Pegasus & Trident

Description: This week, Leo and I catch up with the past week's news including the Dropbox and Opera incidents; a Chinese certificate authority who could not have been more irresponsible; the changing Facebook and WhatsApp information sharing arrangement; the FBI's disclosure of election site hacking; Tavis Ormandy's Dashlane and 1Password vulnerability disclosures, the threat of autonomous weapon systems; WiFi router radio wave spying; and the details behind Pegasus and Trident, the emergency Apple iOS v9.3.5 patch.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-575.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-575-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. And we are going to talk about the latest security news, including spend some time with that big iOS security flaw, the one that's been around for years. How does it work? It's pretty amazing. Coming up next, as always, on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 575, recorded on August 30th, 2016: Pegasus & Trident.

It's time for Security Now!, the show where we cover your security and privacy online with the Explainer in Chief himself, Mr. Steven "Tiberius" Gibson. And today, oh, I'm excited. I'm really excited for Security Now!. You notice, by the way, those of you who watch on video - and I know that's a scant percentage of the total. Most people just listen, and rightly so, since Steve and I are not exactly beauty queens.

Steve Gibson: Talking heads, yes.

Leo: At best talking heads - that we decided not to move. We're in exactly the same studio. Nothing has changed. I thought, why move? When you've got perfection...

Steve: False alarm. False alarm.

Leo: No. This is the new studio, and it looks pretty much indistinguishable.

Steve: Yeah. In fact, clearly, if this had just changed, and everyone was used to seeing where you had been for the last five years...

Leo: If I hadn't said anything, yeah.

Steve: ...nobody would pick up on it.

Leo: All they'd say is, oh, I like the new clock. That's it. I'll tell you some things I like that you can't see. But first of all, the air conditioning works in here. Very nice.

Steve: And was it Lisa who was hot in her office because her AC was...

Leo: Lisa is so hot, even when she's cold. She's hot. No, yes, she couldn't get it right. It would either be too hot or too - the building we were in was 120 years old, and it was funky. And the landlord, the previous owner had not maintained it at all. So it was just falling apart. We now have a parking lot. We now have AC that works. We have - it's just functional in so many more ways, and it's half the rent, so we're saving lots every month. The downside is it doesn't have that quirky charm.

Steve: Quirky charm.

Leo: If you've ever lived in an old house...

Steve: No, I agree.

Leo: ...it's the same thing.

Steve: In fact, your first studio, frankly, I preferred it. I mean, it was really wonderful. This one looks like a studio. It looks much more like a TV studio which is, like, you know, instead of an environment you had a really - you just sort of had an environment.

Leo: And that was always my philosophy, both at The Cottage and The Brick House, was a real sense of place, of a clubhouse. And this is just a TV studio. But that's fine. And frankly, all it really needs to be is a radio studio. And by the way, get ready, because the next incarnation will be me on a boat, and there'll be no video at all. It'll just be me and you, and we'll be talking on a boat, and we'll be in our 70s, and we can go - it'll be great. It'll be so much fun. I'll be, "I'm out here in the middle of the Atlantic, talking to Steve." Now, this has been a good week, I think, for a security professional.

Steve: Oh, yes.

Leo: What are we talking about this week?

Steve: So we have lots of news. I was going to try to do a Q&A this week, but on Thursday afternoon I tweeted to all of my Twitter followers that Apple had just released an emergency iOS update, bringing iOS to 9.3.5. And of course like a week before we had gone from 3.3 to 3.4. Now we were at 3.5. And so I just wanted to let everybody know that that was there. My own, I have, I don't know, maybe seven or eight iOS devices, only today have they begun to offer me, proactively offer me an update. So it's interesting that this, as I had mentioned before, I'm not seeing these devices suggesting that they be upgraded. If you go into...

Leo: You can check and see it.

Steve: Yes, and then it immediately knows it.

Leo: Yeah, but it doesn't tell you. Yeah, that's how Apple does it, which is weird. They don't bug you.

Steve: Yeah. And, I mean, it's certainly a tradeoff. Now they are saying, oh, an update is available. Or there's the little red "1" on the badge of the control panel app. And it's like, huh? And then you go in there, and it's like, oh, yeah, okay, fine. And it does its upgrade.

Anyway, so we're going to talk - the main topic after we catch up with news is the podcast is titled "Pegasus & Trident." Trident is sort of a play on words because there were three, this "tri," vulnerabilities which were patched after they were found being used in the wild. And we have a lot to talk about that. We have, from the Lookout Security guys who did the technical analysis, what they found is really fascinating.

But what is chilling is that they found evidence in the code that this works all the way back to the iPhone 4s. So it has been probably - and these vulnerabilities only last week got fixed. And only due to really a coincidence of the fact that somebody who was already on the lookout was prepared for this eventuality and didn't act on basically a phishing attempt to get him to click a link. Instead, he forwarded the links to Citizen Lab, who looked at it for a couple days, and then they got the Lookout Security guys involved.

Anyway, it's a really interesting story. And, for example, the links are set to only function once, and that's specifically to prevent the exploit from being easily reverse-engineered. So it's something, you have to suspect it and capture it the one and only time it will be primed in order to make it work. So really interesting.

But we had a ton more news. Dropbox and Opera both handled security incidents responsibly, while a Chinese certificate authority could not have been more irresponsible. And we have to talk about whether it's not time to start pulling back from this trust everyone, or trust all CAs, philosophy which we have at the moment. And I know that some of our listeners have experimented with removing questionable-looking root certificates from their certificate root store on their various devices, specifically for this reason.

Then of course we have the news of WhatsApp and Facebook changing their information-

sharing agreement, and a little quick something that people can do if they want to prevent that from happening to them. The FBI discloses two election sites, federal election sites hacked through SQL vulnerabilities. Tavis, our friend at Google, tweets about Dashlane and 1Password upcoming disclosures.

And then we've got two crazy things from the fringe that were just so, well, kind of fun. One is an AI professor at UC Berkeley who writes about autonomous, the forthcoming, the upcoming, the soon-to-be-seen autonomous weapons systems that are on the way. And then a loony-tune, like we need a reality check on this one, the concept of using WiFi radio waves to spy on you, literally because you absorb and reflect radio. So what can that tell you about moving around the room? And, I mean, anyway, that appeared in The Atlantic.

We've got one erratum, a little bit of miscellany, and then we'll do our deep dive into Pegasus, which is, by the way, the name of the exploit kit which is offered, essentially under license, from this Israeli "security firm," unquote, security firm, insecurity firm. And of course Trident with the three exploits.

Leo: Excellent. Okay, Steve. Let's dig in.

Steve: So Dropbox sent its long-time customers...

Leo: Yeah, I got that, yeah.

Steve: Yup. My email said: "Hi, Steve. We're reaching out to let you know that if you haven't updated your Dropbox password since mid-2012, you'll be prompted to update it the next time you sign in." And as it happens, I had, so I guess they didn't tie their mail to last password update. Maybe they didn't have that metadata in their database. Of course because I have messed with LastPass, and my Dropbox password is something that no human could remember.

So they said: "This is purely a preventative measure, and we're sorry for the inconvenience." And I dug in a little bit deeper and found what they were sharing. They said: "If you signed up for a Dropbox prior to mid-2012 and haven't changed your password since, you'll be prompted to update it the next time you sign in. We're doing this purely as a preventative measure, and there is no indication that your account has been improperly accessed. We're sorry for the inconvenience. Our security teams are always watching out for new threats to our users.

"As part of these ongoing efforts, we learned about an old set of Dropbox user credentials" - and then they have in parens - "(email addresses plus hashed and salted passwords) that we believe was obtained in 2012. Our analysis suggests that the credentials relate on an incident we disclosed around that time. Based on our threat reporting and monitoring and the way we secure passwords, we don't believe that any accounts have been improperly accessed. Still, as one of many precautions, we're requiring anyone who hasn't changed their password since 2012 to update it the next time they sign in."

Okay, so now, actually, I'm correcting myself. This sounds like they do know when the password was changed because there is a requirement upon sign-in, if it predates this, that you'll be forced to make a change. And I was curious, so I did sign in last week and

got no requirement to change my password. So it looks like they're handling it automatically. Be nice to know, to be able to read between the lines a little bit more and know, like, where they found what.

Leo: Yeah, they're not telling us what's going on. But I understand that because sometimes you'd prefer not to. Yeah, I didn't get - I just logged in, just to see, because I did get that email. But like you, I didn't get a prompt to change the password because I've changed it in the last year.

Steve: Right.

Leo: Right.

Steve: So anyway, they did everything you could ask someone to do.

Leo: That's the right way to do it.

Steve: Yes, exactly. Opera had a similar handling, although I sort of got a kick out of the way they disclosed it. First of all, there's Opera users. Opera has about 350 million users. And of that 350 million, they have about 1.7 million who use the Opera Sync, which is a cross-browser synchronization feature. So a very, very small, like less than half a percent of or around half a percent of users are taking advantage of Opera Sync.

