SECURITY NOW!

Transcript of Episode #465

iOS Surveillance?

Description: After covering the interesting news of the past week, Steve and Leo reexamine iOS security in the wake of a hacker's presentation at a major conference which brought it all back into question and triggered an avalanche of frightening headlines.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-465.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-465-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We've got some security news. He'll talk about the EFF's, what do they call it, Honey Badger? And he's got a response to the whitepaper we've been talking about all week from Jonathan Zdziarski about iOS security. A little bit in-depth on iOS security, coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 465, recorded July 22nd, 2014: iOS Surveillance?

It's time for Security Now!, the show that protects you and your loved ones online, your privacy, too, with this guy here, Steve Gibson, the guy at GRC.com, our security expert, author of SpinRite, discoverer of spyware, coined the term "spyware," wrote the first antispyware program, and has been doing this show, I think really now we can say that your credentials as a security guru come from eight years of covering the space on Security Now!.

Steve Gibson: I think that's probably true, yeah.

Leo: Yeah, I mean, this is now the platform. Hi, Steve.

Steve: Yeah, well, and the podcast has had me in this long enough that things are beginning to occur to me, like the Perfect Passwords and Off the Grid for a paper-based encryption system, which is really solid, cryptographically strong. And then of course SQRL, which is coming along and may actually get some market or some website share. It's really not market share because it's free. But it's an alternative using state-of-the-art public and private key crypto for obsoleting usernames and passwords. So, yeah, I think I've probably earned my stripes.

Leo: Earned your stripes for sure. And there's an advantage to having done it for a while. You start to see things come around again.

Steve: Yeah, yeah. I think that's exactly right. This week we've got not too much news. I was planning to talk about the web-based password managers. But as happens, in this instance, that got just pushed off to the side by a very attention-grabbing presentation is probably the best way to put it at the HOPE X conference over the weekend.

Leo: Hackers On Planet Earth.

Steve: And I agree with you. I love the acronym. I just love that Hackers On Planet Earth has the acronym of HOPE. That's neat. And so our main topic, the title, as you can see on the screen, is iOS Surveillance?, going up at the end to ask the question. And but we do have some news to talk about. We have new versions of Chrome and Firefox. Level 3 had a blog posting responding to Verizon's that we talked about last week, so we'll talk about that. Microsoft Research got some news by apparently telling people not to always use strong passwords. And of course the press picked that up and said, huh, what?

Also just happening is some news about something called "canvas fingerprinting" that we talked about two years ago when the paper came out. Now everyone's suddenly worried about it, so we'll basically debunk that again. We have some miscellaneous notes and updates. And then we'll plow into Zdziarski? Is that how you pronounce his name, Zdziarski?

Leo: I've been mangling it since the story came out. I'm not sure.

Steve: Yeah, we'll just call him Jonathan. We know his name is Jonathan.

Leo: Or by his hacker handle, which apparently, according to Ars Technica, is NerveGas.

Steve: No, it's absolutely true. There was a bio intro that I saw where he was declaring himself as using the hacker name, the hacker handle NerveGas. And he's a well-known jailbreaker, iPhone jailbreaker. And there was a little aside that I noted in his paper that said, "and there are no iOS 7 jailbreaks so far."

Leo: Really.

Steve: So bravo Apple. I mean, this is what Apple's been working for and trying not to make any mistakes because that's what they would be at this point. With the architecture they have, in theory, it should be jailbreak-proof. And no one, I mean, that's what he was probably doing, where he got all of this intelligence to put together a presentation was, okay, well, I can't demonstrate jailbreaking. So I'm going to talk about things that I found that don't make sense. And so we'll talk about that.

Leo: Wow. Very interesting. As always, lots of things to talk about.

Steve: And the other thing I liked about this is, even for people who aren't on the iOS platform in any way, shape, or form and don't care, this is also all kinds of sort of like teachable moment here because the fundamental principles, and you guys mentioned it on MacBreak Weekly, the tension, which is one of our ongoing themes on this podcast, between convenience and ease of use and security. Arguably, and this is what I will argue when we look at Jonathan's points, I'm sure - and he does raise points that are good. Apple does need, now that he's said this, to respond. But it can also be explained largely, I think not completely - and, see, I guess my feeling is neither of them are going to disagree with anything I have to say. Both sides will say, yup, that's probably the case.

Leo: All right. All right, Steve Gibson. Let's, I guess, kick things off with the security news.

Steve: Yeah. So I was somewhere, I think waiting for some friends at a restaurant or something, where I saw someone tweet the news. And so I was unable to thank them, but they have my thanks for noting that in a recent Chrome update the "check for certificate revocation option" did in fact disappear, as we were expecting it to do. It took some time to move through whatever vetting channel and chain they have. But sure enough, I fired my Chrome up this morning, I went to advanced settings, scrolled down to the HTTPS/SSL big button, under which there used to be an option for turning on or off checking for certificate revocation. And it is no more.

So this is them continuing to say, oh, this is confusing for people, and it doesn't really work anyway, so we're going to remove it. And really it's the case, as we know from our previous coverage of revocation checking, Chrome on Windows and Mac was already getting the benefit of those platforms' robust certificate revocation checking, and they've preempted the browsers running on them, so it didn't matter whether that was set or not, you were protected. So you really didn't need it there. And on Android and iOS, those platforms don't provide the information about revocation up in through the API to the browsers running on the platform. So no revocation checking is being done. And again, the checkbox would have no purpose. You could turn it on, but it wouldn't help you on those platforms. And turning it off wouldn't turn off revocation on Windows and Mac. So I can see the logic of them removing it.

I think the reason everything is different for Firefox is that Mozilla brings its own certificate stack with it. And I'm suspecting that one of the reasons that Google forked the OpenSSL project to create the so-called BoringSSL, as we covered a few weeks ago, is that they're thinking, okay, we need to solve this problem, so we need our own. And so I wouldn't be at all surprised if at some point we do see actual revocation checking in Chrome because they've learned that, in the case of iOS and Android, there's none going on. So they need to provide that themselves if the platforms underneath it don't. And as I understand it, Android is a long way away from having it. iOS has it for EV certs. But I don't understand why it's not being protected, not protecting all of them. Arguably, I think it should be. But so we were expecting that to happen in Chrome, and it has.

Firefox moved from 30 to 31 in its major version. And there's really nothing to write home about. I looked through all the changes. It's just sort of moving forward on all the standards fronts, implementing additional, less-used verbs and semantics for some of the web standards. JavaScript gets a few more things. The math package gets a few more. They've defined constants for greatest integer and smallest integer, those sorts of things. So just sort of a nice thing. When I went to Help in Firefox, it was already open. It always uses that now as sort of the trigger because I don't start and stop Firefox. My Firefox is like my desktop. It's my interface to the world. So it just sits here on. And it's when I go into Help that it kind of wakes up the updater and goes, oh, yeah, we got something here. And then it downloads 16.5MB, and then I restart Firefox, and I'm up and going. So anyone who is a Firefox user, next time you start it I imagine it will update. Or if it's already running, just go to Help and About.

I always try to, as I've said before, or just recently, I'm trying to put some sort of an interesting diagram, sort of like a Diagram of the Week, in the show notes for Security Now!. Since I am publishing them and tweeting the URL, which only varies week by week by the number, so anyone can guess what it is just looking at one of them. In this case, the diagram is from sort of the rebuttal response to Verizon's commentary about what's going on with them and Netflix. Last week's diagram was the one we talked about, which I liked mostly because it just sort of showed us, kind of gave us an inside view into the Verizon network and demonstrated that, sure enough, they were saying what everyone agrees, and that is that internally their network has no problem. Yeah, you found it. But it is the peering to the carriers that are delivering Netflix content where the congestion exists.

Leo: It's the use of color psychology. They've got green everywhere except one arrow, which is the Netflix transit providers to Verizon. They're maxed out. It's their fault. And if you just looked at the graph, you'd assume, hey, well, it must be the red arrow that's the problem because everything else is green.

