SECURITY NOW!

Transcript of Episode #446

iOS Security

Description: On the heels of Apple's major update to their iOS Security whitepaper, Steve and Leo catch up with the week's top security news, including coverage of Edward Snowden's live appearance during the recent SXSW conference. Then they take a deep dive into everything we have learned about the inner workings of iOS. Most is good news, but there's one bit that's VERY troubling!

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-446.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-446-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. It's Patch Tuesday. We'll talk about that. We'll also talk about how Steve plans to stay safe with XP. He says you can, too. And then we'll go through Apple's security whitepaper, the details about how iOS protects you and whether Steve agrees. It's all coming up next on Security Now!.

Leo Laporte: It's time for Security Now! with Steve Gibson, Episode 446, recorded March 11th, 2014: iOS Security.

It's time for Security Now! with Mr. Steven "Tiberius" Gibson, our Explainer in Chief. Steve is the guy who discovered the first spyware, coined the term "spyware," wrote the first antispyware program. He's written many a program to help you protect yourself online. They're all at GRC.com along with SpinRite, the world's best hard drive maintenance and recovery utility. And each and every week he joins us to give us the latest security news, to answer your questions, and to explain how stuff works. Hi, Steve.

Steve Gibson: Hey, Leo. Great to be with you again, as always, for our 446th episode of this weekly security tracking. This week we have so much to talk about, this may end up being a two-part episode. I don't want to shortchange the later stuff, especially very toward the end because this is based on the revision and update that Apple made of their iOS security document. I found reference to one that was dated 2012. This one is February 14th, 2014.

So this contains essentially a current snapshot of Apple's statement about what they've done for iOS security. And I couldn't decide whether just to title the podcast "iOS Security," which is the title that ultimately won, or "Crypto Extravaganza," or "Crypto Heaven." Because, I mean, I am just - it's really interesting, too, because there were some areas where their new thinking exactly tracks the thinking that has been developed for SQRL. I mean, the same, it really was, it was sort of freaky as I was reading through the iOS document, and they were explaining how the fact that they used Touch ID, which allows you to bypass a password, allows you then to use a longer, more painful password which you wouldn't otherwise use.

Remember I was just talking about that with the way SQRL uses a hint, where after you once enter your really long SQRL password, you can then just sort of remind it that you're still you, just by giving it the first few characters of that password. And the fact that you're able to back off from that requirement encourages people to use a really good one that they only have to use very infrequently. And exactly the same tradeoff and logic is laid out in this document. So it's like, okay, well, that sounds very familiar, Apple. And there's other places, too. In fact, except for one place, they've chosen all the same crypto that I chose, the same Dan Bernstein 25519 elliptic curve and so forth.

Leo: That's not completely surprising because that's probably best in class and kind of...

Steve: Oh, it is. Well, yes, yes. Although, and independent discovery we know happens all the time. If people sit down and try to find the best solution, given the same set of starting circumstances, they're apt to come to the same conclusion. And that's what's happened here.

Leo: But speaking as an end-user, it makes me feel better about it that they chose the same thing as my security guru.

Steve: Oh, I was smiling. I was smiling when they say, oh, we're using curve 25519. It's like, oh, well, yes, good. That's what you should be using. Except one place.

Leo: Uh-oh.

Steve: There is a bad NSA...

Leo: Oh, no.

Steve: ...oddity, yeah.

Leo: Oh, no.

Steve: Yeah. I mean, and it just stands out. It's like the one place they use the wrong elliptic curve, and it's a NSA compromised one, and it couldn't be in a worse place.

Leo: What a surprise.

Steve: It's in the iCloud keychain logic to protect everyone's iCloud backed-up password libraries. And so it's like, oh ho ho ho ho. And it just stands out there. And they just sort of casually said, yeah, we use P-256. It's like, what? You didn't use it anywhere else. Why are you using it here?

Leo: Interesting. Isn't that interesting.

Steve: Really freaky. So, yeah. But there's just - I have to say, I mean, my overall take, with that exception, I mean, and it does sort of spoil the whole apple [clearing throat]. But they did everything right. I mean, step by step through the design of this, it is fabulously structured. And at nowhere that I could find are they taking any advantage, are they taking more than they need to. I mean, the architecture demonstrates a comprehensive respect for the user's privacy. I mean, it's just - it's immaculately designed. So I came away feeling really comfortable with it, with this one caveat, which we'll talk about.

But it's also the case that where they have made communication easy, such as with iMessage, you have security, but privacy is completely broken. It's completely broken. So even though the security is good - and all we got was murky information about iMessage before. Now it's laid out. And it is absolutely demonstratably provably secure, except we have to trust Apple because they maintain the directory of public keys. And it explicitly allows them to insert themselves in the middle, to perform a man-in-the-middle attack if they wanted to, and in an environment where they are also prioritizing ease of use.

And I have to tip my hat again. It's amazing how much security they have created and hidden the inherent tradeoff that you normally have with crypto. I mean, what we're holding in our hands, these little iPhones and iPads, they are little crypto bricks. They're amazing instances of applied cryptography. I mean, they really are tremendous, given what we now know from this document. So anyway, as I said, we've got some news to talk about, and we really don't want to hurry through that because we want to talk about Edward Snowden's live appearance on SXSW.

Leo: Oh, good.

Steve: We of course have the question of do we now know who Satoshi is.

Leo: Oh, boy. That's a good one, too.

Steve: Yeah. A little wacky one is Native Americans are jumping on virtual currencies and defying the U.S. The first result of the TrueCrypt audit is out. The SQRL language translation project has just taken off like - it's like I couldn't even believe it. We've got 34 translations, and 80 people signed up to translate the user interface into 30 languages and dialects. And more to talk about. So a great podcast.

Leo: We're going to be busy.

Steve: Which may be Part 1 of a great podcast.

Leo: Two great podcasts. With seven proxies to protect [Snowden], I presume, his - I mean, we know his identity. It's not for anonymity, but to protect him for his location; right?

Steve: He mentioned Tor. We know that he's a believer in Tor. And I'm wondering if it might have been just the Tor network he was using, and he had set it up...

Leo: Deep Tor, I mean, usually you go more than seven.

Steve: Oh, seven? Seven nodes jumping all around? Yeah, that's pretty good. I'm surprised we could even see him.

Leo: Yeah, two frames a second, about, yeah.

Steve: Yeah. Okay. So first of all, I just wanted to briefly follow up on something I mentioned last week. We talked about Foxit, unfortunately, and the fact that they had gone to the dark side, apparently. I received subsequently three tweets, one from Lobby Canada, that's @lobbycanada, said "@SGgrc, it's true about Foxit. They've fallen in with the Conduit.com scumbags."

Leo: Eww.

Steve: Uh-huh. "Now I've got to uninstall from everywhere." Someone, @gh_m3 tweeted, "@SGgrc, I've abandoned Foxit years ago. It's long annoyed me. Sumatra is a great super lightweight alternative." And so that's one of the reasons I wanted to bring these up is that that is the alternative plugin for Firefox, which works very well.

Leo: There are, we should mention, PDF readers. I don't know if we've said what these do.

Steve: Yes. Sorry, yes. I'm just assuming that everyone's following...

Leo: Everybody knows.