So they wrote: "Earlier this week, we detected signs of an attack where access was gained to the Opera Sync system. This attack was quickly blocked. Our investigations are ongoing, but we believe some data, including some of our Sync users' passwords and account information, such as login names, may have been compromised. Although we only store encrypted (for synchronized passwords) or hashed and salted (for authentication) passwords in this system, we have reset all the Opera Sync account passwords as a precaution. We have also sent emails to all Opera Sync users to inform them about the incident and ask them to change the password for their Opera Sync accounts. In an abundance of caution, we have encouraged users to also reset any passwords to third-party sites they may have synchronized with the service."

So that's nice. The only glitch here is that they say "We detected signs of an attack where access was gained." Well, okay. What they're not saying is that "attackers exploited a vulnerability in our system that allowed them to gain access," which would be more correct. So, I mean, it's not like you just use enough force in the attack, and it cracks into the database. As we know, packets are not pointy. They're all kind of, you know, you look at the pictures, they're kind of square, and they move along. And so the packets themselves are not dangerous. It's what they contain.

But the flipside of this is a Chinese certificate authority named WoSign. And, boy, they're properly named. Okay. So get a load of this one. From some of the commentary I sort of pulled things together. One of the largest Chinese root certificate authorities, WoSign, issued many fake certificates due to a vulnerability. WoSign's free certificate service allowed its users - are you sitting down on your ball, Leo?

Leo: I am.

Steve: You're going to want to make sure you're centered for this. WoSign's free certificate service allowed its users to get a certificate for the base domain if they were able to prove control of a subdomain. Whoopsie.

Leo: That's not good.

Steve: This means that, if you can control a subdomain of a major website...

Leo: Leolaporte.squarespace.com, for instance.

Steve: Exactly. Then you can get a certificate from WoSign...

Leo: For Squarespace.

Steve: For Squarespace.com

Leo: That's not good. And a lot of blogging services do that. Blogspot.

Steve: Well, GitHub.

Leo: GitHub, right.

Steve: GitHub.io. And in fact what they found was many certificates for GitHub, Alibaba - which of course is the largest retailer, online retailer - and Microsoft, which of course now has gotten fancy with their logon stuff.

So here's the problem. After the vulnerability was disclosed to WoSign, they never reported this misuse to the root program as required. And their audit report didn't include any mention of this, either. So this has caused some outrage in the community of people who care about the integrity of the CA system. And so commentators have stated that WoSign lacks the security knowledge needed for operating a CA. I would argue maybe they just lack the care required to responsibly operate a CA.

In an online thread discussing potential sanctions against WoSign, WoSign as quoted as saying: "For incident 1, misissued certificate with unvalidated subdomain, total 33 certificates. We have posted to CT log server and listed in crt.sh. Here is the URL. Some certificates are revoked after getting report from subscriber, but some still valid,. If any subscriber think it must be revoked and replaced new one, please contact us in the system. Thanks."

So in my own notes here I said I really think we need to start rethinking our default

"trust everyone" policy because, as we've talked about often on this podcast, the number of root certificates in our systems has exploded. Just during the course of this podcast, I remember at the beginning it was 11. And I hesitate to look. Last time I looked it was more than 400 individual certificates, any of which our browsers will trust unless they're instructed not to.

So I think what we're going to see, because there is really no sign yet of any replacement for this system, this is the system we have - and in fact at the end of the podcast we will be answering last week's Puzzler of the Week. And I have a new Puzzler for this week...

Leo: Oh, good.

Steve: ...which involves some unintended consequences of the CA system, to see if anybody - just to sort of give people another little self-test.

Leo: It's fun to do this. Every week, I wouldn't mind doing a little bit of this. It'd be fun.

Steve: Well, I'll look for them.

Leo: If you come up with them, yeah.

Steve: Yeah. So, you know, right now it's like it was with EXEs. It's like, oh, yeah, download it, run it. Now everyone's got multiple layers of filters, and we're scanning stuff. Our precaution level is way up. Now, executables are tricky because anyone can make them. Signed certificates are much, you know, maliciously signed certificates, or erroneously signed certificates are rarer, but not nonexistent because we've covered this problem of mistakes being made. Sometimes they're made by a company that's otherwise responsible.

Anyway, I'll be very surprised if Google, for example, being one of the major enforcers of these sorts of problems, doesn't immediately yank WoSign out of Chrome and suggest that other browser vendors follow because they've violated the terms of their agreement with the industry implicitly and contractually explicitly by not stepping up and behaving in a responsible fashion. And I would argue that that's a good thing to require, but I don't think it's sufficient. I think - I know that some of our listeners have experimented with removing or disabling huge numbers of certificates and having no problem. I mean, I think if I went to a site that required me to be authenticated or required the server I was visiting to be authenticated by a WoSign certificate, I probably clicked a bad link because I don't speak Chinese. So I probably don't need that.

And the problem is, if they get a certificate for GitHub.io, then that is a place that I would be going. And I would look at my browser, and it would say, yes, you have a secure connection, when in fact I'm not at GitHub. So I will predict that downstream we're going to have to see, something's going to happen. I mean, we could easily - browsers could implement a conditional whitelisting system, much like we have with ads, with like uBlock Origin and so forth, where you selectively say, yes, I want to allow this.

So, for example, you could, if such a thing existed, you could put it in an auditing mode for a couple months where it looks at and, like, counts the use of any root certificates that it ever needs to use during a 60-day period, and you decide that sort of sets your usage profile, then lock that down and say, now require notification, proactive notification, if you encounter a certificate that hasn't already been whitelisted during this learning phase, and decide if that's one that you need to add. Maybe do it a one-time permission, but don't keep it in your validated cert pool.

So as far as I know, nothing like that exists, even as an add-on, because it does require access to some deep plumbing in the browser. So it would have to be something that the browser itself would either provide an API for, which they don't currently, or a feature offered by a more security-conscious browser. But I think we're at the point now where, for some users, that would be a worthwhile tradeoff.

And I know you've talked about it, I think probably on the Sunday show, Facebook and WhatsApp have announced that they're going to get together. And essentially WhatsApp is going to share its data with Facebook. This is a turnaround from everything that WhatsApp said because, at the time that this acquisition by Facebook was announced, WhatsApp was very aware of their users' privacy concerns because that's why people were using WhatsApp beforehand. And so the founder of WhatsApp said, at the time, "Respect for your privacy is coded into our DNA, and we built WhatsApp around the goal of knowing as little about you as possible." Until...

Leo: Until Mark Zuckerberg came along, yeah.

Steve: Until money. So, unfortunately, this appears to be no longer so clear. As I said, they recently announced that they'll be making a change in those policies. Now, when anyone runs WhatsApp, you should be prompted to acknowledge these terms of service change. You can decline. So there's two ways to fix this. If you haven't been in WhatsApp recently, maybe you already did and say, oh, yeah, yeah, fine, just clicked right through, that's okay. You can still fix it.

But if you do get prompted because you haven't used WhatsApp since they added this, you can, instead of clicking on Agree, you click on Read More. And that takes you to a dialogue where you are able to uncheck a checkbox that is labeled "Share my WhatsApp account information with Facebook." So you can turn that off in a subdialog of the agreement to the new terms and conditions. Or, if you've already done that without worrying about it, but now you think, okay, I don't want that, in WhatsApp you can go into the settings menu under the account tab, and you'll find the same thing. It says "Share my account info," which you can disable. Turn that off.

Oh, and I should mention there's a 30-day grace period. So they made the announcement saying 30 days from now the floodgates open, but you have to accept the new terms and conditions and at that time have those options set to, which are set by default, of course, because they want everyone to do it, unless they explicitly opt out. So any of our listeners who don't want that leakage - and I guess one of the big concerns is that WhatsApp has explicitly said that they will be sharing the user's phone number with Facebook. So, and who knows what else. But the good news is you can turn that off, if you think, uh, no thank you.

So we had a case of good old SQL vulnerability attack. The FBI sent out an announcement to all of the states' federal election sites after discovering that Illinois and Arizona had breaches. So the FBI uncovered evidence that foreign hackers had

penetrated, in the announcement it said two state election databases recently. I think, like, late July was one of them. And - I lost my train of thought, sorry. But that bulletin did not indicate which states were affected.

Other people who were familiar with this confirmed that it was Illinois and Arizona. In the Illinois case, officials were forced to shut down the state's voter registration system for 10 days in late July after the hackers managed to download personal data on up to 200,000 state voters. So they confirmed that there was a big exfiltration of their voter registration database through the site.

The Arizona attack was more limited. There was some malicious software that was introduced into the voter registration system, but no exfiltration of data resulted. So the FBI put out a bulletin which actually had a surprising amount of really good technical information. I was very impressed that they laid it out. And what they - I guess what I liked best was that it contained some explicit action items for other IT managers in other states, like specifically what to do, instead of just waving their hands and saying, oh, no, attacks are underway, keep an eye out, they really drilled down.

The bulletin said in late June 2016 an unknown attacker, well, now, or it said an unknown actor, actually it's a little less than unknown because they have the eight IP addresses that the queries came in through, which were logged by the weblog of the system that was attacked. So they said: "An unknown actor scanned" - I guess maybe an unknown individual - "scanned a state's Board of Election website for vulnerabilities using Acunetix" - which was the scanning tool - "and after identifying a Structured Query Language (SQL) injection vulnerability, used SQLMap to target the state's website. The majority of the data exfiltration occurred in mid-July. There were seven suspicious IPs and penetration testing tools Acunetix, SQLMap, and DirBuster used by the actor, detailed in the indicators section below."