Steve: Well, and we've understood that it's the peering. And that's what I liked about the diagram. And in fact we get another level of refinement of that in Level 3's response.

Leo: I mean, I understand that Level 3 has another, you know, they're promoting their position. But...

Steve: Well, yeah. And also the Level 3 guy was a little peevish. For example, nowhere does Verizon name any of these transit providers. Yet Level 3 believes that they're being blamed. I mean, it says...

Leo: Well, they're one of them; right? I mean...

Steve: But they're not named. And so why does Verizon show this red bar? And why do they blame Level 3 and the other network operators contracted by Netflix? So, yeah. What I liked about it was the level of detail that we got, yes, in that diagram. And so this is reading Mark Taylor, who's the VP of content and media for Level 3. He said: "As I explained in my last blog post, the bit that is congested is the place where the Level 3 and Verizon" - now, and this is Level 3 speaking, so they're only speaking of their interconnect. So the "Level 3 and Verizon networks interconnect. Level 3's network interconnects with Verizon's in 10 cities." So again, we need to sort of remember that the Internet is an interconnected network of private networks. And it's these peering, as is the term, where the networks peer with each other, that's where these major networks, like, cross their traffic from one to the other. So Mark says:

"Level 3's network interconnects with Verizon's in 10 cities: three in Europe and seven cities in the United States. The aggregate utilization of those interconnections [so the peering points] in Europe" - and he just takes a date out of, I mean, like a recent date - "on July 8 was 18% [and he notes] a region where Verizon does NOT sell broadband to its customers. The utilization of those interconnections in the United States, where Verizon sells broadband to its customers and sees Level 3 and online video providers such as Netflix as" - now, this is interesting - "as competitors to its own CDN [content delivery network] and pay TV businesses" - so, you know, that's what - I can't comment on that, but that's what Mark is alleging. But he says: "...was about 100%." No one disputes that.

"And to be more specific, as Mr. Young [the Verizon guy] pointed out, that was 100% utilization in the direction of flow from the Level 3 network" - which is to say from the Netflix side - "to the Verizon network." And then Mark continues: "So let's look at what that means in one of those locations, the one Verizon picked in its diagram: Los Angeles. All of the Verizon FiOS customers in Southern California likely get" - and I wish I was one of them, but I can't get FIOS. I only wish it because my alternative is Cox, which is worse - "likely get some of their content through this interconnect location. It is in a single building and boils down to a router Level 3 owns, a router Verizon owns, and four 10Gb Ethernet ports on each router. A small cable runs between each of those ports to connect them together.

Leo: CAT 6a cable.

Steve: Actually, at 10Gb, it's probably fiber.

Leo: No, we use - we get 10Gb on our CAT 6 cable.

Steve: Oh, okay, yeah. So picture this is a building where both Level 3, like Level 3 has a datacenter, and Verizon. And I can't remember the name of the building, but it's a famous building in L.A. that I used to talk to my Verio guys about because, like...

Leo: Probably has no windows. That's how you'll spot it.

Steve: And it's like One Wilshire. Oh, I think that's it. I think it's One Wilshire. And it's like, yeah, it's like this is where it's all happening.

Leo: It's a monolith, yeah.

Steve: Yeah. And so all this cabling is coming up subterranean, up from below, like to different floors where Level 3 has a facility and Verizon has a facility. Then, because they have to connect to each other, I mean, that's what the Internet is, there will be - they'll say, okay. I don't know who provides the actual wire. Maybe they flip for it, or maybe one provides two pair and the other provides the other pair. But they agree, we're going to interconnect our networks. And so this interconnection is literally four 10Gb Ethernet ports on one - and these are not like home routers. These are big iron.

Leo: EGB router, yeah.

Steve: Yeah, these are big monster, you know, you use a forklift or some burly guys to hold them while you screw them into the frame. And they may not even run on AC. Often they run on 48 volts DC because there's, like, there's a room of batteries somewhere, and so they don't bother converting the batteries, inverting them into AC just to have them converted back to DC again. They just provide it. And this is old telco technology, which is sort of still hanging on.

So it's four ports that then probably snake down through some conduit to the next floor up or down, and they plug into the other agency's router. And that's the interconnect. And that's the problem is that, on one side, we've got Netflix. And whether Netflix themselves are on Level 3 - I actually kind of think they're on Cogent because Cogent is a rock-bottom bandwidth provider.

Leo: I think they're on a mix. I think they are on Level 3, among others. It's a mix.

Steve: Yeah. And so the point is that this means that traffic trying to go across that boundary from the Level 3 network to the Verizon network has an absolute cap of 40Gb. That's all that can go.

Leo: There's four of them.

Steve: And, yeah, exactly four 10Gb cables.

Leo: The ones on the left there.

Steve: There they are. And wouldn't you know it, Verizon forgot to paint those red.

Leo: They look fine to me.

Steve: So continuing, Mark says: "Verizon has confirmed that everything between that router in their network and their subscribers is uncongested, in fact has plenty of capacity sitting there waiting to be used."

Leo: See, it's green, it's green, it's green.

Steve: And in truth, that's really only true because of diffusion. I mean, Verizon is also peering with Sprint and AT&T and all these other providers. And so in general, traffic is diffuse. It kind of wanders around. And this is why this has traditionally worked. And what's happened, of course, it's the phenomenon that Netflix is offering this service that suddenly we no longer have diffusion where the normal daily saturation of these links never approaches 100%. Because of this Netflix phenomenon, and as Brett famously said on his podcast, and yes, everybody, I heard myself use the word "famously," as Brett famously said, the first thing his customers ask for is can we get Netflix. It's what people want. And the problem is - and again, we don't know the politics of this. We don't know what's happening behind the scenes. But Brett did tell us that he had asked Netflix if he could cache, and they said no. And I don't know why. Maybe he's too small. Maybe for DRM purposes.

Leo: That's what I suspect. I'm sure Hollywood doesn't want multiple copies of their movies on servers.

Steve: Right. And as we discussed, the problem is the Netflix model strains the traditional diffusion mode that has allowed the Internet to work. If it were possible for Verizon to only have one copy of what people are watching right now, like the "Game of Thrones" things, for example, then not every single person who wanted to watch it would have to pull redundant bandwidth across that point. They'd get it once, and then Verizon would just be able to redistribute it. So maybe there's contracts being negotiated. We don't know. It will be interesting to see how this sorts out.

Anyway, so Mark says that Verizon confirms they have plenty of capacity throughout the rest of their network. And he said: "Above, I confirmed exactly the same thing for the Level 3 network. So in fact we could fix this congestion in about five minutes, simply by connecting up more 10Gb ports on those routers. Simple. Something we've been asking Verizon to do for many, many months, and something other providers regularly do in similar circumstances. But Verizon has refused," says Mark. "So Verizon, not Level 3 or Netflix, causes the congestion. Why is that? Maybe" - now, this is where it gets a little stupid, but I'll continue reading. "Maybe they can't afford a new port card..."

Leo: He's being sarcastic.

Steve: Yes, "because they've run out, even though these cards are very cheap, just a few thousand dollars for each 10Gb card which could support 5,000 streams or more. If that's the case, we'll buy one for them. Maybe they can't afford the small piece of cable between our two ports. If that's the case, we'll provide it. Heck, we'll even install it. But here's the other interesting thing" - and then I'm still reading from Mark, and then this is the end. "But here's the other interesting thing also shown in the Verizon diagram. This congestion only takes place between Verizon and network providers chosen by Netflix." Now, that's not fair, actually, because the congestion only takes place because of the content provided by Netflix.

So anyway, he says: "The providers that Netflix does not use do not experience the same problem." Right, because they don't have the bandwidth burden. And he says: "Why is that? Could it be that Verizon does not want its customers to actually use the high-speed services it sells to them? Could it be that Verizon wants to extract a pound of flesh from its competitors, using the monopoly it has over the only connection to its end-users to raise its competitors' costs?" Well, who knows.