Steve: Right, right. And then @Really_Evil_Rob, with underscores between those words, he tweeted, "Upgraded Foxit, installer added a cloud component program without my knowledge. Uninstalled and switched to Sumatra." So, yup. So beware, anyone who updates. And I heard from other people that, like, a collection of stuff was installed in their machine, so not just one thing. I mean, those guys basically just really decided to monetize and upset their install base. But, again, it's their right to do so. It's our right to choose not to use them.

So I thought that Snowden's hour and the interview with the two guys from the ACLU held, what, I think it was 11:00 a.m. here, 9:00 a.m. Central time, was fabulous. And so I commend everyone who is interested to find this. The ACLU posted a really bad audio. I don't understand where it came from. But the echo on it is so bad, it's unlistenable.

Leo: Oh, that's silly.

Steve: Whereas the live stream was really good. I mean, there was some muffly-ness. But if you're going through seven proxies, you're going to have a little bit of that. But no echo. And the interviewers were super clear. Anyway, someone has captured the live stream, and I've seen a couple of them. The best one I've seen, I created a bit.ly shortcut for. It's bit.ly/sn-snowden. Just the word, just Edward's last name by itself, that was already taken. So sn, as in Security Now!, hyphen snowden, all lowercase. That will bounce you to a very good one-hour YouTube capture of this one-hour interview.

Leo: It's from YouTube, a YouTube account called LeakSource.info, just for people's info.

Steve: Yes, that's the one. And it's completely listenable. It's virtually what we received. You're not going to see Edward's mouth moving in time to the audio. It's like a one-frame-per-minute sort of thing. But the audio gets priority. It's funny, too, because they make a point at one point of saying the irony is not lost on them that they're using Google Hangouts in order to knit this thing together.

Leo: Well, but let's not forget that Julian Assange dared to before that use Skype, and it died in the middle. So probably better to use something and then the proxies and all of that, yeah.

Steve: Yes. And in fact they had a backup presentation in case some forces in the universe decided to kill the interview. They had prerecorded interviews, both from Assange and from Snowden and one other person, that they would be able to plug in if the worst happened. So it didn't. It was an uninterrupted interview. What I liked about it was that it was so sane. It seemed, well, I mean, it just came across. It wasn't rabid. Nobody was huffing and puffing. It just struck me as a very mature sort of reasoned rehash of essentially what we've been talking about off and on through the last year.

The strongest points that Snowden made were he returned to the concept that crypto math works, that is, that the math we know, and there is math that has been - whose provenance is uncertain, like the infamous now elliptic curve deterministic random bit generator, the EC-DRBG, which was famously sort of snuck into RSA and then became the default, which everyone now believes was the result of NSA manipulation. And of course we know that for some reason the RSA that year received $10 million, about a quarter of their annual "revenue," air quotes, apparently in some sort of deal with the NSA.

So but notwithstanding, the things that are known or believed to have been influenced for the NSA's benefit, the fundamental math we really understand works. And the academic community is continuing to develop it and vet it. And he also reiterated another common theme here of the podcast, which is only end-to-end encryption is TNO. Only when you arrange to share keys or to generate a shared key in a way where you're also authenticating each end because that's the other key. Without authentication, you aren't protected from a man-in-the-middle attack. So you have to have that.

And of course we know that there are tools like Threema and like TextSecure that have explicitly deliberately arranged to provide that to users of current mobile platforms. And unfortunately, as I was mentioning at the top of the show, iMessage, in making that tradeoff to just have it work, sacrifices that guarantee, and therefore it isn't a system that we can absolutely know is not being eavesdropped on.

And the thing that he said that I really didn't pick up in his initial presentations was that what he was honoring, what he felt he was honoring was the Fourth Amendment, which of course is our protection in the U.S. against illegal search and seizure, which he said he swore to uphold, and that's what he was continuing to uphold. And his point was that the Fourth Amendment doesn't mean seize everything, but don't search through it, which is arguably how this thing has been interpreted and the regime we've been under now for some number of years.

And I also wanted to make a point that the very first episode this season of "The Good Wife" featured the NSA's involvement in some wiretapping and eavesdropping, and they were back in last Sunday's episode, and it was very episode. I mean, they've upped the stakes of the NSA listening in on their phone conversations. So if anyone's curious, or if it's in some VCR or DVR, and you haven't caught it yet, I can recommend it. It was pretty good. So anyway, it was a great hour, and I really do recommend it. If people haven't heard it yet, it's worth listening to. The interviewers know their stuff. They were really able to ask him the right questions. They also pulled from Twitter live, and the tweets coming in had also great questions. So it was an all-around very worthwhile hour. So I really recommend it. Again, bit.ly/sn-snowden will take you to the link.

Leo: There's also - you said that the ACLU audio was bad. Was that from the YouTube posting? Because somebody's saying that the ACLU's official post is better quality.

Steve: Okay, good. What I saw was something that they put up immediately, very shortly after the show. And it only showed Ed talking. It didn't show the interviewers. And you're right, so this looks very good, what's showing right now.

Leo: This is the feed from the SX direct.

Steve: Okay. Then that's absolutely the one people want. It's probably a little more than an hour, then. How long is that one?

Leo: Yeah, it's an hour and one minute. It's from ACLU Videos, YouTube.com/aclvideos, all one word.

Steve: Good.

Leo: And another note, by the way. Of course it's Ben Wizner, but its also Chris

Soghoian, that's Sal Soghoian's brother, who is the security guy at the ACLU, I think.

Steve: Oh, and again, they were - he was great about technology. And he answered...

Leo: Chris is amazing.

Steve: Yes, he fielded questions, too. I mean, it was a really fabulous hour. So again, any of our listeners who haven't just automatically decided that Edward should be shot, and I know that we do have some listeners who think that way.

Leo: A few of them think that. I think increasingly it's clear that's not the case. I have to say.

Steve: Well, I was thinking about it because I was tweeting these links and getting some feedback from our listeners, or at least from my followers, some of my followers who are just absolutely rabid about this. And as I was thinking about it later, I thought, you know, if this hadn't happened, we wouldn't know what we know now. I think it is vitally useful that we know what we know now. Imagine not knowing any of this that we know. And I also saw that apparently only half of the document release campaign is through. And there's still more coming, which is presumably, I mean, we are told even more significant.

So, yeah, I mean, would anyone choose for this not to have happened? And Edward said, if he had to do it again, he would do it again without a moment. I mean, I think this has worked out, although he's banished from the U.S., I think that's the price he paid for doing what he felt was right. And again, I thank him because I can't imagine not knowing, turning the clock back a year with the blinders on the way we now know it.

Leo: Exactly. Exactly. Yeah. And I don't think you can make the case that there's been any harm done, either. Maybe I don't know, but it doesn't seem as if much has...

Steve: Unfortunately, we don't hear about any harm being done except from the people you absolutely know that's all they're going to say. And so it's like, well, okay, fine, of course you're going to say that. I mean, yes, all the generals and commanders in charge are furious and livid and talking about all the damage that's been done. Certainly we know that it's been done to our reputation. But GCHQ hasn't fared any better than we in this. And anyway, so I just - when I pose it as do we wish it never happened, I can't imagine going back to the way we were. I mean, unfortunately, it's now part of the terrain. But it's the truth of the terrain. So who is Satoshi?

Leo: [Laughing] I'm dying to hear, dying to hear what you think of this one.

