CheckM8

Description: This week we take a look at a sobering supply chain proof-of-concept attack, an update on the ongoing encryption debate, a blast-from-the-past password decryption, an intriguing security and privacy consequence of today's high-resolution consumer cameras, and the sad state of consumer security knowledge. OpenPGP gets a nice boost, Windows Defender gets Tamper Protection, and SQRL gets a very nice mention by Google's Cloud Security architects. We'll share a bit of sci-fi and fun miscellany, then conclude by examining the crucially important, widely available, and completely unpatchable Apple Boot ROM exploit known as "CheckM8."

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-736.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-736-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with lots to talk about, including supply chain security, end-to-end encryption - it's coming to Thunderbird, very excited about that. And they finally cracked Ken Thompson's Unix password. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 736, recorded Tuesday, October 15th, 2019: CheckM8.

It's time for Security Now!, the show where Steve Gibson saves us. Basically, we're going to change all the promotional copy to just "Steve Gibson saves us all." That's all we need to say.

Steve Gibson: Or sometimes worries us, sometimes confuses us.

Leo: Scares the hell out of us and then saves us.

Steve: That's right.

Leo: Goes hand in hand. Hi, Steve. How are you?

Steve: Great. And great to be with you for Episode 736 for the middle of October. Something really interesting has happened that I wanted to put into context because there's been some indubitably - or not indubitably, invariably is what I was looking for - invariably some confusion about...

Leo: Inevitably.

Steve: Inevitably. Inevitably.

Leo: There you go.

Steve: What is probably going to be a significant event for the industry because a security researcher/hacker found a flaw in all Apple iOS boot ROMs...

Leo: Oh, boy.

Steve: ...from the A5 through the A11, which is like from the iPhone 4S to up and including the iPhone X, which I have. Not the next generation, the X whatever they are, XX, the XS and the...

Leo: 12 and 13, the A12 and A13, yeah.

Steve: Yeah. So not the A12 and 13 Bionics, but up through the A11. What's significant about this is that it cannot be fixed because it is in the Boot ROM, which is by definition, you know, it's the ROM that's burned in at the factory. And anyway, so the exploit was called CheckM8, C-H-E-C-K-M-8, which is what we're going to talk about as our main topic at the end of the podcast. And our listeners who are sick and tired of hearing about ransomware will be glad that, except for that, I will never mention the word during this podcast because there's a lot of other stuff to talk about. We have a rather sobering supply chain proof-of-concept attack. And I want to take a minute to use that to talk about supply chain attacks because, although we've had some false reporting, they are real.

We also have an update on the ongoing encryption debate. A fun blast-from-the-past password decryption that you probably already know about. Anyway, we will make sure all of our listeners do. An intriguing security and privacy consequence of today's high-resolution consumer cameras. The not very surprising, but interesting to have numbers, sad state of affairs with the knowledge of security in American consumers. A nice piece of news about OpenPGP. You've always been, Leo, a long PGP booster. And a mainstream email client is going to get OpenPGP built in natively.

Leo: Oh, that's great news. That's awesome.

Steve: Yes, yes. Also Windows Defender gets Tamper Protection. SQRL gets a very nice mention by Google's Cloud Security Architects.

Leo: Ooh.

Steve: We've got a bit of sci-fi. Yeah. A bit of sci-fi and fun miscellany. And then we're going to finish by talking about what it means to end users, what it means to the security

industry, the fact that essentially any earlier devices, many necessarily still supported by Apple, are now jailbreakable, and there's nothing Apple can do. And that's got big consequences.

Leo: CheckM8.

Steve: So I think another - yes, CheckM8. Well named.

Leo: Well named.

Steve: Well named. Another great podcast for our listeners.

Leo: I'm excited. Are you going to talk about the Sudo flaw, the S-U-D-O flaw?

Steve: You know, I meant to. It was in my notes. And I didn't.

Leo: Okay. Because that's a - I don't know if it's something of real concern. My suspicion is it's not. But every Linux is updating. Sudo is used to escalate your non-privileged account to a superuser. And it's a great bug because, if you use the user minus one, it just authenticates without a password, and suddenly you're superuser. And of course you know exactly what happened when you hear "minus one." You go, oh, okay. It's a buffer overflow.

Steve: Yeah, wow.

Leo: And, you know, because Sudo is on everything, including all Macs, as well as all Linux computers, it's only Windows that wouldn't be affected by this.

Steve: Yeah. In fact, I have to use it, and I do, when I'm playing with my Drobos because the normal user account is not privileged, and so you have to elevate yourself in order to do stuff with the file system.

Leo: Yeah. It's probably not going to affect anybody; but nevertheless, everybody is updating right now. So that's something to be aware of. But we'll take a break, and we'll come back, and I want to hear all the story that you've got about this flaw in iOS because god knows we're all using it. All right. Thank you, Mr. Steve.

Steve: So I titled this "A sobering reminder about supply chain attacks." They're not common. And, no, they won't be employed by random 400-pound antisocial hackers living in their moms' basements. But anyone who completely discounts them and imagines that they are impossible is fooling themselves.

Andy Greenberg, writing for Wired magazine, reintroduced the idea by reminding us. He wrote: "More than a year has passed since Bloomberg Businessweek grabbed the lapels of the cybersecurity world with a bombshell claim" - as we covered at the time, as we

well know, and watched them being debunked - "that Supermicro motherboards in servers used by major tech firms, including Apple and Amazon, had been stealthily implanted with a chip the size of a grain of rice that allowed Chinese hackers to spy deep into those networks. Apple, Amazon, and Supermicro all vehemently denied the report. The NSA dismissed it as a false alarm. The DEF CON hacker conference awarded it two Pwnie points for 'the most overhyped bug' and 'the most epic fail.' And no follow-up reporting has yet affirmed its central premise."

Leo: However, it's not been debunked, either.

Steve: Exactly. And what captivated me, as our listeners may remember at the time, was the total feasibility of it. I mean, there was, like, yes, this could happen. And the fact that it was a false positive report this time doesn't mean that it's never going to happen. So, and actually that was sort of the impetus for an event that will be occurring next week in Stockholm, Sweden during the - and there's no way to say this. It's CS3STHLM.

Leo: Yeah. Yeah, baby.

Steve: Yeah, rolls right off the tip of the tongue. It's a security conference which bills itself as the premier cybersecurity conference for the ICS/SCADA and Critical Infrastructure world.

Leo: I'm guessing they probably call it CS3 Stockholm.

Steve: Ah, I bet that's what it is, yes. So during that conference, security researcher Monta Elkins, who works as the self-described "hacker in chief" for the industrial control security firm FoxGuard, will be showing off his handiwork. He will be next week vividly demonstrating just how easily spies, criminals, saboteurs with even minimal skills - I mean, literally the guy in his mother's basement could do this if he were motivated to and had some technical background - working on a shoestring budget can implant a chip in enterprise IT equipment to create a stealthy backdoor for themselves.

So in this instance, Elkins, armed with $150 hot air desoldering tool, a $40 microscope, and some $2 chips ordered online - and you're showing it now. This is the Picture of the Week for the podcast, Leo. That little red circle in the lower left is the chip he added, the $2 chip he added to a Cisco ASA 5505 firewall appliance.

Leo: So this is a small board by itself. It's kind of hard to tell the scale.

Steve: To get a scale for it, yes.

Leo: Yeah, it's just a few inches across.

Steve: Yeah, maybe eight inches, kind of eight by eight. Anyway, so Andy Greenberg interviewed Elkins, who said: "We think this stuff is so magical, but it's not really that hard. By showing people the hardware, I wanted to make it much more real for them." He says: "It's not magical. It's not impossible." He says: "I could do this in my basement.

And there are lots of people," he says, "smarter than me that can do it for almost nothing."

So what he did was he used the ATtiny85 chip, which is about 5mm on a side. It's got four tiny mounting and connecting pins on opposite sides from each other. He pulled the chip from a $2 Digispark Arduino board. He says it's not quite the size of a grain of rice, but it's tiny. After writing his code into that chip, he desoldered it from the Digispark board and soldered it to the motherboard of that Cisco ASA 5505 firewall appliance. He found an inconspicuous spot that required no extra wires, which could give the chip access to power and the firewall's serial port.