And finishing, I would just say it'd be fun to be a fly on the wall. I love something that you said in the last couple days, Leo. I don't remember, I was watching, and you were commenting about this, and you just said, you know, this is happening behind closed doors. Right now there's a he said/he said battle going back and forth. And meanwhile the user is losing because this one guy who's got 75Mb FIOS drop at home is buffering and unable to get 300Kb in order to watch his Netflix video as he would like to because so many people are trying to do that, and that pinch point is congesting. So, I mean, but it's hard to defend the idea that you couldn't simply double the capacity by adding four more...

Leo: Other ISPs do. That's the point.

Steve: Four more interconnects, and then you're not going to have a problem.

Leo: I mean, that's the thing that's compelling to me. Cablevision doesn't have this problem. There are other ISPs that do. And why is Verizon and AT&T, why are these, why are the Big Five the ones having the problems? And as Level 3 pointed out in an earlier blog post on this topic, these are the five that have no competition.

Steve: Yes, I was just going to say, you said this when I was listening to you, and it is absolutely the case. It's monopoly power. It's that they have the ability to say no. This guy has his 75Mb that he is loving, which I'm envious of, except that there's one service he can't get the way he wants to.

Leo: I'd also point out that you also - if you look at the Verizon diagram, everybody else is fine, it's just Netflix. That's the red bar is Netflix. But Netflix is not more than half of the Internet traffic. It's at best 40% in primetime. That means 60% is coming from those interconnects. They're not congested. More traffic is coming over them than is coming from Netflix. It's not like Netflix is all of a sudden using up all the bandwidth. There's plenty of bandwidth. Just put in another router. So it's not like...

Steve: Yeah, actually, if they've got free ports, just string some more wire or some fiber.

Leo: I don't know if they actually have free ports. But that diagram looks like they got, wait, they got all these four ports just sitting there. We're ready to connect. But all they'd have to do is put in a few more routers.

Steve: Okay. So then the idea is that Verizon's position is we should be paid to carry this traffic which is benefiting Netflix. And by refusing to increase our bandwidth, their feeling is they're keeping Netflix's customers, they're making Netflix's customers unhappy, and that puts pressure on Netflix to give money to Verizon for the privilege of this transit.

Leo: They're basically holding it...

Steve: Hostage.

Leo: Yeah. It looks like that. But we don't know for sure. So you're, you see, you're more charitable than I am. You give both sides equal weight and say, well, maybe it's more complicated than that. That doesn't strike me as being - you're very fair. I'm not going to be that fair. I think it's pretty clear who the bad guy is here.

Steve: Yeah, again, without - I just...

Leo: We don't know.

Steve: The evidence. I'm interested in the technology. I love the picture Level 3 painted of we got a router; they got a router. There's four connects right now. That's the problem. That's why our traffic...

Leo: And it really is true, Level 3 has a significant investment in Verizon being at fault here. Because for all we know, although Verizon hasn't given us any evidence, it's Level 3 that's unwilling to buy another router.

Steve: Exactly. All we know is there's a point, due to the unusual fact that an incredible amount of bandwidth wants to come from one location, it's not diffuse. It wants to come from one location. The Internet just wasn't built for that.

Leo: Well, but it is diffuse. That's what I'm saying is they use Cogent. They use Level 3. They even have - Netflix offers its own CDN that they could put into Verizon. They have a box they could put into Verizon's network. Brett talked about that. He didn't like that box, but they do offer these solutions.

Steve: Yeah. Yeah, again, when it goes above the bandwidth, suddenly a whole bunch of unknowns enter. And again, this was...

Leo: I think what Level 3's kind of saying is, look, this is a common thing that happens all the time on the Internet. And by kind of universal unwritten agreement, when there's congestion, you add some hardware. And they're making the point, it ain't that expensive. There's no reason not to eliminate the congestion.

Steve: Except there's this issue of symmetry. And that's something, as I mentioned last week, I've never had anyone explain to me why it's important. But for decades I know that that's an issue, somehow, this notion of generally equal transit in both directions. And so that's something that the Netflix phenomenon breaks because the traffic is not symmetric.

Leo: Well, yeah, but...

Steve: It's all wanting to flow in one direction.

Leo: It's the same thing at a candy...

Steve: All I'm saying it...

Leo: It's the same thing at a candy store. All the Hershey bars go in. But what Verizon's selling is this data.

Steve: Access, in this case.

Leo: That's what they're selling to their customers. So...

Steve: I just think - my point is this is a new thing. This is a new phenomenon for the Internet, the idea that 40%, I mean, that's a huge number of, like, total Internet traffic is one provider? That's crazy. That's completely insane. So it's breaking assumptions. And I think what we're seeing is we're seeing the strain of these assumptions. And maybe it's just large corporations taking a long time to act. Verizon needs to figure out what they're going to do. We'll see how it plays out.

Leo: I think they've figured it out.

Steve: So, okay. This is the weirdest research paper. Microsoft Research published this paper. And it got picked up, and I started getting tweets about it and saw people covering it. The Guardian, their headline was "Microsoft Recommends Against Always Using Strong Passwords." Or sometimes it was just "Microsoft: Stop Using Strong Passwords Everywhere." And people were like, what? What? And so I sort of assumed, actually, the title of the paper is really all you need to know. If you really read the paper's title carefully, it tells you. And I didn't put it down in my notes, but I have it right here. I'm sure I can grab it. There it is.

So the Microsoft Research paper is titled "Password portfolios and the finite effort user: Sustainably managing large numbers of accounts." And, I mean, this is a heavy-duty academic paper. It's 15 pages, and it is steeped in calculus and asymptotes and curves and things. And what it reduces to is nothing that we don't know, which is the absolute worst case that is the hardest for users is the joint recommendation that they never use the same password on more than one site, and their passwords are really complex and impossible for anyone to guess, no matter how much anyone knows about them or their lives. So, I mean, that's the collision of advice.

And of course many things have happened. I'm furiously working away on SQRL as a complete solution to this problem. Meanwhile, password managers have been created to manage this nightmare for us. So Microsoft's paper assumes no existence of aid of any kind, and just sort of assumes that, okay, in today's world is it really true that all passwords are equally valuable? And so what this multiple-page academic research does is it says, no, some sites are not that important. Your bank, yes. You really want to use a strong password there. Random "create a login so you can add a comment to a blog," no. That's not important. So there the policy could be softened, and you could probably get away with using your common "I just created an account so I could comment on a blog" password. Again, they do calculus to show us what we already pretty much knew, was not all websites have equally strong value for our having robust authentication, so on those sites we could soften our rules.

On the other hand, when we do in fact have really good login automation, as we do with LastPass and similar password managers, might as well just use 20 random characters we're never going to be able to memorize because it's doing that for us. So anyway, that's what all that was in the news with Microsoft saying, eh, stop using hard passwords everywhere. And it has lots of calculus to back that up. It's like, yeah, okay.

And just as we were talking last week, Oracle dropped an update to Java. I only mention it just for the sake of people who are using Java. It's Update 65 of Java 7. And I love Brian Krebs. The title of his blog post about this was "Java Update: Patch It or Pitch It." And so he said: "Oracle today released a security update for its Java platform that addresses at least 20 vulnerabilities in the software. Collectively, the bugs fixed in this update earned Oracle's 'critical' rating, meaning they can be exploited over a network without the need for a username and password. In short, if you have Java installed, it is time to patch it or pitch it."

So, and of course we've long talked about this. The good news is you can disable Java. If you need it on your machine, you can disable it in your browser plugins. You can disable it in Java itself. You can tell NoScript you don't want to run, I mean, more and more preemptive actions are taken requiring people to really want to run a Java plugin. It wasn't that many years ago that they just ran all the time by themselves. And we were talking every week about disasters that this was causing for people. It's really not the case anymore.

I'm really sad for the horrible reputation that Java has, only because we're more in a cross-platform environment today than ever, and it would be nice to have a great, strong, reasonably fast, cross-platform tool. I run across things all the time that are available for Mac and for Windows that are in Java, written in Java. You have to have Java installed if you want to run it anywhere because that's where they've got platform independence. And I would love to be using it because it's a powerful platform. It's all I would need to be doing, like, cross-platform network stuff. But I just - I can't because it's hurt itself so much over the years.