Steve: We arguably still don't know. I think we do know. I mean, I think what he first said when he was surprised by the reporter from Newsweek probably was the truth.

Leo: He said, "I don't do that anymore." Later he recanted to the Associated Press, saying, "I meant I don't do engineering anymore." But it was pretty clear. Now, it does rely on the fact that Goodman, the Newsweek reporter, was accurately reporting this. And I wish we had a recording, frankly.

Steve: Yes. And actually what he sounded was - I had it here.

Leo: Let me see if I can find the link.

Steve: Yeah.

Leo: Something like, oh, I don't do this, I'm not doing that anymore. I don't do that anymore.

Steve: So, yes. I do have it here. "I am no longer involved in that and cannot discuss it. It's been turned over to other people." Now, that's damning.

Leo: Yeah.

Steve: I'm sorry, that's very hard to retract that. And so the story is he's a 64-year-old physicist who is on the cover, essentially, the story of Newsweek magazine. He changed his name some years back from Satoshi Nakamoto, the name we - the author of the published document, the paper where all of the Bitcoin architecture and crypto was laid out. He changed his first name from Satoshi to Dorian.

So according to Newsweek reporter Leah McGrath Goodman, when Dorian Nakamoto was confronted at his home before publication and asked about Bitcoin, he responded, quote, "I am no longer involved in that and cannot discuss it. It's been turned over to other people." So subsequently, of course, when a news storm erupted, predictably, around him, he chose one reporter from the crowd that was standing on his front lawn of his home in Temple City, California. And so he chose an AP reporter. They drove off to go get some sushi somewhere, and of course with this vapor trail reporters following behind. So it was quickly clear to him, just looking in the rearview mirror or turning around, that, okay, this wasn't going to work.

Leo: He wasn't going to get away from these guys.

Steve: Have any private sushi anywhere. So instead they went to the AP press headquarters, where they had to - I'm sure the AP guy had to swipe a badge, the gate went up, and obviously it was private property so other reporters were not welcome. And so there he says that he was misunderstood. And he said, "It sounded like I was involved before with Bitcoin and looked like I am not involved now," quotes the AP. "That's not what I meant. I want to clarify that." So of course Newsweek stands behind Goodman's reporting, saying that she did by the book, lived up to all the standards of editorial reporting that they would want. So this has caused, his recanting and rewording and denial have of course caused some controversy. So we don't know. But let's remember that there is still a story here. I mean, there's a lot at stake because we will know him by his digital signature, which he probably, if this is Satoshi, the Satoshi, still has.

Leo: I hope so because he's got $400 million in bitcoin hanging out in his wallet.

Steve: Actually 600.

Leo: 600 now?

Steve: Yes. Analysts who've looked at the Bitcoin ledger have concluded that the creator of the system, which is presumably he, owns about one million coins.

Leo: Holy comoli.

Steve: A million bitcoins.

Leo: Jiminy.

Steve: So it's like, okay. Yeah. So, wow.

Leo: That's all you can say. Wow.

Steve: Yeah. Great story. Meanwhile, the Lakota Native American Indian tribe have decided they're going to adopt the MazaCoin, which is yet another Bitcoin clone. It's actually a true clone. Basically another developer did his own version, or I hope he didn't stray too much from well-proven code. But he came up with another version of Bitcoin and was looking for some good place for it, he said. It's like he wanted to do something good with it. So federal laws granting Native Americans special legal status do provide an argument for a currency totally independent of the U.S. dollar. And Native American sovereignty is legally defined over a patchwork of treaties, laws, and precedent.

Leo: Isn't that interesting. Wow.

Steve: Yeah. It's a little controversial. But a spokesman for the Lakota said, "We're on sovereign soil, so we have the right to have Bitcoin, Litecoin, MazaCoin," whatever coin we want. And legal counsel, South Dakota legal counsel for the Lakota, an individual named Chase Iron Eyes, believes the federal government will push back if MazaCoin succeeds. Yet, he said, "There hasn't been a tribal nation that has declared its own currency and has mandated that currency is used within its borders. But it's because of this pervasive, ever-present asserted dominion of the United States. They'll try to shut us down, try to cite us with law violations." So we'll see how this plays out. And my comment is "A disruptive innovation indeed."

Leo: Very interesting.

Steve: Yeah.

Leo: So you believe that he really is Satoshi.

Steve: I do.

Leo: Yeah. I think I do, too. I feel like, of course, Goodman probably turned her notebooks over to Newsweek, and those would indicate that - but I wish she'd recorded it. I know that's not actually traditional process in print journalism.

Steve: Yeah, and I also feel like he, if he really wants to have his privacy, he has a right to his privacy. I mean, so maybe this will all die down. Maybe over time it can kind of leak out and so the pressure can all be released from this pent-up mystery of who he is. Sounds like she caught him by surprise. He spoke the truth. Then he had a chance to rethink it and say, oh, my goodness, what have I done.

Leo: Well, and understandably. He's been anonymous all this time. And for good - he clearly is a little nervous about how people might react. I don't think he's at risk for his life except that he's very wealthy.

Steve: He's changed his name, too.

Leo: Yeah. Well, that's the thing. To me that's the weird thing, which is why did he use Satoshi Nakamoto? That's his real name.

Steve: Well, and again, this is another place where the Wayback Machine comes in handy. He had no idea this was going to happen.

Leo: Right, right.

Steve: And so it's easy to look back and think, oh, boy. I mean, clearly he wishes, given his news presentation to the Associated Press, clearly he wishes now that he had always been covering his tracks and deliberately being anonymous. But again, no one knew this was going to happen. This was just sort of a wacky Internet concept. There was a prior coin that never really got off the ground, as I recall, and then this one made it. And so it's like, well, he probably...

Leo: He didn't know. He didn't know.

Steve: Yeah, exactly. No reason to take that precaution preemptively. Clearly, now, based on how he feels, he wishes he had. Wow.

So we got a big update to v7 of iOS. It touches on the podcast mostly, not because of all the other tweets, but because maybe Touch ID got fixed. I have had some very early feedback from other people saying that the Touch ID seems to be doing a better job of recognizing. It also seems to be faster. So there seems to be more resilience and more speed. So we don't know what that means. It'll take, just as it did the first iteration, I think it'll take us a while to see if this fade is still a problem for people who aren't overtraining. I even had some reports from people saying that overtraining, as we discussed on the podcast, seems to have a problem over a period of time, but apparently takes a lot longer to die than it did before. So we'll see.

Otherwise, from everything that Apple has said, it's just a whole bunch of UI tweaks. I've noticed a bunch of changes. I mean, I use my iPad constantly. And so I have seen a lot of changes in the UI. Nothing dramatic, but things that really stand out. When you scrunch the app in order to go back to the home screen, it used to show you the home screen. Now it shows you your wallpaper, then the home screen fades in, probably because they had taken a snapshot of the home screen previously, and so technically it was old, and so it was then updating itself quickly, and they were wanting to remove that. So now they update offscreen and then fade it in over your wallpaper. So sort of maturation sorts of things. But as far as we know, no other big things. But of course the new big, what is it, CarJack, I think it's called? No, it's CarPlay.

Leo: CarPlay, not CarJack. I hope it's not CarJack.

