbroken to date the current Apple TV, which is also an iOS device. And I for a long time used jailbreaks on Apple TV to add capabilities to it, FireCore. And I guess you can't do it anymore. I didn't know that.

Steve: Well, remember, it's a function both of how hard the problem is and how great the demand is.

Leo: Yes.

Steve: So if we take the notion of this stuff all to some degree being porous, then how much pressure are you putting against the porosity in order to force your code in?

Leo: Yeah, lots of incentive to break the iPhone. Maybe a little less on the Apple TV, yeah.

Steve: Correct.

Leo: Twitter from Joe, @The_News_Now. And this is the one you were referring to: @SGgrc If SQRL were in active use, would it be vulnerable to the Heartbleed flaw?

Steve: You want a one-word answer?

Leo: No. Yes. No.

Steve: The answer is no. And really quickly, the reason is, and this is one of the very cool things about SQRL, if I don't say so myself, is that SQRL doesn't give a web server any secrets to keep. That's my favorite way of expressing it. SQRL gives web servers no secrets to keep. Normal login, there's a secret. We call it your password. And we're wanting the website to keep your password secret. And because traditionally and historically and famously they can't, that's the problem. SQRL doesn't give them a secret. SQRL gives them an ID. And then, when you come back and allege and assert that ID, the website can test that that ID is true, that you actually are that user. And because your ID changes for every site you go to, and everyone has a different ID, it's just noise. So Heartbleed would have no effect on SQRL whatsoever. Websites could publish the SQRL credentials of all their users. It would tell them nothing.

Leo: Interesting.

Steve: Because they're completely anonymous, and they are of no value to any other site. Only that particular domain. And they just represent a user. But they provide no other information.

Leo: That's awesome.

Steve: I know.

Leo: That's the point, Steve.

Steve: I'll be back to it shortly.

Leo: SQRL, for those who - I don't know how you could not know. But for those who don't know, it's Steve's unique approach to website login.

Steve: Coming soon to one of 53 languages near you.

Leo: Yeah. Yeah. So here's Dave Collins, following up, similar question: Does Heartbleed rely on SSL over UDP to exploit? Will only allowing TCP 443 through my firewall prevent an attack?

Steve: And I think I'm responsible for this confusion, Dave. I apologize for that. Because remember I introduced the Heartbleed conversation by talking about where the "heartbeat" came from, and it was the addition of a TLS heartbeat, which is where the Heartbleed name arose. And it was a mistake in the heartbeat code which is in SSL and which - and in the latest version of SSL/TLS you can transport that over either UDP or TCP. So unfortunately the answer is no. Heartbleed does not rely on SSL over UDP. It's probably equally vulnerable there, but it's definitely vulnerable over TCP. So when you say, "Will only allowing TCP 443 through my firewall prevent the attack," remember that it's not even letting it through your firewall that's a problem. It's what it arrives at.

One thing we never discussed was we talked about the servers, the server-side vulnerability, that is, of a malicious client probing the server, sucking out all of the server's private information, or a great quantity of it. We never talked about the client-side vulnerabilities, and they're symmetrical. That is, if you have a client, which is to say an appliance or any software using that vulnerable version of SSL, first of all, you might be running a server yourself, in which case you're vulnerable just like the Internet servers were. But you also might be running a client who reaches out over SSL to remote servers. So if your client is doing that over TLS connections and is using that vulnerable range of OpenSSL, and you connected to a malicious server, it could reverse probe you, and you wouldn't know it. It could use Heartbleed on you, the client, in order to extract 64K of stuff from your client machine. So remember, this is only TCP.

And so in Dave's case, allowing it through his firewall, it's not the packets passing by that's the problem, it's where they go. So if they went to a server with OpenSSL in that vulnerable range, yes, that's a problem. If you had it open because you wanted client activity, which normally your firewall will open that for you, or the router will dynamically, that's not going to be a problem. So it's important to note that clients with that range of OpenSSL would be subject to attack by a malicious server that they reached out and connected with.

Leo: All right. Moving along to Question 4. And this is a good one, actually. I'd like to know because we've talked about this with you and Bruce Schneier, that some routers are vulnerable to Heartbleed. Our questioner, KarKar the Marklar, tweets: Steve, is there a way to test my home router for an OpenSSL/Heartbleed vulnerability?