And as I mentioned, our Picture of the Week shows it. I have a larger version of it a couple pages down. He noted that he could have used an even smaller chip, but chose the ATtiny85 because it was easier to program. He says he also could have hidden his malicious chip even more subtly inside one of several radio frequency-shielded cans on the board.

Leo: Yeah. I mean, if you looked at the motherboard you would see this.

Steve: Oh, yeah. But looking around, I mean, unless it was circled, we wouldn't know.

Leo: Yeah, yeah. Good point, yeah.

Steve: You know, it looks likes all the other ones.

Leo: It looks a little bit hand-soldered.

Steve: Yeah, there's a little lumpiness, yeah. It's a little lumpy, yeah.

Leo: But still.

Steve: Anyway, so - and he deliberately wanted to be able to show, you know, to hold the board up at the security conference, he says, in order to point to it and say here it is. So what does this tiny, itty-bitty chip do? Elkins programmed his tiny stowaway to carry out an attack as soon as the firewall boots in a target's datacenter. It impersonates a local security administrator who would access the firewall's configuration by connecting their computer directly to the firewall's hardware admin port. So at boot the chip triggers the firewall's password recovery feature to create a new admin account for itself, which then allows subsequent full remote admin access to the firewall's configuration by a remote attacker.

He noted that he chose the Cisco's ASA 5505 firewall for his demo because it was the cheapest one he found on eBay. But he noted that any Cisco firewall that offers that sort of recovery in the case of a lost password - and they all do, you know. They presume, if you're able to attach a serial cable to the serial port of your computer to the firewall, well, then, you know, you're God of that device. You are physically proximate to the device, and you're able to implement serial commands, not Internet port-based commands. So he also noted that with a bit more reverse engineering it would also be possible to reprogram the firmware of the firewall to make it into a more full-featured

foothold for spying on the victim's network, though he didn't go to any trouble of that sort for his simple proof of concept.

So I liked this story because I wanted to highlight the triviality of this proof of concept for our listeners. For non-hardware types, this sort of thing might seem like exotic sci-fi. But in today's world, it really isn't. It is extremely possible to hide this sort of thing in plain sight. Think of it like what's happened with today's operating systems. The graybeards among us - you and I, Leo, and probably half of our listeners - probably fondly recall those days of yesteryear when we knew and could identify every file on our 10MB hard drive of our PC. Back then we knew the five files that MS-DOS was using. And when you installed a piece of code, you installed its executable and then an ini or a config file or something. I mean, we knew exactly what was going on. And we sort of thought of it as that's the way it's supposed to be.

Well, needless to say those days are long gone. Now we have no idea what the heck is loaded on our multi-gig, if not multi-terabyte drives. And if you look at the process monitor, you don't even know what is running on those machines. It's just full of stuff. So talk about hiding in plain sight. Similarly, looking at any modern motherboard, or a security appliance like that firewall, and we saw the picture of it in the Picture of the Week, we see a literal sea of tiny chips. We have no way of knowing whether they are from the factory or not, especially if they are cleanly and carefully added.

And again, I'm not wearing a tinfoil beanie here. I know that this likely doesn't affect any of us, and never will. But the lesson we keep learning, the thing we keep seeing is that what can be done is done. And we know that cyberespionage is today a real thing. And for a nation-state-scale actor, intercepting the physical shipment of a device is not difficult. And the payoff from implanting an undetectable hardware backdoor could be significant. And of course that's why the Bloomberg piece a year ago generated so much fervor is that no one discounted what it would mean if that was actually happening, if somebody had successfully planted a spy chip in a huge number of devices. Well, for a nation-state actor it doesn't have to be a huge number. You're not shooting a shotgun, you're targeting with a sniper scope. And so intercepting just one device bound for a destination that matters is theoretically all you need to do.

So anyway, so I thought about it. If I was really, really worried and had to be safe, what would I do? Well, one possibility for someone who's really concerned would be to ask a favor of a completely unaffiliated company to purchase the hardware that you want on your behalf and then pick it up from them. In other words, avoid the possibility of anybody monitoring the purchases that you're making and intercepting the hardware. And of course another possibility is to note that virtually all of these turnkey appliances are typically just offering prepackaged solutions. Anything they can do can probably be done just by gluing some off-the-shelf open source Linux or Unix apps together in any generic server PC. For example, Linux and Unix firewalls today are extremely capable. You can roll your own to do anything that you might purchase an off-the-shelf appliance to do.

So there are ways to work around this problem, if it was really something you were worried about. I just sort of wanted to take this opportunity, the opportunity of this guy making this presentation next week, so easily making a modification with, I mean, he didn't have engineering plans. He didn't have schematics. He did a bit of reverse engineering of the Cisco device in order to find the place to put the chip. But if he can do it, so can anybody else who is motivated. It's just not that difficult.

And in Andy Greenberg's piece in Wired, he also mentioned something that we didn't cover, which is to prove the point that the Bloomberg report could have been true, somebody did make that change to a Supermicro motherboard subsequently, just to demonstrate that it could be done. It was at, shoot, I can't remember the security

conference now where it was shown. But that was subsequently shown - oh, it was at the Chaos computing conference subsequently where he said, okay, so nobody found one, a Supermicro motherboard with that done. But he held it up. He says: "I'm holding one where I did exactly what the report alleged, just for the sake of saying it's possible."

So the point is it is possible. Again, it's not going to affect the majority of us. But I guess, for some agencies, hopefully they are taking some sorts of measures to protect themselves from this kind of supply chain problem because they are possible.

Leo: But they'd be targeted mostly; right? I mean, you're not going to do a...

Steve: Yes, exactly.

Leo: You're not going to say I'm going to take every - well, I guess you could, if you were the manufacturer, take every Linksys router and mess with it. Much more likely you'd interrupt a shipment of an important piece of gear, a server, say, to the Pentagon. You'd sneak in - that's why they call it "supply chain." You'd sneak in the middle, put something on it, and let it go on. That's much more likely, yeah.

Steve: Right. Right. And...

Leo: It's not something you and I should worry about.

Steve: No, exactly. And our listeners probably have no need to worry about it. But I would imagine - I just sort of wanted to just revisit the issue that that kind of attack is targetable and eminently feasible.

Leo: Absolutely.

Steve: Even though a year ago it was debunked.

Leo: It's almost certainly happening. We know the NSA has done it.

Steve: Yes, yes, exactly, exactly.

Leo: So it's not a proof of concept. It is a concept. It's happened.

Steve: Yeah, it's a demonstration, exactly.

Leo: Yes, exactly.

Steve: So last March 6th, Mark Zuckerberg famously posted his so-called, what was titled "A Privacy-Focused Vision for Social Networking." And I snipped out from a very

long thing sort of the salient piece of what he said. He said: "I believe the future of communication will increasingly shift to private, encrypted services where people can be confident what they say to each other stays secure, and their messages and content won't stick around forever." He says: "This is the future I hope we will help bring about." And of course, yes, it was met with a lot of skepticism, and we talked about it at the time, and I know you did multiple podcasts, used it for content on multiple podcasts because it was an interesting statement coming from Facebook.

He said: "We plan to build this the way we've developed WhatsApp." And of course he didn't. They acquired WhatsApp; right? And doesn't WhatsApp use Signal as its core protocol?

Leo: It does, yeah, yeah.

Steve: Yes. And so of course he was leveraging Moxie Marlinspike's tremendous work that we talked about in detail at the time. He said: "Focus on the most fundamental and private use case, messaging. Make it as secure as possible, and then build more ways for people to interact on top of that, including calls, video chats, groups, stories, businesses, payments, commerce, and ultimately a platform for many other kinds of private services." So all that sounds good, if they actually execute. And his posting was much longer, and it covers many other points. But basically the entire thing was a manifesto for privacy and unbreakable-by-anyone encryption.