Also in the news, a bunch of people picked up on this. This is something we talked about two years ago known as "canvas fingerprinting." Gizmodo covered it with a story saying "What You Need to Know About the Sneakiest New Online Tracking Tool." Well, okay, it's not new. It is sneaky. And it also doesn't work very well. Gizmodo said: "What do the White House and YouPorn have in common? Their websites both use canvas fingerprinting, a newer" - okay, maybe that's true, "er" - "newer form of online tracking designed to make it hard to hide. ProPublica investigated the pervasive shadowing method, developed as an insidious [ooh] alternative to cookies so websites can keep tabs on where their visitors browse online."

Okay, now, and this was actually a consequence of a paper that was just published, I think it was July 1st. And these guys who did the paper said: "By crawling the homepages of the top 100,000 sites, we found that more than 5.5% of the crawled sites include canvas fingerprinting scripts. Although the overwhelming majority (95%) of the scripts belong to a single provider" - and this is top of your "add this to your hosts blocking list," and that's addthis.com, A-D-D-T-H-I-S dotcom. So 95% of the canvas fingerprinting is being done by this company, addthis.com. "We discovered a total of 20 canvas fingerprinting provider domains, active on 5,542" - that's that 5.5% - "of the top 100,000 sites."

Okay. So what is this? Canvas fingerprinting is absolutely clever. "Canvas" is the API term for a platform-independent means of graphics, JavaScript graphics on browsers. And everyone who's been listening to the podcast for a long time will remember when I discovered canvas because I wrote that really cool animation of - like it showed waveforms moving and the polarity switching back and forth and magnetic heads and how - basically it showed how magnetic hard disk recording worked using this animated diagram, all in JavaScript. And that was all canvas. So what these guys are doing is they're rendering objects on this canvas surface, that is, they're, like, printing text. They're doing some WebGL stuff, some 3D stuff. They're drawing circles and rectangles in different shadings and shadows. And basically they're running through at an object level, drawing onto this bitmap. Then they're extracting the contents of the bitmap and hashing it.

So what that does is it creates a fingerprint, which is to say they're finding subtle differences in the drawing of text and objects on the screen and the anti-aliasing that's being done, the grayscaling, the color rounding. Little tiny details will change the actual pixel brightness of specific bits. And they don't care about the details. They draw the standardized thing. Then they suck out the bits of the bitmap that result and hash it to get a fingerprint.

Now, the reason this doesn't work is, despite going to all this effort, is that it says nothing about the user. This doesn't identify the machine, like from other identical systems. And many of us are using the same laptop. Many of us are using the same Mac. Many of us are using, like, the same of a lot of stuff. So it's true that probably Firefox renders, at that level of scrutiny, differently than Chrome; and that Safari renders differently. Or maybe not, if they're both now WebKit-based. Or Opera renders differently. And maybe there's different subtleties of the font that is installed on my machine versus on a Mac. But my machine will also have the same font that many other Windows machines have.

Anyway, it turns out, exactly as our intuition would say, that this, after going to all this work, they get 5.73 bits of entropy, which is to say less than six bits of entropy, and six bits would be 64. So that is to say, looking at all of the users on the Internet, fingerprinting can put each one of those users in one of less than 64 bins. So it doesn't fingerprint people. It fingerprints the machine. And we all, as I said, use a lot of the same machines. So, eh, I guess it's, I mean, as an add-on to existing means of disambiguating for tracking, maybe it's interesting. But it's not pernicious, insidious, and impossible to scrape off your shoe. It's just, yeah, it's one more thing that provides some additional bits of machine-locked identification.

And we've talked about just hashing the headers in your browser is going to, like, the version numbers of all the junk you've got installed typically is sent in the query headers that the browser sends for every single thing it asks for. So there's lots more entropy there. And arguably, that's a much better lock on an individual than this. So anyway, this was a little overblown. Interesting research and, again, people being very clever about trying to figure out how they can track people around the Internet. But I would, however, if you're interested, add addthis.com to whatever blocking you have.

And in pursuing this, I ran across a very interesting EFF project that I hadn't seen before. They call it the Privacy Badger. And so it's at www.eff.org/ and then just all one word, privacybadger, B-A-D-G-E-R. And EFF writes, and this is one of their projects: "How is Privacy Badger different from Disconnect, AdBlock Plus, Ghostery, and other blocking extensions? Privacy Badger was born out of our [the EFF's] desire to be able to recommend a single extension that would automatically analyze and block any tracker or ad that violated the principle of user consent; which would function well without any settings, knowledge, or configuration by the user; which is produced by an organization that is unambiguously working for its users rather than for advertisers; and which uses algorithmic methods to decide what is and is not tracking.

"Although we like Disconnect, Adblock Plus, Ghostery, and similar products - in fact, Privacy Badger is based on the AdBlock Plus code - none of them are exactly what we're looking for. In our testing, all of them required some custom configuration to block non-consensual tracking. Several of these extensions have business models that we weren't entirely comfortable with." For example, and I'm going off-script, Ghostery is a tracker. Even though they show you who's tracking you, they do, too. And they make that clear in their fine print.

And then EFF continues: "And EFF hopes that by developing rigorous algorithmic and policy methods for detecting and preventing non-consensual tracking, we'll produce a codebase that could in fact be adopted by those other extensions, or by mainstream browsers, to give users maximal control over who does and doesn't get to know what they do online." So, frankly, we absolutely know where the EFF stands. If anything, they're over the top on this stuff. And that's where you want someone like this to be. So I would say give Privacy Badger a look, if you're someone who likes to run an experiment with these kinds of add-ons.

I got a kick out of just a - now we're into miscellany. I saw that Dell has begun accepting bitcoin for payment. On their site, on a blog posting, they announced that, in partnership with the clearing house Coinbase, they would now be accepting payment for Dell equipment in bitcoin. And that makes them the largest big name company to do so, so far. And bitcoin does seem to be stabilizing. It's been kind of hovering around just sort of north of $600 and hasn't been doing any of its historical dramatic gyrations, which is what we would expect to see as something matures and settles down.

A number of people tweeted the news that "Particle Fever" is now available on iTunes for $5 to watch, or on Netflix, if you subscribe to Netflix.

Leo: Oh, good.

Steve: Yes, Leo. And if you didn't see it, if you haven't seen it...

Leo: Missed it.

Steve: Oh, my god. Yes, good, good, good, good, good. Please. People will remember that I, walking out of there along with my 76-year-old neighbor, I was already framing what he said when he said, "That's the best $6 I ever spent in my life."

Leo: That's neat.

Steve: That's exactly what I was going to say. It was just - it was an astonishingly great movie. Jenny was traveling, she was out of town in San Francisco. And I texted her, I said, you must, must, must - oh, and I'd seen, I had noted that it was playing up in The City, as we Northern Californians call San Francisco. She broke from her yoga retreat and went to see it and absolutely loved it, too. It's a narrative by the physicists themselves, talking about first beam, when after 15 years of digging tunnels and just incredible - this is the biggest machine mankind has ever built. It's just thrilling on so many levels. And they take you through their stumbling and things that exploded, like when the press had all their cameras rolling, and it's like, oops. And then they had to decide, after they recovered from that, if they wanted to kind of do a sneaky one at night, just so they wouldn't embarrass themselves. Oh, it's just - it's so good.

Leo: I see it's on iTunes, as well. I suspect it's on everywhere you can buy movies at this point, $5.

Steve: Good.

Leo: Rent it for $5, buy it for $15 on iTunes, which is probably a good idea. Own it.

Steve: Yeah.

Leo: Neat.

Steve: And speaking of Netflix, and I thought you'd like this also, Leo, this is something that launched prematurely. It's ABetterQueue.com. And you should bring it up, if you can, ABetterQueue.com. It came up and collapsed under the load. And for about two weeks the guy had a funny picture, it was like a weird Photoshop hybridized shark and goat or something. It was very odd. But it basically got hit by TechCrunch and Gizmodo and everybody and just brought the site down. What this is, is it's a search engine, sort of, or a recommendation engine.