Anyway, so their announcement contained lots of very nice technical detail and, as I mentioned, including the explicit IP addresses that these things came from, probably so that, if somebody wanted to, they could just block them at the border, stick them into a firewall rule and say, you know, under no circumstances do we want to allow any pointy packets from these people coming into our network. Anyway, I was impressed with the nature of the bulletin.

Last Friday evening, actually at 9:31, I was included in a tweet, or I guess maybe it was a retweet, where Tavis Ormandy, Google's famous bug hunter, who we know a couple months ago worked with LastPass to tighten up the minor problem that was found there, he was - Tavis was responding to @SwiftOnSecurity. And Tavis wrote: "I hadn't even heard of True Key. I have upcoming Dashlane and 1Password vulnerabilities." Last sentence: "There's a lot of scary garbage," as he put it.

So we don't have any details. I'm sure if Tavis has found something like that, he's already in touch with Dashlane and 1Password. So we won't know until patches are available, assuming that they're able to do it within 90 days. And of course as we know, LastPass did it, like, that day, within hours. And so the whole notion of any kind of a timeline of them fixing it and maybe a forced disclosure was short-circuited. So the takeaway here, for any of our listeners using Dashlane and 1Password, is keep an eye out for any forthcoming updates because you're going to want to jump on that in order to get the benefit of Tavis's findings.

And this, I have two things, I labeled them "From the Fringe," just because, okay, well, you'll see. And the first, I mean, they're serious. And, I mean, they're seriously written. The first one is written by Stuart Russell, who is a UC Berkeley computer science

professor known for his contributions to AI. And I went through the page, and he's got a thing for what he calls "lethal autonomous weapons systems," which he's very concerned about.

So he writes: "A very, very small quadcopter, one inch in diameter, can carry a one- or two-gram shaped charge. You can order them from a drone manufacturer in China. You can program the code to say: 'Here are thousands of photographs of the kinds of things I want to target.' A one-gram shaped charge can punch a hole in nine millimeters of steel, so presumably you can also punch a hole in someone's head," he writes. "You can fit about three million of those in a semi-tractor-trailer. You can drive up the I95 with three trucks and have 10 million weapons attacking New York City." He's a cheery fellow. "They don't have to be very effective. Only 5 or 10% of them have to find a target."

He writes: "There will be manufacturers producing millions of these weapons that people will be able to buy just like you can buy guns now, except millions of guns don't matter unless you have a million soldiers. You only need three guys to write the program and launch them. So you can just imagine that, in many parts of the world, humans will be hunted. They will be cowering underground in shelters and devising techniques so that they don't get detected. This is the ever-present cloud of lethal autonomous weapons. They could be here in two to three years."

Leo: What? I wish I knew that earlier. I would have fortified the studio a little more.

Steve: Well, yeah. Wow. Now, first of all, we know, for example, we've seen pictures of drones trying to fire a gun. And the problem is that, as Newton explained to us a long time ago, that for every action there is an equal and opposite reaction. So in order to push anything out the gun muzzle in one direction, the drone is going to get shot back based on the relative mass of these two things substantially, which lowers the muzzle velocity and...

Leo: Recoil is a bitch, yeah.

Steve: Exactly. And so I don't know about this one-gram shaped charge. I mean, you'd have to allow this little mosquito thing to land on you and then go off in order to do damage. But the problem is - it's interesting. One of the things I meant to talk to you about, and I forgot to, when I was reading the most recent Peter Hamilton book, "The Great North Road," there was a whole bunch of, like, the notion of grid systems, like they sprayed sensor goo, which formed an autonomous intercommunicating grid, in order to link up to something. And of course was it Daniel Suarez who did the amazing books about like the...

Leo: "Daemon" and "Freedom." Oh, and then "Influx," yeah, yeah, yeah, yeah.

Steve: The drone hordes.

Leo: Yeah.

Steve: And, I mean, these are chilling things. And technically he's not wrong. So, I mean, there are some technical hurdles to overcome. You'd need to have a camera on this little thing. It's got to have smarts. But, you know, neural networks, and I'm hearing you on many of the podcasts, Leo, talking about AI. Many of your guests talk about AI being an area of huge expected growth in the future. We're all hoping that Siri gets smarter somehow. Maybe Apple could make search work, which would be really wonderful for the App Store, apply some AI there. But still, you know, these are - this guy's obviously painting a very gloomy forecast. But there's nothing technically impossible about it.

Leo: He's counting on the hockey stick. That's what he's counting on when he says two years. And that's what, you know, all these AI researchers kind of, when you project that kind of thing, you project an exponential growth at some point which just takes off. And the AI start building better and better stuff, faster than a human could.

Steve: Skynet.

Leo: Yeah, Skynet. And then you're left behind in the dust, with the silicon dust.

Steve: Well, and it is the case, I mean, I was privileged to be at Stanford's AI Lab in the early '70s, sort of at the birth of that first very optimistic AI effort. And we had robot arms, and we had vision systems. And when you turned right to go up this windy road, there was a sign there that said, "Caution: Robot Vehicle in Operation."

Leo: Wow. And this was the '70s.

Steve: Yeah, in '72, '71, '72. And we all were like, it just sort of seemed, okay, we're going to solve these problems. And it turns out it's really hard.

Leo: Yeah.

Steve: I mean, it's - now, of course, we've got ridiculous computing power, compared to what we had back then. But still these are hard problems.

Leo: It feels like it's going to happen. Just we don't know when.

Steve: Speaking of a hard problem, this is now - this is tinfoil hat time.

Leo: Oh, boy.

Steve: The Atlantic had a story that a number of our listeners were worried about and wanted to make sure I was aware of. It's a little fanciful. But, well, because extrapolating

is dangerous for nontechnical writers, is I think probably the best way to put this. The story was titled "All the Ways Your WiFi Router Can Spy on You." Now, we're used to thinking of WiFi routers in terms of packets and firewalls and subnets that are separated and port-forwarding and so forth. That's not what they're talking about here.

This article says: "City dwellers spend nearly every moment of every day awash in WiFi signals." Which we know is true because you take out your phone and look at what's available, and you never don't see any WiFi. "Homes, streets, businesses, and office buildings are constantly blasting wireless signals every which way for the benefit of nearby phones, tablets, laptops, wearables, and other connected paraphernalia. When those devices connect to a router, they send requests for information - a weather forecast, the latest sports scores, a news article - and in turn receive that data, all over the air."

And I'm going to skip some of this because we get down to the meat here: "But it can be used to monitor humans, and in surprisingly detailed ways. As people move through a space with a WiFi signal, their bodies affect it, absorbing some waves and reflecting others in various directions. By analyzing the exact ways that a WiFi signal is altered when a human moves through it, researchers can 'see' what someone writes with their finger in the air" - okay - "identify a particular person by the way they walk, and even read a person's lips with startling accuracy." [Crosstalk].

Leo: We know about gate analysis; right? That can be very unique.

Steve: That was the one I was going to agree with, yes. Because there you're talking about very large signals. But again, in a presumably otherwise sort of prefabricated and established environment, and that's one of the problems.

So this article says: "Several recent experiments have focused on using WiFi signals to identify people, either based on their body shape or the specific way they tend to move. Earlier this month, a group of computer-science researchers at Northwestern Polytechnical University in China posted a paper to an online archive of scientific research, detailing a system that can accurately identify humans as they walk through a door nine times out of ten."

So again, they've purpose-built a system that irradiated a region and had sensors capable of detecting that. The problem is then extending that to a router, which has, like, no imaging capability whatsoever. It'd be like comparing the imaging array of a camera to a single photo cell. Well, yeah, so the photo cell can tell you what the ambient room light is. But you can't point it at the numeral "5," no matter how big it is, and have it tell you what it is. On the other hand, someone could argue, yes, but "5" has a different amount of black in it than "1." And so it could, without even knowing what the shape is, if it understood what the limited nature of what it was seeing was, it could still make a guess. And so that's probably a good analogy for this.

So this writer continues: "The system must first be trained. It has to learn individuals' body shapes so that it can identify them later. After memorizing body shapes, the system, which the researchers named FreeSense, watches for people walking across its line of sight." And so that's, again, that's a giveaway that this is more than just a router stuck on a shelf somewhere. "If it's told that the next passerby will be one of two people" - and get that, if it's told that the next passerby will be one of two people - "the system can correctly identify which it is 95 percent of the time." So it's better than a coin toss. But it's not able to do this for, like, out of any large population. There just isn't enough

information there.

They write: "If it's choosing between six people, it identifies the right one 89 percent of the time. The researchers proposed using their technology in a smart-home setting: If the router senses one person's entry into a room" - now, see again, they used the term "router," which is the problem. They write: "It could communicate with other connected devices - lights, appliances, window shades - to customize the room to that person's preferences." Okay. Except that what if the room had a few other people in it, and they were moving around, too? That is, everything in the room is subject to this WiFi signal.