Well, predictably, in our current environment where this whole question of government and law enforcement legal access to any and all private communications remains very much up in the air and unsettled, several governments have now formally pushed back against Facebook's declared intentions. In a 2.5-page open letter dated October 4th, so about a week and a half ago, which was cosigned by law enforcement authorities in the U.S., the U.K., and Australia, Facebook is strongly urged to halt its plans for stronger end-to-end encryption. And I won't bore our listeners with the entire letter, but I will just share the beginning of it to get a sense for it. It was addressed as an open letter.

"Dear Mr. Zuckerberg. Open Letter: Facebook's Privacy First Proposals." And so the U.S., the U.K., and Australia write: "We are writing to request that Facebook does not proceed with its plan to implement end-to-end encryption across its messaging services without ensuring that there is no reduction" - I love the way they phrased this - "no reduction to user safety and without including a means for lawful access to the content of communications to protect our citizens. In your post of 6 March 2019, 'A Privacy-Focused Vision for Social Networking,' you acknowledged that 'there are real safety concerns to address before we can implement end-to-end encryption across all our messaging services.' You stated that 'we have a responsibility to work with law enforcement and to help prevent,'" and he says, "the use of Facebook for things like child sexual exploitation, terrorism, and extortion."

They write: "We welcome this commitment to consultation. As you know, our governments have engaged with Facebook on this issue, and some of us have written to you to express our views. Unfortunately, Facebook has not committed to address our serious concerns about the impact its proposals could have on protecting our most vulnerable citizens." And so that's where they go with this is basically child sexual abuse.

Leo: Of course. They always do. That's where they go. Think of the children.

Steve: I know. I know. He says: "We support strong encryption, which is used by billions of people every day for services such as banking, commerce, and communications. We also respect promises made by technology companies to protect users' data." And we'll just stop for a second and note that, yes, as far as we know, there is no technology in place that allows the government to intercept our HTTPS connections during banking and commerce and communications. As far as we know.

Leo: As far as we know.

Steve: But we do know because we know how the system works.

Leo: The math works, yeah.

Steve: And that there are no backdoors in TLS at the moment.

Leo: Although we also probably know that they're collecting everything, as much as they can. They built that big server farm so that they can collect more data and then hope that they can crack it down the road, I guess.

Steve: Yeah.

Leo: Does TLS support Perfect Forward Secrecy?

Steve: It does, yes. TLS, yes.

Leo: Oh, so good luck, NSA.

Steve: Exactly. Exactly. You cannot do retrospective decryption once you obtain the key of the server in the future. So then they say: "Law-abiding citizens have a legitimate expectation that their privacy will be protected. However, as your March blog post recognized, we must ensure that technology companies protect their users and others affected by their users' online activities. Security enhancements to the virtual world should not make us more vulnerable in the physical world. We must find a way to balance the need to secure data with public safety and the need for law enforcement to access the information they need to safeguard the public, investigate crimes, and prevent future criminal activity." You know, I mean, even as I'm reading this, I mean, it is self-contradictory; right? I mean, they're contradicting themselves in what they're saying they want.

So they continue: "Not doing so hinders our law enforcement agencies' ability to stop criminals and abusers in their tracks." And I'll just read one more paragraph here because this is important: "Companies should not deliberately design their systems to preclude any form of access to content, even for preventing or investigating the most serious crimes. This puts our citizens and societies at risk by severely eroding a company's ability to detect and respond to illegal content and activity, such as child sexual exploitation and abuse, terrorism, and foreign adversaries' attempts to undermine democratic values and institutions" - oh, yeah, let's go there, too.

Leo: They hit the big three on that one.

Steve: Let's go there, too, yeah: "...preventing the prosecution of offenders and safeguarding of victims. It also impedes law enforcement's ability to investigate these and other serious crimes. Risks to public safety from Facebook's proposals are exacerbated in the context of a single platform that would combine inaccessible messaging services with open profiles, providing unique routes for prospective offenders to identify and groom our children," is what they wrote. So it then congratulates Facebook on the number of child sexual abuse cases they helped law enforcement with and details one particularly poignant case involving an 11 year old, which I'm not going to go into.

It concludes, saying: "Equally important to Facebook's own work to act against illegal activity, law enforcement rely on obtaining the content of communications, under appropriate legal authorization, to save lives, to enable criminals to be brought to justice, and exonerate the innocent. We therefore call on Facebook and other companies to take the following steps."

And we have four bullet points: "Embed the safety of the public in systems designs, thereby enabling you to continue to act against illegal content effectively with no reduction to safety, and facilitating the prosecution of offenders and safeguarding of victims; two, enable law enforcement to obtain lawful access to content in a readable and usable format; three, engage in consultation with governments to facilitate this in a way that is substantive and genuinely influences your design decisions; and, four, not implement the proposed changes until you can ensure that the systems you would apply to maintain the safety of your users are fully tested and operational." I'm not sure what that fourth one is about.

Then they conclude: "We are committed to working with you to focus on reasonable proposals that will allow Facebook and our governments to protect your users and the public while protecting their privacy. Our technical experts," they're writing, "are confident that we can do so while defending cyber security and supporting technological innovation. We will take an open and balanced approach in line with the joint statement of principles signed by the governments of the U.S., the U.K., Australia, New Zealand, and Canada" - as we know, that's the Five Eyes - "in August of 2018 and the subsequent communique agreed in July this year. As you have recognized, it is critical to get this right for the future of the Internet. Children's safety and law enforcement's ability to bring criminals to justice must not be the ultimate cost of Facebook taking forward these proposals."

And Facebook, for its part, just essentially replied with what is mostly a reiteration of some of its previous statements. Facebook replied to this letter, saying: "We believe people have the right to have a private conversation online..."

Leo: Yes.

Steve: "...wherever they are in the world. As the U.S. and U.K. governments acknowledge, the CLOUD Act allows for companies to provide available information when they receive legal valid requests and does not require companies to build backdoors. We respect and support the role law enforcement has in keeping people safe. Ahead of our plans to bring more security and privacy to our messaging apps, we are consulting closely with child safety experts, governments, and technology companies, and devoting

new teams and sophisticated technology so we can use all the information available to us to help keep people safe.

"End-to-end encryption already protects the messages of over a billion people every day. It is increasingly used across the communications industry and in many other important sectors of the economy. We strongly oppose government attempts to build backdoors because they would undermine the privacy and security of people everywhere."

Leo: And I would point out it's not just Facebook saying this. Edward Snowden today published an op-ed piece in The Guardian in which he said: "Without encryption, we will lose all privacy. This is our new battleground." He says: "If Barr's campaign is successful, the communications of billions will remain frozen in a state of permanent insecurity. Users will be vulnerable by design, and those communications will be vulnerable" - and this is the important part - "not only to investigators in the U.S., U.K., and Australia, but also to the intelligence agencies of China, Russia, and Saudi Arabia, not to mention hackers around the world." I think Snowden got it right. If you don't trust Facebook's response, I think it's safe to say Edward Snowden knows a little bit about this.

Steve: Yeah. I mean, really it boils down to will you allow a third party to have access? And, I mean, that's what it is. We've talked about the math. We've explained the math can do anything we want it to do. The question is, what do we want it to do? And as our listeners know, five years ago, when I first talked about SQRL, I explained that what it was was deliberately and explicitly a two-party solution. It's an authentication technology between you and the website that you are authenticating yourself to, without a third party.

And the lack of a third party created huge complexities for me in its design. It's part of what took so long was I had to deal with all of the what-ifs that would arise where there's no one to have an "Oh, I forgot my password" email recovery, you know, because there isn't. There's just the two of you, you and the site. But the point is it's a system, if we want it, for two parties. And the good news is it's not doing anything other than authentication, so it doesn't get into the crosshairs of our government in the same way. But, you know, this is exactly what Edward said, exactly what this letter from Bill Barr and his U.K. and Australia...

Leo: Compadres, compatriots.

Steve: ...compadres are saying. I mean, this is where we are. This is the battle. And the problem is, in the U.S., we have Congress. And the U.S. Congress is where laws originate. Our Congress is demonstrating that it's rather timid at the moment. More so than the U.K. I wouldn't be surprised if something happens first in the U.K. because they seem more willing to break things over there.