Steve: And I have such good news. I didn't realize a very good friend of mine had written this until I tweeted it.

Leo: You know KarKar the Marklar?

Steve: No, no. I know Robin Keir. Robin was at McAfee for while. He was at Foundstone before that. And at Foundstone - and I think McAfee bought Foundstone, I'm pretty sure

- when he was at Foundstone he was producing all kinds of cool utilities. So it's funny because he's now at CrowdStrike.com, and I remembered that he had left - he used to actually be in Southern California, and he left and moved East. So I got word from Twitter. Someone sent me this news. I went to CrowdStrike to get it. So, okay, the link is - I created a bit.ly link for this episode, for this tool. Windows, by the way, although it does run in Wine, so Mac users who have Wine set up and Linux users can also use it. And by the way, the SQRL app is going to be the same way. It'll be definitely, I mean, assured to be Wine compatible. So bit.ly/sn-452, which is today's episode number. So bit.ly/sn-452. That will expand to the link, to the page where you can download this.

So my experience was special because I went there, looked at it, kind of looked okay. I was being careful. I download this thing, and it was like Firefox says you're done. It's like, what? It's like, no. I mean, it's like downloading SpinRite, and it's not something I was used to. So I run it, and there's no setup, no install, it just sort of asks me to agree to the license agreement, and there it is. And it's like, what? Where did this come from? And it's a couple hundred K, I think. I don't remember. It's small. Anyway, so I was just stunned.

Leo: It's Windows only, though, I should point out.

Steve: It is Windows only, both 32- and 64-bit. And but that's my point, was why I mentioned it runs under Wine, is Linux and Mac users can run it. And there's even a - there's a way to bind Wine into the app. I saw, it passed by, and I didn't - yeah. So you're able to...

Leo: But then it wouldn't be 200K.

Steve: No, it would be monstrous.

Leo: You're putting the Windows APIs in there.

Steve: I can vouch a thousand percent for this. This is a scanner, a self-contained scanner for the Heartbleed vulnerability. And if it's from Robin, it works, and it's done right, and it's good. And then when he tweeted, he says, yeah, that's mine. I said, oh, my goodness. Well, now I understand why it's a nice piece of work.

Leo: This is good to know because we had a caller on the radio show who wanted to know if his NAS was susceptible to Heartbleed.

Steve: Yes.

Leo: And you need a way to test it, basically, if the company doesn't mention it. But he could run this, right, and it would show if his NAS was susceptible.

Steve: Yes. What you could do is, most people use a 192.168.0.* or .1.*. You can put in

the range and just turn it loose, and it will scan the entire - that's only 256 IPs, or 253, technically. So it would scan the range, and it will show you no connect, no connect, no connect, no connect, and then when it's able to connect, and then whether or not there's a vulnerability there. Or you could just point it directly at the IP of your NAS, if you knew what it was, and it would just instantly tell you.

Leo: This is such a good thing. Thank you. I will pass that along to our listener. CrowdStrike Heartbleed Scanner. Just Google for that, and you'll find it.

Steve: Perfect.

Leo: Yeah. What a good tip. Oh, let's do some more.

Steve: And thank you, Robin.

Leo: Robin Keir. Lance Reichert, itinerant engineer - have router, will travel - he's home to roost at last. And his subject is: Hold off, hold off on fixing Heartbleed. Some suggest being careful in rolling out fixes to security flaws, echoing Phil Zimmermann's warning: "Treat all new crypto as you would new pharmaceuticals"; or the stronger maxim, "All new crypto is snake oil until proven otherwise."

Steve: And so anyway, that was what I was referring to. I knew it was here somewhere. I just got a big kick out of Phil Zimmermann's very apt analogy, I think: Treat all new crypto as you would new pharmaceuticals. There's so many ways in which that's right because everybody who listens to the podcast knows how I feel about anything new is it's just, oh, let somebody else get the arrows in their back. I'll wait till we're two more OS versions, major versions ahead, and then I'll cautiously move forward.

Leo: Now, we can presume that the OpenSSL fix is good; right?