So again, this isn't vision. This is one parameter. You get some doppler shift in the reflected signal so you can tell, like, the speed with which something's moving. But you can't even reliably tell if it's going towards you or away from you because you could be getting a reflected doppler signal from a far wall, and the person is moving away. So lots of problems with this. And I'm going to skip down here to another little bit of interest.

"A pair of MIT researchers wrote in 2013 that they could use a router to detect the number of humans in a room and identify some basic arm gestures, even through a wall. They could tell how many people were in a room from behind a solid wooden door, a six-inch hollow wall supported by steel beams, or an eight-inch concrete wall" - all of which are transparent to varying degrees to the proper radio - "and detect messages drawn in the air from a distance of five meters, but still in another room."

So anyway, this article goes on, talks about some researchers in 2014 who were able to use WiFi signals to do lip reading by, again, detecting the motions of someone's mouth and doing that within very narrow parameters. Our listeners will remember that I really liked a technology, I can't remember the name of it now, where it was a radio technology - I think Google was an investor - where you could do things like make a clicking motion with your fingers...

Leo: Oh, yeah.

Steve: ...above it, or like roll an imaginary toothpick between two fingers as a dial.

Leo: Yeah, Google showed it at Google I/O, yeah.

Steve: Yeah, it's very cool. And I can see the logo, but I don't remember the name. Anyway, what's so neat about this is that it uses a antenna array to be able to do that. And so you need what I would call imaging more than - yeah, there it is.

Leo: Project Soli, S-O-L-I.

Steve: Ah, yes.

Leo: Cool.

Steve: Just love it.

Leo: And never saw anything again. That was like...

Steve: Well, and in fact we're going to be talking about that because I have some update on XPoint and of course supercapacitors and next-generation batteries and so forth.

Leo: Oh, yeah, that.

Steve: Yeah.

Leo: Oh, yeah, that.

Steve: But anyway, I wanted to sort of give a little reality check. It is absolutely the case that you could stage an environment. But this article suggests that consumer routers are going to be spying, I mean, it says consumer routers are going to be spying on people. They're not going to be. The router is dirt cheap. The router, for example, there's no way it can get doppler information from the radio signal. That would blow its mind. And you absolutely need doppler in order, you know, not just intensity, but motion, in order to pull all this together. So it's like, yeah, one more thing to worry about, kind of.

But I guess one interesting notion is we've seen, for example, in some spy movies, where a laser interferometer is bounced off a window, and the speech in the room vibrates the window just minutely, but that's enough for the laser interferometer to detect the vibrations and turn it back into sound from a great distance away. So you could certainly do something like that, where you're using radio rather than this multistage acoustic process. But again, boy, interpreting any useful signal out of what you would get would be really difficult.

So, one piece of errata. Last week many of our sharp-eared listeners caught a mistake I made when I was reading verbatim from the technical document about the iMessage attack. And I remember looking at the number and thinking, what? But, I mean, I knew it wasn't right, but it's what it said. I said that that attack could be performed in as little as 218 samples. Well, the problem is that, when I copied and pasted from the PDF format into the text format, the caret between the two and the 18 was lost.

Leo: Oh, that's a minor detail.

Steve: Yes, 2^18.

Leo: Yeah, somebody in the chatroom said it, too, and I just didn't want to interrupt.

Steve: Yeah, well...

Leo: I figured people knew. They knew what...

Steve: I would have gone, "Oh, yeah." It does say here 218 in my notes, but I know that's not right. And it's funny, too, because, I mean, I read it earlier, and it said 2^18. And I thought, okay, well, now, that's, what is that, that's 256,000, approximately, because 2^16 we all know is 65,536; 2^17 will be 131,072; and 2^18 will be 262,144. So again...

Leo: But you know what?

Steve: That's not many.

Leo: Exactly. What's the diff? You're getting a lot, you're going to get that many packets pretty quick; right?

Steve: Yeah, it's still - and that was the point. It is a practical attack. It's not like it's 2^426. So anyway, a number of people sent me a tweet saying, "I don't think that's 218." I said, oh, of course it's not. So thank you for the correction. I certainly happily stand corrected here.

A couple pieces of miscellany. Another terrific dumb router write-up has surfaced, a dumb router configuration guide, this one with lots of information and pictures also. We talked about the one from - and I'm blanking on it. I did put - I added the link to the Link Farm page, GRC.com/linkfarm. This one is at nerdcave.littlebytesofpi.com.

Leo: Okay, I really like that URL. Wow.

Steve: That's P-I, by the way, LittleBytesofPi.com.

Leo: Well, of course, yeah.

Steve: And that page details both the two-dumb-router and the three-dumb-router configuration. So the three, of course, is our Y configuration that we have talked about and that has been often used. And again, as time goes on, in some cases these go under the Security Now! episode number. In some cases, like in this case, this is sort of for the ages, so it's at the top of the page in its own little Dumb Router Configuration section of links so people can always find it at GRC.com/linkfarm.

Also I got a tweet from a Willie Howe, who said: "One of your other subscribers wanted me to share my YouTube channel for Ubiquiti products with you." So, and this guy has a cool Ubiquiti EdgeRouter configuration YouTube channel. It's YouTube.com/williehowe, W-I-L-L-I-E-H-O-W-E. And I believe that one is in the show notes under SN-575. So again, YouTube.com/williehowe, Willie Howe. And so, for example, he's got - they're, like, 10- or 11-minute videos. Public WiFi Security, that's actually five different videos, one through five, that ends at "Putting It All Together," so it explains the whole thing.

He did one on a new release of the EdgeOS v1.9, and he notes in there that he was just about to do one on 1.8 when they came out with 1.9. So he goes through the new features there. He's got one on blocking traffic at the EdgeRouter, local traffic blocking, and configuring the EdgeRouter for multiple WAN, that is, Internet-side IP addresses, and more. So I knew that some of our listeners would get a kick out of looking at someone do some walkthroughs for what is actually the replacement for the three-dumb-router paradigm because it's $49, and it's an incredibly powerful little five-port router.

And finally - or actually not, sorry. The penultimate bit of miscellany, Intel just had their 2016 Developer Forum, and many people there were surprised that there were no announcements about the much-anticipated 3D XPoint status. It was a year ago that they and Micron jumped up and down and went crazy. We talked about it just recently, the idea of something that was much faster than flash and much denser than DRAM, so it sort of had a place in between.

And I found really some interesting analysis by someone, Jim Handy, who has a site called TheMemoryGuy.com. And I liked what he had, the way he phrased it, and he raised an interesting point. First of all, he characterized the whole system as layered memory. And we really know, I would say maybe hierarchically organized memory, we know that you have registers in the chip that the processor has instant access to.

Then there's a very complex hierarchy of increasingly slower and larger memory in a series of caches. So there's an L1 cache, the Level or Layer 1 cache, which, for example, all of the cores in the chip can have simultaneous access to also very quickly, although it's not like a register. It's not like working 1+1=2, perform the math. The abstraction is that it is RAM. So the processor sees everything outside of it as just a huge block of RAM. The reality is that there's a series of layers.

Typically now we have three layers of cache - Layer 1, Layer 2, and Layer 3; or Level 1, Level 2, Level 3. And then we have DRAM. And then you could argue outside, you know, the next layer down is flash, and the layer below that is traditional spinning hard drive. And what I liked was that Jim suggested that the proper way to think about this 3D XPoint is another layer in between the DRAM and flash because it is much, much denser than DRAM and much faster than flash. So that fits the requirement of something that would qualify as an additional, as a useful additional layer in the system.

But he wrote, and I really liked this, he said: "Intel really needs for 3D XPoint Memory to work. Without it, the performance of future computing platforms won't scale with processor upgrades. In other words, when a higher performance processor is plugged into the system, that system's performance won't improve because the rest of the system will bog the processor down," meaning its access to memory. "The new 3D XPoint Memory is the key to preventing this from happening. Without it, Intel will be unable to migrate customers to increasingly powerful processors that sell for higher prices and reap higher margins for Intel."

He says: "This is a tough spot for both companies" - meaning Intel and Micron because they've cross-licensed this and jointly developed it - "and there are no indications of any pending breakthrough that will improve the situation. About all we can do is watch from the sidelines with the hopes that Intel and Micron will overcome their technical problems and get back on track."

So essentially what's happened is, and this is very typical, they're having process problems. They have prototypes. They did show prototypes. They are hundreds of times faster than flash in some cases. Well, not quite a hundred. I think it was 1.75 microseconds versus 85 microseconds. But still, you know, way faster. The problem is it's

very different to make one than to, like, get a whole fab line running; and also not only have one out of every thousand actually work, but have a very high level of good parts out of a wafer of these things.

And the problem is this is new material. So everyone's really good about working with silicon. We know how to do that now. But this is some goo that is in between cross points. And you've got to lay the goo down and keep it where you want it, and not have it where you don't want it, and figure out how to actually produce this at volume. So I think there's every reason to believe in the long term it's going to be a good thing that's just not there yet. And they were optimistic a year ago. They weren't actually saying much this time. So, I mean, not even making anymore forecasts.