Leo: Well, hasn't it already happened in Australia? Did they not pass that law at the beginning of the year? I thought they did. I don't know what its upshot is. I don't think any - I think everybody's ignoring it.

Steve: Yeah, exactly. Ignore it as long as you can.

Leo: They said by law you have to be able to provide a cleartext version of any message. They didn't tell you how. Snowden says the true explanation for this is that it's not about public safety, it's about power. End-to-end encryption gives control to individuals and devices they use, not to the companies and carriers that route them. And that means government surveillance would have to become more targeted and methodical instead of their lazy, indiscriminate, universal surveillance. It's about power.

Steve: Yeah.

Leo: They always bring up the child pornography bugaboo. But that's just an excuse, and that's not - they don't care about that. That's not what this is about. We should fight this. Fortunately, the math is out there.

Steve: Yup.

Leo: I don't know if there's any way they can shut the math down. But maybe you can't use Facebook. If you want privacy, why are you using Facebook anyway?

Steve: Right. Well, and so ultimately that's what - we pretty much know what's going to happen. There will end up being legislation, and then our major companies need to decide what they're going to do. I mean, I don't imagine Apple and Facebook and Google can be outlaw organizations. And we've talked about this. And so if somebody then wants secure encrypted communications, you'll use some other tool, some add-on or some third-party solution.

Leo: It's my guess, you know, we see Tim Cook having dinner with the President. We see Mark having dinner with all sorts of interesting characters. It's my guess these companies are already doing it. And anything they say in public, like that letter from Mark, is really a distraction.

Steve: And I've argued that Apple already can. If it wants to add another key to our iMessage conversations, we have no way of knowing that that's what's happened.

Leo: They may well have done that. We don't know.

Steve: Yup.

Leo: We don't know. And I think you'd be foolish to trust them. Can we trust Signal? It's open source; right? You can check the source code.

Steve: Yeah. Although I'm trying to remember, see, I think Signal does some automatic key management. That's the crux. As I've long said, if somebody else is managing your keys, then you're not managing your keys.

Leo: And it uses your phone number as the identifier, which is not good, either.

Steve: Yeah. And in fact we have been seeing lots of SIM attacks and attacks on two-factor authentication that way. You know, it's why I gravitated toward Threema. Threema has the problem that it's not open source.

Leo: It's closed source, so you don't know what they're doing.

Steve: Yeah, exactly.

Leo: Somebody, Steve, needs to write a completely open, encrypted, end-to-end communication technology.

Steve: Where the user manages their own keys.

Leo: The user manages it. It won't be easy to use. You'll have to be responsible. But that's fine.

Steve: Exactly. You're going to have to have a tradeoff between convenience and security. I mean, that's just the way it is.

Leo: That's why Signal uses your phone number, because you can reauthenticate. At least they notify you if the key's been changed. But I think we need some - there must exist - it must exist.

Steve: And I think when we were talking about it, I'm trying to remember, in Signal and/or WhatsApp, you are able to display the user's QR code of your key and then physically show them to each other and pair them and, you know...

Leo: Which is important because you need to verify it. The one problem with WhatsApp is it doesn't let you know if the key's changed. Signal does, so - unless that's changed recently. But the last time I saw, WhatsApp, you can surreptitiously do a key change. Which means somebody could impersonate you, basically, with that. And you could check for that, but most people wouldn't.

Steve: Yup. So get a load of this, Leo.

Leo: Oh, boy.

Steve: Unix's co-creator Ken Thompson's BSD Unix password has finally, after 39 years, been cracked.

Leo: Oh, my god. You mean, they couldn't log into his account?

Steve: No. So, and it was a pretty good password.

Leo: Well, I bet it was.

Steve: So Ken Thompson, who was of course the co-creator of Unix...

Leo: He's still alive; isn't he? I know Ritchie passed away.

Steve: Yes, well, he's at Google. And he did Go, the Go programming language.

Leo: Oh, that's right. Right.

Steve: Okay. So the story begins five years ago when a developer, Leah Neukirchen...

Leo: Neukirchen.

Steve: ...Neukirchen spotted an interesting /etc/passwd file in a publicly available source tree of the historical BSD version 3 from 1980.

Leo: They left the password file in?

Steve: Yes, it was there. It included the hashed passwords belonging to more than two dozen Unix luminaries...

Leo: Oh, my god.

Steve: ...who worked on Unix development, including Dennis Ritchie, Stephen Bourne, Ken Thompson...

Leo: The Bourne shell.

Steve: Yup, the Bourne shell. Eric Schmidt...

Leo: Who was an intern at Bell Labs when they were creating Unix. I love it.

Steve: Yup. The Eric Schmidt, Stuart Feldman, and Brian Kernighan.

Leo: Kernighan.

Steve: Kernighan; right.

Leo: Kernighan and Ritchie is the classic C book, yeah.

Steve: That's right.

Leo: Wow.

Steve: Since those early passwords were protected using the long-since deprecated DES-based descrypt, D-E-S-C-R-Y-P-T...

Leo: That should be crackable.

Steve: ...and they were limited to at most eight input characters, Leah decided to brute-force them for fun. This was five years ago.

Leo: I hold in my hands, by the way, this was a board created by Cryptography Research.

Steve: I remember, yup.

Leo: This was a prototype. It actually failed. They only got one chip on it. But it was going to be filled with chips to crack DES.

Steve: Right.

Leo: This was all about - see, it says EFF, Cryptography Research, DES. They were going to prove that DES was unreliable.

Steve: Nice. Well, and itself it is a 56-bit block, which is now way too short. But that's why Triple DES, essentially three of them, creates a much longer block.

So anyway, so five years ago she started to tackle this. She successfully cracked the passwords for most of those Unix luminaries using standard off-the-shelf tools like John the Ripper and Hashcat. But the toughest ones to crack, which she was unable to crack, belonged to Ken Thompson and five other contributors who helped to build the Unix system, including...

Leo: Ken should be so proud.

Steve: Yes, including Bill Joy, who of course as we know later founded Sun Microsystems and designed Java for us.

Leo: Yeah. Wow.

Steve: So she wrote in a blog posting last Wednesday, she wrote: "Ken's password eluded my cracking endeavor. An exhaustive search back in 2014 through all lowercase letters and digits took several days and yielded no result." She noted that, compared to other password hashing schemes such as NTLM, descrypt turns out to be quite a bit slower to crack. And the problem was there was no special case software for it. No one was, like, bothering. You know, we can crack SHA-256 in a blink; but DES, just no one had bothered to special case it.

So earlier this month she posted all of her findings on the Unix Heritage Society mailing list, requesting help from other members to crack the remaining passwords. And six days later an Australian engineer, Nigel Williams, responded with the plaintext password used by Ken Thompson, which he cracked after four days using an AMD Radeon Vega 64 running Hashcat, which was testing about 930 million hashes per second. Thompson's password consequently has been revealed as...

Leo: Monkey123. Oh, no.

Steve: "P/q2-q4!."

Leo: Huh. That's chess. That's a chess move.

Steve: Exactly. Chess notation describing the move "Pawn from Queen's 2 to Queen's 4."

Leo: Oh, that's hysterical. That's actually a good idea for making - I'll have to remember that for making passwords. That's good.

Steve: It withstood the test of time.

Leo: It's only seven characters. Eight. Was eight the max that you could use?

Steve: Yes. You could only do an eight-character max.

Leo: Okay.

Steve: So the next day another mailing list member, Arthur Krewat, successfully cracked and provided the passwords for four more remaining uncracked hashes. Dennis Ritchie, who of course was the co-inventor of BSD and the creator of the C programming language, his was kind of simple: "dmac," D-M-A-C. Brian Kernighan's was "/.,/.,.."

Leo: That sounds like a shell command. At least a directory.

Steve: Stephen Bourne wasn't too concerned about security. His was "bourne."

Leo: Oh, dear.

Steve: B-O-U-R-N-E.

Leo: Oh, dear. Oh, dear.