Steve: The ability to send your iOS experience to your dashboard. So we'll see how that goes. I noticed that fonts appeared stronger. The keyboard, the big onscreen keyboard fonts, they just look firmer, darker, stronger somehow. Although that's not enumerated among the changes that are reported to have been made. So I don't know about that.

TrueCrypt has had the first results of its audit sent to the developers.

Leo: Oh, yay.

Steve: Yes. Now, I saw that pass by and noted it. But when I went to get more details for links and anything more, I couldn't find any reference to it. Nowhere on the IsTrueCryptAuditedYet.[com] site is there a mention of it. They have everything calendared, but nothing in February 2014. So I'm not even sure if there's anything this year that is posted. But no mention of that. But I'm absolutely sure because this is something that would catch my eye and I would lock onto, that the first results of the audit have been given to the developers. Now, that's all it said. So we don't know anything about what that means. And if there were some things that were found, it would be responsible to allow the developers a chance to respond - maybe there'll be some interaction - or to fix if the result of the audit requires some fixing.

So I will certainly, and I hope all of the people that I've got who tweet me the things they discover, will let me know if anything surfaces about TrueCrypt and the audit because it's moving forward. And it was interesting, I saw someone who just coincidentally tweeted this morning and said, "Hey, Steve, any word on TrueCrypt and where it is?" Because nothing has happened for two years. One of the places I went to look, when I was looking for any update, was the TrueCrypt site. And on news, there's the latest version, dated something in, I think maybe it's October, something in 2012. And my reaction was, well, yeah. They've got it right. It's done. So it's not something you need to be updating constantly. And consequently it's there, and for two years it stood there doing a good job. So anyway, when we know more, we'll certainly let everyone know.

Leo: Excellent.

Steve: Team CYMRU is a nonprofit group that are dedicated to improving the security of the Internet. And they've put out a substantial whitepaper titled "Growing Exploitation of Small Office Routers Creating Serious Risks." And I just wanted to note that it's on my radar. It looks like it may be meaty enough to be a whole podcast. So I wanted to note that I've seen it, and I'm going to plow into it as soon as I'm able to. And we know, we've been talking a lot about the problem with SOHO routers and the firmware that's becoming problematic, even to the point that worms are now beginning to grow out of these things.

I wanted to mention that I have read "Influx," after you talked to Daniel Suarez on your Triangulation episode, and his Audible book reader. And then also it was Paul's mention. I think it was last week he said, I mean, he was asking you for permission, even though Audible wasn't a sponsor...

Leo: It wasn't an ad.

Steve: You know, it's like, I have to talk about this. And it was Paul [Thurrott] who just said, boy. He said it started off a little slow, but then picked up. I didn't think it was slow. I just thought it was...

Leo: I liked it. I loved the science in the beginning. I thought that was fun, yeah.

Steve: Really, yes, really fun. So I have completely read it, and I'm back now to Honor Harrington, sort of as my background reading when I can't do anything else, when I can't work on SQRL. And speaking of SQRL, the UI design is finished. I put it to bed on Monday, posted all the final screens. So that is done. At the same time, when I talked about us using crowd - oh, I've forgotten the name. Crowd...

Leo: It's a crowd-funding thing?

Steve: No. Crowd with two initials after it. Crowdin.

Leo: Crowdin.

Steve: C-r-o-w-d-i-n, Crowdin, the Crowdin guys, who were kind enough to make their facility available to the projects. Still hadn't really made it official. Now, Leo, if you go to Crowdin - is it dot net?

Leo: Yeah.

Steve: Crowdin.net/projects/sqrl. That'll take you to - maybe it's project? Maybe it's not...

Leo: Single, single project.

Steve: Yes, project.

Leo: Let's try that.

Steve: SQRL.

Leo: Yeah, there you go. Project singular.

Steve: Yeah. Now, up in the top...

Leo: Wow, look at this.

Steve: So up in the top are the only languages we don't have translators for.

Leo: Holy cow, look. You need Arabic, Korean, Tagalog, Thai, and Vietnamese. But look at all the languages you've got, including Chinese, Dutch, French, German, Greek, Hebrew. Holy - these are all - Ukrainian? Wow.

Steve: Yes. I put up one string, "Welcome to the SQRL Translation Project." And we now have that string, not that it does anything...

Leo: Oh, that was an easy one. So it's just one, okay.

Steve: Well, I needed somewhere for people to gather 'round. And I really didn't expect that it was going to explode like this. But you can also see, I think it says 80 or however many over there on the right, how many people are a member? There's me and then...

Leo: 81 users. So if you speak English and a second language, and you'd like to help out, this is a great place to go: Crowdin.net/project/sqrl. Wow, that's great.

Steve: Yes. And so I did want to definitely tell people that this exists. We'd like - it'd be better to have - in some cases there's only one person per language. It's better to have a little bit of a team per language. And what this does is it gives you a complete forum to discuss competing translations. I mean, so one person might put something up, and someone says, yeah, you know, that - I mean, since I only speak English, I can't give any really good examples. But I'm sure there are alternative ways of saying the same thing in a different language. And so this really creates a social environment for translators to work.

So here's the deal. The announcement of my finishing the user interface design coincides with my announcement that I am starting coding. So I am finally, when this podcast is over, I literally set to work on writing the code. I'm not going to publish all of the user interface strings because I am sure they are still subject to change. Working through the whole UI allowed me to essentially design the product. I mean, it is the detailed design of when you push this button, this happens, and how it flows, and all of the user interaction. That's nailed down. That's what I did.

It took 10 weeks because, as I have said before, there were places where I realized ease of use was in conflict with the technology. And so I went back and changed, made major changes in the way we manage keys and the way, like in some cases, keys were being independently arrived at. Now there's a parent-child relationship between them that really helped the user flow and the user experience.

So this took 10 weeks. I'm sure, as I'm working through actually implementing this, there will still be things that come up. There will be something I want to break up into two pieces, or I realize, hey, there's a simplification here that will have an impact on the language. So even though I have an initial set of what I would consider alpha user interface strings, I don't want to run people around in circles by posting them all and having everyone work on translations because, I mean, there is a lot. There's a lot of user interface. And I think it makes much more sense for me to push this thing rapidly across the finish line, which is now my goal, and then we will have it running in English in a known final implementation. And then I can immediately publish the UI strings for translation. And a very short time later we'll have it in 34 different languages.

So what I am doing is, I mean, the impact that my determination to make this internationalized, which is really enabled by everyone's willingness to help with the translations, is that I'm externalizing the strings explicitly into a set of files which are what will be translated. And of course I also clicked a button when I was setting up the project to publish the result. So what the other cool thing is that the results of everyone's work is not just for me, but it is for all SQRL clients. That is, the entire multilingual product of this effort will be public. And so to the degree that clients for iOS and Mac and Android and other platforms reuse what I have done, they're also getting for free all the translations into all the languages of the world. So it's an exciting moment here. And believe me, I'm really excited to get going on writing this thing because of course I want to see it happen, and I want to get back to SpinRite.

And speaking of which, I'll just share a tweet, so we know it's short. It's Jeff Harmon, who tweeted from @harmon_jeff at 10:41 p.m. on the 6th of March, so a few days ago, using his iPhone, he said: "Thank goodness for #SpinRite and @SGgrc. Repaired hard drive enough to pull off 350GB of photo/video to a new drive." So another instance of SpinRite saving people's data. It is the case that drives are cheap, but people often have huge investments in their photos and videos. There was another thing that came along relative to Bob Cringely, THE Bob Cringely, who on his blog, I saw some people letting me know that Robert was now running all of his drives through SpinRite...