And against that background I wanted just to note, because I heard you talking about batteries - and I don't remember who you were talking with. I think it was off...

Leo: It wasn't on a show.

Steve: It was off-camera, but after the show.

Leo: Yeah, he was a battery researcher. He was really an interesting guy. He was from Hawaii.

Steve: He knew his stuff, Leo.

Leo: Yeah, yeah. It was fascinating. He was doing really basically empirical battery research. They were testing stuff. And of course it took, you know, it's a time-consuming process.

Steve: Oh, like how are you going determine cycle life unless you cycle?

Leo: Right.

Steve: And so it's going to take some time.

Leo: And it really underscored why it's so difficult to get good information about batteries because even this guy, who is as immersed in it as you could be, was unwilling to give me a unequivocal answer about anything.

Steve: And you know me, I was very impressed. I thought, good, you know, he was telling you when he did not know something.

Leo: Right, right.

Steve: Because that's the correct answer.

Leo: Right.

Steve: Is we don't know. I wanted to mention to you, ever since this is - and I kept forgetting to. I was very impressed when that third-generation Lenovo X1 Carbon that I got, back when I was worried about Windows 7.1 moving past me, after it had been plugged in for maybe two weeks, it popped up a notice. And it said, "This thing seems to be living on the AC line. If this is the way you intend to use it, let's bring the battery down to half."

Leo: Ooh, interesting.

Steve: "Because that will extend its life." And I was so impressed because that is absolutely true.

Leo: Yeah.

Steve: There's a thing called NiCad fast charging. And like those crazy high-performance model cars that zoom around, that are NiCad based, the battery gets hot because the engine, the motor draws so much current that it's essentially one or two ohm dead short. In fact, I remember them actually winding a motor with coat hanger. It was, I mean, it's that...

Leo: Wow.

Steve: You know, they want so much torque out of this motor that it's literally a short for the battery. So, but when the battery runs dead, as it will in 10 minutes, or like after one race it's dead, these guys want to recharge it fast. So they stick them on a fast charger which just pumps the battery with juice. And there's a little trick in NiCad battery chemistry where the terminal voltage on the battery will increase until it's charged and then begin to drop. And so what the fast charger does is sitting there staring at the battery voltage, looking for the first instant that it plateaus, and then it disconnects it, and the battery is charged.

The problem is lithium-ion, that we're all now using because it's got much higher energy density - it doesn't have the NiCad memory problem. You know, for these guys who were draining their NiCad down to the ground and then boosting it back up to full for the next race, there's no memory problem. But most of our devices are being used intermittently. Lithium-ion doesn't have a robust end-of-charge indication. And all of those, everyone's talked about, remember, the hoverboards catching fire and exploding. And people's laptops have done that. That's what happens when you overcharge a lithium-ion battery.

And so there's sort of this devil's bargain being made with everyone who's wanting as much life out of their lithium-ion as possible because the temptation is to get it as close to fully charged as possible, but not more. If you have a flaky charging circuit, and you overcharge, that's when the lithium-ion melts down - and, I mean, destructively so. But

it is the case that, if you want to store a battery, the way to do it is to, after fully charging it, bring it down about a third, between a third and a half. The two-thirds full is what I have most reliably seen. And that's where you want to let it sit.

You don't want to store it fully charged. They're just not as happy as they are with a partial charge. And he was right, Leo, relative to, for example, managing the batteries in your Tesla. Technically, it would be better, because you don't need the maximum mileage, it's easier on the batteries if you don't tax them by giving them a full charge.

Leo: Right.

Steve: Tesla wants to push it full so that you can get as much mileage...

Leo: Well, they recommend, they actually - on the Tesla app you have a setting.

Steve: Nice.

Leo: Yeah, that says daily charge or range charge. And they recommend don't fully charge it every - I did it once, and I got a lot of tweets from people saying - posted on Instagram. They said, oh, no, don't do that. And so I have it set at something like 80 percent. But he also suggested charge it every day. You should keep it charged. And so I started doing that.

Steve: Yeah, there have been people who say, oh, don't leave your devices plugged in.

Leo: I asked him about that, and he was very clear, that's fine.

Steve: Yes. Don't leave a flaky device plugged in. But, for example, Apple has figured this out, and Apple stops charging, and then that's not a problem.

Leo: Don't use a flaky device is probably a better idea.

Steve: Yes.

Leo: Get a device that manages power best.

Steve: You and Ian were talking, or I guess Ian mostly, was talking on Sunday about "Halt and Catch Fire." And I did want to note to our listeners who had watched the first two seasons that, if they had missed it, Season 3 did start last week.

Leo: Harry McCracken was actually on the set and did a long piece about it. That's

who we were talking with is Harry McCracken.

Steve: Okay, right, right, right.

Leo: Yeah. He thought it was amazing. I mean, he was - I don't know if he saw the shows, but he was really impressed by the verisimilitude of the set.

Steve: Yeah. It's tough for me to make time for it. What it has turned out to be is, and I wrote in my notes where I said: "Predominantly character-driven, set in a nominally historical but fictional techie setting, but not particularly compelling characters."

Leo: Yeah.

Steve: That is, it's like, eh, you know, I really don't care about these people. And a good...

Leo: That's important.

Steve: A good show, they really do want you to care about them.

Leo: Funnily enough, the thing that bothers me besides that - and I did try to watch it, and I stopped, I haven't seen the third season - the thing that bothers me is the kind of close to but not actually.

Steve: Yes, exactly.

Leo: And it's like, no, that's not what happened.

Steve: Historical fiction. So it's not actual history.

Leo: Yeah. And we lived it, so we know what happened. And so it's kind of - it bugs me in that sense. It's like a bizarro universe.

Steve: Right, right.

Leo: By the way, good show, just a side note here, I don't know if you've watched it on HBO yet, is "The Night Of."

Steve: It just finished, didn't it.

Leo: Yeah. And I avoided it when I first saw it because I thought, that's too grim. And it is grim. But the acting and the writing is superb.

Steve: I had a number of my followers recommend it, and so I immediately - I think I missed the first two. So I told TiVo - oh, I just did want to mention, I heard you also talking about TiVo. I do not regret my lifetime...

Leo: No. I have two.

Steve: My lifetime subscription.

Leo: I have two. Just bought them.

Steve: I have three. I have three because I have two minis. And they all have them. It is such a good device. I mean, it's just...

Leo: Yeah, it really is. It is the best. There's no doubt.

Steve: It is the one. And so even if it went belly-up a year from now, I'd think, well, okay. It would have been wiser not to go lifetime. But, boy, it's just [crosstalk].

Leo: It's worth it, yeah.

Steve: And I loved your story about Lisa seeing yours and going, "Why do I have this crappy one?"

Leo: "Why didn't you buy me one? Why are you making me use the X1? That's not right." But it's because it's so expensive. I mean, once you buy the TiVo and the lifetime subscription, you've spent 700 bucks.

Steve: Oh, you're biting a bullet, all right.

Leo: Yeah, yeah.

Steve: But, boy, you know, I watch a lot of cable news stuff, especially during this political season. And if I couldn't fast-forward through the commercials, I swear, they're just - it's there as an excuse for advertising.

Leo: Oh, yeah. They don't even make any bones about it. That's clearly what's going

on.

Steve: You can't watch it, like give it your attention. So you have to have that. And, boy, that new skip feature does work nicely that it supports.

Leo: Oh, is that nice, yeah.

Steve: I got a nice tweet from someone about something that, believe it or not, I still haven't ever talked about. The guy's Twitter name is PGP ID 0x01086FDA. Looks like his name might be Cristian Rasch. Anyway, he said...

Leo: Oh, that's a good idea. Use your PGP ID as your Twitter handle. What a great idea. I might change my NIC right now.

Steve: That's kind of cool.

Leo: Yeah.

Steve: Yeah, PGP ID.

Leo: That way people know how to privately reach you.

Steve: Yeah.

Leo: As long as you publish your key somewhere.

Steve: "@SGgrc Thank you for supporting Wine as a valid SpinRite platform. Just purchased my copy, which is currently hard at work." And I never really talked about how, when you buy SpinRite, you get an executable. Just one. Just an EXE. And what's fun is that, when you run it under Windows, or as Cristian notes, Wine, so Linux or Wine on a Mac, it shows you a graphical user interface. It says, "Hello, this is SpinRite." And you then use that to produce boot medium. You're able to burn a CD, to format a thumb drive and then boot the thumb drive. So that's sort of the installer, the creator of the bootable media. But there's only one EXE.

And so after SpinRite creates the boot environment, it copies itself, that same EXE itself, to the device, to the thumb drive or into the CD image. And when you run that same SpinRite EXE from DOS, you get SpinRite. So there aren't two things. There's just one. And the way I did it is kind of cool because, once upon a time in the early days, when Windows was just beginning to happen, and there was a lot of DOS still being run, Microsoft said, okay, wait a minute. What happens if somebody runs - we're going to use the .exe. We're not going to use, like, .win or some other extension for Windows apps. We're going to use EXE, which is, you know, and DOS had .com for command-style

image files up to 64K, which was SpinRite until it outgrew the 64K.