Steve: Eric Schmidt, who of course is the former Google CEO, was "wendy," all lowercase, and three exclamation points.

Leo: Oh, dear. He was a dog even then.

Steve: And Stuart Feldman, who was the author of a Unix automation tool and the first Fortran compiler: "axolotl."

Leo: Axolotl, yeah.

Steve: Yes. Which is that walking fish thing, A-X-O-L-O-T-L.

Leo: That is bad. That's a dictionary word. That's the only one in the bunch that's a dictionary word, so that's the worst one; isn't it.

Steve: Yeah, that's true. Yup. But anyway, Ken Thompson's was a tough one to crack, and it was a chess move.

Leo: Oh, you know what? /.,/.,, that's actually pretty bad, too If you look at the keyboard, it's the lower right, /.,/.,..

Steve: Oh, you're right, three in a row backwards, yes.

Leo: That's a terrible - that's also a terrible password. Every hacker in the world probably has that in their crack set. Oh, man. I like p/q2-q4!.

Steve: That's pretty good.

Leo: And he gives it an exclamation mark, which means it's a brilliant move, so that's even better.

Steve: Pawn Queen 2 to Queen 4.

Leo: Yeah. Nobody would write it that way, but it's memorable. Like the Queen's Gambit. I love it.

Steve: So an interesting consequence, Leo, you'll get a kick out of this because it's about consumer high-resolution cameras. We were beginning to see consequences of this. We've covered the problem, for example, of just some keys being left out on the desk and how camera resolution is now so good, and coupled with computational capabilities, that a photograph of a key at a distance is enough to reproduce that key such that it will be able to open the lock. We've seen an instance of bouncing a laser off of a window to pick up the vibrations of the windowpanes. And in fact on this podcast we covered, you may remember, the balloons, inflated balloons in a room. Pictures of the balloons were used to pick up the conversation in the room...

Leo: Oh, from the vibrations.

Steve: From the vibration of the rubber surface of the balloon, which I thought was very cool. Well, now we have a true story of a stalker locating a celebrity by examining the reflections picked up in her eyes' irises. A Japanese stalker confessed to stalking and unfortunately attacking a young Japanese pop star by zooming in on the reflections that he was able to detect in her eyes...

Leo: Oh, my god.

Steve: ...in photos she posted on social media. After the assault, a 26-year-old man by the name of Sato...

Leo: That's horrible.

Steve: ...was arrested and confessed to police that he'd used the star's selfies to figure out where she lived. Each of her pupils reflected the nearby streetscape, which he was able to plug into the street map function of Google Maps to locate matching bus stops and scenery.

Leo: That's horrible.

Steve: Isn't that, like...

Leo: What a nightmare. That's straight out of "Blade Runner."

Steve: He confessed to observing other reflections in Matsuoka's - I don't know how to pronounce her name, M-A-T-S-U-O-K-A.

Leo: Matsuoka.

Steve: Matsuoka's eyes - curtains, windows, and the angle of the sun, which enabled him to guess which floor she lived on in the building.

Leo: Wow. That's - wow.

Steve: So isn't that something?

Leo: Center, enhance, zoom. Center, enhance, zoom. Wow. That's terrible. What kind of nut - geez.

Steve: I know.

Leo: Horrible.

Steve: But the information is there, which is, like, chilling. Eliot Higgins, the founder of investigations site Bellingcat, which has pioneered online investigative techniques, told the BBC that the better quality the image, the more potential there is for it to be used in geolocating us. He said: "Higher quality images allow for more details to be identified that can help with geolocation; and the more reference imagery there is from services like Google Street View, the greater the chance of determining a location. Even the tiniest details can reveal a lot of information about where a photograph is taken, and information about the individuals in the photograph."

And of course, as we know, the photo's EXIF data, which may include the GPS coordinates of a photo, can be used to do the same thing. And Google's computer vision specialists have worked to train deep learning machines to determine the location of almost any photo just by using its pixels and relying on image retrieval. Google has access to an extraordinary number of images on which to train its deep learning systems. So now we have one more thing to consider when posting photos online, and that is that subtle clues can exist.

Leo: But we know you shouldn't post pictures of your keys.

Steve: Right.

Leo: Because you can from a photo duplicate a key. But, you know, I think back to Tech TV days. This was around 2000. One of our hosts used as her image a picture, you know, a headshot. But it was a topless picture of her that she had cropped down. And what she didn't realize was the image editor that she used to crop it didn't delete the thumbnail of the full image from the file.

Steve: Oh.

Leo: So somebody was able to get the full image and propagate it. It was very embarrassing.

Steve: Wow. Wow.

Leo: These things happen all the time. And, you know, you hear all the time about redacted documents.

Steve: Oh, exactly, where you just take the black layer off.

Leo: So times have moved fast, yeah.

Steve: Yeah, that is a reminder for people, by the way. If you black out areas on a PDF, print the resulting blacked out PDF to another PDF, rather than just send the PDF with the blackouts because those are removable.

Leo: Still cracks me up.

Steve: Yeah. So Pew Research Center shared the results of their survey.

Leo: Oh, this was interesting, yeah.

Steve: Yes, of Americans', I'll say, lack of understanding of technology-related security issues. They found, not surprisingly, that a majority of U.S. adults were able to answer fewer than half of the survey's digital knowledge quiz questions correctly. And, frankly, I'm surprised that a broad swath of America did as well as this shows.

Leo: Well, some of these questions, like the last question, can you identify a picture of Jack Dorsey? They showed them a picture of Jack Dorsey.

Steve: Yeah, well, I would not - I would flunk that one.

Leo: Most people would fail that one. That's...

Steve: I have no idea what...

Leo: That's not digital literacy. That's like digital gossip knowledge.

Steve: Yeah. I would have no idea.

Leo: It was the last question. Maybe they just threw that in for fun. I don't know.

Steve: Yeah. They did note that people tended to say "not sure" rather than guess. So I thought that was sort of, you know, I'm a big fan of saying "I don't know." And so that was - it was nice that that was the case.

So a majority of U.S. adults were able to correctly questions about phishing scams and website cookies. I thought that was good. Because there's been enough in the popular press, I think, about that stuff, and people talking about it at cocktail parties, like, oh, do you flush your cookies? And it's like, my what? And so then they go poke around their browser and find out about that. But other items were more challenging. For example, just 28% of adults were able to identify an example of two-factor authentication. Still, one out of four? I think that's kind of good, actually.

Leo: You know what we don't know, I'd love to see the images. Apparently they used a set of images and said, "Which of these is two-factor?"

Steve: Oh, okay.

Leo: So, you know, this might not - that Jack Dorsey question casts the whole thing into doubt, to be honest with you.

Steve: Yeah, like was Pew appropriately qualified to even put a survey like this together.

Leo: Yeah.

Steve: For me, my own takeaway is I guess what annoys me about these questions is that our moms should never have to know...

Leo: They shouldn't have to. No, no.

Steve: ...any of this stuff.

Leo: Do you know who this is, Steve? Do you know?

Steve: No idea.

Leo: That's Jack Dorsey. You see? You'd have failed. You'd have failed miserably.

Steve: Yeah. Yeah. Yeah.

Leo: By the way, I can see where you are by the reflection in your glasses. I just want to say...

Steve: In fact, you could probably read that screen from my eyeglasses.

Leo: Almost can, yeah. Almost can, yeah.

Steve: Yeah, yeah.

Leo: But I do think - I think it makes perfect sense that only a quarter of America knows what - I think they've probably heard of two-factor. You don't need to know what it is to turn it on.

Steve: True. And you don't even have to know its name.

Leo: Right.

Steve: That's a very good point. It's like, oh, it's that extra six-digit thing I have to put in.

Leo: Yeah, right. I would hope they would know to turn that on. And of course everybody listening to this show does. But tell your friends and family. I just wrote an article for the AARP magazine, the American Association for Retired People, old folk magazine.

Steve: Of which you and I are soon entering.

Leo: We both get it. I'm sure I get it. But on two-factor, explaining what it is and why you should use it. Because I think it is so important that people know this stuff. So Pew's got a good point here. I'm not knocking their point.