Leo: Oh, that's good.

Steve: ...as some big process to, like, keep them alive. And, like, before doing data transfer or data recovery, he was just doing a big housekeeping maintenance project using SpinRite. And he mentioned there, he said: "Still the best hard drive data recovery utility ever written." So thank you, Bob.

Leo: That's awesome.

Steve: You know, I completely forgot something else that is important, this being the second Tuesday of the month. This is Microsoft's Patch Tuesday.

Leo: Oh, today, yeah.

Steve: Yeah, the 11th. And this is the second to the last one that there will ever be for XP. And we knew, I knew, and I know you knew, Leo, last week that my suggesting that it was okay to run XP without a monthly infusion of emergency blood transfusions was safe. And sure enough, I got flak through Twitter from people, and I haven't even checked the mailbag, people telling me I'm crazy. And I did make it very clear that this was for Security Now! listeners, security-aware people, that people's grandmothers probably shouldn't do this. But the cases that we read in last week's Q&A were people where they were being very careful. Their XP was in an internal network. They needed to use it in order to run a remote desktop application, or they were only viewing a different desktop, et cetera. I mean, my argument being it's not like the bits decay without their monthly life support of updates from Microsoft.

And I got a kick out of an F-Secure report which I just ran across this morning when I was tracking - actually it's relative to the relative amount of malware on different mobile platforms. And that's what brought me to it. And it's a fabulous report. It's their - they do a first half of the year they call H1, and a second half of the year H2. This is their H2 report, so it just came out about the second half of 2013. And one page caught my eye, and this is completely germane and relevant to us and this issue, so I wanted to share it. The title of the page was "The End Is Nigh?" And they said Microsoft Windows - now, this is F-Secure, remember, the serious guys who are on top of what's happening with malware in the world.

They said: "Microsoft Windows XP operating system reaches its end of extended support period on April 8th of this year. And after that, no more public system updates. No more public security updates. Users will be on their own. But XP is still a very popular OS, or at least it is prevalent. See other sections of this report for details," they say. Then they said: "Elsewhere in this report are detection statistics which highlight two very serious threats to Windows users, web-based attacks and Java-based attacks. And Windows XP is particularly an issue because, once compromised, it is much more difficult to repair than its siblings. An ounce of prevention is really worth more than a cure in the case of XP. Prediction: The April 8th 'deadline' [they have in quotes] will be picked up by the mainstream press as a type of Y2K apocalypse waiting to happen. And when nothing happens on April 9th, the press will again publicly question what all the fuss was about.

"Meanwhile, in the tech press, reporters will be patiently waiting for the first critical post-

XP vulnerability when, not if, a powerful zero-day exploit makes its way to market. That's when the real concerns begin and important questions will be asked. Can XP be trusted? But all is not lost. Patching XP is not the first line of defense, or it shouldn't be." And they're actually saying it better than I did. I mean, this is essentially what I was saying.

They said: "Some businesses will continue to use XP throughout 2014, either due to contractual obligation or because their customers do so, and they need XP to provide support. In those situations, IT managers have their work cut out for them. Air gapping systems or isolation to separate networks from critical intellectual property is recommended. Businesses should already be making moves such as this for 'Bring Your Own Device' users. XP is just another resource to manage," essentially. "Folks that continue to use XP at home can do so with some reasonable amount of safety for a while still, but they absolutely need to review their Internet, particularly web browsing, and computing habits." And that absolutely is the key. Which is essentially what we're talking about when we say, for example, you don't want to go browsing around the 'Net with IE.

So they have eight points: Install Windows XP's final update. Duh. Install an alternative browser or browsers. And it says, in parens, they're free. Don't solely rely on Internet Explorer, and don't use Internet Explorer as the default. Meaning use Chrome or use Firefox. If installed, make sure Microsoft Office is fully patched. Note that older versions of Office will run things such as Flash by default if embedded in documents. If using an older version of Office, tighten up the security options. Don't open documents from sources you don't trust. So in other words, Office is another point of entry of problems that we've certainly seen for years, and they're not going to go away, especially post zero-day exploit revelation. And of course the same is true of the browser.

Review the third-party software you've installed and uninstall anything that isn't needed. That's always a good idea. We've talked about that before, just essentially lowering your attack surface by having fewer things there. That actually is a point that Apple also made in their iOS security document that I didn't highlight and pull out for the podcast. But they make the point that, because of their design-to-fit approach, they didn't take an existing operating system, like a full-blown UNIX, and just move it over into the iOS mode and thus still have all kinds of things running where you arguably need a firewall in order to protect yourself from exploits that are unknown of those running services. Instead, they have an extremely minimal footprint doing only what's necessary. So they weren't in a position of having to block unknown threats.

So anyway, this just says, for third-party software that you keep - oh, so they say, if you're going to keep XP, do a spring cleaning and get rid of old software because old software very often equals vulnerable software. For the third-party software that you keep, consider disabling or uninstalling the browser plugins. Set the browser to "always ask" what to do about things such as PDF files. And again, I'm using Firefox as my goto browser with NoScript. And in fact Snowden also referred to both NoScript and Ghostery as an ad blocker for things to install in Firefox in order to make that browser more secure.

They said, for the third party software that you keep, consider disabling or uninstalling the browser plugins. Oh, I'm sorry, yeah, I already read that, the browser to always ask what to do for things such as PDF files. Then they ask, do you need Java installed on your home laptop? Probably not. Advanced browser features include "click to play" options. They're worth the extra effort.

Number 6 of 8 is have an up-to-date security product with antivirus and firewall installed. So the point is, again, even though you're not going to be getting things from Microsoft, you could still get things from everybody else, so having something there that is continuing to watch you. And as we know, Microsoft Security Essentials will continue being supported after patches stop flowing from Microsoft for XP. Keep your XP computer connected to a NAT router at home, which will act as a hardware firewall. Standard advice, of course, and great advice, given that your router can be trusted. And finally, consider updating your OS. If you don't want Windows 8, there's always Windows 7. The OEM installation is still available from many fine online retailers. So I know this is of interest to a huge body of our listeners who are still using XP, as I am, and will be for some time.

Okay. iOS. As I said at the top of the show, I am overwhelmed by what Apple has gone through to create what is arguably a "walled garden," as the term is. We know that it is a carefully controlled and curated ecosystem. The result of that level of control is the reason actually that I went to that F-Secure report because, one year ago, in February of 2013, McAfee, which is now the soon-to-be-renamed whatever it is, owned by Intel, they reported a year ago that the mobile malware samples had jumped from 792 seen in 2011 to 36,699 seen in 2012. So a dramatic jump in observed malware. And 97% of those samples were designed to attack the Android platform.

Come forward a year to this second half of 2013 report that F-Secure just released. And they say, quoting their report: "97% of the mobile threats in 2013 were directed at the Android platform, which racked up 804 new families and variants," said F-Secure. "The other 3%" - that's 23 things - "were directed at Symbian. No other platforms had any threats. In contrast, 2012 saw [by F-Secure's count] 238 new Android threats." So 238 new Android threats in 2012; 804 new Android threats in 2013.