Steve: Absolutely. I mean, I wanted to share what Lance shared. But in this case, it's not like this was a mystery, or anyone was scratching their head, or we're not really sure, but we shook it three times, and now the problem went away. No. We know exactly what tiny mistake was made that resulted in a huge event. And there's been some really interesting discussion which is sort of off-topic for us, but the notion of the monoculture, the fact that OpenSSL is so hugely used that a tiny mistake that went unseen for two years could shake up the Internet to the degree that it has. And it's because of the monoculture of, in this case, security. I mean, OpenSSL is the standard library. But in this case a tiny fix was all that was necessary. We don't know that there aren't other problems, but we know that absolutely Heartbleed and this problem is closed. No question.

Leo: But if you're using something like LibreSSL...

Steve: Ah, and I wouldn't. You couldn't make me. No. Nothing could make me use that.

Leo: Because that's a full reimplementation of the library.

Steve: Well, it's a hack job. I mean, they're, like...

Leo: Or even worse.

Steve: ...oh, we've removed 90,000 lines of code. Well, I hope that you didn't remove any important ones. I hope you checked real carefully.

Leo: So stick with the new OpenSSL 1.0.1g.

Steve: Yeah, yeah.

Leo: Use that library. And then we'll watch with interest.

Steve: Somebody Libre will have proven itself. But again, let it prove itself on someone one's network.

Leo: Question 6 comes from Nick Donnelly, who is back in London from Saigon. He says: Shredding is a weak cipher. We talked about shredding. He said: I know you're paranoid, but are you paranoid enough? For those of you who have seen the Nicholas Cage movie "Lord of War," which was a fun movie, he plays a...

Steve: Yeah, an arms dealer.

Leo: Yeah, it's good.

Steve: It's a great movie.

Leo: Shredded paper, given enough time and determination, can be pieced back together by an attacker. The equivalent, perhaps, of using a weak cipher? Shouldn't you be burning it, Steve? What if those garbage men are from the NSA after all? Love the show. Can't wait to try SQRL in the wild, pun intended.

Steve: So, okay. We mentioned this, I don't remember if it was on Security Now! or Triangulation.

Leo: Last week, I think. No, it was, well, I don't know either. But I think it was last week.

Steve: I think it was last week. And you asked me do I have a shredder, and I do. And I mentioned it's a confetti shredder.

Leo: And you said cross-cut.

Steve: Cross-cut, yes. And so I just thought, okay. Well, actually what I wanted to talk about was something else that this put me in mind of. But the documents that I'm shredding are not important, or I wouldn't be throwing them away. I would be archiving them or doing something, maybe using them to start fires, as Nick suggests. I mean, it's just like a snapshot that I get every month of accounts and things. But, I mean, no confidential information. But I've said it before, and I will say it again. I have installed firewalls between my and GRC's and the various types of accounts both I and GRC have. And it annoys my bookkeeper because she is forced to write checks in order to move funds. I will not allow electronic funds transfer between accounts where it's not something that, well, period. In some cases you just have to, like between our merchant account and our main operating account. We've got to be able to move funds there. But accounts that are staging accounts and investment accounts and so forth, they are firewalled.

Which it's funny how much, like the bank goes, what? Now, why don't you want that? That's unusual. It's like, yes. It's because it's not safe. And I want the bank to be responsible, not us, if somehow some fraudulent electronic funds transfer tries to happen. And if they let it through, it's their responsibility, not ours. So I just wanted to say to people, I've mentioned this before, but there are banking trojans in people's machines. People are losing all their money, for which there's no recourse. So although it's not as convenient, if you're interested in security, I would shut down electronic funds transfer between accounts where you really don't have an ongoing business need to have it. It defaults on. It's one of those bad settings in banking. It's on unless you insist on turning it off.

Leo: Hmm. I have it all over the place.

Steve: I know. I mean, it's just, again, everyone does.

Leo: Well, I use it all the time. I mean, I guess I could go to the bank.

Steve: Well, so I would consider that a business purpose. Fortunately, I have Sue, and I can make her write checks. So she grumbles, but she writes checks.

Leo: Like I transfer money into my kids' accounts so that they have money. I don't want to - I need to do that.

Steve: So I guess my point is the thing to do would be to not have all your money in the account from which you're transferring money from.

Leo: Oh, yeah, so a smaller pool to steal from, yeah. That makes sense.

Steve: Precisely. Precisely.

Leo: Is there widespread - and aren't they liable, if there's an automatic funds transfer out of my account?

Steve: No.

Leo: They're not.

Steve: No. And there's no recourse. People lose tens of thousands of dollars, irretrievably.

Leo: I guess I'd better - you can turn it off, huh?