Steve: No. But again, my take is that all of this is, in a sense, deeply abusive. One of their things was that most people don't know what https:// is. It's like, oh, my god.

Leo: Why should they? Why should they?

Steve: Yes, why should anybody have ever been exposed to this?

Leo: Tim Berners-Lee, the guy who created the World Wide Web, has said, "I never expected anybody to see http://. This was for machine-readable purposes only. This is not supposed to be a human-readable [crosstalk]."

Steve: Yes. And in fact I've told the story of how a dear friend of mine who is about the least techie woman on the planet just thinks Google is the Internet. Like she puts it into the Google, and that's how she goes places. And she doesn't know that it's not the Internet.

Leo: She's kind of right.

Steve: So, you know, it was created by techies for techies. But of course most of the world are non-techies.

Leo: Right.

Steve: And when I was thinking it, as I'm sort of mulling this over, remember back in the early days of automobiles, those darned horseless carriages were breaking down all the time.

Leo: I remember that, yeah.

Steve: They were unreliable.

Leo: That was terrible.

Steve: And finicky as hell. So in fact, anyone driving one needed to also be a bit of an auto mechanic.

Leo: That's right.

Steve: And that was part of the fun because those early autos were simple and understandable. And in fact my own first car was a burgundy-colored Fiat 850 Spider. I completely took it apart in our garage and rebuilt it from the ground up.

Leo: Nice.

Steve: So I knew and loved that car inside and out. And if something broke, I knew what and where I could go to get a part and fix it myself. But not anymore. The car I'm driving today is not, as they say, "user serviceable." But what it is in return for being a black box is it is incredibly reliable. I get in, and I turn the key, and it goes. It does what it's supposed to do.

And so my feeling is we're heading in that direction, but we're not there yet, with today's computers. And I think that needs to be our target. These things should be black boxes. It's fine. They are becoming less and less comprehensible. As we were saying before, I look at the process monitor in my PC. I have no idea what that crap is. It's just filled up with stuff. It's all busy doing things. And it's like, okay. And they're not yet as reliable as we need them to be, but it's okay if they become opaque as long as they deliver reliability and security, very much like today's autos. And I know, Leo, you sort of live on the cutting edge of auto technology, so you may not be able to put the key in and drive off.

Leo: I don't have a keyhole.

Steve: Ah. That's even better. You just approach.

Leo: I just walk up to it. You know, that was one thing I miss about the Tesla. There is no start button. You just put it in gear and go. Because you don't need a start button. What is that button for; right?

Steve: You're right. You're right.

Leo: Just go.

Steve: It was just servicing old-school mentality.

Leo: That's exactly right, yeah. If you put it in gear, it should just go.

Steve: So you will be glad to know, Leo, speaking of black boxes, OpenPGP will be built into Mozilla's Thunderbird email client.

Leo: Nice. Nice.

Steve: And it's the one I settled on. When Eudora finally just got too long in the tooth even for me, I tried a whole bunch of different clients, and I settled on Thunderbird, thinking, well, okay. I like Mozilla. It'll be kept up with standards. It'll be moving forward. Thunderbird currently has a third-party plugin, Enigmail, E-N-I-G-M-A-I-L.

Leo: It's kind of hard to use, yeah.

Steve: Yes. And it requires users to separately install additional third-party software like GnuPG or GPG4Win before installing Enigmail itself. And unfortunately, currently the licensing governing those libraries are incompatible with Thunderbird's, MPL v2 versus GPL v3+. So the Thunderbird folks will need to find a compatible library. But they've formally announced they're going to, starting with Thunderbird 78, which I think we're on 65 or something right now. So it's scheduled for release next summer. So summer of 2020, OpenPGP will be built in and fully supported natively.

Leo: So you won't have to download a PGP tool or anything like that, just have the full...

Steve: Anything.

Leo: That's really great.

Steve: Yeah, it'll just be there.

Leo: I mean, there are - I use Claws Mail on Windows, Mac, and Linux, and that supports PGP. But you have to have PGP installed. And it goes out and calls it, kind of like Enigmail would. So I think that's great. I hope they make it a very easy thing to do, to generate and use a key.

Steve: Yeah, and manage your key. I'm sure they will.

Leo: Yeah. That's great, yeah.

Steve: And so we will definitely be talking about that when that happens next summer. And then, gee, Leo, I'll have PGP. Whoo.

Leo: PGP is, I mean, first of all, there are problems with it. It's very old. It's not how you would design it if you were going to design it today. But it's not like we have anything much else out there.

Steve: Right.

Leo: I use it mostly for signing, to kind of verify that I created this.

Steve: Right, right.

Leo: Cool.

Steve: So a little bit of Windows 10 news. Windows 10's Tamper Protection is being enabled by default. And when you think about it, it really doesn't do much good to have Windows Defender watching the store if it can simply be turned off by sufficiently clever malware. And there have been increasing reports of exactly that happening recently.

So yesterday, on the 14th, Microsoft announced that the new Windows 10 Tamper Protection security feature, which was added to Windows 10 in this most recent big update, 1903, also known as the May 2019 update, is now officially available for both enterprise and consumer users. Along with this announcement, Microsoft will be enabling the security feature on all Windows 10 devices, obviously which need to have 1903 in them first, by default.

When enabled, as it will be by default, Tamper Protection prevents Windows Security and Windows Defender settings from being changed by programs, Windows command line tools, registry changes, or group policies. Basically, nothing programmatic can change it. The only way to configure it and/or modify its settings will be directly through the Windows 10 user interface or, in the case of enterprise users, with some management software such as Intune. And before starting the podcast I fired up this Windows 10 machine that I'm talking to you on, Leo, and it was there in the settings. It was off by default.

So for what it's worth, any of our listeners who are interested, who are Windows Defender users, who did update to this most recent Win10 1903, just can go into your

update and security page under Windows Defender or whatever settings it's called. I won't bring it up because I don't want to pause. But anyway, it's obvious where it's there, and you'll find a switch which is called Tamper Protection. Mine was off with a little yellow warning sign. I flipped it on and made it happy. And so now I've got it on, and it cannot be turned off, so says Microsoft, by any means other than doing it manually through the UI. So a nice little bit of forward motion on Microsoft's part for security.

Two little bits of miscellany under the sci-fi and fun category. Lorrie and I and my best friend saw "Ad Astra."

Leo: Oh, I'm curious about that. Brad Pitt's new space opera. What did you think?

Steve: I can't recommend it highly, as I was hoping I might. I described it here in the show notes as "Brad Pitt wanders around through outer space."

Leo: That's what it looks like in the trailers, too. Okay.

Steve: Yeah, it never gets much better than that.

Leo: Oh, that's terrible.

Steve: So it was meant to be set in an interesting future. And as our listeners know, I am just steeped in sci-fi. And so it was an interesting take. There was no warp drive. There was no beaming. It was sort of like the reality moved forward another hundred years. So there was like a Moon base and a Mars base. And we had some bases, like, further out in the solar system. No one quite explained to us why, but there was a base out at Neptune that was doing some things, and it's like, okay. And so it was sort of like, and there was sort of a commercial - you could buy a ticket to go to the Moon. And so it was just sort of an extension of where we are now. But no, like, exotic technology, so it required patience to go anywhere. And Brad was very patient. And the watcher of the movie was asked also to be very patient.

Leo: Oh, dear. No kidding. This does not bode well.

Steve: It's definitely a maybe, you know, wait for it to come out on some free streaming service that you're already subscribed to. And get a big bag of popcorn.

Leo: Good tip.

Steve: Okay. Now, the fun piece is Friday we got an extension to "Breaking Bad."

Leo: Oh, I haven't seen it yet.

Steve: And it was wonderful.

Leo: This is "El Camino."

Steve: Two hours. It is what happened to Jesse after he drove off at the end. The last scene of "Breaking Bad" was he had escaped/gotten out of his cage.

Leo: See, I wasn't clear at the end of "Breaking Bad" whether Jesse had survived.

Steve: Well, in what mental state had he survived.

Leo: Right.