So they were going to use an EXE. But Microsoft thought, what if somebody attempts to run a Windows EXE from DOS? Because DOS is where most people, that's where everyone was initially. So what they did was they actually, they designed the Windows executable so that it had a DOS stub. That is, it had a DOS program. And it was a fixed stub. And when you ran that app under DOS, this little one-liner, it would say, "This program requires Microsoft Windows" to run. That's all it did. And it dropped out and came back to the command prompt. And you'd go, oh, right. And then you'd launch Windows and run that same EXE in Windows.

Well, what I did was I wrote a Windows app which does the install, and SpinRite itself is the DOS stub. So when you run that EXE, which is technically a Windows application, in DOS, DOS runs the DOS stub which is SpinRite, and SpinRite runs. When you run it under Windows, you get the UI that builds SpinRite. So I've never talked about it. It's just kind of a fun little hack.

Leo: That's a nice way to do it, yeah.

Steve: It works beautifully, yeah.

Leo: All right. Let's continue on, Steve Gibson, and we're going to talk about Trident and Pegasus.

Steve: Yeah. So last Thursday at 3:00 in the afternoon I tweeted to my followers: "Apple recently pushed an emergency update for all iOS devices. It has been used against 'targeted victims,' but could see wider use now." So the message was go get yourself updated. And as we've seen, it's five days later, and I'm finally seeing iOS devices that are acknowledging that there's an update to be downloaded. So again, I'm not clear why this wasn't immediately delivered, but it wasn't. So this is a textbook case of a bad exploit and very responsible handling and management of its discovery.

This begins on August 10th, so what is that, 20 days ago. Ahmed Mansoor, who is an internationally recognized human rights defender - he blogs; he's a member of the Human Rights Watch's advisory committee; and he's also received some award, sort of like the equivalent of the Nobel Prize of human rights activists. But he's been harassed for the last five years with various technical attacks on him.

So, as I mentioned at the top of the show, he was already ready to be suspicious, which unfortunately distinguishes him from most people who might be victims of this. And as we'll see, and as I mentioned at the top of the show, this has been apparently in place for quite a while. In the reverse-engineered code, they found references that are three and four years old. So somebody's not happy. Well, in fact, unfortunately we probably know, we're very sure we know who, that this thing got foreclosed on.

The problem is everything we know would have to suggest that there are other unknown vulnerabilities, and that this company that is making a huge amount of money making these available to governments and law enforcement, they say on ethical basis, since they're making this much money, they're highly motivated to continue finding ways in.

What we have in this case is a really good snapshot of the details that I know our

listeners are going to find interesting. So he receives an SMS text message which immediately made him suspicious. He did not click on the link. The next day he received a second similar text. The messages promised him new secrets about detainees tortured in UAE prisons. So that's, like, right up his alley. This was targeted for him. Obviously they had his phone number, whoever it was, and they were sending him something they believed he would want to know. And it contained a hyperlink, these messages did, each of them, to an unfamiliar website.

Well, it was unfamiliar to him, but not to the guys at Citizen Lab. And these arrived on his stock iPhone 6, which was running 9.3.3. Now, as I mentioned also before, and we talked about a couple weeks ago, we just went to 9.3.4, and now we're at 9.3.5 as a consequence of this.

So rather than clicking on either of those messages' links, he forwarded both messages to Citizen Lab for their investigation. Two days later - so that was on August 10th and 11th. Two days later, on the 12th, Citizen Lab brought Lookout Security into the loop for their reverse-engineering and technical analysis, which is very impressive. I don't - in my raw notes I had a link to their PDF. It's a 35-page paper. I will stick it on Link Farm after we're through so that anyone who's interested can go to GRC.com/linkfarm and get the link to the PDF, if anyone's interested, because it was rich in details.

So Citizen Lab brought Lookout Security in for reverse-engineering. Three days afterwards, on August 15th, Apple was brought in and brought up to speed. So these guys figured out what was going on, realized how bad this was, and on the 15th brought Apple in. And then 10 days later, on August 25th, Apple released iOS v9.3.5 to patch and close three previously unknown zero-day vulnerabilities, all of which were used - thus the term "trident" because it was three vulnerabilities in this cyberweapon.

So what do we know? The malware's been in operation for more than a year - probably, like I saw in some notes, at least there were some 2012 and 2013, so as much as four years - which has enabled it, they wrote, to develop a high degree of maturity. As a result, the software is capable of exploiting multiple iOS versions, essentially every version of iOS from 7 - there is a test in the software for iOS 7 - all the way up through 9.3.3. So it would have infected Mansoor's phone. It would not have infected 3.4, only because the software hadn't quite been updated for the two-week-old version. But when they removed that test, it then did infect 9.3.4. So it was just a matter of checking.

It was very much like we were talking about the Cisco PIX firewall the other day, where that malware wouldn't have infected more recent devices, only because it was from 2013. The versions had moved on, but that compound if-then clause hadn't been updated to handle, just wasn't aware of any versions after it had been minted.

So an excerpt that they found in what they called a "magic table" contained in the kernel exploit portion, which maps addresses in the kernel, shows that the exploit supports versions of the phone from the iPhone 4s all the way up through the iPhone 6s Plus. So what they produced was an in-depth technical analysis of what they called a "targeted espionage attack," which is being actively leveraged. And I appreciated what they wrote because they said "leveraged against an undetermined number of mobile users around the world."

That's what we have to realize is that this has been around for years. This guy, just because essentially they targeted - someone targeted someone they shouldn't have, probably the UAE because he's a problem for them. He's been jailed, and when he was released from jail $140,000 were stolen from his bank account and so forth. So he's been subject to a lot of harassment over the years. So - and I lost my train of thought.

Leo: You know, what's interesting, I mean, given that this is going on for three or four years...

Steve: Oh, right, right, right. What I wanted to say was that - thank you. We know of him receiving it. But it is, as I'll cover in a second, extremely stealthy. It is beautifully written. And I hate to use that term for something that is so malicious. But, I mean, it is professional-grade cyberweapon. And we know that it comes from an Israeli company. Is it NFO?

Leo: NSO.

Steve: NSO.

Leo: Yeah.

Steve: NSO Group. And you mentioned on MacBreak Weekly an Israeli company, apparently owned by a San Francisco-based venture capital firm that are now trying to sell their ownership of NSO Group for a billion dollars.

Leo: Well, now is a good time.

Steve: Okay. So what this does is when - if he had clicked either of those links, that would have taken him to a web page that - and I'll get into details in a second - that exploited an unknown, very complex, compound flaw in WebKit that gave the exploit a foothold. And that's one of the three vulnerabilities. Then the other two were used for the second phase, which was obtaining additional pieces through web queries from the remote server and then installing itself in a persistent fashion.

The problem is nobody - it does jailbreak the phone. It dynamically, remotely jailbreaks the phone, giving this tool complete carte blanche access across the phone in a way that chilled these guys. And I'll explain in a second. But it's also completely stealthful so that nobody would know that this had happened. Oh, and if something happens such that the exploit cannot function, a different web page is displayed so the user doesn't know that they just dodged a bullet. And so in no way does it make obvious what has happened. So, I mean, it is absolutely stealthful.

So nobody, since this has worked on iPhone since 4s, and we know that it's been around for years, and it installs itself persistently and survives reboots and upgrades, there is no way of knowing how many people today are currently under monitoring, being monitored by this. Well, until updating to 9.3.5. That sweeps it away and prevents it from happening. But so I love that they said "an undetermined number of mobile users around the world" because we need to remember it's not just this one guy. He's the one, he was the last person to be infected, rather than the first person to be infected.

So they write that "Pegasus is professionally developed and highly advanced in its use of zero-day vulnerabilities, code obfuscation, and encryption. It uses sophisticated function hooking to subvert OS- and application-layer security in voice/audio calls and apps

including remotely accessing text messages, iMessages, calls, emails, logs, and more from apps including Gmail, Facebook, Skype, WhatsApp, Viber, FaceTime, Calendar, Line, Mail.ru, WeChat, Surespot, Tango, Telegram, and others."

Leo: Wow.

Steve: And they know this because embedded in this are individual, per app, essentially DLLs. They're not called DLLs over in Mac land. But they are dynamically loaded libraries. This thing uses the phone's jailbroken status to turn off code sign checking, and then it injects these individual modules per app. So it's got a WeChat hooking module. It's got a Telegram hooking module, a WhatsApp hooking module. And, they note, as we've talked about, it hooks it pre-encryption. So the user is using WhatsApp or Telegram with encryption, or iMessage. It doesn't matter because what's coming out in the clear at their end and what's going in from the keyboard in the clear is captured before the encryption stage because this thing has inserted little stubs, little hooking stubs, custom-written for each of these different apps.

They said: "The attack is very simple in its delivery and silent in delivering its payload. The attack starts when the attacker sends a website URL through SMS, email, social media, or any other message" - and remember, that's another point. This uses WebKit. And so that's the general purpose HTML display engine used not only by Safari, but other web browsers. And it's what gets invoked when an HTTP link is accessed. So through whatever source, whether someone tweets it to you or sends it to you in email, it ends up acting in the same way.