Leo: It's ratings.

Steve: Yes. It takes the very highly regarded Rotten Tomatoes site and adds three sliders and 27 categories. So the Tomato meter you can set between zero and a hundred. So what is the minimum ranking on Rotten Tomatoes that you want to consider, the minimum number of reviews that a film has received, the years of production or release, and you can set a minimum and maximum, so you can, like, only new stuff, only oldies and so forth. And then 27 categories which you can selectively disable. And then it runs through the entire Netflix inventory against the Rotten Tomatoes recommendation to find things for you. And anyway, there's some things I immediately flagged as, oh, great, I didn't - because for me, and I heard you mention this, Leo, whenever I run to Netflix, they don't have what I want. For whatever reason, there's all these movies there, but the one I want, eh, no. And so this sort of solves the problem of wanting to watch something good that Netflix does have. And this ABetterQueue.com will empower people to find them.

Leo: And this is because both Netflix and Rotten Tomatoes have an open API, a public API that anybody can use. This is why APIs are good.

Steve: Yes.

Leo: Everybody benefits.

Steve: This is a total aside, but just worth mentioning again for people because we haven't talked about it for quite a while. I saw a note in my mailbag when I was going through the Q&A a couple weeks ago. And this is not about SpinRite. This is Michelle Roberts, writing about GRC's DNS Benchmark. And she said: "Thank you very much for the DNS Benchmark application." Which of course is freeware that I labored over for many, many months, years ago. She says: "My ISP, although pretty fast, a cable provider," she says, "was still choppy at times. However, since using your application and switching to the top two DNS servers, my Internet experience has been like a dream. I assume I will need to 'benchmark,'" she says, "on occasion, say quarterly, to stay in good shape. Maybe not, since OpenDNS seems to be the top player in my area. Thanks again for a very nice application."

So anyway, I just wanted to remind people that that's there at GRC. It has been downloaded nearly 1.5 million times. And it gets right now about 1,600 downloads a day, every single day. And if you just put "DNS Benchmark" into Google, I own that. I'm the DNS Benchmark because this just did the job. And the nice reminder is that DNS is the first thing our browsers do. And especially now, with sites that are pulling stuff from every direction. I mean, if scripting and assets are coming from 20 different other domains, then when that first page comes in, your browser is madly making DNS queries to get the IP addresses of all those other servers so that it can set up TCP connections and get the stuff that the page needs in order to be complete. And so if your DNS is flaky, everything seems wrong. Just, I mean, it is the first thing that has to be working. That was why I created the benchmark was to bring some awareness to this. So anyway, I just want to remind people that it is there.

And I thought I would also answer a question from a Ben Perrett, or Perrett, who's in Sheffield, England. And he says he's a bit confused, if I can help. He says: "Hi. I'm interested in purchasing SpinRite after listening to Security Now! for the last four months as I have obviously become accustomed to its use, but never needed it until now. Within the last two weeks, my MacBook Pro 1TB hard drive has developed errors, which is causing it to crash more regularly." Okay, now, this is where the world is trying to tell you, take action. If the hard drive is developing errors which is causing your MacBook Pro to, quote, "crash more regularly," you're probably close to the point where it's not going to be regular any longer.

So he says: "I have bought a new 1TB drive to sustain my solid-state urge" - so it sounds like he probably went to an SSD - "but wanted to repair the old drive and bring it back to use." Yeah. "Would it be possible to connect" - he says, I'm sorry, "to correct my old hard drive with SpinRite if I connected it to a Windows machine that had SpinRite installed? Would this method be difficult to accomplish in terms of file systems? Thanks in advance. Keep up the great show."

And the answer is no, not difficult, if you have access to a Windows machine. And obviously you've taken the drive out to put in an SSD. You've got the drive. So you may need an adapter, depending upon what kind of interface your motherboard has - IDE, SATA, and it might be a small connector on the 1TB drive. So you'll need to get the wires connected. But then running SpinRite is trivial. You just run it. And I was telling someone the other day how much pressure I've always been under to add this or that feature to SpinRite. But until v7, which will be a complete ground-up rewrite, I've held to all

SpinRite does is fix your drive. Yes, it could do other things. But that's not what SpinRite does. I know that it could make fabulous pasta. But it's not a pasta maker. It just fixes your drive. So you run it, it's easy to do, and it fixes your drive.

Leo: Please don't use SpinRite to create pasta or any other baked goods. It could not create pasta, Steve. Let's be honest. I don't think there's any way you could write SpinRite...

Steve: No.

Leo: Even in assembly language, the most powerful...

Steve: If I had an - if the pasta machine had an API...

Leo: Ah, yes. There's the key.

Steve: As you have noted, we want open APIs and...

Leo: Everything is possible with an open API.

Steve: And pasta-able.

Leo: I have no reason to think that there aren't pasta makers somewhere with open APIs. I suspect plenty for you to think about. All right, Steve. Let's talk about iOS Security. You did a great two podcasts, as I remember, right, on...

Steve: I think it was three.

Leo: Three, on the original iOS whitepaper.

Steve: Yeah, there was so much to talk about from the whitepaper, yes.

Leo: And at the time you were very impressed with what Apple had done.

Steve: Yes. And that's absolutely the case. And I'm also impressed with what Jonathan has done. This is a great piece of reverse-engineering work. And I guess, as we were saying at the top of the show, I don't think there's any question, with Jonathan having asked some questions and planted some really useful doubts and issues out there, that Apple needs to respond with more than their boilerplate PR. At the same time, the devil is always in the details. And Jonathan's paper, which he originally wrote, I think, in late 2013, late last year was when it was originally published. And he noted that it had been around for a while, but no one had really paid attention to it until he gave a presentation with PowerPoint slides.

And so that's true, and that's a problem because everybody wants sound bites. And the operation of these systems just does not distill to sound bites. Which doesn't prevent headlines from doing that because that's what headlines are. And of course headlines are designed to get readers. And so the popular press has just gone berserk over this.

GigaOM wrote, their headline was "Security Researcher Suggests 600 Million iOS Devices Have Apple-Created Backdoors for Data." ThreatPost.com, their title was "Researcher Identifies Hidden Data Acquisition Services in iOS." MacRumors: "Forensic Expert Questions Covert 'Backdoor' Services Included in iOS by Apple." Ars Technica added a little twist. They said: "Backdoor can be abused by government agents and ex-lovers to gain persistent access." TheRegister.co.uk, never to be outdone for being inflammatory, said: "Hidden network packet sniffer in millions of iPhones, iPads plus host of spying tools." And so it's things like "hidden data acquisition service," "covert backdoor." Those are intention-laden terms that are absolutely unwarranted.

And that's my problem is that these are facilities which are not documented because Apple is a closed system. So no one ever said Apple had to document these. So saying that they're "undocumented" makes them seem like they're secret, and no one is supposed to know about them. Well, these amazing engineers and reverse-engineers and hackers and jailbreakers have done an incredible job of reverse-engineering this technology and figuring out how this works.

So my overall take is that this kind of ruthless analysis is always useful and important for security. All we had going from Apple's whitepaper was sort of like from the clean room, here's all of the amazing stuff that we've built into this. But we've had that for years, and yet the jailbreakers have always found a way in. I noted that in Jonathan's paper he commented that iOS 7 has not been jailbroken. So this is a consequence of the successive refinement that we talked about when we were talking about the iOS security, where Apple was looking at the mistakes they were making and trying to develop more of a defense-in-depth posture where they were creating more layers so that even if something got through, it would have a harder time turning that vulnerability into an exploit and so on.

So but the point is that the work that Jonathan is doing is crucial. I mean, we need someone to do this. This is like somebody auditing the open source of open source code. It's nice that it's open, but until somebody actually attacks it and looks at it, its openness is completely irrelevant until someone goes at it.