And they explain. They said: "For mobile platforms, the continued dominance of the Android operating system makes it almost the exclusive target for mobile threats we've seen this period. Though the relatively low number of vulnerabilities found in Android itself makes the operating system difficult to attack, this security is largely circumvented by the relative ease with which malware authors can provide their 'products' and dupe users into installing it on their own devices, with the necessary permissions to straightforwardly use the device, and the user's data, for the attacker's own benefit."

So this really is the difference today between the Android environment and the iOS environment. And that is, solid and secure as the Android platform is, it literally takes everything that Apple has done to lock a platform down. I mean, given the truth about the amount of pressure there is for exploit of the mobile devices. And you have to know that, were it not for Apple having developed an incredible soup-to-nuts security ecosystem within their platform, which is really what they've done, which is what I learned about reading this 33-page paper and which I'm going to describe, were it not for that, iOS would be a catastrophe. I mean, it would be a disaster because you know bad guys would love to be in there doing what they could. And essentially the architecture that Apple has prevents it, with the exception of the explicit jailbreaking that you can do. If you really want to break these protections, it's still possible. But if you don't do that, the protections Apple provides are, I mean, they're just beautiful.

So essentially, to pull off this closed ecosystem, to actually close the system and to ward off what I think is clear would otherwise be a massive assault on the platform, they have had to take security very seriously. And what impressed me, as I'm looking at this, as I see how much crypto - as I said at the top of the show, I wasn't sure whether to call this "iOS Security" or "Crypto Heaven" because, I mean, there is just, the crypto stuff - there are some descriptions I'm just going to read without even trying to decode them or figure them out, not mostly, but there are a couple, because they just make your eyes cross with what Apple has had to go through in order to achieve the level of security they have.

And that's a lesson, too, because I would contend that nothing short of this is enough. That is, what we've seen time and time again is it just takes one weak link in an otherwise fabulous design to break the whole thing. I mean, as we know, the nature of security, we've talked about this often from a philosophical standpoint, is a chain of interconnecting links, of interdependence from one end to the other, where you're depending upon the proper behavior at every step. And, for example, at one point you're depending upon the processor itself to properly execute instructions.

If it turned out that there was some subtle problem in the division, and we've had those days, we all remember the early days of Intel processors where division problems were discovered, when you got the wrong answer on - I don't know if it was Excel or even before Excel. But a spreadsheet could produce the wrong answer because the divide instruction wasn't working. And even something like that could cause a break in the crypto system. There could be a way that could be leveraged by bad guys.

The point is, doing this, you have to be perfect. And perfect turns out to be really difficult because what we also want is a huge amount of flexibility. And arguably, whereas Apple gives us much less flexibility, they still manage to deliver a lot. And what impressed me was the user is impacted minimally by this. All of us who have used iPads and iPhones have sort of felt like a little - there's sort of an oddness that you encounter, like the way you have to do something. It turns out that underneath that oddness is serious crypto, which Apple has hidden the best way they could. And so it only sort of pokes out a little bit in you having - you kind of grumble about the way you have to do something. Well, it's only there because there was a serious security requirement for it being done that way. And more often than not, a magic is happening underneath.

So what I found - and again, it's hard for me to rave about the security structure of this foundation when I've already raised a question about the security of iCloud key storage. But for this to be useful, I mean, the phone is still useful. The tablets are still useful. The security there is amazing. And we'll get to talking about the use of the P-256 elliptic curve toward the end of this. It has no bearing on what I see as Apple's, in the architecture, Apple's total respect for the user's security and privacy in the design that they document in this 33-page PDF.

At the time of manufacture, the chip fuses a unique ID into itself which is part of the Secure Enclave crypto engine. There is a unique ID and a separate Secure Enclave ID, neither of which expose themselves in any way to the outside. Apple has no idea what it is. There is no idea. There's no way to determine what it is. All that you - your only interaction with it is that things get encrypted by these keys, and these are AES 256-bit keys fused into the application processor during manufacture. It's impossible, no software or firmware is able to read them directly. You can only see the results of their use by the encryption and decryption operations that are performed using them.

So this is sort of a key piece of the structure that Apple then leverages throughout the rest of the architecture. That is, they are often encrypting stuff which needs to get stored or needs to get sent somewhere under the device's unique ID. And no one knows what it is. The device knows. But there is no way, I mean, Microsoft doesn't know what this is. They are oftentimes seeing unique blobs or just pseudorandom blobs of data. So there is a huge amount of this that is really faithful to the TNO paradigm of trusting no one.

Now, as I mentioned, there are some communication services they offer that, again, for the sake of offering the service, they are not TNO. They are, unfortunately, this is a tradeoff that they had to make. And there's a little bit of misdirection, I have to say, in this document, where they're bragging about how they're unable to encrypt the data that's moving through iMessage. Well, that's really not true. But we'll talk about those details here toward the end.

So the first thing that they've done is they have a hardware AES 256-bit, so a 256-bit key, AES crypto engine, in hardware, built into the DMA - DMA is a common acronym, Direct Memory Access - the DMA path between the flash storage and the main system memory. Which is to say that, sitting there as a gatekeeper to a flash, to the nonvolatile memory, is a crypto engine. So that in absolute best practice, in the way that we've talked about from the beginning, never is nonencrypted data written to flash. It is always encrypted. Even when you don't initially have a lock screen or a password or a key set, there's a random number chosen during the first time you turn the system on. That random number is chosen.

It's then used as the key for this DMA encryption. And they make it clear that, if you have absolutely no password of any kind on your device, then you don't have the protection that you can get. But what this does give you is instantaneous wiping because all Apple has to do is to wipe that key, either through some action on the device or any of the mobile device management connectivity or the solutions they offer for reaching out and wiping your device remotely. So this 256-bit random number is available for erasure. And the other thing that they've done, again demonstrating the kind of attention to true security that we want, is they store many very sensitive keys in flash memory.

And then the question, of course, is okay, how do you erase that securely in the face of wear-leveling? And they have an explicit memory system that they call "Effaceable." Effaceable flash allows them to bypass the NAND-style wear-leveling and never have that data relocated somewhere else. So they're able to lock it in place and know that they are securely wiping a key when they want it to no longer be valid and never have to worry that the NAND flash memory manager has swapped that key off in order to balance the rights across the space of the NAND flash. So again, they did that right.

In the boot ROM is Apple's root CA public key. So that sits in the boot ROM and allows, throughout the system, allows every stage of iOS coming up to have its integrity checked. So the boot ROM loads from flash into RAM the working kernel image and verifies its signature. So Apple knows that what you're loading is this kernel image that they originally signed. One of the cool things about, again, the way Apple has thought this through, is the way they handle updates. Imagine the problem of this whole ecosystem moving forward, fixing over time security problems. We started with, what, we had v5 a while ago, or v4, then 5, then 6, and now we're at 7. And many of those earlier versions will still run on the hardware platform. But they have at this point many known security weaknesses, which have since been fixed. Well, and famously, the SSL certificate verification problem, the double goto fail problem.