Steve: It's a little over two hours. It was written by what's his name, Gilligan, who wrote the original, produced by - and of course now I'm blanking on the actor's name.

Leo: Bryan Cranston.

Steve: Bryan is in it.

Leo: Oh, he is?

Steve: Yes. There are some - they had so much fun. They went back and filled in some back story that we weren't originally given in order to carry the storyline forward. So this is like - it's like the missing chapter.

Leo: Oh, I can't wait.

Steve: It's the final closure that all "Breaking Bad" fans have been waiting for, and it will not disappoint you.

Leo: I'm so happy.

Steve: And this actor is so good. Holy crap. I mean, he is so good. Aaron Paul, that's the name I was trying to remember.

Leo: Aaron Paul, yeah, Jesse Pinkman, yeah.

Steve: Yes.

Leo: And Vince Gilligan is the author.

Steve: Yes. And he wrote this.

Leo: He created it, yeah.

Steve: And so I just wanted to make sure that all of our listeners who were "Breaking Bad" fans knew that Netflix did this two-hour movie. It's not another series. It's a two-hour final conclusion to what happened after Jesse drove off in his El Camino. And it's...

Leo: I have to say, I mean, I watched "Better Call Saul." I mean, I'm such a "Breaking Bad" fan. I can't wait. I'm saving it because Lisa's in New York. So when she gets back...

Steve: Yup, do it.

Leo: I look forward to it, yeah.

Steve: Do it.

Leo: Good. "El Camino."

Steve: It's got top recommendation. "Ad Astra," no. "El Camino," all thumbs up.

Leo: Yeah.

Steve: And a Twitter follower of mine, certainly a follower of the podcast, tweeted to me a pointer to something that I was so thankful for. There are two documents produced by Google, "Modern Password Security for Systems Designers" and a complementary "Modern Password Security for End Users." And this was what to consider when building a password-based authentication system, written by Ian Maddox and Kyle Moschetto, who are Google Cloud Solutions Architects. And there's a chunk on SQRL.

Leo: All right. All right.

Steve: So Google Cloud Solutions Architects, under "Alternatives to passwords," for SQRL they wrote: "The Secure Quick Reliable Login protocol is a recent addition to the security space. It is designed for end-user authentication to websites and applications. SQRL users run a small client application on their modern password security for system" - this looks like I've scrambled up - "run a small client application on their computer..."

Leo: It looks like some garbled words, yeah.

Steve: Yeah, I mangled the text somehow when I did a copy and paste - "...or in their browser. Instead of giving servers a password that they must keep secret, the client

provides a public key that is unique to the application or domain the user wants to authenticate to. The server provides a unique value to the client, and the client must then use their private key to sign and return that secret. The server verifies the signature by using their public key and authenticates the user. Most importantly, a compromised site or service cannot expose its users' credentials in a way that impacts any other site or service." So these guys got it perfectly. They said: "Users can expect to see the option for SQRL login to appear in more places in the coming years."

Leo: Right on, right on, right on.

Steve: "Developers and software architects will appreciate the deep level of technical detail written with an eye toward an evolving security landscape and the way real humans interact with security controls." And that deep level of technical detail was a link in the PDF, in Google's PDF, to the SQRL explainer doc.

Leo: Right on. Bravo, Steve. That's great.

Steve: That's very cool. And thank you to Ian and Kyle for the inclusion of SQRL in this official Google Cloud Solutions Architects documents. So very, very cool.

Leo: That's really good, yeah. All right. Let's talk about CheckM8. And we ain't talking p/q2-q4!.

Steve: No.

Leo: No.

Steve: A recently discovered, unpatchable, iOS Boot ROM exploit. Two weeks ago the iOS jailbreaking community received a welcome surprise when a security researcher going by the handle "axi0mX," A-X-I-0-M-X, dropped what's been described as a "game-changing," and I agree, new exploit affecting, well, pretty much all of Apple's mobile platform. He called it "CheckM8." It's a Boot ROM exploit which is being widely proclaimed the most important single exploit ever released for iPhone, iPad, Apple TV, and Apple Watch devices.

So what does that mean for us? First of all, CheckM8 will most likely, at least initially and directly, prove to be a massive boon for the security researchers who are wishing to peek under the hood of iOS devices, despite Apple's every attempt to prevent that. The only real threat to end-users might be that, by allowing researchers and hackers to get in under the hood, it will almost certainly facilitate the discovery of other weaknesses, which are, as we know, all too possible, since Apple is constantly patching discoveries which occur. This is going to make those discoveries probably more likely.

But for now, at least, CheckM8 itself does not directly affect end users. For one thing, there's no remote execution path. An attacker cannot use CheckM8 to compromise an untethered device. Which means that anyone who wanted to use this exploit would need to physically have the device and have it tethered. Also it does not allow any sort of threat actor to bypass a device's Touch ID or built-in existing PIN protections. In other

words, it does not permit any compromise of the Secure Enclave which exists in these devices.

So the user's personal data continues to remain safe from attackers who are lacking the device's unlock credentials, that is, notwithstanding the possibility of there being other zero days that are not known. But this is in itself a testament to Apple's multilayered security design philosophy. The idea that even a full boot compromise, which is what CheckM8 enables, would still leave the user's private data secure should be comforting to all Apple device users. And there's no persistence mechanism. If an attacker were to gain physical access to an affected device and use the Boot ROM exploit to compromise that device, rebooting the device would restore its normal boot chain security, and any changes that the attacker may have made in RAM would be lost as Apple's security checks, which would then again be effective, would either delete any files modified by the attacker or would refuse to run them.

So, okay. Which devices? As I mentioned before, only the most recent devices are immune because Apple did discover this summer before last. So the most recent devices - an iPhone XR, XS, XS Max, or any iPhone 11 series - all of which use the A12 Bionic or later chip, the A13, that Boot ROM exploit will not work because Apple did find it.

There is a use-after-free vulnerability which axi0mX found which appears only in devices using the A11 chips or earlier. So that's from the iPhone 4S, which uses the A5 chip, all the way through the iPhone 8 and X models, as well as any iPad, Apple TV, or Apple Watch using A11 or earlier chips. But on the other hand, think about that. I mean, that's most generations of iPhones and iPads which are vulnerable.

Leo: Now, how vulnerable is vulnerable, though? I mean, a jailbreak nominally means you can use some store besides Apple Store. Is this really like rooting the phone more?

Steve: So yes. So what it means is that researchers and hackers will be able to get in and look around. They'll be able to dump these files that Apple has protected and reverse-engineer them.

Leo: But they'd have to get physical access to your phone to do it.

Steve: Well, okay. So that's the point. I don't think people care about our phones. They care about Apple's code.

Leo: Ah.

Steve: What this means...

Leo: So this is more bad for Apple than it is for us.

Steve: Oh, my god. It is horrible for Apple.

Leo: I get it. I get it.

Steve: Yes.

Leo: Okay.

Steve: So on September 11th, so like two weeks ago, axi0mX tweeted - this is the fifth in a series of his tweets. I have it in the show notes. He said: "During iOS 12 betas in summer of 2018, Apple patched a critical use-after-free vulnerability in iBoot USB code. This vulnerability can only be triggered over USB and requires physical access. It cannot be exploited remotely." He says: "I am sure many researchers have seen that patch."

So what has happened now is, on GitHub is the full exploit of this use-after-free vulnerability which Apple fixed last summer. So what does it mean for end-users? Nothing. Apple is definitely not happy, whereas researchers and hackers are dancing a jig. But that said, if a paranoid person, or perhaps someone who might be explicitly targeted, wants some takeaways from this, the only risk would be if their device were outside of their control. In such a case, it would be possible for the phone to be jailbroken and for transient surveillance malware to be loaded into the phone.

Leo: Right, right.

Steve: So if you've left your phone unattended and powered on in your whatever, a hotel room or on a desk in a shared office environment or whatever, or for example if it was temporarily confiscated by border security guards while you were traveling into China, you might consider rebooting the iOS device once it's back in your possession. And you should do like a forced restart to ensure that malware hasn't found a way of simulating a fake reboot. So do a full deep restart. That will reestablish the secure boot chain, and you'll be okay.