Now, on a closed platform like Apple's iOS platforms, it takes somebody prying under the hood in order to figure out what's going on. So this kind of attack is absolutely important. And really it's what we need in order to achieve trust. It's one thing, I mean, it's a good starting point is for Apple to say, with their white coats on, look at all of the cool technology we've got. But what really matters is how it works, like what it does when the bits start flowing through the processor, which is what Jonathan has done.

And my very favorite anecdote about this was Steve Ballmer, back in the summer before Windows XP's release, where he was prancing around the stage and loudly proclaiming that Windows XP was the most secure operating system Microsoft had ever produced. And I said at the time, I think this was pre-podcast, although I feel like I was saying it somewhere, but maybe this was during the podcast, that you can't proclaim security. That's not something that - especially the person who created something cannot proclaim security. That's something that only history can judge.

And so, for example, it's history that is now judging earlier versions of iOS and judging that they were not as secure as Apple was hoping they would be. People were finding wedges, ways in. And so there have been a succession of iOS versions over time. But again, so the fundamental position is we absolutely need people like Jonathan to focus on this and do their worst, do everything that they can. At the same time, I really don't think that anything Jonathan said should take anyone by surprise. We'll parse a lot of this here in the next half hour.

But while I hope Apple will be induced to explain some of their decisions, and it may very well be that they can further tighten things up, in your discussion during MacBreak Weekly about this, Rene said a couple times, I think it was Rene, that maybe it was the case that Apple rushed this a little bit. Or, for example, Rene recognized, who also read the paper, didn't just rely on the slides as I have, that it was maybe for adding features that the enterprise needed, and maybe Apple went too far. So I don't think that Jonathan got anything wrong. In the paper he was very careful with his facts. The slides have the problem that they're not the paper. And they have the problem that they were meant to accompany his presentation.

Leo: Yeah, Rene mentioned that he ended up going to the whitepaper to really understand what it was all about.

Steve: Yeah. And you have to.

Leo: Jonathan's paper, yeah, yeah.

Steve: Yes. And that's the point is essentially the slides take the facts out of context. And if they're taken out of context, then they are inflammatory. I mean, they look much more scary. Whereas, for example, Jonathan - and we'll talk about pairing in a second because pairing is the keys to the kingdom. Pairing is everything. And Jonathan makes that very clear in his paper early on. And the problem is that, because you can't keep restating that, like as a caveat to everything you then say, it's easy to forget that. So, for example, the fact that the pcap daemon is running doesn't matter at all. I mean, it absolutely is superfluous. There's all kinds of stuff running in there, none of which is accessible unless you have a trusted pairing. That's crucial. And Apple understands this.

Now, I completely agree with some of Jonathan's recommendations and some of Rene's, that what would be valuable, because pairing is so important, would be that users get more control of that, that it be made more visible, that we're helped to understand, or maybe able, for example, to easily flush all of the outstanding pairing that exists. I would love to be able to do that because over time you just sort of acquire these things, these pairing relationships. And as Jonathan makes very clear, that absolutely creates security vulnerabilities.

The reason all of this exists is this tension that fundamentally exists between usability and security. Toward the end of his paper, Jonathan recommends a number of things, one of which is tightening things down more securely with passwords, using passwords more because a passcode or a password is a fundamentally powerful tool. It's something that exists in the user's mind which, if managed properly by the system, is very potent as a protection mechanism.

The problem is it's also very burdensome. And I'm sure that Apple is struggling with trying to offer the features that the system does while keeping it secure. I don't see anything here that looks like deliberate surveillance. And nothing that Jonathan has said, I mean, and this is why I push back at the idea of these things being covert. Well, they're Apple's. They're not, I mean, they're covert because it's a closed system. That's what Apple is selling. That's what users are buying is a system which they purchase, and with minimal interaction and requiring them to do very little, it works. And it doesn't get in their way.

And so, for example, during MacBreak you guys brought up the perfect example from Jonathan's presentation, this notion that, once the device has been powered up or rebooted and unlocked, and then relocked, all of this is going on behind the scenes. Well, if we think about it, of course that's the case because we want to be able to receive SMS messages, and we want to be able to receive email. And the user actually has a device which is operating with its screen turned on. Basically this is a UI lock. But big chunks of the operating system are decrypted. I mean, the system has come alive. It's in use.

And so we could say, for example, the same thing, and we have, about full-drive encryption. It's only safe when you're not using it, when the drive is not in use and not decrypted. That's the only time that it's providing you with any protection against its non-use case, not when it's in use because it has to be decrypted in order to be in use. So essentially these phones are - it's a functioning, radio-connected computer while we're using it. I mean, and while it's in our pocket with the screen turned off.

Leo: So really it's a fundamental misunderstanding, not Apple's fault, about what the screen lock does. It's a UI lock, not an encryption or a phone lock. It's just locking down the UI.

Steve: Yeah. Now, Jonathan, in being careful, again, did Apple justice, I thought. On Slide 10 he said: "Encryption in iOS 7 Not Changed Much." And he says: "Once the device is first unlocked after reboot, most of the data-protection encrypted data can be accessed until the device is shut down." So he makes the point "screen lock not equal to encrypted," which is important, as you say, Leo, for users to understand. Well, I mean, for maybe our listeners to understand. And this is really people with absolutely no understanding of security, none of these things that we talk about, are happily using the phone and their iOS devices, and it's pretty much protecting them. I mean, and this is why I was so pleased with the clear intent and the technology which is now in the iPhone 5 and in use in iOS 7 is there's incredible resources Apple has brought to bear to push that tension to the point where users are still not being harassed, yet they really are very well protected.

And Jonathan's next bullet point on Slide 10 says: "The undocumented" - now, okay, undocumented services, which, yes, they are because Apple didn't document them because everyone has reverse-engineered them. But that doesn't make them spooky or secret or anything, just it's a closed system that these guys have pried the lid off of. So "The undocumented services running on every iOS device help make this possible." Right. Then he says: "Your device is almost always at risk of spilling 'all,'" in bold, "data, since it's almost always authenticated even while locked." And my immediate response, the picture that flashed through my head was, yes. And while we're walking upright, we're almost always at risk of falling down.

Leo: That's fair.

Steve: And sure enough, as babies and old people, that's a problem. Babies aren't good at walking. Neither are old people. But pretty much we're okay.

Leo: Everybody else is all right, yeah.

Steve: So I love the work Jonathan did. But words like "spilling all data." "Spilling" implies like it's all going to come tumbling out if the user shakes it wrong, which, you know, and he knows that's far from the truth because he explains in his paper, and if not on these slides, probably in his presentation, the nature of what's necessary to make this happen. And we'll talk about that in a second. But so, again, "Your device is almost always at risk of spilling all data." No. And in fact it's incredibly well protected against ever spilling any data. That's the truth, incredibly well protected against ever spilling any data. There's a chain of things necessary that have to precede the access to any data from the outside of the phone. And the first thing...

Leo: You need physical access to the phone.

Steve: A physical USB connection. Not radio. You can't do it through RF of any kind. You have to have a USB wire connection.

Now, again, because of tradeoffs and also bad implementation, I love - Jonathan made a point of the fact that, for example, users could plug into a power, a third-party power adapter. And iOS 7 now prompts you. There's no longer a - it's not possible to do a prompt-free pairing. Pairing requires user interaction. But you should never have to pair with a power plug. That's ridiculous. And that's the point is it could be an evil power plug with a microprocessor in there, not just a transformer. And it's established a security-critical pairing relationship.

And so again Apple very cleverly, and this was the point that I made during those three podcasts about iOS security, they did everything they could to hide the amazing crypto which is going on under the covers. But there's a problem with them hiding it because then people don't appreciate it. They don't know that when they plug into a power adapter, that they should not trust that device. It's saying, you know, "Trust this device?" No. Don't. You shouldn't need to to get juice. And in fact that could be - and again, that's the tradeoff. So Jonathan made some great points. Also it's absolutely the case that the iOS attack surface should always be made as minimal as possible. And Jonathan found services for which he could find no obvious mating software, which he looked for, to his credit.