So imagine the problem if it were possible to take an image of a prior build, like if you could take 7.0.6, which everybody in the world had loaded on their devices two weeks ago. And you just simply reload that, grab that from somewhere and stick that on someone's phone, make them deliberately downgrade their version of iOS. It would be signed cryptographically from Apple. So it's a valid image. It's a valid piece of code for this model and version of hardware. But we now know that there's a problem, and you wouldn't want to be running it.

The way they solve this is that every iOS device, when it wants to upgrade, is it generates an inventory of all the different packages, the pieces that it has. It generates a pseudorandom nonce, which will only be used once, and a version of its unique ID. This is not the UID, which is the device unique ID, which nobody knows. This is called the ECID. But it is also unique to the device. So the device takes its inventory, the nonce, and the unique ID. The nonce is there to make all of these requests unique, so that another one would always be different, so that there would never be a repetition. This goes off to Apple. Basically it's our device making a claim for updates from Apple, saying, "These are all the things I've got. Does anything need to be fixed?"

If so, Apple builds for this device, per device, a custom update package, including the ECID in this bundle, and signs the result. So what comes back is a bundle that is signed by Apple, that contains this ECID, not shared by any other device, unique to this one device, and this device will only run this bundle and accept it if it's both signed by Apple and contains its ECID. So that very handily prevents downgrade attacks. That is, you cannot take any of the update software from any other device and get a different device to accept it because essentially the device requests it, and Apple generates a custom package for that specific instance of a device. Apple does know the ECID, but not the unique ID from which the ECID was derived. So they've got that part nailed.

Leo: On we go. Steve Gibson, Steven "Tiberius" Gibson. And we're going through this most recent Apple security whitepaper. Not the first they've done, though. I didn't realize that.

Steve: Correct. Yes. They did one a couple years ago. And so this is an update, sort of where they stand. It's funny because the beginning of it was a little bit like they were, I mean, as I was reading it, they were talking about the ecosystem and all of this as if they knew this was where they were going to be. But we were there, it wasn't that long ago, and when the iPhone was a closed platform. There was no developers. Nobody was - Apple didn't support creating apps. They sort of had to do it. This was pushed on them, as you'll remember, Leo, from the outside world, people saying we want to make apps for this. Because it used to be just that little home screen with a little more than half of it filled with the standard icons.

Leo: Yeah, it was just going to be web apps, yeah. App Store took over. Now there's more than a billion - how many billions have they sold? It just is unbelievable. More than a million apps out there.

Steve: So as a consequence of studying this paper, when I hold the phone now, I'm impressed by it. It's just incredible what a nice piece of work this is. And I would argue that this sets the bar for mobile platform. This is what you have to do if you're going to do secure mobile. If you don't do this, and this was a point I hope I got across, if you, I mean, literally, don't do every single one of these things, then that creates an opportunity for, unfortunately, the world's malicious operators, and we obviously have plenty of them, to pry their way in. It has to be absolutely air tight. And making something air tight which is also as open as the iPhone ecosystem is, you know, you and I just downloaded a new app. We found out about the water log app from Andy, and it was like, wow, okay. Hey, now we both have it.

So there's a lot here that protected us and makes it safe for us to do this. The fact that we can only get apps from the App Store is, yes, it's a limit. But as F-Secure points out, it is the lack of that which represents the huge problem that Android has and why 97% of the mobile malware is on Android, is that people are going to install things from all over the place. And of course that is why people jailbreak their iPhone or iPad or whatever, is they don't like being contained by this walled garden. But if you accept the need, if you want the security and the safety, you're going to trade off some freedom. But what you really get is phenomenal security.

One thing new that we know got added was the so-called "Secure Enclave." That's a logically separate but physically resident processor that exists on the same chip. It's on the same A7 processor. But it is not connected to the processor except in sharing silicon and power. It has its own independent secure boot process that it goes through, much as the A7 does with the kernel microcode. It has a hardware true random number generator, so we have true random numbers. As we know, that's crucial for good security.

Prior to this you were able, thanks to all of the I/O that a smartphone has, you've got the gyro and the accelerometer, and you've got the camera and audio, there's a lot, and specific micro coordinates of where users are touching and things. You've got lots of nondeterministic input that help to create really good random numbers. Now, we probably have a reverse diode junction that's being biased at high voltage, and we're seeing electrons crossing it and counting them, which down at the quantum level is absolutely unpredictable. So that, there in the Secure Enclave, operates with - they use a counter-based deterministic random bit generator, not the EC-DRBG, but the CTR-DRBG, which is one of the good ones, because it's often the case that you need more random numbers more quickly than your source of entropy can actually provide.

So what they use is they use the source of entropy to seed and continually reseed the counter-based DRBG. Its state, then, is not known. So the future of its generated numbers is not known. Technically, they're related to each other, but in an unknowable way. And there's a limit to how long you can run one of these before your security guarantees begin to fade. And so the idea is that the real random source reseeds this much sooner than that limit, which is generally pretty large. Therefore, you end up always using numbers that no one can predict. And it's that predictability that is what we're looking for.

And Leo, remember you mentioned the word "tangling" last week? Because you must have seen this or read something about it. And you were right. They use this bizarre term. They call it "tangling." And, for example, quoting a line from there, they said, "When the device starts up, an ephemeral key is created, tangled with its UID, and used to encrypt the Secure Enclave's portion of the device's memory space." Now, I read the entire document several times. They use "tangle" all over the place. They never describe it. And no one's ever heard of it.

So I think what happened is they used to be saying HMAC, or they said we used a key message authentication code. And some proofreader came along and said, no, no, no, no, no one knows what an HMAC is. Come up with a different term. And so someone said, well, how about "tangled"? That sounds good. Anyway, so I think, for anyone reading this document on their own, when you come across "tangled," I think it's probably safe to replace that with HMAC, and we understand what a keyed hash is, a keyed message authentication code. So that appears to be - and I developed that supposition, and then in every instance where I saw the word "tangled," HMAC made sense. So I think that's what...

Leo: They define it in the glossary at the bottom.

Steve: Oh, do they.

Leo: Yeah, there's a glossary.

Steve: Okay, I didn't get here. I got, I mean, believe me...

Leo: You didn't need it. That's why.

Steve: By the time I was done, I was like, oh, my god.

Leo: And it's their own definition, obviously. I don't think there's - I asked you before, is there such a thing as tangled? And you said, I don't - never heard that before.

Steve: No. And I was curious. So of course there's something for everyone. There was once some tangled hash that ended up not surviving scrutiny for long.

Leo: Here it is. Here's what they say: "The process by which a user's passcode is turned into a cryptographic key and strengthened with the device's UID. This ensures that a brute-force attack must be performed on a given device and thus is rate-limited and cannot be performed in parallel. The tangling algorithm" - and you'll recognize this - "is PBKDF2, which uses AES as a pseudorandom function with a UID-derived key." So they're - it's like a hash, right, with the UID.

Steve: Well, yeah. And in fact, again, this is one of the other weird overlaps between their work and SQRL is I have something I call EnScrypt, which is this iterative use of the scrypt function in order explicitly to create a much more time-consuming encryption of the user's password. So we are tangling. And what's really interesting is that I was struggling for a term for that. And I asked the newsgroup folks, who were following along and working with me on this, I said, we're really not encrypting the password. We're, like, really hashing it. But no one knows what a hash is. So I'm just going to say "encrypt," even though it's not technically correct because we think of encryption as being reversible. It doesn't have to be reversible. It could be irreversible encryption. But anyway, so their solution to the same exact problem I had was to make up a new term, "tangling."