"Once the user clicks the link, the software silently carries out a series of exploits against the victim's device to remotely jailbreak it so that the espionage software packages can be installed. The user's only..."

Leo: Oh, it's not the website they're going to. So there's WebKit. WebKit modified renders then a malicious website in that link?

Steve: No. Yeah. The link itself is innocuous.

Leo: Oh, interesting.

Steve: It was SMS dot and it was web.avs.co or something. Now, that didn't mean anything to Mansoor, but the Citizen Lab guys instantly recognized this as a domain where they had seen spyware tools operating before. So they immediately knew this was probably malicious. So there's a remote server somewhere - oh, and then it had - it was slash and then like a six- or seven-digit serial number. And the two messages contained different serial numbers. And these are one-time-use links.

Leo: Right.

Steve: And that's in order to prevent any kind of an analysis post-infection. So you can get it once, and in clicking it, it disables it at the sending end. So the remote server

sends you the exploit. In fact, Leo, this is the licensing model. When someone...

Leo: They have a licensing model.

Steve: Yes. When some government says "We want to use this," in their documentation, I think it might have been Citizen Lab's write-up, they indicated that someone purchased 900 or some hundred number of infections for X amount of dollars.

Leo: Oh, my god.

Steve: And so you receive a Pegasus workstation which connects you into this infrastructure. But NSO runs the infrastructure. They source the exploit from their servers. So if your check bounces, you don't get any of the information. So, I mean, so they're an active...

Leo: This NSO Group really is like a criminal organization, frankly. I mean, this is just appalling.

Steve: Well, yeah, under the guise of political expedience, I guess. So they say: "To accomplish [what it does], after jailbreaking the user's phone, the spyware does not download malicious versions of these apps to the victim's device, but rather compromises the original apps already installed on the device. This includes preinstalled apps such as FaceTime and Calendar" and blah blah blah, the ones I already mentioned, by installing spying hooks into those.

So a user infected with this spyware is under complete surveillance by the attacker because, in addition to the apps that it has specific hooks for, it also spies on all phone calls, call logs, SMS messages the victim sends or receives, and audio and video communications. And in fact, and in the words of one of the founders of the NSO Group, it turns the phone into a walkie-talkie.

Leo: Jesus. How can this be legal?

Steve: It's amazing to me.

Leo: It's amazing.

Steve: And again, so get a load of this. So Stage 1 is the delivery and the WebKit vulnerability. This stage comes down over the initial URL in the form of an HTML file that exploits a vulnerability. And so there were three vulnerabilities given: the CVE-2016, because that's the year we're in, and so these are vulnerabilities 4655, 4656, and 4657. So Stage 1 is click the link, and it uses a WebKit vulnerability.

Then the jailbreak is Stage 2. This stage is downloaded from the first stage code based on the device type. So the user agent in the web query tells the remote server what it is,

who it is, what device and technology is making the query. So it then sends a customized attack package for that device. So Stage 2 is downloaded as an obfuscated and encrypted package. Each package is encrypted with unique keys at each download, making traditional network-based controls ineffective. As we know, if you encrypt the same file with a different key, you get a completely different result. That's what encryption does. And so by always encrypting under a new key, you're never able to get any pattern match, and every single instance of the same thing looks completely different.

"Stage 3, espionage software. This stage is downloaded by Stage 2 and is also based on the device type, 32-bit, 64, et cetera. Stage 3 contains the espionage software, daemons, and other processes that are used after the device has been jailbroken by Stage 2. Stage 3 installs the hooks into the applications the attacker wishes to spy on. Additionally, Stage 3 detects if the device was previously jailbroken through another method; and, if so, removes any access to the device that the previous jailbreak provided, such as SSH." So it's also jealous. It says, no, anybody else who's in here, we're kicking you out. We're taking over. It's ours; this phone is ours now.

"The software also contains a failsafe to remove itself if certain conditions are present." And what I loved is the third stage deploys a number of files, and it enumerates them. The one that caught my eye was ca.crt. It brings along its own root TLS certificate, which it adds to the keystore. So on top of everything else, they've installed their own key so that they can get up to other mischief in the future, if they choose. Now this phone will trust any certificates, signed by them, spoofing any other sites.

So they write: "The attack works on iOS up to 9.3.4, and the developers maintain a large table in their code that attacks all iOS versions from 7.0 up to and including 9.3.3. While the code investigated did not contain the appropriate values to initially work on iOS 9.3.4, the exploits we investigated do still work, and it is trivial for the attackers to update the table so that the attack will work on 9.3.4." As I mentioned before, it's just a function of Apple's been revving iOS at a pace that this malware has, like, stayed just a few weeks behind.

They said: "One other unique property of this attack is that standard jailbreak detections fail to report that the device has been exploited. The attack and installation of the spying software is designed to be as silent as possible to the target." They write: "Pegasus is well designed in terms of its modularity and efficiency. For example, the kernel exploits call upon magic tables for each of the platforms that map out kernel memory for each version and phone model. The code is extremely modular, relative to other malware our researchers have encountered." And this is Lookout Security talking.

"We found common libraries and common formats with similar naming conventions. Unlike most malware authors, the code in Pegasus is clean and efficient, with evidence of professional and careful design. We see evidence of a robust quality assurance process for the development. Even their first-stage exploit contains both debugging and QA-specific functions of the type one would expect from an enterprise-class software development organization."

And so I just had a couple little notes about these three exploits. The first one, 4655, is the memory corruption in Safari via WebKit. They write: "A memory corruption vulnerability exists in Safari WebKit that allows an attacker to execute arbitrary code. Pegasus exploits this vulnerability to obtain initial code execution privileges within the context of the Safari web browser. This vulnerability is complex, and Lookout continues to work on analyzing this vulnerability and will publish additional findings as they become available."

And I thought of you, Leo, because this is address space layout randomization defeat. 4656, kernel information leak circumvents KASLR, the "K" in this case for kernel. "Before Pegasus can execute its jailbreak, it must determine where the kernel is located in memory. Kernel Address Space Layout Randomization (KASLR) makes this task difficult by mapping the kernel into different and unpredictable locations in memory. In short, before attacking the kernel, Pegasus has to find it. The attacker has found a way to locate the kernel by using a function call that leaks a non-obfuscated kernel memory address in the return value, allowing the kernel's actual memory location to be mapped."

So that's a beautiful example of, as we've said on the podcast, yes, address space layout randomization makes it more difficult. Everybody has that now. And they're still getting defeated because, unfortunately, it ups the ante, but it doesn't prevent the problem. And in this case there was a function whose return value allowed the inference of where the kernel was, and that completely defeated that.

Leo: Wow. So much for ASLR.

Steve: Yeah. And finally, 4657, memory corruption in kernel leads to jailbreak. "The third vulnerability in Pegasus's Trident is the one that is used to jailbreak the phone. A memory corruption vulnerability in the kernel is used to corrupt memory in both the 32 and 64-bit versions. The exploits are performed differently on each version."

So again, that's why different packages are downloaded, depending upon the bitness of the target device. So these guys, it's really four, when you think about it, or more, exploits because there have - so they needed, for the jailbreak, they needed individual ways of jailbreaking differently for the 32- versus the 64-bit code. So they have both, and they choose the one that they need, depending upon what phone the person is holding when they click the link. "The vulnerability is complex, and Lookout continues to work on analyzing this vulnerability and will publish additional findings as they become available."

So, and then I conclude by talking about the jailbreak persistence. They said: "Once the kernel has been exploited, both exploits perform similar tasks" - that is, either 32- or 64-bit - "to prepare the system to be jailbroken. They disable kernel security protections, including code signing. They remount the system partition. They clear the Safari browser caches to help cover their tracks and then write the jailbreak files, which then give them persistence across reboots and future version iOS updates." Wow. So a look into the real world of, I mean, as these guys said, it's enterprise-class, enterprise-grade remote attack on the most secure mobile platform that man has developed so far.

Leo: Well, I mean, how secure is "most secure"? I mean, it's kind of meaningless; right?

Steve: And that's the problem, exactly.

Leo: "Most secure" is probably not much more secure than less secure.

Steve: Well, as we know, there's policy, and there's implementation. And so sloppy

policy there's no excuse for. Implementation errors, well, unfortunately, we all want these things to do a lot. And that makes them complicated. And, as we know, complexity is the enemy of security. So, wow.

Leo: Amazing.

Steve: And so I think the takeaway from this is anyone who is a high-value target would need to avoid the use of everything.

Leo: [Laughing] Ta-da.

Steve: Put on a bathing suit and walk into the middle of Central Park and meet somebody who is also in a bathing suit and whisper to each other and cover up your mouth moving so that lip readers can't get you. I mean, it's the reality. I'm not worried that much because I don't do anything that anyone wants. And I exercise all prudent cautions, but I don't stay awake at night.

Leo: Wow. Well...

Steve: Wow. Okay, so...

Leo: Yes. Now there's Pegasus, the flying horse.