So the question is, and this is really what, if Apple generates a technical response - and we should understand, too, they're not under any obligation to. I mean, their system was hacked, and someone has said, "I found stuff I don't understand." And unfortunately, scary words have been hung on these things that we don't understand. Well, that doesn't make them malicious or nefarious or anything. It just means we don't know why they're there, within a system which in every other way has shown it's trying to protect its users. So he raises the point that it looks like some of these things, like the packet capture, could be made unavailable unless the system is in developer mode. It's hard to argue against that. I would agree. Unless maybe something that's not in developer mode uses it.

And again, this notion of packet capture, that sounds like a big deal. But there's packets running all over the place everywhere. And so anything that is in a long chain of stack, of security and protocol at various levels, is able to capture whatever is passing by. So we shouldn't read more into that than is there. And again, it's none of this, nothing is available without first this pairing relationship. Which is the only thing that will convince the phone to trust something else. But I do agree Apple needs to probably - it would be great if Apple would respond with more than their canned PR blurb.

So Jonathan says, in his paper he explained that pairing is the linchpin. And because this is key, I'm just going to read directly from the end of page four and into page five, where he says: "Pairing: The Keys to Do Everything. In order to understand how an attacker could penetrate an iPhone" - now, okay. So as I'm reading this, everyone listening needs to understand this has been written with an adversarial posture. So, for example, at the end of this first paragraph he says: "There are a few frightening things to know about the pairing mechanism." Well, okay. So anyway, but - and I also write how Apple would describe the same thing in a second.

So Jonathan says: "In order to understand how an attacker could penetrate" - penetrate, there's that, again - okay. I'm not going to stop on every one of these words. You understand what I'm saying - "could penetrate an iPhone from the owner's desktop computer, it's important to understand how pairing works. A pairing is a trusted relationship with another device, where the client device is granted privileged, trusted access. In order to have the level of control to download personal data, install applications, or perform other such tasks on an iOS device" - and I'll just stop here and note that's what iTunes does.

We essentially have this weird thing where we have an incredibly sophisticated pocket computer that has grown out from a music player, yet it still sort of has this music player user experience. You know, where you used to connect your iPod to your computer, and iTunes would come up, and you'd manage your music. Well, now you're managing way more than that. And Apple has basically kept the same paradigm of operation so that this phone, awkward as it often is, is tethered to a computer. And we are seeing Apple gradually breaking that strange bond; and, for example, updates can how happen without having to be re-tethered to iTunes and so forth. But that's sort of where this came from. But many of these things that sound spooky are just - it's what iTunes does. I mean, it's what you want, if you're not a bad guy.

So continuing, Jonathan says: "This is done through a very simple protocol, where the desktop and the phone create and exchange a set of keys and certificates. These keys are later used to authenticate and establish an encrypted SSL channel to communicate with the device." Okay, that's all great. "Without the correct keys, the attempted SSL handshake fails, preventing the client from obtaining privileged access. A copy of the keys and certificates are stored in a single file, both on the desktop machine and on the paired mobile device. The pairing file is never deleted from the device except when the user performs a restore or uses Apple's Erase All Content and Settings feature. In other words, every desktop that a phone has been plugged into, especially prior to iOS 7, is given," and he describes it as "a skeleton key to the phone." That's true. But that sounds scary.

"This pairing record allows either the desktop or any client who has copied the file to connect to the subject's mobile device and perform a number of privileged tasks that can access personal data [like iTunes does], install software [like iTunes does], analyze network content, and so on. This one pairing file identifies someone as the owner of the phone, and with this file gives anyone trust and access as the device's owner." Then he says: "There are a few frightening things to know about the pairing mechanism in iOS." And this is where I say this is great to shed light on. We should know, we security-focused people should know and understand this. Apple has deliberately kept my sister and mother from needing to know any of this, yet at the same time provided them with very good security.

Anyway, Jonathan continues: "Pairing happens automatically, without any user interaction, up until iOS 7." So he noted that that was something Apple clearly understood was too easy, and so they made it explicit, requiring a deliberate acknowledgement from the user starting in iOS 7. You could argue it took them too long, but it's there now - "and only takes a few seconds. Pairing can be performed by anyone on the other end of the USB cable. The mobile device must either have no passcode or be unlocked." So again, it's either got to be unlocked or have no passcode. So anyone with any security on their phone can't pair. Again, Apple trying to put up every barrier they can so that users don't even know there are barriers.

"If the user has Require Passcode set to anything other than Immediate, then it is also possible to pair with the device after it is turned off until the lock timer expires. So if the user has a device unlocked to play music, and connects it to an alarm clock or a charger running malicious code, whatever it's connected to can establish a pairing record that can later on be used to gain access to the device, at any point in time, until the device is restored or wiped."

Now, I don't know whether that can happen, again, whether this caveat, up until iOS 7 and the requirement of an explicit acknowledgment, still applies. I hope it does. But that's the other problem with the slide presentation, and even with the paper, is it's not possible for Jonathan to keep reminding everybody only, only if the device has been paired and the user has acknowledged blah blah blah blah blah. So that stuff gets left out. And then things seem vastly scarier, like, again, like data could just be spilling out if you hold the phone upside down. So but these overriding caveats are always there.

So finishing this on Jonathan's part: "While the pairing process itself must take place over USB, at any time after that the phone can be accessed over either USB or WiFi, regardless of whether or not WiFi sync is turned on. This means that an attacker only needs a couple of seconds to" - now he says "pair," but we really mean re-pair. That is - oh, no, I'm sorry, I read that wrong, "needs a couple of seconds to pair with a device," which is true, assuming that they can with all of the caveats that protect from inadvertent pairing, "and can later on access the device to download personal data or wreak other havoc, if they can reach the phone across a network. Additionally, an attacker can easily find the target device on a WiFi network by scanning [for port] TCP:62078 and attempting to authenticate with this pairing record. As the pair validation process is very quick, sweeping a LAN's address space for the correct iOS device generally only takes a short amount of time."

So again, I love that Jonathan is poking in every corner and has done all this experimentation to show us the exact boundary of the security perimeter that Apple has established. But then I wrote how Apple, if they were describing this, how Apple would describe the same thing. They would say: Pairing is an important and security-sensitive system which we have made as strong as possible while attempting to strike a balance between users' absolute intolerance of anything getting in their way while still working to protect them as much as possible. To that end, it is impossible to ever pair wirelessly. Physical USB cable connection must always be present. Since iOS v7, any pairing also requires that the device be unlocked, and the user must acknowledge and accept the pairing request from the physically attached device."

So you can see there were two very different ways to state essentially the same thing. Again, I'm glad to have the aggressive, you know, is this really what we want? And out of this comes some good questions, like with pairing - for really security-conscious people, not all of us, but for those of us who care, who are listening to this podcast, wouldn't you like to be able to examine all of the records in this pairing file in your iOS devices and delete the ones that...

Leo: Revoke them, yeah.

Steve: Yeah, revoke them. Just, it's like, wait a minute. I don't know what that is, so I don't want it to have my private keys, which essentially it has keys that would enable it in the future to re-pair. So and in fact I'm wondering about this whole notion of re-pairing because it's not clear to me why that has to be a non-UI event ever, like from now on, like trust once and forever? No. How about just a little tiny window that pops up...

Leo: Each time. All you have to do is enter your passcode, and you're in.

Steve: Well, you just type, yeah, tap on "Yes, I trust this, I still trust this device." So it's not clear, I mean, it seems to me that making these as persistent as they are, which Jonathan wonderfully highlights, maybe is going too far. Maybe they should...

Leo: And by the way, you can, with a tool that Apple offers for download, turn that on. They just don't have it on by default.