We've previously talked about the way secure boot chains work. I have a graphic here on page 14 of the show notes, which is from Apple's 2016 Worldwide Developer Conference presentation, showing the concept of the flow of the secure boot chain from power-on of an uncompromised device, where the Boot ROM, which is actually a ROM, gets control of the processor. It then looks up at the Low-Level Bootloader, the so-called LLB. And before it loads it and transfers control, it verifies Apple's signature. And if the signature verifies, then it loads it and transfers control.

It then does its Low-Level Bootloader work, beginning to initialize and set up the hardware. And then it wants to transfer control to iBoot. Before it does that, it goes out and, similarly, verifies the Apple signature on that code which nobody knows how, thanks to a cryptographic signature, nobody knows how to spoof that. That cannot be modified from the time Apple has done it. And then it loads it and transfers control to it. iBoot does the same thing for the kernel. So it brings the kernel in, verifies the signature, gets the kernel going. The kernel does the same thing for iOS, verifies the iOS signature, loads it, and gets it going.

So each step of that startup process verifies the signature, the digital signature signed only by Apple, using a super secret private key that doesn't exist in the device, never exists outside of Apple, in order to sort of literally to bootstrap itself, one module successively after another, until it's up. What makes the CheckM8 so devastating is that it exploits a flaw which exists in the ROM that nothing can patch. Apple can't go back and fix it. It would require a factory recall, a physical recall of those devices because it's in the ROM. So it intercepts it before the first signature check, which essentially allows the

entire boot chain to be intercepted and all of that code to be studied, reverse engineered, dumped out.

And Apple now, all of these devices that are the more recent A11 and A10 devices, they're still receiving updates, right, from Apple. That means that the instant an update is released, it can now be fully reverse engineered, analyzed, and compared against the previous version, which will allow both security researchers, but also bad guys, to figure out what Apple has changed, what it is exactly that Apple fixed. And if they're able to get an exploit out into the wild before a targeted device has been updated, they could take advantage of that.

So Apple is famously rather quiet about their security updates. We've often talked about how Apple just says, oh, we fixed this and this and this, but with no details. Well, it hasn't been possible until now to easily get those details. Now any device, the instant it is updated, can be jailbroken through this publicly disclosed and unfixable Boot ROM exploit in order to crack the device open, allow all the updated files to be dumped and analyzed.

So we probably are entering a new era in Apple device security. As we know, they recently announced they would be making some selected special iPhones or iOS devices available to a few handpicked researchers who wish to explore iOS for bugs and security flaws. Well, that's no longer necessary. The entire world has that level of access now, and Apple has no choice but to continue supporting these devices which are vulnerable to this boot time bug. So it isn't the end for - it's not directly a problem for iOS end users, but it really does change the security landscape for Apple, so long as devices that are still being updated and have this unpatchable Boot ROM flaw are out in the world.

Leo: Now, it doesn't persist, but it's enough to get something else on your phone that would persist, like a spy program or that kind of thing.

Steve: Correct, until it's rebooted. So if you lost physical control, you can imagine, I mean, imagine that the guys at...

Leo: When you come into the United States, and they take your phone, and they go into another room.

Steve: Yup. Yup.

Leo: This happens all the time, not just entering the border. If you're arrested, they take your phone, lock you up, modify your phone, give it back to you, say thanks, sorry, send you on your way.

Steve: Yup. And Cellebrite, the guys who do the iPhone jailbreaking...

Leo: We know they know about this; right?

Steve: They'll jump on this in a heartbeat. And so it does mean that the world of jailbreaking, I mean, like being a problem, it's just not anymore. Yes, on the A12 and A13 devices Apple fixed this. And it was known...

Leo: So the real risk to an end user would be if somebody got your phone, took it away. They'd hack it with this thing, jailbreak it, which would allow them to sideload a surveillance app.

Steve: Yes.

Leo: They could even reboot it, which would un-jailbreak it, but the surveillance app would still be there, and give it back to you.

Steve: Correct. Correct. So that's the direct effect of physical access to your device. The secondary effect is this makes bugs way easier to find, and it makes Apple's changes easy to reverse engineer. You know, we know because we're talking about it all the time, the instant Microsoft releases an update, the old version of the DLL is compared against the new version. What it is they changed is found, and an exploit exists within hours of Windows release. We're talking about this all the time.

Apple hasn't been exposed to that because they've been able to lock down their platform to a degree that Microsoft inherently can't. That just changed. Apple can no longer lock down their platform. It is going to be open for anyone to reverse engineer any changes Apple makes to devices which are necessarily still being supported and are receiving updates.

Leo: Even iOS 13 you could look into that and see what they're doing on the newest iOS, what new updates do.

Steve: On the older devices, yes.

Leo: But you have to have an older device, yeah.

Steve: I have an iPhone 10. I can do it on mine, if I cared.

Leo: Wow.

Steve: Yeah. It's a game changer.

Leo: I had no idea. I mean, we talked about it, but I didn't realize the real risk was this risk to Apple, more than to end users.

Steve: Yeah, I think so. You know, Apple has enjoyed being coy, and they've been able to because they've had complete control of the boot chain. And they just lost control.

Leo: This is why we need Steve Gibson in the world, and Security Now! is your "must listen to" podcast every week. Thank you, Steve. We do Security Now! Tuesdays, about 1:30 Pacific, sometimes later because it's a long day. 1:30 Pacific,

4:30 Eastern, 20:30 UTC. There we go. I got all the times right. You can watch live at TWiT.tv/live. If you do that, join us in the chatroom, irc.twit.tv. Those are the kids talking about the show behind the scenes as we record it. But there's also a great place to go now to talk about the show after we deliver it, because it is a podcast. That's our new forums at twit.community. Check it out, twit.community.

Steve has 16Kb audio versions of this show, 64Kb audio. Plus, unique to his site, he's got the transcripts. That's at GRC.com. While you're there, pick up a copy of SpinRite, the world's best hard drive maintenance and recovery utility. He didn't talk about it, but I will. There's also lots of other great free stuff there, including ShieldsUP! and lots of information about SQRL: GRC.com. You want to ask Steve a question or talk at him, you can do that on Twitter: @SGgrc. He accepts DMs: @SGgrc. It's also a great place to follow Steve. The show notes go up there ahead of time. And anything he finds out during the week he tweets: @SGgrc.

Our website for the show, TWiT.tv/sn. You can get every episode ever recorded there, all what is it, 736 now. You could also subscribe. In fact, that's the best thing to do. Get your complete set. Just subscribe in your favorite podcast application, and it will be delivered to you the minute it's done. But now we are done. Thank you, Steve. I will not be here next week, or the week after.

Steve: Ooh, your cruise begins. Your cruise begins.

Leo: My vacation.

Steve: It's only two weeks you're missing, not three?

Leo: I'm missing a lot. I'll be back. I will be back November 20th, I think. The first Tuesday - we get back November 15th. The first Tuesday after that.

Steve: I think you're missing several weeks.

Leo: I'm missing quite a few. I'm sorry. What can I say? It's vacation. I apologize. I won't be here. I'm not sure who's handling the show. I'm guessing it's Jason Howell. He's my usual fill-in. But of course we tune in, not for me, but for Steve. So he'll be here; right?

Steve: I'll be here. I never go anywhere.

Leo: I know.

Steve: Well, okay, not quite never.

Leo: Yeah, that's changing, isn't it. That's changing.

Steve: Not quite never, but I strongly resist.

Leo: I got some bad news/good news for you. LastPass enjoyed our visit to Boston so much they want to do two more events next year. I'm not sure where they'll be. They don't have to be with Steve, but I'm thinking they probably ought to be. Anyway, I'll fill you in on that down the road.

Steve: Sounds good.

Leo: Yeah. We had a good time, didn't we.

Steve: We did. And I will commit now.

Leo: Oh, wow. You just made somebody at LastPass very, very happy. Thank you, Steve. We'll see you next time on Security Now!.

Steve: Thanks, Leo.