Steve: Yes.

Leo: Guess I'd better turn it off.

Steve: Yeah.

Leo: Hey, you didn't hear this. I don't have EFT. These are not the droids you're looking for. Question 7 comes to us from Murray Court in Edinburgh. He offers an ITPro.TV testimonial. Now, we should mention this is a sponsor, ITPro.TV. They're great guys. Tim and Don were in here last week. And what they do is they do training, kind of cert training, but they do it kind of in the style that we do with a live stream and video and stuff. And so it's really fun. Anyway, he says he's a fan of the show. He's been listening for about two years, and he subscribed for about six months to ITPro.TV.

He says: I don't have an IT background, but I started listening to your show in order to learn about crypto. So when I heard the ad for ITPro.TV on the show, it seemed like too good a deal to pass over. And he started working towards his CompTIA Security+ certification. I am now CompTIA A+ and Network+ certified, and I plan on taking the Security+ test next month. That's awesome. Good man. What's his name? Well done, Murray. I'm happy that I can now better enjoy your show as I now understand the terminology you guys frequently use, like NAT and IPv4 and PPTP, et cetera. Furthermore, I understand the underlying concepts.

I realize you guys have probably defined all of this terminology in your years on the air together; however, the guys at ITPro.TV have condensed the basics of IT and networking into an enjoyable framework, with the added benefit of preparing you for

professional certification. I would recommend ITPro.TV to anyone who enjoys Security Now!, but has the occasional difficulty keeping the propeller hat on and keeping up with Steve. Feel free to read this on air. Well, they're not on. They actually don't have an ad this week. But, hey, there you go. I think that's great. I appreciate the testimonial.

Steve: Well, and I ran across this, unsolicited, in the mailbag as I was going through to build our Q&A for the day. And I thought, yeah, I mean, it was so well written and such a great first-hand account, I thought, yeah, let's share that.

Leo: It's a nice story. They now have the, what is it, IST2? There's a new certificate, even more high-end security certificate, that they also are doing. And it's nice, it's easy to watch, and you learn it kind of almost - just the same way you learn from listening to the show.

Steve: Yeah, and he is right. We've covered all of those terminologies in excruciating depth in the past. And they're available. You can go find them. But you can also get, I guess, maybe a condensed version, and a little more targeted. Targeted toward more what you have to know rather than...

Leo: It's actually breaking it down by the questions and the chapters and stuff, so you can get exactly, yeah, you're learning that stuff. Teaching to the test is not always a bad thing.

Tom Behrens, our last question, comes from Long Island, New York. He wonders about Chrome's certificate revocation checkbox. Steve, you said, I believe, in the last Security Now! we should be clicking on the box to allow Chrome to check for revocation of certificates. I'm reading articles that say differently, and they're quoting Google directly. I'd love to hear your input on this since I have checked the said box. Tom B., Long Island. As have I. We showed you how to do it on the show.

Steve: Yep. And I didn't realize when I - this was going to be my lead-in question for next week, and we ended up I got into it a little bit more than I intended to at the beginning of the show. So I won't drag everyone through that again. Just to say that, by the end of next week, everyone who listens will absolutely understand the whole truth and nothing but the truth, I mean, the absolute story on revocation. Because I have learned so much in the last week from the consequences of the revoked.grc.com site and trying to understand, for example, today Chrome cannot block it on Android. Absolutely can't. But it is on iOS, but nobody else is on iOS. And Firefox is the only thing that works on Android. I will explain why, next week, why Firefox is, in terms of certificate revocation, the only secure browser that exists today, and how we change that for everybody else.

Leo: Now somebody posted in the chatroom, oh, Steve, revoked.grc.com's broken. No, this is what's supposed to happen. It's not broken. You're supposed to get a warning that it can't connect. That means your browser's smart and is recognizing it; right?

Steve: Right. But if you try that with Chrome on Android, you will see the page which you should not see. If you try that with Safari on iOS, you will see the page, which you should not see. So revocation - and, see, Google's complaint is that it's not perfect. And nothing that Adam has said, Adam Langley has said, is factually incorrect. But it is very biased. When he says that Chrome's certificate rev- its only private CRL set doesn't contain all the revoked certificates. I mean, boy, is he telling the truth because it's like saying that this thimble of water doesn't contain all of the water in the ocean.

Leo: Right.