Steve: Right. The system Configurator, yeah. Anyway, I think I've pretty much covered all of this. Oh, and I did - Jonathan just said: "Because of the way WiFi works on these devices, an attacker can take advantage of the device" - oh, okay - "a device's 'known wireless networks' to," as he phrased it, "force a phone to join their network" - okay, well, yeah - "when within range." But he does note that Apple recognizes by SSID, like Linksys or ATT WiFi, not by MAC address. And that's another important thing to note.

And he posits: "It may even be possible for a government agency with privileged access to a cellular carrier's network to connect to the device over cellular, although I cannot verify this due to the carrier's firewalls." So he can't see. This port that he mentions, 62078, is a port where that master lockdownd service runs. And he describes it as a service that all low-level Linux guys will know about, the inetd daemon, which is then - it's sort of like the main listening service that then is able to spawn other services as required. If you don't want to have a web server running all the time, you can have, like, I don't know who doesn't, but you could have the inetd daemon start it for you on the fly when TCP traffic, web traffic comes in and wants to talk to something on port 80, for example.

So finally he says: "Essentially, that tiny little pairing record is the key to downloading, installing, and even manipulating data and applications on the target device." Again, yes, that's iTunes. "That is why I have advised law enforcement agencies to begin seizing desktop machines, so that they can grab a copy of this pairing record in order to unlock the phone. A number of forensic imaging products, including some I've written, and even open source tools are capable of acquiring data from a locked mobile device, so long as the desktop's pairing record has been recovered. The pairing record also contains an escrow keybag so that it can unlock data that is protected by data-protection encryption. This is good news for the 'good' cops, who do crazy things like get warrants. It's very bad for anyone who is targeted by spy agencies or malicious hackers looking to snoop on their data."

But again, absolutely to do that there must have been a physical connection with the machine. So again, the takeaway, valuable, that Jonathan provides is that the machines we routinely pair with are sources of vulnerability for access to our phones. So that's definitely something that we want to keep in mind.

And I really think I've covered - I'm scrolling through the rest of the slides that I had here in my notes, but although we sort of did some of this out of sequence, I think I've pretty much covered it. So the summary slide says: "Apple is dishing out a lot of data behind our backs." And I think now people have a better way of understanding that statement on the summary slide. Much like the data spilling, bursting forth from the phone. Well, no. Jonathan's position is: "It's a violation of the customer's trust and privacy to bypass backup encryption." He says: "There's no valid excuse to leak personal data or allow packet sniffing without the user's knowledge and permission."

And again, this is really twisting the nature of the security Apple's provided and the way they provided it. Could they make it way more difficult? Yes. But then it would affect every single user of the system. And so really the way to think of this I think, is if we reduce the vulnerability to what Jonathan himself, to his credit, lays out as the linchpin, which is pairing, and requires a physical connection. And we take away from this that, in that being done, key persistent access to the device in the future is granted. That's really good to know. Does that represent a violation of the customer's trust and privacy? No. That's the compromise Apple made to otherwise have the system utterly locked down, yet users never even know. Just email is flowing, and text is coming in, and they're busy swiping left and right to decide whether they like the way this guy looks or not. And, I mean, they're having a great time.

Meanwhile, Apple has done, I think, everything they can to protect people from themselves. And there is no indication that Apple does have anything you could allege even as a secret backdoor or other access. It's true you could exploit that aspect, the pairing, the nature of long-term host pairing trust. You could exploit that to get access to people's devices. But it's not clear unless you - I guess I would say I'd love to have a setting which expired those pairing records or allowed me to audit, well, actually, expired them for normal people, and also, for those of us who wanted to drill down deeper, allowed us to audit and revoke them as we see necessary, for those of us who want more security.

And just so you get a sense of this, Jonathan finishes, saying: "Much of this data simply should never come off the phone, even during a backup. Apple has added many conveniences for enterprises that make tasty attack points for .gov and criminals." Although he says: "Overall, the otherwise great security of iOS has been compromised, by Apple, by design." And I would take issue with that. I would say Apple has created as burden-free a secure system as is possible today, with the small exception that maybe they could raise the bar and break this pairing, give users - require per pairing acknowledgment, even on future re-pairings, which Jonathan alleges is not there now. So overall, great work. I understand the adversarial posture that is here. And I think we now understand that it's a matter of tradeoffs.

Leo: I guess Apple's point of view would be, look, if you really care this much, get the Apple Configurator. It's free. It's for enterprise. Because obviously enterprise cares a lot about this. So they do make a tool available. And configure it and secure yourself. It's also good for people to understand that, unless you power the thing off, stuff's not encrypted. But again - and so the issue would be..

Steve: Then it's not working. It's not working. If the file system is not entirely encrypted...

Leo: Yeah, right. It has to be powered - I understand. It's the same thing with TrueCrypt or anything else. Once you log in, everything's unencrypted. And there's an attack for law enforcement and others there, too. But they have to get physical access to the machine. I guess, you know, remember, if you've been arrested, they can do it.

Steve: Yes. And remember, too, that the phone is only as secure as your passcode. That is, we know that Apple can brute-force crack phones, that there's a long waiting list.

Leo: Takes them a while to do it.

Steve: Yes.

Leo: And Zdziarski talks about this, by the way, in this.

Steve: Yes. And in fact, Apple, to their credit, they added PBKDF to slow that down. They made the phone as crack-proof as they can. That was why we spent three podcasts talking about how impressed I was by this. So they can, if they're given a phone, they can apply their tools. Takes a while. They can get in. So still, users are protected by having a really strong passcode, ultimately. And by, if they really care, exactly, and I'm glad you brought that up again, turn it off. Not just blank the screen. Power it down, if you want your phone not in radio contact, and to have flushed even those working decryption keys for the file system.

Leo: Yeah, I think this is good. I mean, you have the expertise to look at this and parse it. And that's something I lacked. And so I'm glad to hear your analysis of this.

Steve: Now, and we should mention to our listeners that Rene said he was going to post a link to an article he wrote about the PC or the Configurator tool. I'm going to go track that down because I want that. I want that granularity of control.

Leo: Yeah. It's on iMore.com.

Steve: Great.

Leo: Let me see if I can find it just by searching iMore for "Configurator." Well, there's some older articles. I'm sure you could find it.

Steve: Or you probably just google "iOS Configurator.

Leo: iMore, and it's called the "Apple Configurator." And you do need a Mac for it. It's in the Mac Store for free. I bet you our audience could figure it out. But it'd be nice to have a roadmap for others.

Steve: I'm going to get it.

Leo: Yeah. Yeah, play with it. Maybe we could talk about that next time.

Steve: Yeah.

Leo: You want to do Q&A next week?

Steve: Absolutely. We've got them piling up. So we'll handle them. We'll do news and Q&A.

Leo: If you have a question for Steve, GRC.com/feedback, that's the feedback form. And that's the only way to do it. Don't email him. He won't see it. But while you're there, check out SpinRite, world's best hard drive maintenance and recovery utility, and all the free stuff Steve offers at GRC.com in a lot of different areas. I mean, it's really a very rich site, getting richer all the time. And it's free and easy to use: GRC.com. He also has 16Kb versions of the audio of this show for the bandwidth-impaired. He has transcripts written by a human, an actual human being, a day or so after the show. We have 64K MP3s, as well as hi-def video, standard-def video of the show, at our site, TWiT.tv/sn. Or subscribe using Stitcher or the TWiT apps or iTunes and all of that. Just don't pair your phone to your iTunes, and then you'll be sorry. You don't need to do that anymore. There's really no occasion that you need to do that. Used to be to activate an iPhone you had to hook it up to a machine. You don't even have to do that anymore.

Steve: Although it - oh, that's true because it'll now back up to the cloud. So as long as it has WiFi, it'll do cloud backup. You don't need to pair it in order to back it up.

Leo: Right, exactly. Or to get podcasts, for that matter. We do Security Now! every Tuesday, 1:00 p.m. Pacific, 4:00 p.m. Eastern time. That's 2000 UTC at TWiT.tv. Do tune in live and watch if you can. If not, on-demand audio and video available, as I've told you, any time, of any of our shows. We'll be back here next Tuesday. Thank you, Steve.

Steve: Thanks, Leo.