Leo: And the point is making it computationally difficult to solve.

Steve: Yes. And notice also that they're mixing in the UID once again. So part of this entanglement mixes in the unique device ID so that even two users using the same password on different devices would end up with completely different tangled results.

Leo: Excellent. That's even better.

Steve: And that's exactly what you want.

Leo: That's even better.

Steve: Interesting. And finally, just to finish off on this notion of the Secure Enclave, so what they've done is, again, this is - you could call it engineering overdesign, crypto overdesign, I mean, they really, really care about delivering on their promise here of creating a platform that is just bulletproof. So the communication between the Secure Enclave and the main A7 processor is just their ability to raise a semaphore. They use an interrupt-driven mailbox and shared buffers. So either one of them can put something into a shared buffer and then basically say there's something here for you. And so they share an interrupt where that brings the event of data being available to the other's attention. That's the only communication they have.

So there is no way for - so the Secure Enclave is itself a very well-protected walled garden which performs crypto operations, holds the master keys to the user's identity, and there isn't a mechanism for asking it to divulge them. There isn't a way. You can merely hand it data and say encrypt this for me in the following way, please. And then, when it's done, it interrupts you back saying, okay, here's your result. And that's all you can do.

So as an example of how much almost overboard they've gone, here's an example of the Secure Enclave's usage. It's responsible, that is, the Secure Enclave is responsible for processing fingerprint data from the Touch ID sensor. So there's been a lot of concern from the release of the 5s, what's being done with my thumbprint on this sensor? We believe Apple is going to make it secure. They've said they are. But we never really had any details. So now in this document we know. They write:

"The Secure Enclave is responsible for processing fingerprint data from the Touch ID sensor, determining if there is a match against registered fingerprints, and then enabling access or purchase on behalf of the user. Communication between the A7 [processor] and the Touch ID sensor takes place over a serial peripheral interface bus. The A7 forwards the data to the Secure Enclave, but cannot read it. It's encrypted and authenticated" - they actually use a well-known authenticated encryption block mode, AES-CCM, which is counter with CBC-MAC. So it's encrypted and authenticated. I mean, we're talking about the data moving across, what, about an inch and a half of little tiny micro-thin trace of wire as it comes out of the Touch ID sensor and before it goes into the A7 chip. It's been encrypted and authenticated for that inch and a half of travel at lightspeed "with a session key that is negotiated using the device's shared key that is built into the Touch ID sensor and the Secure Enclave."

So there's three parties involved. The Secure Enclave, remember, it has no I/O. It has no interaction with the world because that would be unsafe. So the A7 processor does have a serial peripheral interface, and it's able to talk to the Touch ID sensor and ask it for readings. But it doesn't have the key that's necessary for decrypting them. That key lives in the Secure Enclave. So all the A7 processor, the main guts of the iOS device, all it can do is serve as the intermediary, essentially deserializing the data over the little one-wire interface line, filling up the buffer in this mailbox that it shares, and then tripping an interrupt saying, okay, I got a fingerprint here for you. I don't know what it says, but it's up to you now.

So then they say: "The session key exchange uses AES key wrapping with both sides providing a random key that establishes the session key and" - as we said before - "uses AES-CCM mode transport encryption." So just for this half-an-inch journey inside the iPhone 5s, there is a secure key negotiation performed using random numbers generated at each end. And this data moves into the Secure Enclave. Now, in terms of the Touch ID unit itself, it uses an 88x88-pixel array at 500 points per inch in a raster scan, which is temporarily stored in encrypted memory within - so that's what it generates. It takes a snapshot, 88 pixels at 500 ppi. And I don't know what the per pixel intensity depth is on that. It's not said here. So that gets transferred into the Secure Enclave's encrypted memory, where it is vectorized for analysis and then discarded. So the bitmap is vectorized and discarded.

The analysis itself uses subdermal ridge flow angle mapping, as we've talked about initially, originally when this was released, which is a lossy process that discards minutia data that would be required to reconstruct a user's actual fingerprint. So they're making it clear, as we assumed, but now we know, that they're translating the bitmap image into this what they call "ridge flow angle mapping" representation, which doesn't allow you to rebuild the fingerprint. And they say the resulting map of nodes never leaves iPhone 5s, is stored without any identity information in an encrypted format that can only be read by the Secure Enclave. So that's private memory to it.

So during training, all the fingerprint images are encrypted as they go into it, dealt with by it, and stored encrypted in memory that only it has access to. And they never go anywhere else. All that can happen is that it decides if you are you and sends a little mailbox message to the A7 processor saying - actually it's even more sophisticated than that because it involves keys, which Apple just goes crazy with.

And that's where we're going to stop for this week. We will pick up on locking and unlocking the phone, keys, the file system, how much Siri has to get from you because it turns out Siri needs to know a lot more about you than is readily apparent. And we'll also talk about what happens with iMessage and the little bit of a concern that's raised by an odd choice of crypto, one place in the cloud. And we'll do that as Part 2 of this next week.

Leo: Great stuff. I'm really glad. And who knew it would take this much detail to parse this. But that's great. I'm really glad you're doing it. We need to know this.

Steve: Oh, and it's a fabulous document. I mean, they really did - I think this, you know, how much criticism have we and everyone given Apple over their historic refusal to tell us what's going on? And while there are some technical details left out, there's enough here, I mean, to really understand. We don't know how they're obtaining their initialization vectors for the AES-CCM, for example. But everything else was done right. We have to assume that they did that right, too. And arguably that's a level of detail that really doesn't fit in a whitepaper. I think this is set at just the right level, especially now that we know what tangling is.

Leo: Steve Gibson's at GRC.com. That's where you could find this show, 16Kb audio versions thereof; transcriptions, too, handwritten by a human being, Elaine Farris. You can also, while you're at GRC.com, get - there's so much other stuff. Of course there's SpinRite, the world's best hard drive maintenance and recovery utility. But there's, oh, SQRL is there. Perfect Paper Passwords. Password padding, he calls them "Haystacks" in general. Lots of great - it's more and more become a richer and richer resource. If you want audio, 32 or I guess 64K audio, or full-quality video, HD and SD, we have that at our site, TWiT.tv/sn.

You can also subscribe. We have RSS feeds for it in your favorite netcast catcher, Stitcher or iTunes, whatever you use. And share it with your friends, too. I think there are probably a lot of people on iOS who have the security wherewithal to understand what we've been talking about. Next week more, including how iMessages works. Because I'm really curious about that.

Steve: Yeah. And I have to say, too, even, I mean, we're going to talk about locking and unlocking. It's just incredible what they go through. I mean, these little boxes are so secure, so well designed. And my point is that, if they weren't, they'd be crawling with an infestation of malware, and they're just not.

Leo: Right, right. Steve, thank you so much. We do Security Now! Tuesdays now, our new time 1:00 p.m. Pacific, 4:00 p.m. Eastern time, that's 20:00 UTC because we are in summertime. UTC doesn't change, but we do.

Steve: Yay, I love summertime.

Leo: We're an hour earlier now, and it's whatever I said, 20:00 UTC.

Steve: Okay, Leo. Talk to next week.

Leo: Thank you, Steve Gibson. See you next time on Security Now!.