Steve: Because the Chrome CRL set is limited to about 24,000 certs total. And that's how many are revoked daily. So it's an incredibly minimal subset and absolutely doesn't do the job.

Leo: Well, we're going to get into this debate. That's going to be very interesting next week. And, by the way, there it is. That's what it looks like if it's done wrong. I'm surfing to it on my Android phone, and I'm getting a message from Steve, "Revocation awareness test. If you can see this, and apparently you can, you are using..."

Steve: Because you're reading it.

Leo: Yes, "a revocation unaware web browser." So you want to see this. That looks like it's broken. But the way this is, that's the test. You want to see the block with the big...

Steve: The router preventing you from getting to this page, exactly.

Leo: You don't want to see the page. And if you do, then you're going to want to listen next week and find out why Google has decided that this is not a box that you should check.

Steve: Yeah. We have - it's a fabulous podcast.

Leo: I can't wait. And everybody's going to listen. But just should I still check that box in Chrome?

Steve: Yes. It does you no harm. What Adam's argument is - and again, he's not wrong. It's that...

Leo: It does you no good is what he's saying.

Steve: Yes. He's saying, if a bad guy did actually want to exploit a revoked certificate,

they could arrange to defeat the revocation test. But it's not absolutely true. You have to be a powerful, properly positioned bad guy in the right setting. So he's absolutely right that revocation, Google's revocation with that checkbox can be defeated. Firefox's cannot because everybody else is using what's called "soft fail," where if you can't verify revocation, then you assume it's okay. If you check that second checkbox on the Firefox dialogue, where they say, if you cannot affirm that the certificate is okay, treat it as failed, that's called a hard fail. Only Firefox offers it.

Leo: Ah. So what I'm seeing here on Chrome, this is not a hard fail.

Steve: Correct.

Leo: This is check for server certificate revocation. But it will not fail if they can't verify that it's been revoked, even if it's not a correct certificate.

Steve: Exactly.

Leo: Well, that's not right.

Steve: Exactly. No. I mean, and so my whole goal, you know, Internet engineers have been saying, I've read Adam say there doesn't seem to be much demand for this. I argue it's because nobody knows.

Leo: Yes. I didn't know.

Steve: This is being hidden from us.

Leo: No, yeah.

Steve: Yes. And so by the end of next week everybody is going to understand this. It's really interesting.

Leo: And there'll be a sudden increase in the demand.

Steve: And that's all we need. We need, I mean, Apple, Apple doesn't allow browsers to do this. And Android absolutely doesn't allow it. Chrome can't do it. They can't even use their own small 24K certs block list on Chrome, or on Android. So it's very complicated until the end of next week. And then everyone's going to go, okay, that wasn't so hard. I understand all the issues.

Leo: I saw Adam's post. Of course it was the first thing I was going to bring up on

this show because I said, whoa, wait a minute, it's more complicated than we thought. And it is. But of course there's nobody better than the Explainer in Chief, and so we'll talk about it next week.

Steve: We will.

Leo: You'll find Steve at GRC.com. That's where you'll also find SpinRite, the world's best hard drive maintenance and recovery utility. That's Steve's bread and butter. There's a lot of free stuff there, too, including the browser revocation test, and I presume lots of information about that coming up. Previous episodes, show notes, full transcriptions so you can read along as you listen along. He also very kindly puts a 64K, I'm sorry, 16K version of that on the website, low-quality audio for bandwidth-impaired folks. But we do have 64K audio, MP3 audio, as well as full HD and SD video at TWiT.tv/sn for every one of our 452 consecutive episodes.

Steve: It's funny, Elaine waits for me to create the compressed version. I mean, she's waiting now until...

Leo: She needs a 16K version.

Steve: Because she's got a satellite link and is really bandwidth constrained. So she does the transcripts from the lower quality audio version.

Leo: Well, I appreciate all the work you do for us, Steve, in each and every week. And we'll be back here, as we are every Tuesday, 1:00 p.m. Pacific, 4:00 p.m. Eastern time, 20:00 UTC for our next edition of Security Now!. And I hope everybody will join us then. Thanks, Steve.

Steve: The Revocation Revelation.

Leo: Ooh, baby. You should call Adam ahead of time and let him know.

Steve: We're already talking.

Leo: Yeah, I can't wait. Thanks, Steve. Take care.

Steve: Bye.