Steve: And the problem is I'm sure this was an expensive thing to have happen. I don't believe these were the only unknown problems. And I would imagine this company has such strong incentives and such a demand for these tools that the moment this happened they brought their next generation vulnerabilities online, and they're offering something, they didn't have to reprint the brochure. It still says we'll do the same thing we did before. Technically they're doing it in a different way. But I'll bet you that there is still a link that somebody is going to receive next month, or tomorrow, and it'll still take over their phone, still install all this stuff, same modules get installed in Stage 3, all that work's been done. That's going to be protected because there's going to be some other way in.

Leo: Okay. That's good. You know, it's a lesson to be learned; right?

Steve: I think...

Leo: What's frustrating to me is that they know about these vulnerabilities for four years, don't say a word to Apple to fix them because of course not, they're selling them. But who's to stop somebody else from coming along and discovering it and weaponizing it? Right? I mean, these guys can't be the only people with the skill set.

Steve: No, no. That's absolutely right. And so certainly, hats off to Apple. A 10-day turnaround, I mean, it's not like this is a huge thing to fix. I appreciate what Rene said, talking about this on MacBreak Weekly. Certainly they need to be careful. We've had some disasters where updates have caused phones to stop functioning correctly and that, you know, back in the early days. We haven't had that happen for a long time. But so you always have to be careful. But Apple was, I think, very responsive to this. In order to go from first notification to pushing a patch out in 10 days across their entire iOS platform, hats off. The problem is there's probably other problems.

Leo: Oh, probably. Almost certainly.

Steve: Yeah, yeah. So last week we left our listeners with a puzzler. Why do we use primes? And I said there's a little subtlety to it. So the best way to explain it is that the security of prime number-based, or prime number factoring dependent public-key encryption, depends upon a one-way function. The guys who invented this came up with the term "trapdoor," which never really seemed that clear to me, but it's called a "trapdoor function" in cryptography ever since they invented the term in the '70s. So but it's a one-way function, meaning that it is easy to do, but intractable to undo in the other direction.

And I ran across a nice analogy the other day. When you're describing this to your non-techie friends, use the analogy of a padlock. It's easy to close the padlock. But opening it requires a key. So a padlock is a one-way function, easy to do in one direction, hard to undo in the other direction. In the case of math, two very large - and that's the key. Remember that it's very large prime numbers satisfy this requirement because multiplying numbers of any size is easy. We've got algorithms that just multiply, and they can be arbitrarily long numbers. We know how to multiply them. But then the security entirely depends upon not being able to unmultiply them, which we call "factoring."

So if the numbers multiplied were not prime, that would mean that one or the other or both of these nonprime numbers themselves would have smaller factors. And what could happen is that it's much easier to find a smaller factor than a larger factor. It's the size of the factors that is intractable because, despite decades now of very clever mathematicians speaking bizarre languages of math, we have made very little progress on speeding up prime factorization. It has turned out that is really resistant to any kind of, I mean, and you can imagine it's a huge pot of gold if someone could figure out how to crack a big number into the two numbers that multiplied to get it. Nobody has figured that out.

Leo: How big do the primes need to be to be effective?

Steve: They're thousands of bits.

Leo: Yeah. Is that what we say when we say 4,096 bits? That's the size of the prime factor? Or the size of the total, the number? The number they're factors of?

Steve: It's the modulus. So it's multiplied, and then essentially the division by 4,096 or 2,048 or whatever is what is kept behind.

Leo: Oh, interesting.

Steve: The residue from the modulus.

Leo: Got it, okay.

Steve: So anyway, so the point is, it is the fact that the individual factors cannot be decomposed; that what the cryptographer has to do is decompose the entire thing into two primes. If they were not primes, if they had other - if those numbers being multiplied themselves had factors, then you could chip away at it. You could find a smaller factor and then remove that, find the next factor and remove it, and whittle this thing down.

Leo: The obvious example would be is, if this big number is even, you'd go, oh, well, there's a two in there.

Steve: Perfect, yes. Now it's half as long as it was before.

Leo: Right, right. So you multiply two primes together, you've got to get the primes. There's nothing smaller you can go for.

Steve: Right.

Leo: That's what I thought. I'm glad I got that right.

Steve: You got it right. The puzzler for this week...

Leo: Good. These are fun. I want you to do this every week. Folks, tweet Steve with puzzlers. I want more of these.

Steve: Okay. So, and this one came yesterday in a tweet from the owner of a company who had a tricky problem. He listens to the podcast. He said: "We're in trouble. What do you think?" So a family of network-connected, embedded-style devices having unalterable firmware is hardwired to make TLS connections to a server at this company's publicly available, fully qualified domain name. So there's a device, there are devices...

Leo: They sold these devices; right?

Steve: ...a family of devices out in the world.

Leo: And they sold them; right? And they hardwired the address into the device.

Steve: Yes. And the firmware cannot be altered.

Leo: Right.

Steve: But the firmware only understands how to satisfy and verify SHA-1 hash certificate signatures.

Leo: Uh-oh. Oh, crap.

Steve: The server these devices are connecting to at the domain name burned into the device's memory is currently serving an SHA-1 cert. But as we know, when it expires, it will be impossible, and it happens to be a Symantec VeriSign cert, it will be impossible to get Symantec or any other CA to sign an SHA-1 for a standard public fully qualified domain name.

Leo: Because we're all going SHA-2.

Steve: Exactly, SHA-256.

Leo: 256.

Steve: Those are the only certs, as of midnight of last year, or January 1st of this year. So no CA will issue an SHA-1 signed cert in 2016 or after. Is there anything this company can do, any trick that can be played externally to satisfy the needs of this unmodifiable hardware to allow it to continue to function?

Leo: And kids, this is why you don't lock down the firmware on your IoT device, and you provide an update mechanism.

Steve: Exactly. It's a classic example of whoops.

Leo: Wow.

Steve: Yeah. So there is no update for this.

Leo: So you need to somehow spoof - the address is written in stone. And of course your cert...

Steve: Oh, you want to do this now, Leo?

Leo: No, no, no, I'm just thinking. I just want to clarify the problem. I'm not going to solve...

Steve: Okay.

Leo: Believe me, I ain't solving this one. You'd have to somehow still function with a nonfunctional cert, basically. Your cert's going to expire.

Steve: Yes.

Leo: And I presume that the device requires a secure connection to operate, right, it's going to balk.

Steve: Correct.

Leo: It's going to say, well, you're not secure. Okay, stupid question. Can you set the time on the device? Okay. I'm not going to ask you more questions. Let's presume not. The answer obviously is not setting the device into - putting the device in 2009. Hmm. Interesting. You can't modify the device. You can't modify the URL. It needs be a secure connection.

Steve: It's burned into this thing that does not have it firmware updated.

Leo: But sometime soon they're not going to be able to go back to VeriSign and say I need a new cert. It'll have to be - it won't be an SHA-1 cert. That's a good one.

Steve: So think about it for a week. We will discuss it.

Leo: Oh, I know. You go to Wo.co.

Steve: Yes.

Leo: They'll give you an SHA-1 cert. Steve Gibson's at GRC.com. That's his home on the Internet, the Gibson Research Corporation. If you go there, just do us a favor, make a yabba-dabba doo happen. Buy a copy of SpinRite, the world's best hard drive maintenance and recovery utility. That's Steve's bread and butter. And while you're there do take advantage of all the freebies. Find out about SQRL; get some Perfect Paper Passwords; check your shields, are they up or down; all of that. Including this show. He's got audio of the show. And of course transcripts. Give it a

couple of days. Elaine has to write these, so it takes a couple of days after the show.

Steve: Yeah.

Leo: You can find 64Kb audio at his site. You can find that and video at our site, TWiT.tv/sn. Otherwise they're identical. They're the same files. And we also make sure that they're available on all your favorite podcatch apps, including the TWiT apps, which are on every platform - Windows and Mac and iOS and Android and Roku and everywhere you want to be. There's even five different Apple TV apps. And you just go to Security Now!, you could sit there and watch it in the comfort of your own home. Don't miss an episode. This is a good show. Now, we might do questions next week.

Steve: Let's hope.

Leo: Let's hope.

Steve: Because otherwise a catastrophe, only a catastrophe will keep us from doing a Q&A.

Leo: Right, that's a good point. That's a good point. Security willing. So go to - he's on Twitter, @SGgrc. You can always tweet him there. You can also go to GRC.com/feedback and ask your questions, and we'll take the best ones, the most commonly asked, and answer them next week.

Steve, lots of fun. And I love the stumper. And I have no idea at all. The last week one I thought was - that was easy. This one, I don't know. There's a clever way, though, huh? All right.

Steve: Could be the Kobayashi Maru.

Leo: Ooh. Mmm.

Steve: Could be.

Leo: Could be.

Steve: Could be.

Leo: No. You wouldn't do that to us.

Steve: We'll see. Stay tuned.

Leo: Steven "Tiberius" Pike. By the way, anniversary; right? Next week the 50th anniversary of the launch of "Star Trek," I believe, 1966. Is it September? I think it's in September. I'm not sure exactly when. Denise Howell tweeted it. Thank you, Steve. We'll see you next time on Security Now!.

Steve: Thanks, Leo.